Package: CVN (via r-universe)

September 11, 2024

Title Covariate-varying Networks

Version 1.1

Date 2022-09-05

Author Louis Dijkstra [aut, cre]

Maintainer Louis Dijkstra <dijkstra@leibniz-bips.de>

Description A package for inferring high-dimensional Gaussian graphical networks that change with multiple discrete covariates

Depends R (>= 4.0.2), Rcpp, doSNOW, visNetwork, parallel, progress, Matrix, huge, glasso, crayon, ggplot2, reshape2, dplyr

LinkingTo Rcpp

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

URL https://github.com/bips-hb/CVN

BugReports https://github.com/bips-hb/CVN/issues

Repository https://bips-hb.r-universe.dev

RemoteUrl https://github.com/bips-hb/CVN

RemoteRef HEAD

RemoteSha 01a689108fbac014a1ea345126dd0252e1496ef6

Contents

`check_correctness_input` *Check whether Input is Valid*

Description

Checks whether the input for the function `CVN` is valid. This function does not return anything. The execution of the function halts when an issue has been detected.

Usage

```
check_correctness_input(raw_data, W, lambda1, lambda2, gamma1, gamma2, rho)
```

Arguments

raw_data	A list with matrices. The number of columns should be the same for each matrix
W	The $(m \times m)$-dimensional upper-triangular weight matrix W
lambda1	A vector of λ_1's LASSO penalty terms
lambda2	A vector of λ_2's global smoothing parameters
gamma1	A vector of γ_1's LASSO penalty terms. Note that $\gamma_1 = \frac{2\lambda_1}{mp(1-p)}$
gamma2	A vector of γ_2's global smoothing parameters. Note that $\gamma_2 = \frac{4\lambda_2}{m(m-1)p(p-1)}$
rho	The ρ ADMM's penalty parameter

See Also

CVN

---

create_edges_visnetwork

Create a `data.frame` *for the Edges for* `visNetwork`

Description

In order to visualize a graph, we need to create a `data.frame` that can be used by the `visNetwork` package. This function returns the needed `data.frame` given a adjacency matrix.

Usage

```
create_edges_visnetwork(adj_matrix)
```

Arguments

adj_matrix	A symmetric adjacency matrix

Value

Data frame that be used as input for `visNetwork` `create_matrix_D` *Create matrix D to be used for the Generalized LASSO*

Description

Generates a matrix D to be used for the generalized LASSO. We solve a generalized LASSO problem for each edge (s,t) for each update step for Z.

Usage

```
create_matrix_D(W, lambda1, lambda2, rho = 1, remove_zero_row = TRUE)
```

Arguments

`W`	The $(m \times m)$-dimensional upper-triangular weight matrix W
`lambda1`	The λ_1 LASSO penalty term
`lambda2`	The λ_2 global smoothing parameter
`rho`	The ρ ADMM's penalty parameter (Default: `1`)
`remove_zero_row`	If `TRUE`, rows with zeros are removed. (Default: `TRUE`)

Value

A $((m \cdot (m+1)/2) \times m)$-dimensional matrix

References

Tibshirani, R. J., & Taylor, J. (2011). The solution path of the generalized lasso. Annals of Statistics, 39(3), 1335–1371. https://doi.org/10.1214/11-AOS878

Examples

```
m <- 4 # number of graphs
W <- matrix(1, nrow = m, ncol = m)

# penalty terms:
lambda1 <- .2
lambda2 <- .4
rho <- 1

CVN::create_matrix_D(W, lambda1, lambda2, rho)
#      [,1] [,2] [,3] [,4]
# [1,]  0.2  0.0  0.0  0.0
# [2,]  0.0  0.2  0.0  0.0
# [3,]  0.0  0.0  0.2  0.0
# [4,]  0.0  0.0  0.0  0.2
# [5,]  0.4 -0.4  0.0  0.0
# [6,]  0.4  0.0 -0.4  0.0
```

```
# [7,]  0.4  0.0  0.0 -0.4
# [8,]  0.0  0.4 -0.4  0.0
# [9,]  0.0  0.4  0.0 -0.4
# [10,]  0.0  0.0  0.4 -0.4
```

`create_nodes_visnetwork` *Nodes for the* `visNetwork` *package*

Description

Creates a data frame that can be used for the `visNetwork` package.

Usage

```
create_nodes_visnetwork(n_nodes, labels = 1:n_nodes)
```

Arguments

`n_nodes`	Number of nodes in the graph
`labels`	The labels for the individual nodes (Default: `1:n_nodes`)

Value

Data frame with two columns: `id` and `title`

`create_weight_matrix` *Different Weight Matrices*

Description

This function generates different weight matrices, W. There are several types:

- `full` All graphs are fully connected with weight 1
- `glasso` All graphs are disconnected with weight 0. This mimicks the GLASSO, where each graph is estimated independently
- `grid` A weight matrix for a 3×3 grid
- `uniform-random` Fully-connected, but the entries are drawn from a uniform distribution

Usage

```
create_weight_matrix(
  type = c("full", "glasso", "grid", "uniform-random"),
  m = 9
)
```

Arguments

`type` The type of weight matrix

`m` Number of graphs

Value

Weight matrix

CVN *Estimating a Covariate-Varying Network (CVN)*

Description

Estimates a covariate-varying network model (CVN), i.e., m Gaussian graphical models that change with (multiple) external covariate(s). The smoothing between the graphs is specified by the $(m \times m)$-dimensional weight matrix W. The function returns the estimated precision matrices for each graph.

An estimator for graphical models changing with multiple discrete external covariates

Usage

```
CVN(
  data,
  W,
  lambda1 = 1:2,
  lambda2 = 1:2,
  gamma1 = NULL,
  gamma2 = NULL,
  rho = 1,
  eps = 1e-04,
  maxiter = 100,
  truncate = 1e-05,
  rho_genlasso = 1,
  eps_genlasso = 1e-10,
  maxiter_genlasso = 100,
  truncate_genlasso = 1e-04,
  n_cores = min(length(lambda1) * length(lambda2), parallel::detectCores() - 1),
  normalized = FALSE,
  warmstart = TRUE,
  minimal = FALSE,
  verbose = TRUE
)
```

Arguments

data	A list with matrices, each entry associated with a single graph. The number of columns should be the same for each matrix. Number of observations can differ
W	The $(m \times m)$-dimensional symmetric weight matrix W
lambda1	Vector with different λ_1 LASSO penalty terms (Default: 1:2)
lambda2	Vector with different λ_2 global smoothing parameter values (Default: 1:2)
gamma1	A vector of γ_1's LASSO penalty terms, where $\gamma_1 = \frac{2\lambda_1}{mp(1-p)}$. If gamma1 is set, the value of lambda1 is ignored. (Default: NULL).
gamma2	A vector of γ_2's global smoothing parameters, where that $\gamma_2 = \frac{4\lambda_2}{m(m-1)p(p-1)}$. If gamma2 is set, the value of lambda2 is ignored.(Default: NULL).
rho	The ρ penalty parameter for the global ADMM algorithm (Default: 1)
eps	If the relative difference between two update steps is smaller than ϵ, the algorithm stops. See relative_difference_precision_matrices (Default: 1e-4)
maxiter	Maximum number of iterations (Default: 100)
truncate	All values of the final $\hat{\Theta}_i$'s below truncate will be set to 0. (Default: 1e-5)
rho_genlasso	The ρ penalty parameter for the ADMM algorithm used to solve the generalized LASSO (Default: 1)
eps_genlasso	If the relative difference between two update steps is smaller than ϵ, the algorithm stops. (Default: 1e-10)
maxiter_genlasso	Maximum number of iterations for solving the generalized LASSO problem (Default: 100)
truncate_genlasso	All values of the final $\hat{\beta}$ below truncate_genlasso will be set to 0. (Default: 1e-4)
n_cores	Number of cores used (Default: max. number of cores - 1, or the total number penalty term pairs if that is less)
normalized	Data is normalized if TRUE. Otherwise the data is only centered (Default: FALSE)
warmstart	If TRUE, use the huge package for estimating the individual graphs first (Default: TRUE)
minimal	If TRUE, the returned cvn is minimal in terms of memory, i.e., Theta, data and Sigma are not returned (Default: FALSE)
verbose	Verbose (Default: TRUE)

Value

A CVN object containing the estimates for all the graphs for each different value of (λ_1, λ_2). General results for the different values of (λ_1, λ_2) can be found in the data frame results. It consists of multiple columns, namely:

lambda1	λ_1 value
lambda2	λ_2 value
converged	whether algorithm converged or not

value	value of the negative log-likelihood function
n_iterations	number of iterations of the ADMM
aic	Aikake information criterion
gamma1	γ_1 value
gamma2	γ_2 value
id	The id. This corresponds to the indices of the lists
bic	Bayesian information criterion

The estimates of the precision matrices and the corresponding adjacency matrices for the different values of (λ_1, λ_2) can be found

Theta	A list with the estimated precision matrices $\{\hat{\Theta}_i(\lambda_1, \lambda_2)\}_{i=1}^m$, (only if minimal = FALSE)
adj_matrices	A list with the estimated adjacency matrices corresponding to the estimated precision matrices in Theta. The entries are 1 if there is an edge, 0 otherwise. The matrices are sparse using package Matrix

In addition, the input given to the CVN function is stored in the object as well:

Sigma	Empirical covariance matrices $\{\hat{\Sigma}_i\}_{i=1}^m$, (only if minimal = FALSE)
m	Number of graphs
p	Number of variables
n_obs	Vector of length m with number of observations for each graph
data	The data, but then normalized or centered (only if minimal = FALSE)
W	The $(m \times m)$-dimensional weight matrix W
maxiter	Maximum number of iterations for the ADMM
rho	The ρ ADMM's penalty parameter
eps	The stopping criterion ϵ
truncate	Truncation value for $\{\hat{\Theta}_i\}_{i=1}^m$
maxiter_genlasso	Maximum number of iterations for the generarlzed LASSO
rho_genlasso	The ρ generalized LASSO penalty parameter
eps_genlasso	The stopping criterion ϵ for the generalized LASSO
truncate_genlasso	Truncation value for β of the generalized LASSO
n_lambda_values	Total number of (λ_1, λ_2) value combinations
normalized	If TRUE, data was normalized. Otherwise data was only centered
warmstart	If TRUE, warmstart was used
minimal	If TRUE, data, Theta and Sigma are not added
hits_border_aic	If TRUE, the optimal model based on the AIC hits the border of (λ_1, λ_2)
hits_border_bic	If TRUE, the optimal model based on the BIC hits the border of (λ_1, λ_2)

Reusing Estimates

When estimating the graph for different values of λ_1 and λ_2, we use the graph estimated (if available) for other λ_1 and λ_2 values closest to them.

Author(s)

Louis Dijkstra

Examples

```
data(grid)
m <- 9 # must be 9 for this example

#' Choice of the weight matrix W.
#' (uniform random)
W <- matrix(runif(m*m), ncol = m)
W <- W %*% t(W)
W <- W / max(W)
diag(W) <- 0

# lambdas:
lambda1 = 1:2
lambda2 = 1:2

(cvn <- CVN::CVN(grid, W, lambda1 = lambda1, lambda2 = lambda2, eps = 1e-3, maxiter = 1000, verbose = TRUE))
```

`determine_information_criterion_cvn`

Information Criteria for a `cvn` *object*

Description

Determines a given information criteria for a cvn object, see `CVN`.

Usage

```
determine_information_criterion_cvn(cvn, type = c("AIC", "BIC"))
```

estimate *Estimate a CVN*

Description

A function for estimating a CVN for a single (λ_1, λ_2)-value. See for more details CVN

Usage

```
estimate(
  m,
  p,
  W,
  Theta0,
  Z0,
  Y0,
  a,
  eta1,
  eta2,
  Sigma,
  n_obs,
  rho,
  rho_genlasso,
  eps,
  eps_genlasso,
  maxiter,
  maxiter_genlasso,
  truncate,
  truncate_genlasso,
  verbose = FALSE
)
```

find_core_graph *The Core Graph*

Description

Finds the 'core graph', i.e., those edges that are present in all of the estimated graphs

Usage

```
find_core_graph(cvn)
```

Arguments

cvn	A cvn object

Value

A list of adjacency matrix, one for each value of (lambda1, lambda2)

---

genlasso_wrapper *Wrapper for* genlassoRcpp

Description

See for details genlassoRcpp

Usage

```
genlasso_wrapper(y, W, m, c, eta1, eta2, a, rho, max_iter, eps, truncate)
```

See Also

genlassoRcpp

---

glasso *Estimating Multiple Networks Separately*

Description

A wrapper for the GLASSO in the context of CVNs. Each graph is estimated individually. There is NO smoothing between the graphs. This function relies completely on the glasso package. The output is, therefore, slightly different than for the CVN function.

Usage

```
glasso(
  data,
  lambda1 = 1:2,
  eps = 1e-04,
  maxiter = 10000,
  n_cores = min(length(lambda1), parallel::detectCores() - 1),
  normalized = FALSE,
  verbose = TRUE
)
```

Arguments

data	A list with matrices, each entry associated with a single graph. The number of columns should be the same for each matrix. Number of observations can differ
lambda1	Vector with different λ_1 LASSO penalty terms (Default: 1:2)
eps	Threshold for convergence (Default: 1e-4; the same as in the glasso package)
maxiter	Maximum number of iterations (Default: 10,000)
n_cores	Number of cores used (Default: max. number of cores - 1, or the total number penalty term pairs if that is less)
normalized	Data is normalized if TRUE. Otherwise the data is only centered (Default: FALSE)
verbose	Verbose (Default: TRUE)

Value

A CVN object containing the estimates for all the graphs for different value of λ_1. General results for the different value of λ_1 can be found in the data frame results. It consists of multiple columns, namely:

lambda1	λ_1 value
value	value of the negative log-likelihood function
aic	Aikake information criteration
id	The id. This corresponds to the indices of the lists

The estimates of the precision matrices and the corresponding adjacency matrices for the different values of λ_1 can be found

Theta	A list with the estimated precision matrices $\{\hat{\Theta}_i(\lambda_1)\}_{i=1}^m$
adj_matrices	A list with the estimated adjacency matrices corresponding to the estimated precision matrices in Theta. The entries are 1 if there is an edge, 0 otherwise. The matrices are sparse using package Matrix

In addition, the input given to this function is stored in the object as well:

Sigma	Empirical covariance matrices $\{\hat{\Sigma}_i\}_{i=1}^m$
m	Number of graphs
p	Number of variables
n_obs	Vector of length m with number of observations for each graph
data	The data, but then normalized or centered
maxiter	Maximum number of iterations for the ADMM
eps	The stopping criterion ϵ
n_lambda_values	Total number of λ_1 values
normalized	If TRUE, data was normalized. Otherwise data was only centered

Examples

```
data(grid)
m <- 9 # must be 9 for this example

#' Choice of the weight matrix W.
#' (uniform random)
W <- matrix(runif(m*m), ncol = m)
W <- W %*% t(W)
W <- W / max(W)
diag(W) <- 0

# lambdas:
lambda1 = 1:4

(glasso_est <- CVN::glasso(grid, lambda1 = lambda1))
```

---

grid — *Data for a grid of graphs (3 x 3)*

Description

Data generated for 9 graphs in total, organized in a grid of (3x3). See the package `CVNSim` for more information on how the grid is constructed: https://github.com/bips-hb/CVNSim

Usage

```
data(grid)
```

Format

List

References

https://github.com/bips-hb/CVNSim

---

hamming_distance — *Structural Hamming Distance for a* `cvn` *Object*

Description

Returns the structural Hamming distances

Usage

```
hamming_distance(cvn, verbose = TRUE)
```

Arguments

`cvn`	A `cvn` or `cvn:glasso` object created by either the `CVN::CVN` or the `CVN::glasso` function
`verbose`	If `TRUE`, shows a progress bar

Value

A list of symmetric matrices. Each matrix contains the structural Hamming distances between the different graphs. Each item in the list corresponds to one (λ_1, λ_2) pair

---

`hamming_distance_adj_matrices` *Structural Hamming Distance*

Description

Returns the structural Hamming distance between multiple graphs

Usage

```
hamming_distance_adj_matrices(adj_matrices)
```

Arguments

`adj_matrices`	A list of adjacency matrices

Value

Matrix of Hamming distances

`hits_end_lambda_intervals` *Hitting the end points of* (λ_1, λ_2) *interval*

Description

One often selected the optimal model for the (λ_1, λ_2)-values based on the AIC and BIC. This function checks and warns when the optimal value lies on the border of the values (λ_1, λ_2) takes.

Usage

```
hits_end_lambda_intervals(results)
```

Arguments

`results`	Results of the `CVN` function

Value

List with two values:

`hits_border_aic`
: If `TRUE`, hits the border for the AIC

`hits_border_bic`
: If `TRUE`, hits the border for the BIC

`interpolate` *Interpolation of a Graph*

Description

Estimates a graph for which there are no observation based on a previously fitted CVN model

Usage

```
interpolate(cvn, weights, truncate = NULL)
```

Arguments

`cvn`	A CVN fit with m graphs
`weights`	A vector of length m with the regression coefficients
`truncate`	Truncation value. When a value in the precision matrix is considered 0. If `NULL`, the same truncation is used as for the fitted CVN model (Default)

Value

A list with

`adj_matrices`	A list of adjacency matrix. One for each pair of (λ_1, λ_2) values. The entries are `1` if there is an edge, `0` otherwise. The matrices are sparse using package `Matrix`
`m`	Number of graphs
`p`	Number of variables
`weights`	The weights used for interpolation
`truncate`	Truncation value
`n_lambda_values`	Total number of (λ_1, λ_2) value combinations

`results`. It consists of two columns:

`lambda1`	λ_1 value
`lambda2`	λ_2 value

matrix_A_inner_ADMM — *Determine matrix A for inner-ADMM for the Z-update step*

Description

The Z-update step, see updateZ, requires us to solve a special Generalized LASSO problem of the form

$$\hat{\beta} = \text{argmin } \frac{1}{2}||y - \beta||_2^2 + ||D\beta||_1$$

where β and y are m-dimensional vectors and D is a $(c \times m)$-matrix where $c = (m^2 + m)/2$. We solve this optimization problem using an adaption of the ADMM algorithm presented in Zhu (2017). This algorithm requires the choice of a matrix A such that $A - D'D$ is positive semidefinite. In order to optimize the ADMM, we choose the matrix A to be diagonal with a fixed value a. This function determines the smallest value of a such that $A - D'D$ is indeed positive semidefinite. We do this be determining the largest eigenvalue

Usage

```
matrix_A_inner_ADMM(W, eta1, eta2)
```

Arguments

W	Weight matrix W
eta1, eta2	The values $\eta_1 = \lambda_1/\rho$ and $\eta_2 = \lambda_2/\rho$

Value

Value of a

References

Zhu, Y. (2017). An Augmented ADMM Algorithm With Application to the Generalized Lasso Problem. Journal of Computational and Graphical Statistics, 26(1), 195–204. https://doi.org/10.1080/10618600.2015.111449

plot.cvn — *Plot Function for CVN Object Class*

Description

Plot Function for CVN Object Class

Usage

```
## S3 method for class 'cvn'
plot(cvn, ...)
```

plot_hamming_distances *Heat Map of the Distances between Graphs*

Description

Returns a heat map of the distance matrix for a particular CVN

Usage

```
plot_hamming_distances(
  distance_matrix,
  absolute = TRUE,
  limits = c(NA, NA),
  title = "",
  legend_label = "Hamming Distance",
  add_counts_to_cells = TRUE,
  add_ticks_labels = TRUE,
  t = -6,
  r = -8
)
```

Arguments

`distance_matrix`	Symmetric matrix with distances
`absolute`	If `FALSE`, rescaled to [0,1]
`limits`	The limits for the values of the Hamming distance
`title`	Title plot (Default is none)
`legend_label`	Title of the legend (Default: "Hamming Distance")
`add_counts_to_cells`	If `TRUE`, counts from the matrix are added to the plot (Default: `TRUE`)
`add_ticks_labels`	If `TRUE`, the number corresponding to the graph is add to the plot (Default: `TRUE`)
`t`	Distance between tick labels and x-axis (Default: -6)
`r`	Distance between tick labels and y-axis (Default: -8)

Value

A heatmap plot

`plot_hamming_distances_cvn`
Heatmaps for a CVN

Description

Creates all the heatmaps for a CVN, a heatmap for each pair of (λ_1, λ_2)

Usage

```
plot_hamming_distances_cvn(
  cvn,
  absolute = TRUE,
  same_range = TRUE,
  titles = rep("", cvn$n_lambda_values),
  legend_label = "Hamming Distance",
  add_counts_to_cells = TRUE,
  add_ticks_labels = TRUE,
  t = -6,
  r = -8,
  verbose = TRUE
)
```

Arguments

`cvn`	A cvn object
`absolute`	If `FALSE`, rescaled to [0,1]
`same_range`	If `TRUE`, all heatmaps have the same range of values of the Hamming distance shown (Default: TRUE)
`titles`	Title of the plots (Default is none)
`legend_label`	Title of the legend (Default: "Hamming Distance")
`add_counts_to_cells`	If `TRUE`, counts from the matrix are added to the plot (Default: `TRUE`)
`add_ticks_labels`	If `TRUE`, the number corresponding to the graph is add to the plot (Default: `TRUE`)
`t`	Distance between tick labels and x-axis (Default: -6)
`r`	Distance between tick labels and y-axis (Default: -8)
`verbose`	If `TRUE`, shows progress bar (Default: `TRUE`)

Value

List of plots

plot_information_criterion *Heat Map of an Information Criterion (AIC or BIC)*

Description

Returns a heat map of the AIC or BIC for a fitted CVN

Usage

```
plot_information_criterion(
  cvn,
  criterion = c("bic", "aic"),
  use_gammas = TRUE,
  show_minimum = TRUE,
  title = "",
  xlabel = NULL,
  ylabel = NULL,
  legend_label = NULL,
  limits = c(NA, NA)
)
```

Arguments

`cvn`	Fitted CVN, see `CVN`
`criterion`	The information criterion, must be either `'aic'` or `'bic'`. Default: `'bic'`
`use_gammas`	If `TRUE`, plots the γ-values. Otherwise, the λ-values are used
`show_minimum`	If `TRUE`, an orange dot is put on the point with the minimum value of the information criterion is. If `FALSE`, no dot is added. Default: `TRUE`.
`title`	Title plot (Default is none)
`xlabel`	Label for the x-axis. Default depends on `use_gammas`. If `use_gammas = TRUE`, then the label is 'gamma1'. Otherwise, 'lambda1'
`ylabel`	Label for the x-axis. Default depends on `use_gammas`. If `use_gammas = TRUE`, then the label is 'gamma1'. Otherwise, 'lambda1'
`legend_label`	Title for the legend. Default depends on `criterion`. If `'aic'`, then the label is 'AIC'. Otherwise, 'BIC'.
`limits`	The limits for the values of the Hamming distance

Value

A heatmap plot

plot_weight_matrix *Plot Weight Matrix*

Description

Returns a heat map of a weight matrix

Usage

```
plot_weight_matrix(
  W,
  title = "",
  legend_label = "weight",
  add_counts_to_cells = TRUE,
  add_ticks_labels = TRUE,
  t = -6,
  r = -8
)
```

Arguments

`W`	Symmetric weight matrix
`title`	Title plot (Default is none)
`legend_label`	Title of the legend (Default: "weight")
`add_counts_to_cells`	If `TRUE`, counts from the matrix are added to the plot (Default: `TRUE`)
`add_ticks_labels`	If `TRUE`, the number corresponding to the graph is add to the plot (Default: `TRUE`)
`t`	Distance between tick labels and x-axis (Default: -6)
`r`	Distance between tick labels and y-axis (Default: -8)

Value

A heatmap plot

print.cvn *Print Function for the CVN Object Class*

Description

Print Function for the CVN Object Class

Usage

```
## S3 method for class 'cvn'
print(cvn, ...)
```

`relative_difference_precision_matrices`
The Relative Difference between two Precision Matrices

Description

Returns the relative $L1$ difference between precision matrix $\Theta(k+1)$ (parameter `Theta_new`) and $\Theta(k)$ (parameter `Theta_old`).

Usage

```
relative_difference_precision_matrices(Theta_new, Theta_old)
```

Arguments

`Theta_new`	A list with matrices with the updated values of Θ
`Theta_old`	A list with matrices with the old values of Θ

Details

This is used for checking whether the stopping condition has been met.

Value

The relative difference between $\Theta(k+1)$ and $\Theta(k)$

`set_attributes_to_edges_visnetwork`
Add Attributes to Subset of Edges for `visNetwork`

Description

A subset of edges can be assign a different thickness or color.

Usage

```
set_attributes_to_edges_visnetwork(
  edges,
  subset_edges,
  width = c(NA, NA),
  color = c(NULL, NULL)
)
```

Arguments

edges	A data.frame create by create_edges_visnetwork
subset_edges	A list with the elements from and to. Both from and to are vectors of the same length denoting the different edges
width	Vector with two values. The first is assigned to the edges in the subset given by subset_edges. The second value is assigned to the rest. If width = c(NA,NA), no width is assigned
color	Vector with two values. The first is assigned to the edges in the subset given by subset_edges. The second value is assigned to the rest. If color = c(NULL,NULL), no color is assigned

Value

A data frame that can be used by the visNetwork package

---

strip_cvn *Strip CVN*

Description

Function that removes most of the items to make the CVN object more memory sufficient. This is especially important when the graphs are rather larger

Usage

```
strip_cvn(cvn)
```

Arguments

cvn	cvn object

Value

Reduced cvn where Theta, data and Sigma are removed

updateTheta *The Θ-update step for the ADMM*

Description

`updateTheta` returns the updated value of Θ for the ADMM given the previously updated values of Z and Y

Usage

```
updateTheta(m, Z, Y, Sigma, n_obs, rho = 1)
```

Arguments

`m`	Number of graphs
`Z`	A list with matrices with the current values of Z
`Y`	A list with matrices with the current values of Y
`Sigma`	A list with empirical covariance matrices Σ
`n_obs`	A m-dimensional vector with the number of observations per graph
`rho`	The ρ ADMM's penalty parameter (Default: 1)

Value

A list with matrices with the new values of Θ

updateY *The Y-update step for the ADMM*

Description

Returns the updated value of Y for the ADMM given the previously updated values of Θ and Z

Usage

```
updateY(Theta, Z, Y)
```

Arguments

`Theta`	A list with matrices with the current values of Θ
`Z`	A list with matrices with the current values of Z
`Y`	A list with matrices with the current values of Y

Value

A list with matrices with the new values of Y

updateZRcpp *The Z-update Step*

Description

A C implementation of the Z-update step. We solve a generalized LASSO problem repeatedly for each of the individual edges

Usage

```
updateZRcpp(m, p, Theta, Y, W, eta1, eta2, a, rho, max_iter, eps, truncate)
```

Arguments

m	The number of graphs
p	The number of variables
Theta	A list of matrices with the Θ-matrices
Y	A list of matrices with the Y-matrices
eta1	Equals λ_1/rho
eta2	Equals λ_2/rho
a	Value added to the diagonal of $-D'D$ so that the matrix is positive definite, see matrix_A_inner_ADMM
rho	The ADMM's parameter
max_iter	Maximum number of iterations
eps	Stopping criterion. If differences are smaller than ϵ, algorithm is halted
truncate	Values below truncate are set to 0

Value

The estimated vector $\hat{\beta}$

See Also

updateZ_wrapper

updateZ_wrapper *Wrapper for the Z-update Step for the ADMM*

Description

A wrapper for the C function that returns the updated value of Z for the ADMM given the previously updated values of Θ and Y

Usage

```
updateZ_wrapper(
  m,
  p,
  nrow_D,
  Theta,
  Y,
  W,
  eta1,
  eta2,
  a,
  rho_genlasso,
  maxiter_genlasso,
  eps_genlasso,
  truncate_genlasso
)
```

Arguments

m	Number of graphs
p	Number of variables
nrow_D	Number of rows of the D-matrix
Theta	A list with matrices with the current values of Θ
Y	A list with matrices with the current values of Y
W	Weight matrix
eta1	
rho_genlasso	The ρ penalty parameter for the ADMM algorithm
maxiter_genlasso	Maximum number of iterations for solving the generalized LASSO problem
eps_genlasso	If the relative difference between two update steps is smaller than ϵ, the algorithm stops
truncate_genlasso	All values of the final $\hat{\beta}$ below truncate_genlasso will be set to 0.

Value

A list with matrices with the new values of Z

See Also

`create_matrix_D`

---

visnetwork — *A* `visNetwork` *plot*

Description

Creates a `visNetwork` plot given a list of nodes and edges. The nodes data frame can be created with `create_nodes_visnetwork`; the edges with `create_edges_visnetwork`. In order to highlight edges, you can use `set_attributes_to_edges_visnetwork`.

Usage

```
visnetwork(
  nodes,
  edges,
  node_titles = nodes$id,
  title = "",
  igraph_layout = "layout_in_circle"
)
```

Value

A `visNetwork` plot

Examples

```
nodes <- CVN::create_nodes_visnetwork(n_nodes = 5, labels = LETTERS[1:5])

adj_matrix <- matrix(c(0, 1, 0, 1, 0,
                       1, 0, 1, 0, 0,
                       0, 1, 0, 0, 0,
                       1, 0, 0, 0, 1,
                       0, 0, 0, 1, 0), ncol = 5)

edges <- CVN::create_edges_visnetwork(adj_matrix)

edges <- set_attributes_to_edges_visnetwork(edges,
                                      subset_edges = list(from = c(1, 2), to = c(4, 3)),
                                           width = c(3, .5),
                                           color = c("red", "blue"))

CVN::visnetwork(nodes, edges)
```

visnetwork_cvn *All visNetwork plots for a CVN object*

Description

Creates all visNetwork plots, see CVN::visnetwork, for all graphs in a cvn object

Usage

```
visnetwork_cvn(
  cvn,
  node_titles = 1:cvn$p,
  titles = lapply(1:cvn$n_lambda_values, function(i) sapply(1:cvn$m, function(j) "")),
  show_core_graph = TRUE,
  width = c(3, 1),
  color = c("red", "blue"),
  igraph_layout = "layout_in_circle",
  verbose = TRUE
)
```

Arguments

cvn	A cvn object, see CVN::CVN or CVN::glasso
node_titles	Vector with title of the nodes (Default: 1:p)
titles	A list with n_lambda_values vectors. Each vector is of the lenght m. Regulates the titles of the graphs (Default: no title)
show_core_graph, width, color	Show the core graph using the width and colors

Value

List

Index