Package: cmfilter (via r-universe)

September 3, 2024

Type Package

Title Coordinate-Wise Mediation Filter

Version 0.9.5

Date 2023-03-30

Author Erik-Jan van Kesteren <e.vankesteren1@uu.nl>

Maintainer Erik-Jan van Kesteren <e.vankesteren1@uu.nl>

Description Functions to discover, plot, and select multiple mediators from an x -> M -> y linear system. This exploratory mediation analysis is performed using the Coordinate-wise Mediation Filter as introduced by Van Kesteren and Oberski (2019) <doi:10.1080/10705511.2019.1588124>.

License MIT + file LICENCE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 2.10)

Imports Matrix, sparseMVN, pbapply, MASS, parallel, Rcpp (>= 0.12.13)

LinkingTo Rcpp, RcppArmadillo, RcppProgress

Suggests testthat

Repository https://vankesteren.r-universe.dev

RemoteUrl https://github.com/vankesteren/cmfilter

RemoteRef HEAD

RemoteSha c789334c11c95d6712af9c5cb85df38d863a634f

Contents

`+.cmf` *Combine the results of multiple cmf objects into one*

Description

This function combines two cmf objects and returns one cmf object with the combined results. This helps with combining results done over multiple runs, for example in high-performance computing.

Usage

```
## S3 method for class 'cmf'
x + y
```

Arguments

`x`	a cmf object
`y`	a cmf object

Value

a cmf object with combined results

Examples

```
# generate some data
dat <- generateMed(a = (1:10)/20, b = (1:10)/20)
# create two different cmf objects on this data
res_1 <- cmf(dat, nStarts = 500)
res_2 <- cmf(dat, nStarts = 500)
# Combine the results using the + operator
res_1 + res_2
```

`cmf` *Coordinate-wise Mediation Filter*

Description

This function performs CMF on a set of potential mediators, given an input and an output variable. It selects those mediators that are deemed relevant by a default or a user-defined decision function, *conditional* on the other mediators in the model. By doing this cyclically, and with multiple random starts, the algorithm outputs an estimate of the best mediators.

Usage

```
cmf(
  x,
  M,
  y,
  decisionFunction = "prodcoef",
  nStarts = 1000,
  nCores = 2,
  cutoff = 0.5,
  maxIter = 25,
  stableLag = 5,
  pb = TRUE,
  ...
)
```

Arguments

x	exogenous variable; numeric vector or data frame with x, y, and at least one M column
M	potential mediators; data frame with column names
y	outcome variable; numeric vector
decisionFunction	either a function with as inputs x, m, y, parameters, and as output a TRUE (include) or FALSE (exclude) statement or a string indicating the built-in decision function to use (see details)
nStarts	how many times to start the algorithm
nCores	how many threads (cores) to use for parallel processing (default 2)
cutoff	a cutoff value for selection: variables are selected if they display a selection rate higher than this value. Only relevant when multiple starts are specified. Can also be specified post-hoc using `setCutoff`.
maxIter	the maximum number of iterations for each start
stableLag	how long does the selection need to be stable before deciding the algorithm has converged
pb	Whether to display a progress bar (default TRUE). Only available with built-in decision functions
...	parameters passed to decisionFunction

Details

Available decision functions. These functions are implemented in C++ to speed up computation. Between brackets the additional parameter that may be passed to the function in the ... argument of this function. (`arguments = defaultvalue`):

- `prodcoef` (`p.value = 0.1`): Test for the product of coefficients, Sobel test

- `causalsteps` (`p.value = 0.1`): Causal steps test min(Ta, Tb)

Value

an object of class cmf. See cmf-methods

Examples

```
# generate some data
dat <- generateMed(a = (1:10)/20, b = (1:10)/20)
# Run CMF on this data
cmf(dat)
```

---

cmf-methods *Methods for cmf results lists*

Description

Plotting, summarising, and printing cmf objects

Usage

```
## S3 method for class 'cmf'
plot(
  x,
  select,
  line = TRUE,
  labelSelected = TRUE,
  defaultColour = "#00008b",
  highlightColour = "#e2bd36",
  ...
)

## S3 method for class 'cmf'
screeplot(x, topn, ...)

## S3 method for class 'cmf'
summary(object, ...)

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

Arguments

x	cmf object
select	optional selection vector of variables to show
line	whether to show a line at the chosen cutoff
labelSelected	whether to label the selected mediators in the plot, not used when fewer than 20 bars are shown.

defaultColour	the colour for the bars lower than the cutoff
highlightColour	the colour for the bars of the selected mediators
...	other arguments passed to barplot and summary
topn	only show the top n mediators
object	cmf object

See Also

cmf

Examples

```
# generate some data
dat <- generateMed(a = (1:10)/20, b = (1:10)/20)
res <- cmf(dat)
# screeplot of the result
screeplot(res)
# manhattan style plot the result
plot(res)
```

---

generateMed *Generate a high-dimensional mediation dataset*

Description

This function generates a dataset from an x -> M -> y model, where M may be of any size with any correlation matrix.

Usage

```
generateMed(
  n = 100L,
  a = 0.3,
  b = 0.3,
  r2y = 0.5,
  dir = 0,
  Sigma,
  residual = FALSE,
  empirical = FALSE,
  scaley = FALSE,
  forma = identity,
  formb = identity
)
```

Arguments

n	Sample size
a	Vector of a path coefficients within 0 and 1
b	Vector of b path coefficients within 0 and 1
r2y	Proportion of explained variance in y. Set to b %*% Sigma %*% b for var(y) == 1.
dir	Direct path from x to y
Sigma	Desired true covariance matrix between the mediators M
residual	Whether Sigma indicates residual or marginal covariance
empirical	Ensure observed data matrix has exactly the requested covmat (only if Sigma is specified)
scaley	Whether to standardise y (changes b path coefficients)
forma	Functional form of the a paths. Function that accepts a matrix as input and transforms each column to the desired form.
formb	Functional form of the b paths. Function that accepts a vector.

Value

A data frame with columns x, M.1 - M.p, y

Examples

```
# Generate a suppression dataset where M.2 is suppressed
sup <- generateMed(n = 100,
                   a = c(-0.4, 0.4),
                   b = c(0.8, 0.48),
                   Sigma = matrix(c(1, -0.6, -0.6, 1), 2))
```

---

methylation *DNA Methylation, childhood trauma, and stress response*

Description

A dataset containing variables on childhood trauma, stress response, and 1000 preprocessed epigenetic methylation values. This dataset was synthesised using the synthpop package to remove any identifying information.

Usage

```
methylation
```

Format

A data frame with 85 rows and 1002 variables: a composite score on a childhood trauma questionnaire (x), a measure of increase in area under the curve of cortisol after stress task (y), and beta values of methylation at 1000 different locations in the genome.

References

Houtepen, L. C., Vinkers, C. H., Carrillo-Roa, T., Hiemstra, M., Van Lier, P. A., Meeus, W., ... & Schalkwyk, L. C. (2016). Genome-wide DNA methylation levels and altered cortisol stress reactivity following childhood trauma in humans. *Nature communications*, 7, 10967.

van Kesteren, E. J., & Oberski, D. L. (2018). Exploratory Mediation Analysis with Many Potential Mediators. *arXiv preprint* arXiv:1810.06334.

`setCutoff` *Set the cutoff for mediator selection*

Description

This function sets the cutoff value on a cmf object for mediator selection. Any cutoff value between 0 and 1 is allowed, where potential mediators with empirical selection probability (selection rate) above the cutoff will be considered mediators and the others will not. The cutoff can be entered manually or, when set to "mc", be based on a monte carlo simulation. See "details"

Usage

```
setCutoff(object, cutoff = 0.5)
```

Arguments

`object`	a cmf object
`cutoff`	either a number between 0 and 1 or "mc" - see details

Details

The monte carlo determination is based on a procedure in PCA and Factor Analysis called "Parallel Analysis". We generate data from the null hypothesis with the same dimensionality as the original dataset and perform the algorithm 100 times. This creates a distribution of nonmediator selection rates from which we can determine the cutoff (the 99.9th percentile)

Value

a cmf object with updated cutoff value

Examples

```
# generate some data
dat <- generateMed(a = (1:10)/20, b = (1:10)/20)
res <- cmf(dat)
# set the cutoff for this result at 0.1
setCutoff(res, 0.1)
```

`update.cmf` *Add samples to existing cmf results object*

Description

This function adds additional samples to the existing results object

Usage

```
## S3 method for class 'cmf'
update(object, nStarts = 100, ...)
```

Arguments

`object`	a cmf object
`nStarts`	the number of starts to add (default 100)
`...`	not used

Examples

```
# generate some data
dat <- generateMed(a = (1:10)/20, b = (1:10)/20)
res <- cmf(dat, nStarts = 200)
# double the samples
res <- update(res, 500)
```

Index