

# Package ‘sindyr’

June 9, 2020

**Type** Package

**Title** Sparse Identification of Nonlinear Dynamics

**Version** 0.2.3

**Date** 2020-06-08

**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>) 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, in press <doi:10.1016/j.cogsys.2018.06.020>).

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

**Imports** crqa, plot3D, pracma

**License** GPL (>= 2)

**Collate** 'windowed\_sindy.R' 'sindy.R' 'features.R'  
'finite\_differences.R' 'finite\_difference.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-06-09 05:20:03 UTC

## R topics documented:

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

|              |          |
|--------------|----------|
| <b>Index</b> | <b>7</b> |
|--------------|----------|

**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, in press, doi: 10.1016/j.cogsys.2018.06.020).

**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.

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. (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.

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(crqa::lorenzattractor(numsteps, dt, sigma, r, b, plots=FALSE))
colnames(xs) = list('x','y','z')
xs = xs[2000:nrow(xs),] # cut out initialization

plot3D::points3D(xs$x,xs$y,xs$z,type='l',col='black')
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      *Estimate derivative of variable with finite differences*

---

**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      *Estimate derivatives of multiple variables with finite differences*

---

**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

sindy

*Run main SINDy function***Description**

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

**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.)                                   |
| B.expected    | The function will compute a goodness of fit if supplied with an expected coefficient matrix B; default = NULL |
| verbose       | Verbose mode outputs Theta and dx values in their entirety; default = FALSE                                   |
| fit.its       | Number of iterations to conduct the least-square threshold sparsification; default = 10                       |
| plot.eq.graph | 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

\*Topic **misc**

- features, [3](#)
- finite\_difference, [4](#)
- finite\_differences, [4](#)
- sindy, [5](#)
- windowed\_sindy, [6](#)

\*Topic **package**

- sindyr-package, [2](#)

features, [3](#)

finite\_difference, [4](#)

finite\_differences, [4](#)

sindy, [5](#)

sindyr-package, [2](#)

windowed\_sindy, [6](#)