Package 'dineR'

October 13, 2022

Title Differential Network Estimation in R

Version 1.0.1

Description An efficient and convenient set of functions to perform differential network estimation through the use of alternating direction method of multipliers optimization with a variety of loss functions.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

URL https://github.com/RicSalgado/dineR

BugReports https://github.com/RicSalgado/dineR/issues

Imports MASS, progress, Matrix

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Ricardo Daniel Marques Salgado [aut, cre]
(<https://orcid.org/0000-0002-6415-7379>),
Andriette Bekker [ctb],
Mohammad Arashi [ctb]

Maintainer Ricardo Daniel Marques Salgado <ricardodansalgado@gmail.com>

Repository CRAN

Date/Publication 2021-11-15 09:20:12 UTC

R topics documented:

data_generator *Data Generator*

Description

This functions generates two n by p size samples of multivariate normal data. In doing this it also determines and provides the relevant covariance matrices.

Usage

```
data_generator(n, p, Delta = NULL, case = "sparse", seed = NULL)
```

Arguments

n	The number of observations generated.
p	The number of dimensions for the generated samples.
Delta	Optional parameter - Provides the differential network that will be used to obtain the sample covariance matrices.
case	Optional parameter - Selects under which case the covariance matrices are determined. Possible cases are: "sparse" - Sparse Case or "asymsparse"- Asymptotically Sparse Case. Defaults to "sparse".
seed	Optional parameter - Allows a seed to be set for reproducibility.

Value

A list of various outputs, namely:

- case - The case used.
- seed_option - The seed provided.
- X - The first multivariate normal sample.
- Y - The second multivariate normal sample.
- Sigma_X - The covariance matrix of X.
- Sigma_Y - The covariance matrix of Y.
- Omega_X - The precision matrix of X.
- Omega_Y - The precision matrix of Y.
- diff_Omega - The difference of precision matrices.
- Delta - The target differential network.

Examples

```
data <- data_generator(n = 100, p = 50, seed = 123)
data <- data_generator(n = 10, p = 50, case = "asymsparse")
```

`estimation` *Estimation*

Description

This function performs alternating direction method of multipliers optimization for a variety of loss functions to estimate the differential network given two samples of multivariate normal data.

Usage

```
estimation(
  X,
  Y,
  lambdas = NULL,
  lambda_min_ratio = 0.3,
  nlambda = 10,
  a = NULL,
  loss = "lasso",
  tuning = "none",
  perturb = FALSE,
  stop_tol = 1e-05,
  max_iter = 500,
  correlation = FALSE,
  Delta_init = NULL,
  rho = NULL,
  gamma = NULL,
  verbose = FALSE
)
```

Arguments

`X`	The first multivariate normal sample.
`Y`	The second multivariate normal sample.
`lambdas`	Optional parameter - A list of the regularization values to be used within the loss functions.
`lambda_min_ratio`	Optional parameter - Defines the smallest regularization values as this proportion of the largest regularization value. Defaults to 0.3.
`nlambda`	Optional parameter - The number of regularization values considered. Defaults to 10.
`a`	Optional parameter - The thresholding parameter used in SCAD and MCP loss functions. Defaults to 3.7 with SCAD, and 3 with MCP respectively.
`loss`	Optional parameter - The loss function of choice to implement. The function allows for four choices, namely "lasso", "scad", "mcp" and "d-trace". Defaults to "lasso".

`tuning`	Optional parameter - The tuning method selected to determine the optimal value for the regularization parameter. Options are "none", "AIC", "BIC" and "EBIC". Defaults to "none".
`perturb`	Optional parameter - When set to TRUE perturbation as done by the CLIME software to improve performance is implemented. Options are TRUE or FALSE, with the function defaulting to FALSE.
`stop_tol`	Optional parameter - The stop tolerance to determine whether convergence has occurred. Defaults to 1e-5.
`max_iter`	Optional parameter - The maximum number of iterations that can be perform for any one regularization value. Defaults to 100.
`correlation`	Optional parameter - Determines whether the sample correlation matrices should be used in the place of the sample covariance matrices. Choices are TRUE and FALSE with the function defaulting to FALSE.
`Delta_init`	Optional parameter - Allows for the algorithm to provided an initial estimate of the differential network to ease computation.
`rho`	Optional parameter - Allows the user to adjust the ADMM step-size. Defaults to 1.
`gamma`	Optional parameter - Allows the user to adjust the EBIC value when EBIC is the selected tuning method. Defaults to 0.5.
`verbose`	Optional parameter - Allows the user to obtain a summary of the estimation results. Options are TRUE or FALSE, where FALSE indicates the summary is not provided. Defaults to FALSE.

Value

A list of various outputs, namely:

- n_X - The number of observations in X.
- n_Y - The number of observations in Y.
- Sigma_X - The covariance matrix of X.
- Sigma_Y - The covariance matrix of Y.
- loss - The loss function implemented.
- tuning - The tuning method utilized.
- lip - The value of the lipschitz constant.
- iter - The iterations until convergence for each of the regularization values.
- elapse - The total system time (in seconds) elapsed from initialization to completion of the optimization.
- lambdas - The regularization parameter values used.
- sparsity - The level of sparsity of the differential network for each regularization value.
- path - The set of all differential networks for all regularization values considered.
- ic - The output obtained from any possible tuning.
- ic_index - The index at which the tuning is optimized.

- ic_value - The tuning method optimal value.
- chosen_lambda_ic - The regularization value that occurs at **ic_index**.
- loss_index - The index at which the loss function is optimized.
- loss_value - The loss function optimal value.
- chosen_lambda_loss - The regularization value that occurs at **loss_index**.

Examples

```
data <- data_generator(n = 100, p = 50, seed = 123)
X <- data$X
Y <- data$Y
result <- estimation(X,Y)
```

---

npn *NPN - Non paranormal Transformation*

Description

This functions allows us to transform non-normal multivariate data to that of non paranormal data.

Usage

```
npn(x, npn_func = "shrinkage", npn_thresh = NULL, verbose = TRUE)
```

Arguments

x	The multivariate non-normal data to be transformed.
npn_func	Optional parameter - The method of transformation to be applied. Can either be "shrinkage" or "truncation" but defaults to "shrinkage".
npn_thresh	Optional parameter - The truncation threshold that is used when making use of truncation.
verbose	Optional parameter - Prints additional output of the selected approach. Can either be "TRUE" or "FALSE" and defaults to "TRUE".

Value

Returns the transformed data matrix.

Examples

```
data <- data_generator(n = 100, p = 50, seed = 123)
X <- data$X
X_transformed <- npn(X, npn_func = "truncation")
```

Index