Package ‘networktree’

March 9, 2019

Title  Recursive Partitioning of Network Models
Version  0.2.1
Date    2019-03-09
Description  Methods to create tree models with correlation-based network models (multivariate normal distributions).
Depends    R (>= 3.0.0)
License    GPL-3
Encoding   UTF-8
LazyData   true
Imports    partykit, qgraph, stats, utils, Matrix, mvtnorm, Formula, grid, graphics, gridBase
RoxygenNote 6.1.1
Suggests   R.rsp, knitr, rmarkdown, fxregime, zoo
URL        NA
BugReports http://github.com/paytonjones/networktree/issues
NeedsCompilation no
Author      Payton Jones [aut, cre] (<https://orcid.org/0000-0001-6513-8498>),
            Thorsten Simon [aut] (<https://orcid.org/0000-0002-3778-7738>),
            Achim Zeileis [aut] (<https://orcid.org/0000-0003-0918-3766>)
Maintainer  Payton Jones <payton_jones@g.harvard.edu>
Repository   CRAN
Date/Publication 2019-03-09 13:10:03 UTC

R topics documented:

getnetwork ................................................................. 2
mvnfit ................................................................. 2
networktree ............................................................ 3
plot.networktree ...................................................... 5
workaholic ............................................................. 5

Index  7
getnetwork

Description
Easily extract a network from one of the nodes in a networktree object

Usage
getnetwork(tree, id = 1L, type = "detect", ...)

Arguments
- tree: a networktree object
- id: the node in the tree to extract. Use summary(tree) to see id numbers for each split
- type: "cor", "pcor", or "glasso". Defaults to automatic detection
- ...: arguments passed to qgraph (e.g., "tuning", "threshold")

Examples

```r
setNseed(1)
d <- data.frame(trend = 1:200, foo = runif(200, -1, 1))
d <- cbind(d, rbind(
  mvtnorm::rmvnorm(100, mean = c(0, 0, 0)),
  sigma = matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), ncol = 3)),
  mvtnorm::rmvnorm(100, mean = c(0, 0, 0)),
  sigma = matrix(c(1, 0, 0.5, 0, 1, 0.5, 0.5, 0.5, 1), ncol = 3))
)
colnames(d)[3:5] <- paste0("y", 1:3)

## Now use the function
tree1 <- networktree(nodevars=d[,3:5], splitvars=d[,1:2])

getnetwork(tree1, id=1)
```

mvnfit

Maximum Likelihood Estimation for Multivariate Normal Model

Description
Fit a multivariate normal model without covariates or covariance restrictions. In addition to the (straightforward) parameter estimates the fitted log-likelihood and corresponding score contributions are computed.
Usage

```r
mvnfit(y, x = NULL, start = NULL, weights = NULL, offset = NULL, 
model = c("correlation", "mean", "variance"), ..., estfun = FALSE, 
object = FALSE)
```

Arguments

- `y`: A matrix or data.frame where each row corresponds to a k-dim observation.
- `x`: Not used yet
- `start`: Not used yet
- `weights`: Not used yet
- `offset`: Not used yet
- `model`: Vector of characters. Specifies which estimated parameters are returned.
- `...`: Not used yet
- `estfun`: Logical. Should the matrix of score contributions (aka estimating functions) be returned?
- `object`: Not used yet

Details

Used internally in when method="mob"

Description

Computes a tree model with networks at the end of branches. Can use model-based recursive partitioning or conditional inference.

Wraps the mob() and ctree() functions from the partykit package.

Note: this package is in its early stages and the interface may change for future versions.

Usage

```r
networktree(...)  
```

## Default S3 method:
```r
networktree(nodevars, splitvars, type = c("cor", 
  "pcor", "glasso"), method = c("mob", "ctree"), model = "correlation", 
na.action = na.pass, weights = NULL, ...)
```

## S3 method for class 'formula'
```r
networktree(formula, data, type = c("cor", "pcor", 
  "glasso"), method = c("mob", "ctree"), na.action = na.pass, 
model = "correlation", ...)```
Arguments

... additional arguments passed to mob_control (mob) or ctree_control (ctree)
nodevars the variables with which to compute the network. Can be vector, matrix, or dataframe
splitvars the variables with which to test split the network. Can be vector, matrix, or dataframe
type the type of network to compute. Can be "cor", "pcor", or "glasso". Note that networks are always stored internally as correlation matrices, but will be auto-adjusted in plots etc. according to type
method "mob" or "ctree"
model can be any combination of c("correlation", "mean", "variance") splits are determined based on the specified characteristics
na.action a function which indicates what should happen when the data contain missing values (NAs).
weights weights
formula A symbolic description of the model to be fit. This should either be of type y1 + y2 + y3 ~ x1 + x2 with node vectors y1, y2, and y3 or y ~ x1 + x2 with a matrix response y. x1 and x2 are used as partitioning variables.
data a data frame containing the variables in the model

Examples

set.seed(1)
d <- data.frame(trend = 1:200, foo = runif(200, -1, 1))
d <- cbind(d, rbind(
  mvtnorm::rmvnorm(100, mean = c(0, 0, 0)),
  sigma = matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), ncol = 3)),
  mvtnorm::rmvnorm(100, mean = c(0, 0, 0)),
  sigma = matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), ncol = 3))
)
colnames(d)[3:5] <- paste0("y", 1:3)

## Now use the function
tree1 <- networktree(nodevars=d[,3:5], splitvars=d[,1:2])

## Formula interface
tree2 <- networktree(y1 + y2 + y3 ~ trend + foo, data=d)

## Conditional version
tree3 <- networktree(nodevars=d[,3:5], splitvars=d[,1:2],
  method="ctree")

## Change control arguments
tree4 <- networktree(nodevars=d[,3:5], splitvars=d[,1:2],
  alpha=0.01)
**plot.networktree**

Plotting 'treenetwork' objects

---

**Description**

Wraps `plot.party` to plot a tree model with networks on the ends. Networks are plotted with qgraph, and additional arguments are passed there.

**Usage**

```r
## S3 method for class 'networktree'
plot(x, type = NULL, ...)
```

**Arguments**

- `x`: an object of type 'networktree'
- `type`: "cor", "pcor", or "glasso". If set to NULL, type detected from `x`
- `...`: additional arguments passed to `qgraph`

---

**workaholic**

Workaholism and Psychiatric Symptoms

---

**Description**

This dataset includes 16,426 workers who were assessed on symptoms of psychiatric disorders (ADHD, OCD, anxiety, depression) and workaholism.

**Usage**

`workaholic`

**Format**

a dataframe. Columns represent symptoms and rows represent individuals.

**Details**

Scales: Adult ADHD Self-Report Scale, Obsession-Compulsive Inventory-Revised, Hospital Anxiety and Depression Scale, and the Bergen Work Addiction Scale.

Also includes demographics such as age, gender, work status, position, sector, annual income.

The dataset is publicly available at https://doi.org/10.1371/journal.pone.0152978 and can be cited as:

Examples

head(workaholic)

## Example networktree with OCI-R scale

data(workaholic)
nodeVars <- paste("OCIR",1:18,sep="")
splitVars <- c("Workaholism_diagnosis","Gender")
myTree<-networktree(workaholic[,nodeVars], workaholic[,splitVars])
myTree
plot(myTree)
Index

*Topic datasets
  workaholic, 5

ctree_control, 4
getnetwork, 2
mob_control, 4
mvnfit, 2
networktree, 3
plot.networktree, 5
workaholic, 5