Package 'dTBM'

October 13, 2022

Title Multi-Way Spherical Clustering via Degree-Corrected Tensor Block Models

Version 2.0

Date 2022-01-10

Maintainer Jiaxin Hu <jhu267@wisc.edu>

Description Implement weighted higher-order initialization and angle-based iteration for multi-way spherical clustering under degree-corrected tensor block model.

Imports tensorregress, WeightedCluster, EnvStats

License GPL (>= 2)

Encoding UTF-8

LazyData true

Author Jiaxin Hu [aut, cre, cph],
Miaoyan Wang [aut, cph]

RoxygenNote 7.1.1

Depends R (>= 3.5.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-06 03:50:02 UTC

R topics documented:

`angle_iteration` *Angle-based iteration*

Description

Angle-based iteration for multiway spherical clustering under degree-corrected tensor block model. This function takes the tensor/matrix observation, initial clustering assignment, and a logic variable indicating the symmetry as input. Output is the refined clustering assignment.

Usage

```
angle_iteration(Y, z0, max_iter, alpha1 = 0.01, asymm)
```

Arguments

`Y`	array/matrix, order-3 tensor/matrix observation
`z0`	a list of vectors, initial clustering assignment; see "details"
`max_iter`	integer, max number of iterations if update does not converge
`alpha1`	number, substitution of degenerate core tensor; see "details"
`asymm`	logic variable, if "TRUE", assume the clustering assignment differs in different modes; if "FALSE", assume all the modes share the same clustering assignment

Details

`z0` should be a length 2 list for matrix and length 3 list for tensor observation; observations with non-identical dimension on each mode are only applicable with `asymm = T`;

When the estimated core tensor has a degenerate slice, i.e., a slice with all zero elements, randomly pick an entry in the degenerate slice with value `alpha1`.

Value

a list containing the following:

`z` a list of vectors recording the estimated clustering assignment

`s_deg` logic variable, if "TRUE", degenerate estimated core tensor/matrix occurs during the iteration; if "FALSE", otherwise

Examples

```
test_data = sim_dTBM(seed = 1, imat = FALSE, asymm = FALSE, p = c(50,50,50), r = c(3,3,3),
                     core_control = "control", s_min = 0.05, s_max = 1,
                     dist = "normal", sigma = 0.5,
                     theta_dist = "pareto", alpha = 4, beta = 3/4)

initialization <- wkmeans(test_data$Y, r = c(3,3,3), asymm = FALSE)

iteration <- angle_iteration(test_data$Y, initialization$z0, max_iter = 20, asymm = FALSE)
```

dtbm *Multiway spherical clustering for degree-corrected tensor block model*

Description

Multiway spherical clustering for degree-corrected tensor block model including weighted higher-order initialization and angle-based iteration. Main function in the package. This function takes the tensor/matrix observation, the cluster number, and a logic variable indicating the symmetry as input. Output contains initial and refined clustering assignment.

Usage

```
dtbm(Y, r, max_iter, alpha1 = 0.01, asymm)
```

Arguments

`Y`	array/matrix, order-3 tensor/matrix observation
`r`	vector, the cluster number on each mode; see "details"
`max_iter`	integer, max number of iterations if update does not converge
`alpha1`	number, substitution of degenerate core tensor; see "details"
`asymm`	logic variable, if "TRUE", assume the clustering assignment differs in different modes; if "FALSE", assume all the modes share the same clustering assignment

Details

`r` should be a length 2 vector for matrix and length 3 vector for tensor observation;

all the elements in `r` should be integer larger than 1;

symmetric case only allow `r` with the same cluster number on each mode;

observations with non-identical dimension on each mode are only applicable with `asymm = T`.

When the estimated core tensor has a degenerate slice during iteration, i.e., a slice with all zero elements, randomly pick an entry in the degenerate slice with value `alpha1`.

Value

a list containing the following:

`z` a list of vectors recording the refined clustering assignment with initialization `z0`

`s_deg` logic variable, if "TRUE", degenerate estimated core tensor/matrix occurs during the iteration; if "FALSE", otherwise

`z0` a list of vectors recording the initial clustering assignment

`s0` a list of vectors recording the index of degenerate entities with random clustering assignment in initialization

Examples

```
test_data = sim_dTBM(seed = 1, imat = FALSE, asymm = FALSE, p = c(50,50,50), r = c(3,3,3),
                     core_control = "control", s_min = 0.05, s_max = 1,
                     dist = "normal", sigma = 0.5,
                     theta_dist = "pareto", alpha = 4, beta = 3/4)

result = dtbm(test_data$Y, r = c(3,3,3), max_iter = 20, asymm = FALSE)
```

HCP *HCP data*

Description

The HCP data is obtained by preprocessing the data from Human Connectome Project (HCP); see https://wiki.humanconnectome.org/display/PublicData/.

Usage

```
data(HCP)
```

Format

A list. Includes a 68-68-136 binary array named "tensor" and a 136-573 data frame named "attr".

Details

The array "tensor" is a $68 \times 68 \times 136$ binary tensor consisting of structural connectivity patterns among 68 brain regions for 136 individuals. All the individual images were preprocessed following a standard pipeline (Zhang et al., 2018), and the brain was parcellated to 68 regions-of-interest following the Desikan atlas (Desikan et al., 2006). The tensor entries encode the presence or absence of fiber connections between those 68 brain regions for each of the 136 individuals.

The data frame "attr" is a 136×573 matrix consisting of 573 personal features for 136 individuals. The full list of covariates can be found at: https://wiki.humanconnectome.org/display/PublicData/

---

peru *Peru Legislation data*

Description

The Peru Legislation data is obtained by preprocessing the original data in Lee et al., 2017.

Usage

```
data(peru)
```

Format

A list. Includes a 116-2 data frame named "attr_data", a 5844-7 data frame named "laws_data", and a 116-116-116 binary array named "network_data".

Details

The data frame "attr_data" is a 116 x 2 matrix consisting the name and party affiliation of 116 legislators in the top five parties. The legislators IDs are recorded in the row names of the matrix.

The data frame "laws_data" is a 5844 x 7 matrix recording the co-sponsorship of 116 legislators of 802 bills during the first half of 2006-2007 year.

The array "network_data" is a 116 x 116 x 116 binary tensor recording the presence of order-3 co-sponsorship among legislators based on "laws_data". Specfically, the entry (i,j,k) is 1 if the legislators (i,j,k) have sponsored the same bill, and the entry (i,j,k) is 0 otherwise.

---

select_r *Cluster number selection*

Description

Estimate the cluster number in the degree-corrected tensor block model based on BIC criterion. The choice of BIC aims to balance between the goodness-of-fit for the data and the degree of freedom in the population model. This function is restricted for the Gaussian observation.

Usage

```
select_r(Y, r_range, asymm = F)
```

Arguments

`Y`	array/matrix, order-3 Gaussian tensor/matrix observation
`r_range`	matrix, candidates for the cluster number on each row; see "details"
`asymm`	logic variable, if "TRUE", clustering assignment differs in different modes; if "FALSE", all the modes share the same clustering assignment

Details

`r_range` should be a two-column matrix for matrix and three-column matrix for tensor observation;

all the elements in `r_range` should be integer larger than 1;

symmetric case only allow candidates with the same cluster number on each mode;

observations with non-identical dimension on each mode are only applicable with `asymm = T`.

Value

a list containing the following:

`r` vector, the cluster number among the candidates with minimal BIC value

`bic` vector, the BIC value for each candidiate

Examples

```
test_data = sim_dTBM(seed = 1, imat = FALSE, asymm = FALSE, p = c(50,50,50), r = c(3,3,3),
                     core_control = "control", s_min = 0.05, s_max = 1,
                     dist = "normal", sigma = 0.5,
                     theta_dist = "pareto", alpha = 4, beta = 3/4)

r_range = rbind(c(2,2,2), c(3,3,3),c(4,4,4),c(5,5,5))
selection <- select_r(test_data$Y, r_range, asymm = FALSE)
```

`sim_dTBM` *Simulation of degree-corrected tensor block models*

Description

Generate order-3 tensor/matrix observations with degree heterogeneity under degree-corrected tensor block models.

Usage

```
sim_dTBM(
  seed = NA,
  imat = F,
  asymm = F,
  p,
  r,
  core_control = c("random", "control"),
  delta = NULL,
  s_min = NULL,
  s_max = NULL,
  dist = c("normal", "binary"),
  sigma = 1,
  theta_dist = c("abs_normal", "pareto", "non"),
  alpha = NULL,
  beta = NULL
)
```

Arguments

`seed`	number, random seed for generating data
`imat`	logic variable, if "TRUE", generate matrix data; if "FALSE", generate order-3 tensor data
`asymm`	logic variable, if "TRUE", clustering assignment differs in different modes; if "FALSE", all the modes share the same clustering assignment
`p`	vector, dimension of the tensor/matrix observation
`r`	vector, cluster number on each mode

core_control	character, the way to control the generation of core tensor/matrix; see "details"
delta	number, Frobenius norm of the slices in core tensor if core_control = "control"
s_min	number, value of off-diagonal elements in original core tensor/matrix if core_control = "control"
s_max	number, value of diagonal elements in original core tensor/matrix if core_control = "control"
dist	character, distribution of tensor/matrix observation; see "details"
sigma	number, standard deviation of Gaussian noise if dist = "normal"
theta_dist	character, distribution of degree heterogeneity; see "details"
alpha	number, shape parameter in pareto distribution if theta_dist = "pareto"
beta	number, scale parameter in pareto distribution if theta_dist = "pareto"

Details

The general tensor observation is generated as

Y = S x1 Theta1 M1 x2 Theta2 M2 x3 Theta3 M3 + E,

where S is the core tensor, Thetak is a diagonal matrix with elements in the k-th vector of theta, Mk is the membership matrix based on the clustering assignment in the k-th vector of z with r[k] clusters, E is the mean-zero noise tensor, and xk refers to the matrix-by-tensor product on the k-th mode, for k = 1,2,3.

If imat = T, Y,S,E degenerate to matrix and Y = Theta1 M1 S M2^T Theta2^T + E.

If asymm = F, Thetak = Theta and Mk = M for all k = 1,2,3.

core_control specifies the way to generate S:

If core_control = "control", first generate S as a diagonal tensor/matrix with diagonal elements s_max and off-diagonal elements s_min; then scale the original core such that Frobenius norm of the slices equal to delta, i.e, delta = sqrt(sum(S[1,,]^2)) or delta = sqrt(sum(S[1,]^2)); ignore the scaling if delta = NULL; option "control" is only applicable for symmetric case asymm = F.

If core_control = "random", generate S with random entries following uniform distribution U(0,1).

dist specifies the distribution of E: "normal" for Gaussian and "binary" for Bernoulli distribution; sigma specifices the standard deviation if dist = "normal".

theta_dist firstly specifies the distribution of theta: "non" for constant 1, "abs_normal" for absoulte normal distribution, "pareto" for pareto distribution; alpha, beta specify the shape and scale parameter if theta_dist = "pareto"; then scale theta to have mean equal to one in each cluster.

Value

a list containing the following:

Y array (if imat = F)/matrix (if imat = T), simulated tensor/matrix observations with dimension p

X array (if imat = F)/matrix (if imat = T), mean tensor/matrix of the observation, i.e., the expectation of Y

S array (if imat = F)/matrix (if imat = T), core tensor/matrix recording the block effects with dimension r

theta a list of vectors, degree heterogeneity on each mode

z a list of vectors, clustering assignment on each mode

Examples

```
test_data = sim_dTBM(seed = 1, imat = FALSE, asymm = FALSE, p = c(50,50,50), r = c(3,3,3),
                     core_control = "control", s_min = 0.05, s_max = 1,
                     dist = "normal", sigma = 0.5,
                     theta_dist = "pareto", alpha = 4, beta = 3/4)
```

---

wkmeans *Weighted higher-order initialization*

Description

Weighted higher-order initialization for multiway spherical clustering under degree-corrected tensor block model. This function takes the tensor/matrix observation, the cluster number, and a logic variable indicating the symmetry as input. Output is the estimated clustering assignment.

Usage

```
wkmeans(Y, r, asymm)
```

Arguments

Y	array/matrix, order-3 tensor/matrix observation
r	vector, the cluster number on each mode; see "details"
asymm	logic variable, if "TRUE", assume the clustering assignment differs in different modes; if "FALSE", assume all the modes share the same clustering assignment

Details

r should be a length 2 vector for matrix and length 3 vector for tensor observation;

all the elements in r should be integer larger than 1;

symmetric case only allow r with the same cluster number on each mode;

observations with non-identical dimension on each mode are only applicable with asymm = T.

Value

a list containing the following:

z0 a list of vectors recording the estimated clustering assignment

s0 a list of vectors recording the index of degenerate entities with random clustering assignment

Examples

```
test_data = sim_dTBM(seed = 1, imat = FALSE, asymm = FALSE, p = c(50,50,50), r = c(3,3,3),
                     core_control = "control", s_min = 0.05, s_max = 1,
                     dist = "normal", sigma = 0.5,
                     theta_dist = "pareto", alpha = 4, beta = 3/4)

initialization <- wkmeans(test_data$Y, r = c(3,3,3), asymm = FALSE)
```

Index