Package 'HTT'

May 7, 2026

Type Package

Title Hypothesis Testing Tree

Version 0.1.2

Date 2023-03-05

Author Jiaqi Hu [cre, aut],
Zhe Gao [aut],
Bo Zhang [aut],
Xueqin Wang [aut]

Maintainer Jiaqi Hu <hujiaqi@mail.ustc.edu.cn>

Description A novel decision tree algorithm in the hypothesis testing framework. The algorithm examines the distribution difference between two child nodes over all possible binary partitions. The test statistic of the hypothesis testing is equivalent to the generalized energy distance, which enables the algorithm to be more powerful in detecting the complex structure, not only the mean difference. It is applicable for numeric, nominal, ordinal explanatory variables and the response in general metric space of strong negative type. The algorithm has superior performance compared to other tree models in type I error, power, prediction accuracy, and complexity.

License GPL-3

VignetteBuilder knitr

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown, MASS

Imports Rcpp (>= 1.0.6), ggraph, igraph, ggplot2

LinkingTo Rcpp

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-03-12 14:30:02 UTC

Contents

ENB *Energy efficiency dataset*

Description

The data is about energy performance of buildings, containing eight input variables: relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, glazing area distribution and two output variables: heating load (HL) and cooling load (CL) of residential buildings. The goal is to predict two real valued responses from eight input variables. It can also be used as a multi-class classification problem if the response is rounded to the nearest integer.

Usage

```
data("ENB")
```

Format

A data frame with 768 observations on the following 10 variables.

`X1` Relative Compactness

`X2` Surface Area

`X3` Wall Area

`X4` Roof Area

`X5` Overall Height

`X6` Orientation

`X7` Glazing Area

`X8` Glazing Area Distribution

`Y1` Heating Load

`Y2` Cooling Load

Source

UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Energy+efficiency.

References

A. Tsanas, A. Xifara: 'Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools', Energy and Buildings, Vol. 49, pp. 560-567, 2012

Examples

```
data(ENB)
set.seed(1)
idx = sample(1:nrow(ENB), floor(nrow(ENB)*0.8))
train = ENB[idx, ]
test = ENB[-idx, ]
htt_enb = HTT(cbind(Y1, Y2) ~ . , data = train, controls = htt_control(pt = 0.05, R = 99))
# prediction
pred = predict(htt_enb, newdata = test)
test_y = test[, 9:10]
# MAE
colMeans(abs(pred - test_y))
# MSE
colMeans(abs(pred - test_y)^2)
```

---

htt.object *Hypothesis Testing Tree Object*

Description

A class for representing hypothesis testing tree.

Value

`frame` a dataframe about the split information. It contains following information:

`node`: the node numbers in a split order.

`parent`: the parent node number.

`leftChild`: the left daughter node number, `NA` represents leaf.

`rightChild`: the right daughter node number, `NA` represents leaf.

`statistic`: the maximum test statistic of all possible splits within the node.

`split`: the rule of the split. It is numeric for continuous covariate split, it is a character for non-numeric covariate split, the levels of two child nodes are stored in two braces.

`pval`: approximate p-values estimated from permutation test.

`isleaf`: 1 denotes terminal node and 0 denotes internal node.

`n`: the number of observations reaching the node.

`var`: the names of the variables used in the split at each node (leaf nodes are denoted by the label `"<leaf>"`).

`yval`: the fitted value of the response at the node, if the dimension of response is larger than 1, it will presents as yval1, yval2, ...

`prob`: the probability of each class at the node, only visible for classification tree.

where	an integer vector of the same length as the number of observations in the root node, containing the row number of frame corresponding to the leaf node that each observation falls into.
method	the method used to grow the hypothesis testing tree, "regression" or "classification".
control	a list of options that control the HTT algorithm. See htt_control.
X	a copy of the input X in a dataframe format.
var.type	a vector recording for each variables, 0 represents continuous, 1 represents ordinal and 2 represents nominal variables.

See Also

HTT, plot.htt, print.htt, predict.htt

---

htt_control *Control for Hypothesis Testing Tree*

Description

Various parameters that control aspects of the HTT function.

Usage

```
htt_control(teststat = c("energy0", "energy1"),
            testtype = c("permutation", "fastpermutation"),
            alpha = 1, pt = 0.05, minsplit = 30,
            minbucket = round(minsplit/3),
            R = 199, nmin = 1000)
```

Arguments

teststat	a character specifying the type of the test statistic to be applied. It can be teststat = "energy0" or teststat = "energy1". Default is teststat = "energy0".
testtype	a character specifying how to compute the distribution of the test statistic. It can be testtype = "permutation" or testtype = "fastpermutation". For testtype = "fastpermutation", it will not perform the permutation tests on the node with more than nmin observations. Default is testtype = "permutation".
alpha	the exponent on Euclidean distance in (0,2] (for regression tree). Default is alpha = 1.
pt	the p-value of the permutation test must be less than in order to implement a split. If pt = 1, hypothesis testing tree will fully split without performing the permutation tests. Default is pt = 0.05.
minsplit	the minimum number of observations in a node in order to be considered for splitting. Default is minsplit = 30.
minbucket	the minimum number of observations in a terminal node. Default is minbucket = round(minsplit/3).

`R`	the number of permutation replications are used to simulated the distribution of the test statistic. Default is `R = 199`.
`nmin`	the minimum number of observations in a node that does not require the permutation test (for `testtype = "fastpermutation"`). Default is `nmin = 1000`.

Details

The arguments `teststat`, `testtype` and `pt` determine the hypothesis testing of each split. The argument `R` is the number of permutations to be used. For the dataset with more than 2000 observations, `testtype = "fastpermutation"` will be useful to save time.

Value

A list containing the options.

See Also

`HTT`, `htt.object`

Examples

```
## choose the teststat as "energy1"
htt_control(teststat = "energy1")

## choose the p-value 0.01
htt_control(pt = 0.01)

## choose the alpha to 0.5
htt_control(alpha = 0.5)

## change the minimum number of observations in a terminal node
htt_control(minbucket = 7)

## reduce the number of permutation replications to save time
htt_control(R = 99)
```

`Hypothesis Testing Tree`

Hypothesis Testing Tree

Description

Fit a hypothesis testing tree.

Usage

```
HTT(formula, data, method, distance, controls = htt_control(...), ...)
```

Arguments

formula	a symbolic description of the model to be fit.
data	a data frame containing the variables in the model.
method	"regression" or "classification". If method is missing then the routine tries to make an intelligent guess. If Y is factor, then method = "classification". If Y is numeric vector or numeric matrix, then method = "classification".
distance	If distance is missing, then Euclidean distance with exponent alpha is used for regression tree, 0-1 distance is used for classification tree. Otherwise, use the distance as the distance matrix of Y.
controls	a list of options that control details of the HTT algorithm. See htt_control.
...	arguments passed to htt_control.

Details

Hypothesis testing trees examines the distribution difference over two child nodes by the binary partitioning in a hypothesis testing framework. At each split, it finds the maximum distribution difference over all possible binary partitions, the test statistic is based on generalized energy distance. The permutation test is used to estimate the p-value of the hypothesis testing.

Value

An object of class htt. See htt.object.

Author(s)

Jiaqi Hu

See Also

htt_control, print.htt, plot.htt, predict.htt

Examples

```
## regression
data("Boston", package = "MASS")
Bostonhtt <- HTT(medv ~ . , data = Boston, controls = htt_control(R = 99))
plot(Bostonhtt)
mean((Boston$medv - predict(Bostonhtt))^2)

## classification
irishtt <- HTT(Species ~., data = iris)
plot(irishtt)
mean(iris$Species == predict(irishtt))
```

plot.htt *Plot an htt Object*

Description

Visualize a htt object, several arguments can be passed to control the color and shape.

Usage

```
## S3 method for class 'htt'
plot(x, digits = 3,
    line.color = "blue",
    node.color = "black",
    line.type = c("straight", "curved"),
    layout = c("tree", "dendrogram"), ...)
```

Arguments

x	fitted model object of class htt returned by the HTT function.
digits	the number of significant digits in displayed numbers. Default is digits = 3.
line.color	a character specifying the edge color. Default is line.color = "blue".
node.color	a character specifying the node color. Default is node.color = "black".
line.type	a character specifying the type of edge, line.type = "straight" or line.type = "curved". Default is line.type = "straight".
layout	a character specifying the layout, layout = "tree" or layout = "dendrogram". Default is layout = "tree".
...	additional print arguments.

Details

This function is a method for the generic function plot, for objects of class htt.

Value

Visualize the hypothesis testing tree.

See Also

print.htt, printsplit, predict.htt

Examples

```
irishtt = HTT(Species ~., data = iris)
plot(irishtt)

# change the line color and node color
plot(irishtt, line.color = "black", node.color = "blue")

# change the line type
plot(irishtt, line.type = "curved")

# change the layout
plot(irishtt, layout = "dendrogram")
```

`predict.htt` *Predictions from a Fitted htt Object*

Description

Compute predictions from `htt` object.

Usage

```
## S3 method for class 'htt'
predict(object, newdata,
        type = c("response", "prob", "node"),
        ...)
```

Arguments

`object`	fitted model object of class `htt` returned by the `HTT` function.
`newdata`	an optional data frame in which to look for variables with which to predict, if omitted, the fitted values are used.
`type`	a character string denoting the type of predicted value returned. For `type = "response"`, the mean of a numeric response and the predicted class for a categorical response is returned. For `type = "prob"` the matrix of class probabilities is returned for a categorical response. `type = "node"` returns an integer vector of terminal node identifiers.
`...`	additional print arguments.

Details

This function is a method for the generic function `predict` for class `htt`. It can be invoked by calling `predict` for an `object` of the appropriate class, or directly by calling `predict.htt` regardless of the class of the object.

Value

A list of predictions, possibly simplified to a numeric vector, numeric matrix or factor.

If `type = "response"`:
the mean of a numeric response and the predicted class for a categorical response is returned.

If `type = "prob"`:
the matrix of class probabilities is returned for a categorical response.

If `type = "node"`:
an integer vector of terminal node identifiers is returned.

See Also

`predict`, `htt.object`

Examples

```
irishtt <- HTT(Species ~., data = iris)

## the predicted class
predict(irishtt, type = "response")

## class probabilities
predict(irishtt, type = "prob")

## terminal node identifiers
predict(irishtt, type = "node")
```

---

print.htt *Print a Fitted htt Object*

Description

This function prints a `htt.object`. It is a method for the generic function `print` of class `htt`. It can be invoked by calling `print` for an object of the appropriate class, or directly by calling `print.htt` regardless of the class of the object.

Usage

```
## S3 method for class 'htt'
print(x, ...)
```

Arguments

`x`	fitted model object of class `htt` returned by the `HTT` function.
`...`	additional print arguments.

Details

A semi-graphical layout of the contents of x$frame is printed. Indentation is used to convey the tree topology. Information for each node includes the node number,split rule, size and p-value. For the "class" method, the class probabilities are also printed.

Value

Visualize the hypothesis testing tree in a semi-graphical layout.

See Also

htt.object, printsplit

Examples

```
irishtt = HTT(Species ~., data = iris)
print(irishtt)
```

---

printsplit *Displays split table for Fitted htt Object*

Description

Display the split table for fitted htt object.

Usage

```
printsplit(object)
```

Arguments

object	fitted model object of class htt returned by the HTT function.

Value

Display the split table.

See Also

HTT, htt.object

Examples

```
irishtt = HTT(Species ~., data = iris)
printsplit(irishtt)
```

Index