Package 'fusedMGM'

October 17, 2024

Type Package

Title Implementation of Fused MGM to Infer 2-Class Networks

Version 0.1.2

RoxygenNote 7.3.2

Maintainer Jaehyun Park <J.31.Park@gmail.com>

License MIT + file LICENSE

Imports fastDummies, parallel, bigmemory, gplots, bigalgebra, biganalytics

Description Implementation of fused Markov graphical model (FMGM; Park and Won, 2022). The functions include building mixed graphical model (MGM) objects from data, inference of networks using FMGM, stable edge-specific penalty selection (StEPS) for the determination of penalization parameters, and the visualization. For details, please refer to Park and Won (2022) <doi:10.48550/arXiv.2208.14959>.

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Jaehyun Park [aut, cre, cph] (<https://orcid.org/0000-0003-2900-8340>), Sungho Won [ths]

Repository CRAN

Date/Publication 2024-10-17 15:20:02 UTC

Contents

`data_all` *An example of 2-group mixed data*

Description

A dataset containing 50 numeric and 50 categorical variables Includes 250 observations in each group

Usage

```
data_all
```

Format

'data_all' A data frame with 500 rows and 100 columns.

`data_mini` *A toy example of 2-group mixed data*

Description

A dataset containing 4 numeric and 6 categorical variables Includes 250 observations in each group

Usage

```
data_mini
```

Format

'data_mini' A data frame with 500 rows and 10 columns.

FMGM_mc *Main function of fused MGM*

Description

Infers networks from 2-class mixed data

Usage

```
FMGM_mc(
  data,
  ind_disc,
  group,
  t = 1,
  L = NULL,
  eta = 2,
  lambda_intra,
  lambda_intra_prior = NULL,
  lambda_inter,
  with_prior = FALSE,
  prior_list = NULL,
  converge_by_edge = TRUE,
  tol_edge = 3,
  tol_mgm = 1e-05,
  tol_g = 1e-05,
  tol_fpa = 1e-12,
  maxit = 1e+06,
  polish = TRUE,
  tol_polish = 1e-12,
  cores = parallel::detectCores(),
  verbose = FALSE
)
```

Arguments

data	Data frame with rows as observations and columns as variables
ind_disc	Indices of discrete variables
group	Group indices, must be provided with the observation names
t	Numeric. Initial value of coefficient that reflect 2 previous iterations in fast proximal gradient method. Default: 1
L	Numeric. Initial guess of Lipschitz constant. Default: missing (use backtracking)
eta	Numeric. Multipliers for L in backtracking. Default: 2
lambda_intra	Vector with 3 numeric variables. Penalization parameters for network edge weights

`lambda_intra_prior`
: Vector with 3 numeric variables. Penalization parameters for network edge weights, applied to the edges with prior information

`lambda_inter`
: Vector with 3 numeric variables. Penalization parameters for network edge weight differences

`with_prior`
: Logical. Is prior information provided? Default: FALSE

`prior_list`
: List of prior information. Each element must be a 3-column data frames, with the 1st and the 2nd columns being variable names and the 3rd column being prior confidence (0,1)

`converge_by_edge`
: Logical. The convergence should be judged by null differences of network edges after iteration. If FALSE, the rooted mean square difference (RMSD) of edge weights is used. Default: TRUE

`tol_edge`
: Integer. Number of consecutive iterations of convergence to stop the iteration. Default: 3

`tol_mgm`
: Numeric. Cutoff of network edge RMSD for convergence. Default: 1e-05

`tol_g`
: Numeric. Cutoff of iternations in prox-grad map calculation. Default: 1e-05

`tol_fpa`
: Numeric. Cutoff for fixed-point approach. Default: 1e-12

`maxit`
: Integer. Maximum number of iterations in fixed-point approach. Default: 1000000

`polish`
: Logical. Should the edges with the weights below the cutoff should be discarded? Default: TRUE

`tol_polish`
: Numeric. Cutoff of polishing the resulting network. Default: 1e-12

`cores`
: Integer. Number of cores to use multi-core utilization. Default: maximum number of available cores

`verbose`
: Logical. If TRUE, the procedures are reported in real-time manner. Default: FALSE

Details

If the value of Lipschitz constant, L, is not provided, the backtracking will be performed

Value

The resulting networks, in the form of a list of MGMs

Examples

```
chk <- tolower(Sys.getenv("_R_CHECK_LIMIT_CORES_", ""))

if (Sys.info()['sysname'] != 'Linux') {
  cores=1L
} else {
  chk = tolower(Sys.getenv("_R_CHECK_LIMIT_CORES_", ""))
  if (nzchar(chk) && (chk != "false")) {
    cores=2L
  } else {
```

```
    cores=parallel::detectCores() - 1 ;
  }
}

## Not run:
data(data_all) ;  # Example 500-by-100 simulation data
data(ind_disc) ;

group <- rep(c(1,2), each=250) ;
names(group) <- rownames(data_all) ;

res_FMGM <- FMGM_mc(data_all, ind_disc, group,
                    lambda_intra=c(0.2,0.15,0.1), lambda_inter=c(0.2,0.15,0.1),
                    cores=cores, verbose=TRUE)

## End(Not run)


data(data_mini) ; # Minimal example 500-by-10 simulation data
data(ind_disc_mini) ;

group <- rep(c(1,2), each=250) ;
names(group) <- rownames(data_mini) ;

res_FMGM_mini <- FMGM_mc(data_mini, ind_disc_mini, group,
                    lambda_intra=c(0.2,0.15,0.1), lambda_inter=c(0.2,0.15,0.1),
                    cores=cores, verbose=TRUE)
```

FMGM_plot *A plot function for a list of MGMs. The output is usually from FMGM main function.*

Description

This function is written based on R base function 'heatmap'.

Usage

```
FMGM_plot(
  MGM_list,
  sortby = "diff",
  highlight = c(),
  tol_polish = 1e-12,
  tol_plot = 0.01,
  sideColor = FALSE,
  distfun = dist,
  hclustfun = hclust,
  reorderfun = function(d, w) reorder(d, w),
```

```
  margins = c(2.5, 2.5),
  cexRow = 0.1 + 0.5/log10(n),
  cexCol = cexRow,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  verbose = getOption("verbose")
)
```

Arguments

MGM_list	A list of graphs from 2 groups. Usually a result of FMGM main function.
sortby	Determines the standard of sorting & dendrograms. Either 1, 2, or "diff" (default).
highlight	A vector of variable names or indices to highlight
tol_polish	A threshold for the network edge presence
tol_plot	Only network edges above this value will be displayed on the heatmap
sideColor	A named vector determining a sidebar colors. Set NULL to make the colors based on the variable types (discrete/continuous). Default: FALSE (no sidebars)
distfun	A function for the distances between rows/columns
hclustfun	A function for hierarchical clustering
reorderfun	A function of dendrogram and weights for reordering
margins	A numeric vector of 2 numbers for row & column name margins
cexRow	A visual parameter cex for row axis labeling
cexCol	A visual parameter cex for column axis labeling, default to be same as cexRow
main	Main title, default to none
xlab	X-axis title, default to none
ylab	Y-axis title, default to none
verbose	Logical. Should plotting information be printed?

Value

None

Examples

```
chk <- tolower(Sys.getenv("_R_CHECK_LIMIT_CORES_", ""))

if (Sys.info()['sysname'] != 'Linux') {
  cores=1L
} else {
  chk = tolower(Sys.getenv("_R_CHECK_LIMIT_CORES_", ""))
  if (nzchar(chk) && (chk != "false")) {
    cores=2L
  } else {
```

```
    cores=parallel::detectCores() - 1 ;
  }
}

## Not run:
data(data_all) ;  # Example 500-by-100 simulation data
data(ind_disc) ;

group <- rep(c(1,2), each=250) ;
names(group) <- seq(500) ;

res_FMGM <- FMGM_mc(data_all, ind_disc, group,
                    lambda_intra=c(0.2,0.15,0.1), lambda_inter=c(0.2,0.15,0.1),
                    cores=cores, verbose=TRUE)

FMGM_plot(res_FMGM)

## End(Not run)


data(data_mini) ; # Minimal example 500-by-10 simulation data
data(ind_disc_mini) ;

group <- rep(c(1,2), each=250) ;
names(group) <- rownames(data_mini) ;

res_FMGM_mini <- FMGM_mc(data_mini, ind_disc_mini, group,
                    lambda_intra=c(0.2,0.15,0.1), lambda_inter=c(0.2,0.15,0.1),
                    cores=cores, verbose=TRUE)

FMGM_plot(res_FMGM_mini)
```

FMGM_StEPS *StEPS: train subsamples and calculate edge instabilities*

Description

From large to small values of candidates, calculate the edge inference instabilities from subsamples The smallest values with the instabilities under the cutoff are chosen. See Sedgewich et al. (2016) for more details

Usage

```
FMGM_StEPS(
  data,
  ind_disc,
  group,
  lambda_list,
  with_prior = FALSE,
```

```
  prior_list = NULL,
  N = 20,
  b = NULL,
  gamma = 0.05,
  perm = 10000,
  eps = 0.05,
  tol_polish = 1e-12,
  ...,
  cores = parallel::detectCores(),
  verbose = FALSE
)
```

Arguments

data	Data frame with rows as observations and columns as variables
ind_disc	Indices of discrete variables
group	Group indices, must be provided with the observation names
lambda_list	Vector with numeric variables. Penalization parameter candidates
with_prior	Logical. Is prior information provided? Default: FALSE
prior_list	List of prior information. Each element must be a 3-column data frames, with the 1st and the 2nd columns being variable names and the 3rd column being prior confidence (0,1)
N	Integer. Number of subsamples to use. Default: 20
b	Integer. Number of observations in each subsample. Default: ceiling(10*sqrt(number of total observations))
gamma	Numeric. Instability cutoff. Default: 0.05
perm	Integer. Number of permutations to normalize the prior confidence. Default: 10000
eps	Numeric. Pseudocount to calculate the likelihood of edge detection. Default: 0.05
tol_polish	Numeric. Cutoff of polishing the resulting network. Default: 1e-12
...	Other arguments sent to fast proximal gradient method
cores	Integer. Number of cores to use multi-core utilization. Default: maximum number of available cores
verbose	Logical. If TRUE, the procedures are reported in real-time manner. Default: FALSE

Value

The resulting networks, in the form of a list of MGMs

Examples

```
chk <- tolower(Sys.getenv("_R_CHECK_LIMIT_CORES_", ""))

if (Sys.info()['sysname'] != 'Linux') {
  cores=1L
} else {
  chk = tolower(Sys.getenv("_R_CHECK_LIMIT_CORES_", ""))
  if (nzchar(chk) && (chk != "false")) {
    cores=2L
  } else {
    cores=parallel::detectCores() - 1 ;
  }
}

## Not run:
data(data_all) ;  # Example 500-by-100 simulation data
data(ind_disc) ;

group <- rep(c(1,2), each=250) ;
names(group) <- rownames(data_all) ;

lambda_list <- 2^seq(log2(.08), log2(.32), length.out=7) ;
lambda_list <- sort(lambda_list, decreasing=TRUE) ;

res_steps <- FMGM_StEPS(data_all, ind_disc, group,
                    lambda_list=lambda_list,
                    cores=cores, verbose=TRUE)

data(data_mini) ; # Minimal example 500-by-10 simulation data
data(ind_disc_mini) ;

group <- rep(c(1,2), each=250) ;
names(group) <- rownames(data_mini) ;

lambda_list <- 2^seq(log2(.08), log2(.32), length.out=7) ;
lambda_list <- sort(lambda_list, decreasing=TRUE) ;

res_steps_mini <- FMGM_StEPS(data_mini, ind_disc_mini, group,
                    lambda_list=lambda_list,
                    cores=cores, verbose=TRUE)

## End(Not run)
```

---

ind_disc *An example of 2-group mixed data*

Description

A vector indicating which columns in 'data_all' have categorical variables

Usage

```
ind_disc
```

Format

'ind_disc' A 50-length vector with discrete variable indices.

---

ind_disc_mini *A toy example of 2-group mixed data*

Description

A vector indicating which columns in 'data_mini' have categorical variables

Usage

```
ind_disc_mini
```

Format

'ind_disc_mini' A 6-length vector with discrete variable indices.

---

make_MGM_list *Make MGM lists from input data*

Description

Make MGM lists from input data

Usage

```
make_MGM_list(X, Y, group)
```

Arguments

X	data frame or matrix of continuous variables (row: observation, column: variable)
Y	data frame or matrix of discrete variables (row: observation, column: variable)
group	group variable vector, with the sample names

Value

A list of MGM objects. The length is equal to the unique number of groups.

MGM *Defining S3 object "MGM"*

Description

Defining S3 object "MGM"

Usage

```
MGM(X, Y, g)
```

Arguments

X	data frame or matrix of continuous variables (row: observation, column: variable)
Y	data frame or matrix of discrete variables (row: observation, column: variable)
g	group index, needed for temporary files

Value

An S3 ‘MGM‘ object, containing data, network parameters, and the 1st derivatives

Index