Package 'sr'

March 10, 2023

Type Package

Title Smooth Regression - The Gamma Test and Tools

Version 0.1.0

Description Finds causal connections in precision data, finds lags and embeddings in time series, guides training of neural networks and other smooth models, evaluates their performance, gives a mathematically grounded answer to the over-training problem. Smooth regression is based on the Gamma test, which measures smoothness in a multivariate relationship. Causal relations are smooth, noise is not. 'sr' includes the Gamma test and search techniques that use it. References: Evans & Jones (2002) <doi:10.1098/rspa.2002.1010>, AJ Jones (2004) <doi:10.1007/s10287-003-0006-1>.

License GPL (>= 3)

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.3

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 3.5.0)

Imports ggplot2, dplyr, progress, RANN, stats, vdiffr

Suggests knitr, magrittr, nnet, rmarkdown, testthat (>= 3.0.0)

URL https://smoothregression.com, https://github.com/haythorn/sr/

BugReports https://github.com/haythorn/sr/issues

NeedsCompilation no

Author Wayne Haythorn [aut, cre],
Antonia Jones [aut] (Principal creator of the Gamma test),
Sam Kemp [ctb] (Wrote the original code for the Gamma test in R)

Maintainer Wayne Haythorn <support@smoothregression.com>

Repository CRAN

Date/Publication 2023-03-10 08:00:03 UTC

R topics documented:

`fe_search` *Full Embedding Search*

Description

Calculates Gamma for all combinations of a set of input predictors

Usage

```
fe_search(predictors, target, prog_bar = TRUE, n_neighbors = 10, eps = 0)
```

Arguments

`predictors`	A vector or matrix whose columns are proposed inputs to a predictive function
`target`	A vector of double, the output variable that is to be predicted
`prog_bar`	Logical, set this to FALSE if you don't want progress bar displayed
`n_neighbors`	Integer number of near neighbors to use in RANN search, passed to gamma_test
`eps`	The error limit for the approximate near neighbor search. This will be passed to gamma_test, which will pass it on to the ANN near neighbor search. Setting this greater than zero can significantly reduce search time for large data sets.

Details

Given a set of predictors and a target that is to be predicted, this search will run the gamma test on every combination of the inputs. It returns the results in order of increasing gamma, so the best combinations of inputs for prediction will be at the beginning of the list. As this is a fully combinatoric search, it will start to get slow beyond about 16 inputs. By default, `fe_search` will display a progress bar showing the time to completion.

`fe_search()` returns a data.frame with two columns: Gamma, a sorted vector of Gamma values, and mask, an integer column containing the masks representing the inputs used to calculate each Gamma. To reconstruct the predictor set for a Gamma, use its mask with int_to_intMask and select_by_mask as shown in their examples.

Value

An invisible data frame with two columns, mask - an integer mask representing a subset of the predictors, and Gamma, the value of Gamma using those predictors. The rows are sorted from lowest to highest Gamma. The return value also has an attribute named target_V, the target variance. To get the vratio (estimated fraction of target variance due to noise), divide any of the Gammas by target_v.

Examples

```
e6 <- embed(mgls, 7)
t <- e6[ ,1]
p <- e6[ ,2:7]
full_search <- fe_search(predictors = p, target = t)
full_search <- dplyr::mutate(full_search,
                             vratio = Gamma / attr(full_search, "target_v"))
```

`gamma_histogram` *Plot Histogram of Gammas*

Description

Produces a histogram showing the distribution in a population of Gamma values, used to examine the result of a full embedding search. Pass the result of fe_search() to this function to look for structure in the predictors. For example, it this histogram is bimodal, there is probably one input variable which is absolutely required for a good predictive function, so the histogram divides into the subset containing that variable, and the others that don't.

Usage

```
gamma_histogram(fe_results, bins = 100, caption = "")
```

Arguments

`fe_results`	The result of fe_search or full_embedding_search. A matrix containing a column labeled Gamma, of Numeric Gamma values. It also contains an integer column of masks, but that is not used by this function.
`bins`	Numeric, number of bins in the histogram
`caption`	Character string caption for the plot

Value

a ggplot object, a histogram showing the distribution of Gamma values full embedding search output

Examples

```
e6 <- embed(mgls, 7)
t <- e6[ ,1]
p <- e6[ ,2:7]
full_search <- fe_search(predictors = p, target = t)
gamma_histogram(full_search, caption = "my data")
```

---

gamma_test *Estimate Smoothness in an Input/output Dataset*

Description

The gamma test measures mean squared error in an input/output data set, relative to an arbitrary, unknown smooth function. This can usually be interpreted as testing for the existence of a causal relationship, and estimating the expected error of the best smooth model that could be built on that relationship.

Usage

```
gamma_test(
  predictors,
  target,
  n_neighbors = 10,
  eps = 0,
  plot = FALSE,
  caption = "",
  verbose = FALSE
)
```

Arguments

predictors	A Numeric vector or matrix whose columns are proposed inputs to a predictive function.
target	A Numeric vector, the output variable that is to be predicted
n_neighbors	An Integer, the number of near neighbors to use in calculating gamma
eps	The error term passed to the approximate near neighbor search. The default value of zero means that exact near neighbors will be found, but time will be O(M^2), where an approximate search can run in O(M*log(M))
plot	A Logical variable, whether to plot the delta/gamma graph.
caption	A character string which will be the caption for the plot if plot = TRUE
verbose	A Logical variable, whether to return details of the computation

Value

If verbose == FALSE, a list containing Gamma and the vratio, If verbose == TRUE, that list plus the distances from each point to its near neighbors, the average of squared distances, and the value returned by lm on the delta and gamma averages. Gamma is Coefficient 1 of lm.

References

https://royalsocietypublishing.org/doi/10.1098/rspa.2002.1010, https://link.springer.com/article/10.1007/s10287-003-0006-1, https://smoothregression.com

Examples

```
he <- embed(henon_x, 3)
t <- he[ , 1]
p <- he[ ,2:3]
gamma_test(predictors = p, target = t)
```

---

get_Mlist *Discover how Gamma varies with sample size*

Description

Investigates the effect of sample size by calculating Gamma on larger and larger samples. Gamma will converge on the true noise in the relationship as sampling density on the function increases. `get_Mlist` produces a showing M values (sample sizes), and the associated Gammas and vratios. It produces a graph by default, and also returns an invisible data.frame. The successive samples are taken starting at the beginning of the inputs. There is no option to sort the input data; if you want the data to be randomized, do that before calling `get_Mlist`. The graph will become stable when the sample size is large enough. If the M list does not become stable, there is not enough data for either the Gamma test or a successful smooth model.

Usage

```
get_Mlist(
  predictors,
  target,
  plot = TRUE,
  caption = "",
  show = "Gamma",
  from = 20,
  to = length(target),
  by = 20
)
```

Arguments

`predictors`	A Numeric vector or matrix whose columns are proposed inputs to a predictive relationship
`target`	A Numeric vector, the output variable that is to be predicted
`plot`	A logical, set this to FALSE if you don't want the plot
`caption`	Character string to be used as caption for the plot

show	Character string, if it equals "vratio", vratios will be plotted, otherwise Gamma is plotted
from	Integer length of the first data sample, as passed to seq
to	Integer maximum length of sample to test, passed to seq
by	Integer increment in lengths of successive windows, passed to seq

Value

An invisible data frame with three columns: M (a sample size), Gamma and the associated vratio. This is ordered by increasing M.

Examples

```
he <- embed(henon_x, 13)
t <- he[ , 1]
p <- he[ ,2:13]
get_Mlist(p, t, by = 2, caption = "this data")
```

---

henon_x *Henon Map*

Description

1000 x data points from the Henon Map

Usage

```
henon_x
```

Format

An object of class `numeric` of length 1000.

References

See Wikipedia entry on "Henon map"

Examples

```
henon_embedded <- embed(as.matrix(henon_x), 3)
targets <- henon_embedded[ ,1]
predictors <- henon_embedded[ ,2:3]
gamma_test(predictors, targets)
```

`increasing_search` *Increasing Embedding Search engine, used by get/plot increasing_search*

Description

Adds variables one at a time to the input set, to see how many are needed for prediction.

Usage

```
increasing_search(
  predictors,
  target,
  plot = TRUE,
  caption = "",
  show = "Gamma"
)
```

Arguments

`predictors`	A vector or matrix whose columns are proposed inputs to a predictive function
`target`	A vector of double, the output variable that is to be predicted
`plot`	Logical, set plot = FALSE if you don't want the plot
`caption`	Character string to identify plot, for example, data being plotted
`show`	Character string, if it equals "vratio", vratios will be plotted, otherwise Gamma is plotted

Details

An increasing embedding search is appropriate when the input variables are ordered, most commonly in analyzing time series, when it's useful to know how many previous time steps or lags should be examined to build a model. Starting with lag 1, the search adds previous values one at a time, and saves the resulting gammas. These results can be examined using plot_increasing_search()

Value

An invisible data frame with three columns, Depth of search, from 1 to ncol(predictors), Gamma calculated using columns 1:Depth as predictors, and vratio corresponding to that Gamma (Gamma / var(target))

Examples

```
he <- embed(henon_x, 13)
t <- he[ , 1]
p <- he[ ,2:13]
increasing_search(p, t, caption = "henon data embedded 16")
df <- increasing_search(predictors=p, target=t, plot = FALSE)
```

int_to_intMask *Integer to Vector Bitmask*

Description

Converts the bit representation of an integer into a vector of integers

Usage

```
int_to_intMask(i, length)
```

Arguments

i	A 32 bit integer
length	Integer length of the bitmask to produce, must be <= 32

Details

Converts an integer to a vector of ones and zeroes. Used as a helper function for full_embedding_search, it allows more compact storage of bit masks. The result reads left to right, so the one bit will have index of one in the vector corresponding to lag 1 in an embedding. Works for masks up to 32 bits

Value

A vector of integer containing 1 or 0

Examples

```
he <- embed(henon_x, 17)
t <- he[ , 1]
p <- he[ ,2:17]
mask <- int_to_intMask(7, 16)     # pick out the first three columns
pn <- select_by_mask(p, mask)
gamma_test(predictors = pn, target = t)
```

mask_histogram *Mask Histogram*

Description

Display a histogram of mask bits.

Usage

```
mask_histogram(fe_result, dimension, tick_step = 2, caption = "")
```

Arguments

fe_result	Output data frame from fe_search. Normally you would filter this by, for example, selecting the top 100 results from that output. If the whole fe_search result was passed in, all of the mask bits would have the same frequency and the histogram would be flat.
dimension	Integer number of effective columns in a mask, ncol of the predictors given to the search
tick_step	Integer, where to put ticks on the x axis
caption	A character string you can use to identify this graph

Details

After a full embedding search, it is sometimes useful to see which bits appear in a subset of the masks, for example, the masks with the lowest Gamma values. Filtering of the search results should be done before calling this function, which uses whatever it is given. The histogram can show which predictors are generally useful. For selecting an effective mask it isn't as useful as you might think - it doesn't show interactions between predictors, for mask selection it would only work for linear combinations of inputs.

Value

A ggplot object, a histogram showing the mask bits used in the fe_search results that are passed to it

Examples

```
e6 <- embed(mgls, 7)
t <- e6[ ,1]
p <- e6[ ,2:7]
full_search <- fe_search(predictors = p, target = t)
goodies <- head(full_search, 20)
mask_histogram(goodies, 6, caption = "mask bits in top 20 Gammas")
baddies <- tail(full_search, 20)
mask_histogram(baddies, 6, caption = "bits appearing in 20 worst Gammas")
```

---

mgls *Mackey-Glass time delayed differential equation*

Description

4999 data points

Usage

```
mgls
```

Format

An object of class numeric of length 4999.

References

See Wikipedia entry on "Mackey-Glass equations"

Examples

```
mgls_embedded <- embed(as.matrix(mgls), 25)
targets <- mgls_embedded[ ,1]
predictors <- mgls_embedded[ ,2:25]
```

---

moving_window_search *Moving Window Search*

Description

Calculate Gamma values for a window moving through the data.

Usage

```
moving_window_search(
  predictors,
  target,
  window_size = 40,
  by = 1,
  plot = TRUE,
  caption = "",
  show = "Gamma"
)
```

Arguments

predictors	A Numeric vector or matrix whose columns are proposed inputs to a predictive function
target	A Numeric vector, the output variable that is to be predicted
window_size	Integer width of the window that will move through the data
by	The increment between successive window starts
plot	Logical, set this to FALSE if you don't want the plot
caption	Character string, caption for plot
show	Character string, if it equals "vratio", vratios will be plotted, otherwise Gamma is plotted

Details

This is used for data sets that are ordered on one or more dimension, such as time series or spatial data. The search slides a window across the data set, calculating gamma for the data at each step. A change in causal dynamics will appear as a spike in gamma when the causal discontinuity is in the window.

Value

An invisible data frame containing starting and ending positions of each window with its associated gamma

Examples

```
he <- embed(henon_x, 13)
t <- he[ , 1]
p <- he[ ,2:13]
moving_window_search(p, t, by = 5, caption = "my data")
```

---

select_by_mask *Select by Mask*

Description

Select columns from a matrix using an integer bitmap

Usage

```
select_by_mask(data, intMask)
```

Arguments

data	A numeric matrix in tidy form
intMask	An Integer vector whose length equals number of columns in data

Details

Selects columns from a matrix. A column is included in the output when the corresponding mask value is 1.

Value

A matrix containing the columns of data for which intMask is 1

Examples

```
e12 <- embed(mgls, 13)
tn <- e12[ , 1]
pn <- e12[ ,2:13]
msk <- integer(12)
msk[c(1,2,3,4,6,7,9)] <- 1  # select these columns
p <- select_by_mask(pn, msk)
gamma_test(predictors = p, target = tn)

msk <- int_to_intMask(15, 12)     # pick out the first four columns
p <- select_by_mask(pn, msk)
gamma_test(predictors = p, target = tn)
```

Index