

Package ‘sindyr’

May 9, 2026

Type Package

Title Sparse Identification of Nonlinear Dynamics

Version 0.2.4

Date 2024-04-30

Author Rick Dale and Harish S. Bhat

Maintainer Rick Dale <racdale@gmail.com>

Description This implements the Brunton et al (2016; PNAS <[doi:10.1073/pnas.1517384113](https://doi.org/10.1073/pnas.1517384113)>) sparse identification algorithm for finding ordinary differential equations for a measured system from raw data (SINDy). The package includes a set of additional tools for working with raw data, with an emphasis on cognitive science applications (Dale and Bhat, 2018 <[doi:10.1016/j.cogsys.2018.06.020](https://doi.org/10.1016/j.cogsys.2018.06.020)>). See <<https://github.com/racdale/sindyr>> for examples and updates.

Depends R (>= 3.4), arrangements, matrixStats, igraph, graphics, grDevices

Imports pracma

License GPL (>= 2)

Collate 'windowed_sindy.R' 'sindy.R' 'features.R'
'finite_differences.R' 'finite_difference.R'
'lorenzattractor.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2024-05-01 06:00:02 UTC

Contents

sindyr-package	2
features	3
finite_difference	4
finite_differences	4
lorenzattractor	5
sindy	6
windowed_sindy	7

`sindyr-package`*Sparse Identification of Nonlinear Dynamics*

Description

This implements the Brunton et al (2016; PNAS, doi: 10.1073/pnas.1517384113) sparse identification algorithm for finding ordinary differential equations for a measured system from raw data (SINDy). The package includes a set of additional tools for working with raw data, with an emphasis on cognitive science applications (Dale and Bhat, 2018, doi: 10.1016/j.cogsys.2018.06.020). See <<https://github.com/racdale/sindyr>> for examples and updates.

Details

Package: `sindyr`
Type: `Package`
Version: `0.2.1`
Date: `2018-09-10`
License: `GPL >= 2`

`sindy`: Main function to infer coefficient matrix for set of ODEs.

`windowed_sindy`: Sliding window function to obtain SINDy results across segments of a time series.

`features`: Function for generation feature space from measured variables.

`finite_differences`: Numerical differentiation over multiple columns.

`finite_difference`: Numerical differential of a vector.

Author(s)

Rick Dale and Harish S. Bhat

References

Dale, R. and Bhat, H. S. (2018). Equations of mind: data science for inferring nonlinear dynamics of socio-cognitive systems. *Cognitive Systems Research*, 52, 275-290.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15), 3932-3937.

For further examples and links to other materials see: <https://github.com/racdale/sindyr>

Examples

```
# example to reconstruct of
# the Lorenz system

library(sindyr)

set.seed(666)
dt = .001
numsteps = 10000; dt = dt; sigma = 10; r = 28; b = 2.6;
xs = data.frame(lorenzattractor(numsteps, dt, sigma, r, b))
colnames(xs) = list('x', 'y', 'z')
xs = xs[2000:nrow(xs),] # cut out initialization

Theta = features(xs,3) # grid of features
par(mfrow=c(7,3),oma = c(2,0,0,0) + 0.1,mar = c(1,1,1,1) + 0.1)
for (i in 2:ncol(Theta)) {
  plot(Theta[,i],xlab='t',main=gsub(':', '', colnames(Theta)[i]),type='l',xaxt='n',yaxt='n')
}

sindy.obj = sindy(xs=xs,dt=dt,lambda=.5) # let's reconstruct
sindy.obj$B # Lorenz equations
```

 features

Build a matrix of features for SINDy

Description

Takes a raw matrix of data and converts into polynomial features

Arguments

x	Raw data to be converted into features
polyorder	Order of polynomials (including k-th self products)
intercept	Include column of 1s in features to represent intercept (default = TRUE)

Details

Expands raw data into a set of polynomial features.

Value

Returns a new matrix of data with features from raw data

Author(s)

Rick Dale and Harish S. Bhat

finite_difference	<i>Estimate derivative of variable with finite differences</i>
-------------------	--

Description

Estimates first-order derivatives of a vector

Arguments

x	Raw data to be differentiated
S	Sample rate of data to return derivatives using raw time

Details

Uses simplest version of finite-difference method (window size 2) to numerically estimate derivative of a time series.

Value

Returns first-order numerical derivatives estimated from data.

Author(s)

Rick Dale and Harish S. Bhat

finite_differences	<i>Estimate derivatives of multiple variables with finite differences</i>
--------------------	---

Description

Estimates first-order derivatives of column vectors of a matrix

Arguments

xs	Raw data to be differentiated (matrix)
S	Sample rate of data to return derivatives using raw time

Details

Uses simplest version of finite-difference method (window size 2) to numerically estimate derivative of multiple columnar time series.

Value

Returns first-order numerical derivatives estimated from data.

Author(s)

Rick Dale and Harish S. Bhat

lorenzattractor *Simulate the Lorenz Attractor*

Description

An implementation of the Lorenz dynamical system, which describes the motion of a possible particle, which will neither converge to a steady state, nor diverge to infinity; but rather stay in a bounded but 'chaotically' defined region, i.e., an attractor.

Usage

```
lorenzattractor(numsteps, dt, sigma, r, b)
```

Arguments

numsteps	The number of simulated points
dt	System parameter
sigma	System parameter
r	System parameter
b	System parameter

Value

It returns a matrix with the 3 dimensions of the Lorenz

Author(s)

Moreno I. Coco (moreno.cocoi@gmail.com)

References

Lorenz, Edward Norton (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences* 20(2) 130-141.

Examples

```
## initialize the parameters
numsteps = 2 ^ 11; dt = .01; sigma = 10; r = 28; b = 8/3;

res = lorenzattractor(numsteps, dt, sigma, r, b)
```

sindy

*Run main SINDy function***Description**

Estimates coefficients for set of ordinary differential equations governing system variables.

Arguments

<code>xs</code>	Matrix of raw data
<code>dx</code>	Matrix of main system variable derivatives; if NULL, it estimates with finite differences from <code>xs</code>
<code>dt</code>	Sample interval, if data continuously sampled; default = 1
<code>Theta</code>	Matrix of features; if not supplied, assumes polynomial features of order 3
<code>lambda</code>	Threshold to use for iterated least squares sparsification (Brunton et al.)
<code>B.expected</code>	The function will compute a goodness of fit if supplied with an expected coefficient matrix <code>B</code> ; default = NULL
<code>verbose</code>	Verbose mode outputs <code>Theta</code> and <code>dx</code> values in their entirety; default = FALSE
<code>fit.its</code>	Number of iterations to conduct the least-square threshold sparsification; default = 10
<code>plot.eq.graph</code>	When set to TRUE, prints an igraph plot of variables as a graph structure; default = FALSE

Details

Uses the "left-division" approach of Brunton et al. (2016), and implements least-squares sparsification, and outputs coefficients after iterations stabilize.

Value

Returns a matrix `B` of coefficients specifying the relationship between `dx` and `Theta`

Author(s)

Rick Dale and Harish S. Bhat

References

- Dale, R. and Bhat, H. S. (in press). Equations of mind: data science for inferring nonlinear dynamics of socio-cognitive systems. *Cognitive Systems Research*.
- Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15), 3932-3937.

windowed_sindy *Run SINDy over time windows*

Description

Run SINDy on raw data with a sliding window approach

Arguments

xs	Matrix of raw data
dx	Matrix of main system variable derivatives; if NULL, it estimates with finite differences from xs
dt	Sample interval, if data continuously sampled; default = 1
Theta	Matrix of features; if not supplied, assumes polynomial features of order 3
lambda	Threshold to use for iterated least squares sparsification (Brunton et al.)
fit.its	Number of iterations to conduct the least-square threshold sparsification; default = 10
B.expected	The function will compute a goodness of fit if supplied with an expected coefficient matrix B; default = NULL
window.size	Size of window to segment raw data as separate time series; defaults to deciles
window.shift	Step sizes across windows, permitting overlap; defaults to deciles

Details

A convenience function for extracting a list of coefficients on segments of a time series. This facilitates using SINDy output as source of descriptive measures of dynamics.

Value

It returns a list of coefficients Bs containing B coefficients at each window

Author(s)

Rick Dale and Harish S. Bhat

References

Dale, R. and Bhat, H. S. (in press). Equations of mind: data science for inferring nonlinear dynamics of socio-cognitive systems. Cognitive Systems Research.

Index

* **misc**

- features, [3](#)
- finite_difference, [4](#)
- finite_differences, [4](#)
- sindy, [6](#)
- windowed_sindy, [7](#)

* **package**

- sindyr-package, [2](#)

* **ts**

- lorenzattractor, [5](#)

features, [3](#)

finite_difference, [4](#)

finite_differences, [4](#)

lorenzattractor, [5](#)

sindy, [6](#)

sindyr-package, [2](#)

windowed_sindy, [7](#)