Package 'locallasso'

August 2, 2021

Type Package

Title What the Package Does (Title Case)

Version 0.1.0

Author Gilles Cattani

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line)
Use four spaces when indenting paragraphs within the Description.

Depends MASS,
RANN,
glmnet,
KSgeneral,
lattice,
plyr,
Matrix,
doParallel,
foreach

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

R topics documented:

ann_search *Ann search function*

Description

Ann search function

Usage

```
ann_search(
  data,
  ll_set,
  search,
  grid = NULL,
  ll_object = NULL,
  threshold = 0.75
)
```

Arguments

data	Data set.
ll_set	Local lasso settings produced by ll_set function.
search	"first" or "second".
grid	Query points to find k-ANN.
ll_object	Global variable relevancy produced by local_lasso function.
threshold	Threshold defining variable relevancy (default = 0.75).

Details

Find the k-ANN for query points.

Value

k-ANN observations number and euclidean distance.

comb *Combine function*

Description

Combine function

Usage

```
comb(...)
```

Details

Combined different object (used in parallel computing).

Value

Combined elements.

`epanechnikov` *epanechnikov function*

Description

epanechnikov function

Usage

```
epanechnikov(u)
```

Arguments

`u`	numerical variable.

Details

Evaluate the value of epanechnikov function at point u.

Value

epanechnikov function at point u

`gen_grid` *Generate grid function*

Description

Generate grid function

Usage

```
gen_grid(ng, ll_set, x_var1, x_var2)
```

Arguments

`ng`	Sequential length of `x_var1` and `x_var2` variables. The resulting grid has `ng*ng` rows.
`ll_set`	Object produced by `ll_setting` function.
`x_var1`	Name of the first variable of interest.
`x_var2`	Name of the second variable of interest.

Details

Generate a grid of length `ng*ng`. Where variables of interest `x_var1` and `x_var2` variables have a sequence of `ng` values and the remaining continuous variables are set to 0. The categorical variable are set to one of the level observed.

Value

Generated grid.

`ll_settings` *Local lasso setting function*

Description

Local lasso setting function

Usage

```
ll_settings(sample, response, k, e, eps = 1)
```

Arguments

`sample`	Random sample drawn from full data set with `sampling` function.
`response`	Name of the response variable.
`k`	Number of approximate nearest neighbours.
`e`	Number of evaluation points.
`eps`	Value of `epsilon` for the ANN-search (default = 1).

Details

Compute a robust estimation of standard deviations. Generate random set of evaluation points.

Value

Settings for local lasso function.

`ll_summary` *Local lasso summary function*

Description

Local lasso summary function

Usage

```
ll_summary(ll_object)
```

Arguments

`ll_object`	Local lasso object produced by the `local_lasso` function.

Details

Display global relevancy scores and KS-statistics for testing uniformity of relevancy points.

Value

Summary of local lasso estimation results.

local_lasso *Local Lasso function*

Description

Local Lasso function

Usage

```
local_lasso(ann, ll_set)
```

Arguments

ann	Approximate nearest neighbours obtained by ann_search function.
ll_set	Object produced by ll_setting function.

Details

A Local Kernel Weighted Linear Regression with LASSO penalty is then fitted to each evaluation points using ann observations. The method uses local bandwidth such that each local model is fitted only with the k-ANN observations of each evaluation points. The value of lambda is selected with cross-validation using the glmnet package.

Value

local_lasso object containing table of global relevence and KS statistic.

Examples

```
library(locallasso)

# # - - - - - - - - - - - - - - - -#
# # Numerical Example
# # - - - - - - - - - - - - - - - -#

n=10^6   # number of observations
nc=19       # number of continuous variables

# Continous variables
X = mvrnorm(n,rep(0,nc),diag(1,nc))

# Categorical variable
D1 = factor(as.numeric(runif(n)<=.5))

# Response variable
y <- -X[,1]^2 - 2*sin((pi/2)*X[,2]) + X[,3]*X[,4] + rnorm(n,0,1)

data = data.frame(D1,X,y)

# Randomly spliting data
ndata=10    # number of distributed dataset
ss <- sample(1:ndata, size=nrow(data),replace=TRUE)
```

```
DD <- list()
for (i in 1:ndata) {
 DD[[i]] = data[ss==i,]
 }

# Random sampling
rand_sample = foreach(i = 1:10, .combine = "rbind") %do% {
 sample = sampling(DD[[i]], ss=1000)
 return(sample)
 }

# Settings
ll_set = ll_settings(rand_sample, response = 'y', k=1000, e=1000, eps = 1)

# First ann search
 first_ann = foreach(i = 1:ndata, .combine = "comb", .packages=c('RANN')) %do% {
   ann1 = ann_search(DD[[i]], ll_set, search='first')
   ann_list = lapply(1:ll_set$e,
                  function(d) cbind(DD[[i]][ann1$nn.idx[d,],], ann1$nn.dists[d,]))
 return(ann_list)
}

# Local Lasso
ll = local_lasso(first_ann, ll_set)

ll_summary(ll)
```

`plot_ll` *Plot function*

Description

Plot function

Usage

```
plot_ll(ann, grid, ll_object, x_var1, x_var2, threshold = 0.75)
```

Arguments

`ann`	Approximate nearest neighbours obtained by `ann_search` function.
`grid`	Grid points to evaluate produced by `gen_grid` function.
`ll_object`	Local lasso object produced by the `local_lasso` function.
`x_var1`	Name of the first variable of interest.
`x_var2`	Name of the second variable of interest.
`threshold`	Threshold defining variable relevancy (default = 0.75).

Details

Predict the value of response for `grid`. Plot of the conditional expectation of response variable given `x_var1` and `x_var2` variables.

Value

Plot of the conditional expectation of response variable given `x_var1` and `x_var2` variables.

`pred_ll` *Predict function*

Description

Predict function

Usage

```
pred_ll(ann, newdata, ll_object, threshold = 0.75)
```

Arguments

`ann`	Approximate nearest neighbours obtained by `ann_search` function.
`newdata`	Set of points of interest to predict
`ll_object`	Local lasso object produced by the `local_lasso` function.
`threshold`	Threshold defining variable relevancy (default = 0.75).

Details

Predict the value of response for new set of point provided.

Value

Prediction of response variable for `newdata`

`sampling` *Sampling function*

Description

Sampling function

Usage

```
sampling(data, ss = 1000, fixed = T, frac = 0.01)
```

Arguments

`data`	Data set.
`ss`	Number of observation to sample from full dataset (default = 1000)
`fixed`	Sample a fixed number of observation (fixed = T) or a fraction of the data set (fixed = F)
`frac`	Fraction of observation to sample from dataset (default = 0.01)

Details

Draw a random sample of observation from data set.

Value

Random sample of observation

Index