Package 'rope'

July 23, 2025

Title Model Selection with FDR Control of Selected Variables

Version 1.0

Author Jonatan Kallus [aut, cre]

Maintainer Jonatan Kallus <kallus@chalmers.se>

Description Selects one model with variable selection FDR controlled at a specified level. A q-value for each potential variable is also returned. The input, variable selection counts over many bootstraps for several levels of penalization, is modeled as coming from a beta-binomial mixture distribution.

Depends R (>= 3.1.0)

Suggests Matrix, parallel, knitr, rmarkdown

License GPL-3

LazyData true

VignetteBuilder knitr

RoxygenNote 6.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-16 07:55:41

Contents

explore *Run first step of model fitting to find good penalization interval*

Description

Run first step of model fitting to find good penalization interval

Usage

```
explore(data, B, mc.cores = getOption("mc.cores", 2L))
```

Arguments

data	Matrix of variable presence counts. One column for each variable, one row for each parameter value (e.g. levels of regularization).
B	Number of bootstraps used to construct data. At least 21 are needed for u-shape test heuristic to work, but in general it is recommended to use many more.
mc.cores	Number of threads to run in parallel (1 turns of parallelization)

Value

A list with components

pop.sep	vector of values saying how separated true and false variables are for each level of penalization

exploregraph *Convenience wrapper for* explore *for adjacency matrices*

Description

When modeling graphs it may be more convenient to store data as matrices instead of row vectors.

Usage

```
exploregraph(data, B, ...)
```

Arguments

data	List of symmetric matrices, one matrix for each penalization level
B	Number of bootstraps used to construct data. At least 21 are needed for u-shape test heuristic to work, but in general it is recommended to use many more.
...	Additional arguments are passed on to explore.

Value

A list with components

pop.sep	vector of values saying how separated true and false variables are for each level of penalization

---

plotrope *Plot rope results*

Description

Plot rope results

Usage

```
plotrope(result, data, types = c("global"), ...)
```

Arguments

result	An object returned by rope or explore
data	Matrix of variable presence counts. One column for each variable, one row for each parameter value (e.g. levels of regularization).
types	List of names of plots to draw (alternatives 'global', 'q-values' or 'fits')
...	Pass level=v for a vector v of indices when drawing the fits plot to only plot for penalization levels corresponding to v

---

rope *FDR controlled model selection*

Description

Estimates a model from bootstap counts. The objective is to maximize accuracy while controlling the false discovery rate of selected variables. Developed for high-dimensional models with number of variables in the order of at least 10000.

Usage

```
rope(data, B, fdr = 0.1, mc.cores = getOption("mc.cores", 2L),
  only.first = FALSE)
```

Arguments

data	Matrix of variable presence counts. One column for each variable, one row for each parameter value (e.g. levels of regularization).
B	Number of bootstraps used to construct data. At least 21 are needed for u-shape test heuristic to work, but in general it is recommended to use many more.
fdr	Vector of target false discovery rates to return selections for
mc.cores	Number of threads to run in parallel (1 turns of parallelization)
only.first	Skip second part of algorithm. Saves time but gives worse results.

Value

A list with components

selection	matrix (one row for each fdr target, one column for each variable)
q	vector of q-values, one for each variable
level	index of most separating parameter value
alt.prop	estimated proportion of alternative variables

Author(s)

Jonatan Kallus, <kallus@chalmers.se>

Examples

```
## Not run:
data # a matrix of selection counts, for 100 bootstraps, with ncol(data)
  # potential variables counted for nrow(data) different penalization levels
fdr <- c(0.05, 0.1)
result <- rope(data, 100, fdr)

## End(Not run)
```

---

ropegraph *Convenience wrapper for* rope *for adjacency matrices*

Description

When modeling graphs it may be more convenient to store data as matrices instead of row vectors.

Usage

```
ropegraph(data, B, ...)
```

Arguments

data	List of symmetric matrices, one matrix for each penalization level
B	Number of bootstraps used to construct data. At least 21 are needed for u-shape test heuristic to work, but in general it is recommended to use many more.
...	Additional arguments are passed on to rope.

Value

A list with components

selection	list of symmetric matrices, one matrix for each fdr target
q	symmetric matrix of q-values
level	index of most separating parameter value
alt.prop	estimated proportion of alternative variables

Examples

```
## Not run:
data # a list of symmetric matrices, one matrix for each penalization level,
  # each matrix containing selection counts for each edge over 100 bootstraps
fdr <- c(0.05, 0.1)
result <- rope(data, 100, fdr)

## End(Not run)
```

---

scalefree *A simulated data set for a scale-free network of 200 nodes*

Description

The data set contains 175 observations for each node, the true network structure dat was used to generate data and edge presence counts from glasso over 100 bootstraps.

Usage

```
scalefree
```

Format

A list containing:

x A matrix of 175 observations (rows) for 200 variabels (columns)

g The generating network structure (as a vector)

B 100, the number of bootstraps used when counting edge presence

lambda The range of penalization used for glasso (the first 9 generate U-shaped histograms)

W A matrix of length(lambda) rows and 200*199/2 columns containing presence counts for each edge and each level of penalization

Wlist A list of length(lamdba) containing matrices of size 200 by 200, the data in W but in an alternative format

gmatrix A 200 by 200 matrix, the data in g but in an alternative format

---

`symmetric.matrix2vector` *Take upper half of matrix and convert it to a vector*

Description

If variable selection counts are in a matrix this function converts them into vector to input into rope. Can be useful when variables correspond to edges in a graph.

Usage

```
symmetric.matrix2vector(m)
```

Arguments

`m`	A symmetric matrix

`vector2symmetric.matrix` *Convert vector that represents half of a symmetric matrix into a matrix*

Description

This can be convenient for using output when rope is used for selection of graph models.

Usage

```
vector2symmetric.matrix(v)
```

Arguments

`v`	A vector with length p*(p-1)/2 for some integer p

Index