Package 'spEDM'

August 31, 2025

Title Spatial Empirical Dynamic Modeling

Version 1.8

Description Inferring causation from spatial cross-sectional data through empirical dynamic modeling (EDM), with methodological extensions including geographical convergent cross mapping from Gao et al. (2023) <doi:10.1038/s41467-023-41619-6>, as well as the spatial causality test following the approach of Herrera et al. (2016) <doi:10.1111/pirs.12144>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

URL https://stscl.github.io/spEDM/, https://github.com/stscl/spEDM

BugReports https://github.com/stscl/spEDM/issues

Depends R (>= 4.1.0)

LinkingTo Rcpp, RcppThread, RcppArmadillo

Imports dplyr, ggplot2, methods, sdsfun (>= 0.7.0), sf, terra

Suggests knitr, Rcpp, RcppThread, RcppArmadillo, rmarkdown, readr, plot3D

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 2025-08-31 11:30:02 UTC

Contents

`detectThreads` *detect the number of available threads*

Description

detect the number of available threads

Usage

```
detectThreads()
```

Value

An integer

Examples

```
detectThreads()
```

`embedded` *embedding spatial cross sectional data*

Description

embedding spatial cross sectional data

Usage

```
## S4 method for signature 'sf'
embedded(data, target, E = 3, tau = 1, style = 1, nb = NULL, detrend = FALSE)

## S4 method for signature 'SpatRaster'
embedded(data, target, E = 3, tau = 1, style = 1, detrend = FALSE)
```

Arguments

data	observation data.
target	name of target variable.
E	(optional) embedding dimensions.
tau	(optional) step of spatial lags.
style	(optional) embedding style (0 includes current state, 1 excludes it).
nb	(optional) neighbours list.
detrend	(optional) whether to remove the linear trend.

Value

A matrix

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))
v = embedded(columbus,"crime")
v[1:5,]

cu = terra::rast(system.file("case/cu.tif", package="spEDM"))
r = embedded(cu,"cu")
r[1:5,]
```

---

fnn *false nearest neighbours*

Description

false nearest neighbours

Usage

```
## S4 method for signature 'sf'
fnn(
  data,
  target,
  lib = NULL,
  pred = NULL,
  E = 1:10,
  tau = 1,
  style = 1,
  dist.metric = "L1",
  nb = NULL,
  rt = 10,
  eps = 2,
```

```
  threads = detectThreads(),
  detrend = TRUE
)

## S4 method for signature 'SpatRaster'
fnn(
  data,
  target,
  lib = NULL,
  pred = NULL,
  E = 1:10,
  tau = 1,
  style = 1,
  dist.metric = "L1",
  rt = 10,
  eps = 2,
  threads = detectThreads(),
  detrend = TRUE
)
```

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 spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
`nb`	(optional) neighbours list.
`rt`	(optional) escape factor.
`eps`	(optional) neighborhood diameter.
`threads`	(optional) number of threads to use.
`detrend`	(optional) whether to remove the linear trend.

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

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

fnn(columbus,"crime")
```

---

gccm *geographical convergent cross mapping*

Description

geographical convergent cross mapping

Usage

```
## S4 method for signature 'sf'
gccm(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  tau = 1,
  k = E + 2,
  theta = 1,
  algorithm = "simplex",
  lib = NULL,
  pred = NULL,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  nb = NULL,
  threads = detectThreads(),
  parallel.level = "low",
  bidirectional = TRUE,
  detrend = TRUE,
  progressbar = TRUE
)

## S4 method for signature 'SpatRaster'
gccm(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  tau = 1,
```

```
  k = E + 2,
  theta = 1,
  algorithm = "simplex",
  lib = NULL,
  pred = NULL,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  threads = detectThreads(),
  parallel.level = "low",
  bidirectional = TRUE,
  detrend = TRUE,
  progressbar = TRUE
)
```

Arguments

data	observation data.
cause	name of causal variable.
effect	name of effect variable.
libsizes	(optional) number of spatial units used.
E	(optional) embedding dimensions.
tau	(optional) step of spatial 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.
style	(optional) embedding style (`0` includes current state, `1` excludes it).
dist.metric	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
dist.average	(optional) whether to average distance.
nb	(optional) neighbours list.
threads	(optional) number of threads to use.
parallel.level	(optional) level of parallelism, `low` or `high`.
bidirectional	(optional) whether to examine bidirectional causality.
detrend	(optional) whether to remove the linear trend.
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

Gao, B., Yang, J., Chen, Z. et al. Causal inference from cross-sectional earth system data with geographical convergent cross mapping. Nat Commun 14, 5875 (2023).

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

g = gccm(columbus,"hoval","crime",libsizes = seq(5,45,5),E = 6)
g
plot(g, ylimits = c(0,0.85))
```

---

gcmc *geographical cross mapping cardinality*

Description

geographical cross mapping cardinality

Usage

```
## S4 method for signature 'sf'
gcmc(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  tau = 1,
  k = pmin(E^2),
  lib = NULL,
  pred = NULL,
  style = 1,
  dist.metric = "L2",
  nb = NULL,
  threads = detectThreads(),
  parallel.level = "low",
  bidirectional = TRUE,
  detrend = FALSE,
  progressbar = TRUE
)

## S4 method for signature 'SpatRaster'
gcmc(
  data,
  cause,
```

```
  effect,
  libsizes = NULL,
  E = 3,
  tau = 1,
  k = pmin(E^2),
  lib = NULL,
  pred = NULL,
  style = 1,
  dist.metric = "L2",
  threads = detectThreads(),
  parallel.level = "low",
  bidirectional = TRUE,
  detrend = FALSE,
  progressbar = TRUE
)
```

Arguments

data	observation data.
cause	name of causal variable.
effect	name of effect variable.
libsizes	(optional) number of spatial units used.
E	(optional) embedding dimensions.
tau	(optional) step of spatial lags.
k	(optional) number of nearest neighbors.
lib	(optional) libraries indices.
pred	(optional) predictions indices.
style	(optional) embedding style (0 includes current state, 1 excludes it).
dist.metric	(optional) distance metric (L1: Manhattan, L2: Euclidean).
nb	(optional) neighbours list.
threads	(optional) number of threads to use.
parallel.level	(optional) level of parallelism, low or high.
bidirectional	(optional) whether to examine bidirectional causality.
detrend	(optional) whether to remove the linear trend.
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

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

g = gcmc(columbus,"hoval","crime",E = 7,k = 18)
g
```

---

ic *intersection cardinality*

Description

intersection cardinality

Usage

```
## S4 method for signature 'sf'
ic(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  nb = NULL,
  threads = detectThreads(),
  parallel.level = "low",
  detrend = FALSE
)

## S4 method for signature 'SpatRaster'
ic(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  threads = detectThreads(),
```

```
  parallel.level = "low",
  detrend = FALSE
)
```

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 spatial lags.
k	(optional) number of nearest neighbors used.
style	(optional) embedding style (0 includes current state, 1 excludes it).
dist.metric	(optional) distance metric (L1: Manhattan, L2: Euclidean).
nb	(optional) neighbours list.
threads	(optional) number of threads to use.
parallel.level	(optional) level of parallelism, low or high.
detrend	(optional) whether to remove the linear trend.

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

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

ic(columbus,"hoval","crime", E = 7, k = 15:25)
```

multiview *multiview embedding forecast*

Description

multiview embedding forecast

Usage

```
## S4 method for signature 'sf'
multiview(
  data,
  column,
  target,
  nvar,
  lib = NULL,
  pred = NULL,
  E = 3,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  nb = NULL,
  top = NULL,
  threads = detectThreads(),
  detrend = TRUE
)

## S4 method for signature 'SpatRaster'
multiview(
  data,
  column,
  target,
  nvar,
  lib = NULL,
  pred = NULL,
  E = 3,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  top = NULL,
  threads = detectThreads(),
  detrend = TRUE
)
```

Arguments

data	observation data.
column	name of library variable.
target	name of target variable.
nvar	number of variable combinations.
lib	(optional) libraries indices.
pred	(optional) predictions indices.
E	(optional) embedding dimensions.
tau	(optional) step of spatial lags.
k	(optional) number of nearest neighbors used.
style	(optional) embedding style (0 includes current state, 1 excludes it).
dist.metric	(optional) distance metric (L1: Manhattan, L2: Euclidean).
dist.average	(optional) whether to average distance.
nb	(optional) neighbours list.
top	(optional) number of reconstructions used in MVE forecast.
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.

Value

A vector (when input is sf object) or matrix

References

Ye H., and G. Sugihara, 2016. Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. Science 353:922-925.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

multiview(columbus,
          column = c("inc","crime","open","plumb","discbd"),
          target = "hoval", nvar = 3)
```

sc.test *spatial causality test*

Description

spatial causality test

Usage

```
## S4 method for signature 'sf'
sc.test(
  data,
  cause,
  effect,
  k,
  block = 3,
  boot = 399,
  seed = 42,
  base = 2,
  lib = NULL,
  pred = NULL,
  nb = NULL,
  threads = detectThreads(),
  detrend = TRUE,
  normalize = FALSE,
  progressbar = FALSE
)

## S4 method for signature 'SpatRaster'
sc.test(
  data,
  cause,
  effect,
  k,
  block = 3,
  boot = 399,
  seed = 42,
  base = 2,
  lib = NULL,
  pred = NULL,
  threads = detectThreads(),
  detrend = TRUE,
  normalize = FALSE,
  progressbar = FALSE
)
```

Arguments

data	observation data.
cause	name of causal variable.
effect	name of effect variable.
k	(optional) number of nearest neighbors used in symbolization.
block	(optional) number of blocks used in spatial block bootstrap.
boot	(optional) number of bootstraps to perform.
seed	(optional) random seed.
base	(optional) logarithm base.
lib	(optional) libraries indices.
pred	(optional) predictions indices.
nb	(optional) neighbours list.
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.
normalize	(optional) whether to normalize the result.
progressbar	(optional) whether to show the progress bar.

Value

A list

sc statistic for spatial causality

varname names of causal and effect variable

References

Herrera, M., Mur, J., & Ruiz, M. (2016). Detecting causal relationships between spatial processes. Papers in Regional Science, 95(3), 577–595.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

sc.test(columbus,"hoval","crime", k = 15)
```

simplex *simplex forecast*

Description

simplex forecast

Usage

```
## S4 method for signature 'sf'
simplex(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  nb = NULL,
  threads = detectThreads(),
  detrend = TRUE
)

## S4 method for signature 'SpatRaster'
simplex(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 2:10,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  threads = detectThreads(),
  detrend = TRUE
)
```

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 spatial lags.
k	(optional) number of nearest neighbors used.
style	(optional) embedding style (0 includes current state, 1 excludes it).
dist.metric	(optional) distance metric (L1: Manhattan, L2: Euclidean).
dist.average	(optional) whether to average distance.
nb	(optional) neighbours list.
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.

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

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

simplex(columbus,"inc","crime")
```

slm *spatial logistic map*

Description

spatial logistic map

Usage

```
## S4 method for signature 'sf'
slm(
  data,
  x = NULL,
  y = NULL,
  z = NULL,
  k = 4,
  step = 15,
  alpha_x = 0.28,
  alpha_y = 0.25,
  alpha_z = 0.22,
  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,
  interact = "local",
  nb = NULL
)

## S4 method for signature 'SpatRaster'
slm(
  data,
  x = NULL,
  y = NULL,
  z = NULL,
  k = 4,
  step = 15,
  alpha_x = 0.28,
  alpha_y = 0.25,
  alpha_z = 0.22,
  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,
  interact = "local"
)
```

Arguments

data	observation data.
x	(optional) name of first spatial variable.
y	(optional) name of second spatial variable.
z	(optional) name of third spatial variable.
k	(optional) number of neighbors to used.
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.
interact	(optional) type of cross-variable interaction (local or neighbors).
nb	(optional) neighbours list.

Value

A list

References

Willeboordse, F.H., The spatial logistic map as a simple prototype for spatiotemporal chaos, Chaos, 533–540 (2003).

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))
columbus$inc = sdsfun::normalize_vector(columbus$inc)
slm(columbus,"inc")
```

smap *smap forecast*

Description

smap forecast

Usage

```
## S4 method for signature 'sf'
smap(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 3,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  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),
  nb = NULL,
  threads = detectThreads(),
  detrend = TRUE
)

## S4 method for signature 'SpatRaster'
smap(
  data,
  column,
  target,
  lib = NULL,
  pred = NULL,
  E = 3,
  tau = 1,
  k = E + 2,
  style = 1,
  dist.metric = "L2",
  dist.average = TRUE,
  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 = detectThreads(),
  detrend = TRUE
)
```

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 spatial lags.
k	(optional) number of nearest neighbors used.
style	(optional) embedding style (0 includes current state, 1 excludes it).
dist.metric	(optional) distance metric (L1: Manhattan, L2: Euclidean).
dist.average	(optional) whether to average distance.
theta	(optional) weighting parameter for distances.
nb	(optional) neighbours list.
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.

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

```
columbus = sf::read_sf(system.file("case/columbus.gpkg", package="spEDM"))

smap(columbus,"inc","crime",E = 5,k = 6)
```

Index