Package ‘tEDM’

July 15, 2025

Title Temporal Empirical Dynamic Modeling

Version 1.0

Description Inferring causation from time series data through empirical dynamic modeling (EDM), with methods such as convergent cross mapping from Sugihara et al. (2012) <doi:10.1126/science.1227079>, partial cross mapping as outlined in Leng et al. (2020) <doi:10.1038/s41467-020-16238-0>, and cross mapping cardinality as described in Tao et al. (2023) <doi:10.1016/j.fmre.2023.01.007>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

URL `https://stscl.github.io/tEDM/`, `https://github.com/stscl/tEDM`

BugReports `https://github.com/stscl/tEDM/issues`

Depends R (>= 4.1.0)

LinkingTo Rcpp, RcppThread, RcppArmadillo

Imports dplyr, ggplot2, methods, Rcpp

Suggests RcppThread, RcppArmadillo, readr, plot3D, spEDM, knitr, rmarkdown, purrr, tidyr, cowplot

VignetteBuilder knitr

NeedsCompilation yes

Author Wenbo Lv [aut, cre, cph] (ORCID: <`https://orcid.org/0009-0002-6003-3800`>)

Maintainer Wenbo Lv <`lyu.geosocial@gmail.com`>

Repository CRAN

Date/Publication 2025-07-15 12:00:02 UTC

Contents

`ccm` *convergent cross mapping*

Description

convergent cross mapping

Usage

```
## S4 method for signature 'data.frame'
ccm(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  tau = 0,
  k = E + 1,
  theta = 1,
  algorithm = "simplex",
  lib = NULL,
  pred = NULL,
  threads = length(libsizes),
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE
)
```

Arguments

`data`	observation data.
`cause`	name of causal variable.
`effect`	name of effect variable.
`libsizes`	(optional) number of time points used.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.

`k`	(optional) number of nearest neighbors.
`theta`	(optional) weighting parameter for distances, useful when `algorithm` is `smap`.
`algorithm`	(optional) prediction algorithm.
`lib`	(optional) libraries indices.
`pred`	(optional) predictions indices.
`threads`	(optional) number of threads to use.
`parallel.level`	(optional) level of parallelism, `low` or `high`.
`bidirectional`	(optional) whether to examine bidirectional causality.
`progressbar`	(optional) whether to show the progress bar.

Value

A list

`xmap` cross mapping results

`varname` names of causal and effect variable

`bidirectional` whether to examine bidirectional causality

References

Sugihara, G., May, R., Ye, H., Hsieh, C., Deyle, E., Fogarty, M., Munch, S., 2012. Detecting Causality in Complex Ecosystems. Science 338, 496–500.

Examples

```
sim = logistic_map(x = 0.4,y = 0.4,step = 45,beta_xy = 0.5,beta_yx = 0)
ccm(sim,"x","y",libsizes = seq(5,35,5),E = 8,k = 7,threads = 1)
```

`cmc` *cross mapping cardinality*

Description

cross mapping cardinality

Usage

```
## S4 method for signature 'data.frame'
cmc(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
```

```
  tau = 0,
  k = pmin(E^2),
  lib = NULL,
  pred = NULL,
  threads = length(libsizes),
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE
)
```

Arguments

`data`	observation data.
`cause`	name of causal variable.
`effect`	name of effect variable.
`libsizes`	(optional) number of time points used.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.
`k`	(optional) number of nearest neighbors.
`lib`	(optional) libraries indices.
`pred`	(optional) predictions indices.
`threads`	(optional) number of threads to use.
`parallel.level`	(optional) level of parallelism, `low` or `high`.
`bidirectional`	(optional) whether to examine bidirectional causality.
`progressbar`	(optional) whether to show the progress bar.

Value

A list

`xmap` cross mapping results

`cs` causal strength

`varname` names of causal and effect variable

`bidirectional` whether to examine bidirectional causality

References

Tao, P., Wang, Q., Shi, J., Hao, X., Liu, X., Min, B., Zhang, Y., Li, C., Cui, H., Chen, L., 2023. Detecting dynamical causality by intersection cardinal concavity. Fundamental Research.

Examples

```
sim = logistic_map(x = 0.4,y = 0.4,step = 45,beta_xy = 0.5,beta_yx = 0)
cmc(sim,"x","y",E = 4,k = 15,threads = 1)
```

embedded — *embedding time series data*

Description

embedding time series data

Usage

```
## S4 method for signature 'data.frame'
embedded(data, target, E = 3, tau = 0)
```

Arguments

`data`	observation data.
`target`	name of target variable.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.

Value

A matrix

Examples

```
embedded(data.frame(t = 1:5),"t",3)
```

fnn — *false nearest neighbours*

Description

false nearest neighbours

Usage

```
## S4 method for signature 'data.frame'
fnn(
  data,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 0,
```

```
  rt = 10,
  eps = 2,
  threads = length(E)
)
```

Arguments

`data`	observation data.
`target`	name of target variable.
`lib`	(optional) libraries indices.
`pred`	(optional) predictions indices.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.
`rt`	(optional) escape factor.
`eps`	(optional) neighborhood diameter.
`threads`	(optional) number of threads to use.

Value

A vector

References

Kennel M. B., Brown R. and Abarbanel H. D. I., Determining embedding dimension for phase-space reconstruction using a geometrical construction, Phys. Rev. A, Volume 45, 3403 (1992).

Examples

```
sim = logistic_map(x = 0.4,y = 0.4,step = 45,beta_xy = 0.5,beta_yx = 0)
fnn(sim,"x",threads = 1)
```

`ic` *intersection cardinality*

Description

intersection cardinality

Usage

```
## S4 method for signature 'data.frame'
ic(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 0,
  k = E + 2,
  threads = length(pred),
  parallel.level = "low"
)
```

Arguments

`data`	observation data.
`column`	name of library variable.
`target`	name of target variable.
`lib`	(optional) libraries indices.
`pred`	(optional) predictions indices.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.
`k`	(optional) number of nearest neighbors used in prediction.
`threads`	(optional) number of threads to use.
`parallel.level`	(optional) level of parallelism, `low` or `high`.

Value

A list

`xmap` cross mapping performance

`varname` name of target variable

`method` method of cross mapping

`tau` step of time lag

References

Tao, P., Wang, Q., Shi, J., Hao, X., Liu, X., Min, B., Zhang, Y., Li, C., Cui, H., Chen, L., 2023. Detecting dynamical causality by intersection cardinal concavity. Fundamental Research.

Examples

```
sim = logistic_map(x = 0.4,y = 0.4,step = 45,beta_xy = 0.5,beta_yx = 0)
ic(sim,"x","y",E = 4,k = 15:30,threads = 1)
```

logistic_map *logistic map*

Description

logistic map

Usage

```
logistic_map(
  x,
  y = NULL,
  z = NULL,
  step = 15,
  alpha_x = 3.6,
  alpha_y = 3.72,
  alpha_z = 3.68,
  beta_xy = 0.05,
  beta_xz = 0.05,
  beta_yx = 0.2,
  beta_yz = 0.2,
  beta_zx = 0.35,
  beta_zy = 0.35,
  threshold = Inf,
  transient = 1
)
```

Arguments

x	value x.
y	(optional) value y.
z	(optional) value z.
step	(optional) number of simulation time steps.
alpha_x	(optional) growth parameter for x.
alpha_y	(optional) growth parameter for y.
alpha_z	(optional) growth parameter for y.
beta_xy	(optional) cross-inhibition from x to y.
beta_xz	(optional) cross-inhibition from x to z.
beta_yx	(optional) cross-inhibition from y to x.
beta_yz	(optional) cross-inhibition from y to z.
beta_zx	(optional) cross-inhibition from z to x.
beta_zy	(optional) cross-inhibition from z to y.
threshold	(optional) set to `NaN` if the absolute value exceeds this threshold.
transient	(optional) transients to be excluded from the results.

Value

A data.frame

Examples

```
logistic_map(x = 0.2)
```

`multispatialccm` *multispatial convergent cross mapping*

Description

multispatial convergent cross mapping

Usage

```
## S4 method for signature 'list'
multispatialccm(
  data,
  cause,
  effect,
  libsizes,
  E = 3,
  tau = 0,
  k = E + 1,
  boot = 99,
  seed = 42,
  threads = length(libsizes),
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE
)
```

Arguments

`data`	observation data.
`cause`	name of causal variable.
`effect`	name of effect variable.
`libsizes`	number of time points used in prediction.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.
`k`	(optional) number of nearest neighbors used in prediction.
`boot`	(optional) number of bootstraps to perform.

`seed`	(optional) random seed.
`threads`	(optional) number of threads to use.
`parallel.level`	(optional) level of parallelism, `low` or `high`.
`bidirectional`	(optional) whether to examine bidirectional causality.
`progressbar`	(optional) whether to show the progress bar.

Value

A list

`xmap` cross mapping results

`varname` names of causal and effect variable

`bidirectional` whether to examine bidirectional causality

References

Clark, A.T., Ye, H., Isbell, F., Deyle, E.R., Cowles, J., Tilman, G.D., Sugihara, G., 2015. Spatial convergent cross mapping to detect causal relationships from short time series. Ecology 96, 1174–1181.

Examples

```
set.seed(42)
obs = runif(15,0,0.1)
sim = vector("list",15)
for (i in seq_along(obs)){
  sim[[i]] = logistic_map(x = obs[i],y = obs[i],step = 15,beta_xy = 0.5,beta_yx = 0)
}
lst = list(x = do.call(cbind, lapply(sim, function(df) df$x)),
           y = do.call(cbind, lapply(sim, function(df) df$y)))
multispatialccm(lst,"x","y",libsizes = 5:15,E = c(7,2),k = 6,threads = 1)
```

`pcm` *partial cross mapping*

Description

partial cross mapping

Usage

```
## S4 method for signature 'data.frame'
pcm(
  data,
  cause,
  effect,
  conds,
  libsizes = NULL,
  E = 3,
  tau = 0,
  k = E + 1,
  theta = 1,
  algorithm = "simplex",
  lib = NULL,
  pred = NULL,
  threads = length(libsizes),
  parallel.level = "low",
  bidirectional = TRUE,
  cumulate = FALSE,
  progressbar = TRUE
)
```

Arguments

`data`	observation data.
`cause`	name of causal variable.
`effect`	name of effect variable.
`conds`	name of conditioning variables.
`libsizes`	(optional) number of time points used.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.
`k`	(optional) number of nearest neighbors.
`theta`	(optional) weighting parameter for distances, useful when `algorithm` is `smap`.
`algorithm`	(optional) prediction algorithm.
`lib`	(optional) libraries indices.
`pred`	(optional) predictions indices.
`threads`	(optional) number of threads to use.
`parallel.level`	(optional) level of parallelism, `low` or `high`.
`bidirectional`	(optional) whether to examine bidirectional causality.
`cumulate`	(optional) serial or cumulative computation of partial cross mapping.
`progressbar`	(optional) whether to show the progress bar.

Value

A list

`pxmap` partial cross mapping results

`xmap` cross mapping results

`varname` names of causal and effect variable

`bidirectional` whether to examine bidirectional causality

References

Leng, S., Ma, H., Kurths, J. et al. Partial cross mapping eliminates indirect causal influences. Nat Commun 11, 2632 (2020).

Examples

```
sim = logistic_map(x = 0.4,y = 0.4,z = 0.4,step = 45,
                   beta_xy = 0.5, beta_xz = 0,
                   beta_yx = 0, beta_yz = 0.5,
                   beta_zx = 0, beta_zy = 0)
pcm(sim,"x","z","y",libsizes = seq(5,35,5),E = 10,k = 7,threads = 1)
```

`simplex` *simplex forecast*

Description

simplex forecast

Usage

```
## S4 method for signature 'data.frame'
simplex(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 0,
  k = E + 1,
  threads = length(E)
)

## S4 method for signature 'list'
simplex(
  data,
```

```
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 0,
  k = E + 1,
  threads = length(E)
)
```

Arguments

`data`	observation data.
`column`	name of library variable.
`target`	name of target variable.
`lib`	(optional) libraries indices.
`pred`	(optional) predictions indices.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.
`k`	(optional) number of nearest neighbors used in prediction.
`threads`	(optional) number of threads to use.

Value

A list

`xmap` forecast performance

`varname` name of target variable

`method` method of cross mapping

`tau` step of time lag

References

Sugihara G. and May R. 1990. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. Nature, 344:734-741.

Examples

```
sim = logistic_map(x = 0.4,y = 0.4,step = 45,beta_xy = 0.5,beta_yx = 0)
simplex(sim,"x","y",k = 7,threads = 1)
```

smap *smap forecast*

Description

smap forecast

Usage

```
## S4 method for signature 'data.frame'
smap(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 3,
  tau = 0,
  k = E + 1,
  theta = c(0, 1e-04, 3e-04, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 0.5, 0.75, 1, 1.5, 2, 3,
    4, 6, 8),
  threads = length(theta)
)
```

Arguments

`data`	observation data.
`column`	name of library variable.
`target`	name of target variable.
`lib`	(optional) libraries indices.
`pred`	(optional) predictions indices.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of time lags.
`k`	(optional) number of nearest neighbors used in prediction.
`theta`	(optional) weighting parameter for distances.
`threads`	(optional) number of threads to use.

Value

A list

`xmap` forecast performance

`varname` name of target variable

`method` method of cross mapping

References

Sugihara G. 1994. Nonlinear forecasting for the classification of natural time series. Philosophical Transactions: Physical Sciences and Engineering, 348 (1688):477-495.

Examples

```
sim = logistic_map(x = 0.4,y = 0.4,step = 45,beta_xy = 0.5,beta_yx = 0)
smap(sim,"x","y",E = 8,k = 7,threads = 1)
```

Index