Package: nodeHarvest (via r-universe)

November 15, 2026

Type Package

Title Node Harvest for Regression and Classification

Version 0.7.3.9001

Author Nicolai Meinshausen [aut, cre]

Maintainer Nicolai Meinshausen <meinshausen@stat.math.ethz.ch>

Imports graphics, quadprog,randomForest

Description Node harvest is a simple interpretable tree-like estimator for high-dimensional regression and classification. A few nodes are selected from an initially large ensemble of nodes, each associated with a positive weight. New observations can fall into one or several nodes and predictions are the weighted average response across all these groups. The package offers visualization of the estimator. Predictions can return the nodes a new observation fell into, along with the mean response of training observations in each node, offering a simple explanation of the prediction. Compatibility updates by Renova.

License GPL-3

URL http://stat.ethz.ch/~nicolai

NeedsCompilation no

Repository https://renova-r.r-universe.dev

Date/Publication 2026-11-15 20:26:58 UTC

RemoteUrl https://github.com/renova-r/nodeHarvest

RemoteRef renova-0.7.3.9001

RemoteSha e3e00dede6baad5643a6b6d4ce77a1b27b4dd492

Contents

BostonHousing 2
nodeHarvest 3
plot.nodeHarvest 5
predict.nodeHarvest 6

Index **8**

`BostonHousing` *BostonHousing*

Description

Housing data for 506 census tracts of Boston from the 1970 census. The dataframe 'BostonHousing' contains the original data by Harrison and Rubinfeld (1979), the dataframe 'BostonHousing2' the corrected version with additional spatial information (see references below).

Usage

```
data(BostonHousing)
```

Format

The original data are 506 observations on 14 variables, 'medv' being the target variable:

`crim` per capita crime rate by town

`zn` proportion of residential land zoned for lots over 25,000 sq.ft

`indus` proportion of non-retail business acres per town

`chas` Charles River dummy variable (= `1` if tract bounds river; `0` otherwise)

`nox` nitric oxides concentration (parts per 10 million)

`rm` average number of rooms per dwelling

`age` proportion of owner-occupied units built prior to 1940

`dis` weighted distances to five Boston employment centres

`rad` index of accessibility to radial highways

`tax` full-value property-tax rate per USD 10,000

`ptratio` pupil-teacher ratio by town

`b` 1000(B - 0.63)^2 where B is the proportion of blacks by town

`lstat` percentage of lower status of the population

`medv` median value of owner-occupied homes in USD 1000's

Details

The original data have been taken from the UCI Repository Of Machine Learning Databases at

`http://www.ics.uci.edu/~mlearn/MLRepository.html`,

See Statlib and references there for details on the corrections. Converted to R format by Friedrich Leisch.

References

Harrison, D. and Rubinfeld, D.L. (1978). Hedonic prices and the demand for clean air. _Journal of Environmental Economics and Management_, *5*, 81-102.

Gilley, O.W., and R. Kelley Pace (1996). On the Harrison and Rubinfeld Data. _Journal of Environmental Economics and Management_, *31*, 403-405. [Provided corrections and examined censoring.]

Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Pace, R. Kelley, and O.W. Gilley (1997). Using the Spatial Configuration of the Data to Improve Estimation. _Journal of the Real Estate Finance and Economics_, *14*, 333-340. [Added georeferencing and spatial estimation.]

`nodeHarvest` *Node Harvest*

Description

Computes the node harvest estimator

Usage

```
nodeHarvest(X, Y, nodesize = 10,
            nodes = 1000,
    maxinter = 2,
    mode = "mean",
    lambda = Inf,
    addto = NULL,
    onlyinter = NULL,
    silent = FALSE,
    biascorr = FALSE)
```

Arguments

`X`	A n x p - dimensional data matrix, where n is sample size and p is the dimensionality of the predictor variable. Factorial variables are currently converted to numerical variables (will be changed in the future). Missing values are supported.
`Y`	A numerical vector of length n, containing the observations of the response variable. Can be continuous (regression) or binary 0/1 (classification).
`nodesize`	Minimal number of samples in each node.
`nodes`	Number of nodes in the initial large ensemble of nodes.
`maxinter`	Maximal interaction depth (1 = only main effects; 2 = two-factor interactions etc).

mode	If mode is equal to "mean", predictions are weighted group means. If equal to "outbag" (experimental version), the diagonal elements of the smoothing matrix are set to 0.
lambda	Optional upper bound on the inverse of the average weighted fraction of samples within each node.
addto	A previous node harvest estimator to which additional nodes should be attached (useful for iterative growth of the estimator when hitting memory constraints).
onlyinter	Allow interactions only for this list of variables.
silent	If TRUE, no comments are printed.
biascorr	Use bias correction? Experimental. Can be useful for high signal-to-noise ratio data.

Details

The number of nodes should be chosen as large as possible under the available computational resources. If these resources are limited, an estimator can be build by iteratively calling the function, adding the previous estimator via the addto argument.

Feedback and feature requests are more than welcome (email below).

Value

A list with entries

nodes	A list of all selected nodes
predicted	Predicted values on training data
connection	Connectivity matrix between selected nodes (used for plotting)
varnames	Variable names
Y	The original observations

Author(s)

Nicolai Meinshausen <meinshausen@stats.ox.ac.uk>

http://www.stats.ox.ac.uk/~meinshau

References

Node harvest: simple and interpretable regression and classification' (arxiv:0910.2145)

http://arxiv.org/abs/0910.2145

See Also

predict.nodeHarvest, plot.nodeHarvest

Examples

```
## Load Boston Housing dataset
    data(BostonHousing)
    X <- BostonHousing[,1:13]
    Y <- BostonHousing[,14]

## Divide data into training and test data
    n <- nrow(X)
    training <- sample(1:n,round(n/2))
    testing <- (1:n)[-training]


## Train Node Harvest and plot and print the estimator
    NH <- nodeHarvest( X[training,], Y[training], nodes=500 )
    plot(NH)
    print(NH, nonodes=6)

## Predict on test data and explain prediction of the first sample in the test set
    predicttest <- predict(NH, X[testing,], explain=1)
    plot( predicttest, Y[testing] )
```

`plot.nodeHarvest` *plot method for Node Harvest objects*

Description

Node Harvest visualization. Each node with a non-zero weight is plotted (weights is proportional to area of node). The average response in each node is shown on the horizontal axis. The number of observations in each node is shown on the vertical axis.

Usage

```
## S3 method for class 'nodeHarvest'
plot(x,  XTEST = NULL,
  highlight = NULL,
         varnames = NULL,
  yoffset = 0.12,
  labels = "all",
         cexfaclab = 1,  ...)
```

Arguments

`x`	An object of class `nodeHarvest`.
`XTEST`	New observations (for highlighting relevant nodes).
`highlight`	The nodes of this observation in `X` are highlighted and possibly annotated, depending on argument `labels`.

varnames	The variable names can be changed here.
yoffset	The vertical offset in the annotation of interaction nodes.
labels	If 'none', no annotation is made. If 'all', all nodes are annotated. Otherwise the nodes given in highlight are annotated only.
cexfaclab	Character expansion factor for node annotation.
...	Additional arguments passed to plot.

Value

None.

Author(s)

Nicolai Meinshausen <meinshausen@stats.ox.ac.uk>

http://www.stats.ox.ac.uk/~meinshau

See Also

nodeHarvest, plot.nodeHarvest

---

predict.nodeHarvest *predict method for Node Harvest objects*

Description

Given new observations, compute the prediction of a node harvest estimator.

Usage

```
## S3 method for class 'nodeHarvest'
predict(object, newdata = NULL, explain = NULL,
                maxshow = 5,
                weight = sapply(object[["nodes"]], attr, "weight"), ...)
```

Arguments

object	An object of class nodeHarvest.
newdata	A data matrix with predictor variables. If missing, the predictions on the training data are returned.
explain	Row numbers for newdata for which the predictions should be 'explained'. If NULL, no explanation is given.
maxshow	When explaining a prediction, show at most this many nodes (the most important ones).
weight	Optional changed weight vector for the nodes.
...	Additional arguments passed to predict

Details

If explains is equal to NULL, no output is printed. If explain is a numeric vector (with values in 1 to the number of samples in newdata), for each observation in newdata with a sample number in vector explain, the following is done: all nodes that the observation belongs to are printed on screen, along with their node mean (the mean of all training observations who fell into this node) and weight. The prediction for this new observation is the weighted average across these node means. The number of nodes show is given in descending order of their importance (weight) and the number of nodes shown is limited to maxshow.

Value

A numeric vector with the predicted response.

Author(s)

Nicolai Meinshausen <meinshausen@stats.ox.ac.uk>

http://www.stats.ox.ac.uk/~meinshau

See Also

nodeHarvest, plot.nodeHarvest

Index

∗ **datasets**
BostonHousing, 2

BostonHousing, 2

nodeHarvest, 3, *6, 7*

plot.nodeHarvest, *4*, 5, *6, 7*
predict.nodeHarvest, *4*, 6