Package: visTree (via r-universe)

September 6, 2024

Title Visualization of Subgroups for Decision Trees

Version 0.8.1

Description Provides a visualization for characterizing subgroups defined by a decision tree structure. The visualization simplifies the ability to interpret individual pathways to subgroups; each sub-plot describes the distribution of observations within individual terminal nodes and percentile ranges for the associated inner nodes.

Depends R (>= 3.4.0)

License GPL-3

Encoding UTF-8

Imports partykit, rpart, colorspace

LazyData true

RoxygenNote 6.1.0

Suggests covr, knitr, rmarkdown, testthat

VignetteBuilder knitr

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

RemoteUrl https://github.com/ashwinikv/vistree

RemoteRef HEAD

RemoteSha 0faebdfb853f827154e38bc57a88c132c1838c31

Contents

blsdata 2
l_node 3
makeTransparent 3
minmax_mat 4
path_node 4
plot_minmax 5
ptree_criteria 6
ptree_left 6

ptree_right 7
ptree_y 7
trim 7
visTree 8

Index **10**

`blsdata` *Box Lunch Study - Baseline dataset*

Description

The variables are as follows:

Usage

```
data(blsdata)
```

Format

A data frame with 226 rows and 26 variables

Details

- trt. Treatment
- sex. Sex
- bmi0. BMI
- snackkcal0. Snacking kilo calories
- srvgfv0. Serving size of fruits and vegetables
- srvgssb0. Serving size of beverages
- kcal24h0.
- edeq01.
- edeq02.
- edeq13.
- edeq14.
- edeq15.
- edeq22.
- edeq23.
- edeq25.
- edeq26.
- cdrsbody0. Body image
- weighfreq0. Weighing frequency
- freqff0. Fast food frequency

- age. Age
- tfactor1.
- tfactor2.
- tfactor3.
- mlhfbias0.
- fwahfbias0.
- rrvfood. Relative reinforcement of food

Examples

```
data(blsdata)
```

`l_node` *Function for determining a pathway*

Description

Decision tree structure

Usage

```
l_node(newtree, node_id = 1, start_criteria = character(0))
```

Arguments

`newtree`	Decision tree generated as a party object
`node_id`	Node ID
`start_criteria`	Character vector

`makeTransparent` *Color Scheme*

Description

Function to adjust the transparency and define the color scheme within the visualization.

Usage

```
makeTransparent(colortype, alpha)
```

Arguments

`colortype`	Color palette
`alpha`	Transparency

`minmax_mat` *Minmax matrix*

Description

Identifies splits and relevant criteria

Usage

```
minmax_mat(str, varnms, Y, interval)
```

Arguments

`str`	Structure of pathway from the root node in the decision tree to each terminal node
`varnms`	Names of covariates
`Y`	Response variable in the dataset
`interval`	logical. Continuous response (interval = FALSE) and Categorical response (interval = TRUE).

`path_node` *Function for determining a pathway*

Description

Generates the pathway from the root node to individual terminal nodes of a decision tree generated as a party object using the partykit package.

Usage

```
path_node(newtree, idnumber = 0)
```

Arguments

`newtree`	Decision tree generated as a party object
`idnumber`	Terminal ID number

`plot_minmax` *Generate individual subplots within the graphical visualization*

Description

This function is utilized to generate a series of sub-plots, where each subplot corresponds to individual terminal nodes within the decision tree structure. Each subplot is composed of a histogram (or a barchart) that displays the distribution for the relevant subgroup and colored horizontal bars that summarize the set of covariate splits.

Usage

```
plot_minmax(My, X, Y, str, color.type, alpha, add.p.axis, add.h.axis,
  cond.tree, text.main, text.bar, text.round, text.percentile,
  density.line, text.title, text.axis, text.label)
```

Arguments

`My`	A matrix to define the split points within the decision tree structure
`X`	Covariates
`Y`	Response variable
`str`	Structure of pathway from the root node in the decision tree to each terminal node
`color.type`	Color palettes. (rainbow_hcl = 1; heat_hcl = 2; terrain_hcl = 3; sequential_hcl = 4; diverge_hcl = 5)
`alpha`	Transparency of individual horizontal bars. Choose values between 0 to 1.
`add.p.axis`	logical. Add axis for the percentiles (add.p.axis = TRUE), remove axis for the percentiles (add.p.axis = FALSE).
`add.h.axis`	logical. Add axis for the outcome (add.h.axis = TRUE), remove axis for the outcome (add.h.axis = FALSE).
`cond.tree`	Tree as a party object
`text.main`	Change the size of the main titles
`text.bar`	Change the size of the text in the horizontal bar and below the bar plot
`text.round`	Round the threshold displayed on the bar
`text.percentile`	Change the size of the percentile title
`density.line`	Draw a density line
`text.title`	Change the size of the text in the title
`text.axis`	Change the size of the text of axis labels
`text.label`	Change the size of the axis annotation

`ptree_criteria` *Splitting Criteria*

Description

Identifies the splitting criteria for the relevant node leading to lower level inner nodes or a terminal node.

Usage

```
ptree_criteria(newtree, node_id, left)
```

Arguments

`newtree`	Decision tree
`node_id`	Node id
`left`	Splits to the left

`ptree_left` *Left split*

Description

Identifies a node that corresponds to the left split

Usage

```
ptree_left(newtree, start_id)
```

Arguments

`newtree`	Decision tree generated as a party object
`start_id`	Character vector

`ptree_right` *Right Split*

Description

Identifies a node that corresponds to the right split

Usage

```
ptree_right(newtree, start_id)
```

Arguments

`newtree`	Decision tree generated as a party object
`start_id`	Character vector

`ptree_y` *Function for determining a pathway*

Description

Identifies the predicted outcome value for the relevant node.

Usage

```
ptree_y(newtree, node_id)
```

Arguments

`newtree`	Decision tree generated as a party object
`node_id`	Node ID

`trim` *Function for determining a pathway*

Description

Parsing function

Usage

```
trim(x)
```

Arguments

`x`	String

`visTree` *Visualization of subgroups for decision trees*

Description

This visualization characterizes subgroups defined by a decision tree structure and identifies the range of covariate values associated with outcome values in each subgroup.

Usage

```
visTree(cond.tree, rng = NULL, interval = FALSE, color.type = 1,
  alpha = 0.5, add.h.axis = TRUE, add.p.axis = TRUE,
  text.round = 1, text.main = 1.5, text.bar = 1.5,
  text.title = 1.5, text.label = 1.5, text.axis = 1.5,
  text.percentile = 0.7, density.line = TRUE)
```

Arguments

`cond.tree`	Decision tree generated as a party object.
`rng`	Restrict plotting to a particular set of nodes. Default value is set as NULL.
`interval`	logical. Continuous outcome (interval = FALSE) and Categorical outcome (interval = TRUE).
`color.type`	Color palettes (rainbow_hcl = 1; heat_hcl = 2; terrain_hcl = 3; sequential_hcl = 4 ; diverge_hcl = 5)
`alpha`	Transparency for horizontal colored bars in each subplot. Values between 0 to 1.
`add.h.axis`	logical. Add axis for the outcome distribution (add.h.axis = TRUE), remove axis for the outcome (add.h.axis = FALSE).
`add.p.axis`	logical. Add axis for the percentiles (add.p.axis = TRUE) computed over covariate values, remove axis for the percentiles (add.p.axis = FALSE).
`text.round`	Round the threshold displayed on the horizontal bar
`text.main`	Change the size of the main titles
`text.bar`	Change the size of the text in the horizontal bar
`text.title`	Change the size of the text in the title
`text.label`	Change the size of the axis annotation
`text.axis`	Change the size of the text of axis labels
`text.percentile`	Change the size of the percentile title
`density.line`	logical. Draw a density line. (density.line = TRUE).

Author(s)

Ashwini Venkatasubramaniam and Julian Wolfson

Examples

```
data(blsdata)
newblsdata<-blsdata[,c(7,21, 22,23, 24, 25, 26)]
## Continuous response
ptree1<-partykit::ctree(kcal24h0~., data = newblsdata)
visTree(ptree1, text.axis = 1.3, text.label = 1.2, text.bar = 1.2, alpha = 0.5)

## Repeated covariates in the splits of the decision tree
ptree2<-partykit::ctree(kcal24h0~skcal+rrvfood+resteating+age, data = blsdata)
visTree(ptree2, text.axis = 1.3, text.label = 1.2, text.bar = 1.2, alpha = 0.5)

## Categorical response
blsdataedit<-blsdata[,-7]
blsdataedit$bin<-0
blsdataedit$bin<-cut(blsdata$kcal24h0, unique(quantile(blsdata$kcal24h0)),
include.lowest = TRUE, dig.lab = 4)
names(blsdataedit)[26]<-"kcal24h0"
ptree3<-partykit::ctree(kcal24h0~hunger+rrvfood+resteating+liking, data = blsdataedit)
visTree(ptree3, interval = TRUE,  color.type = 1, alpha = 0.6,
text.percentile = 1.2, text.bar = 1.8)

## Other decision trees (e.g., rpart)
ptree4<-rpart::rpart(kcal24h0~wanting+liking+rrvfood, data = newblsdata,
control = rpart::rpart.control(cp = 0.029))
visTree(ptree4, text.bar = 1.8, text.label = 1.4, text.round = 1,
density.line = TRUE, text.percentile = 1.3)

## Change the color scheme and transparency of the horizontal bars
ptree1<-partykit::ctree(kcal24h0~., data = newblsdata)
visTree(ptree1, text.axis = 1.3, text.label = 1.2, text.bar = 1.2, alpha = 0.65,
color.type = 3)

## Remove the axes corresponding to the percentiles and the response values.
ptree1<-partykit::ctree(kcal24h0~., data = newblsdata)
visTree(ptree1, text.axis = 1.3, text.label = 1.2, text.bar = 1.2, alpha = 0.65,
color.type = 3, add.p.axis = FALSE, add.h.axis = FALSE)

# Remove the density line over the histograms
ptree1<-partykit::ctree(kcal24h0~., data = newblsdata)
visTree(ptree1, text.axis = 1.3, text.label = 1.2, text.bar = 1.2, alpha = 0.65,
color.type = 3, density.line = FALSE)
```

Index

∗ **datasets**
  - blsdata, 2

∗ **decision**
  - l_node, 3
  - makeTransparent, 3
  - minmax_mat, 4
  - path_node, 4
  - plot_minmax, 5
  - ptree_criteria, 6
  - ptree_left, 6
  - ptree_right, 7
  - ptree_y, 7
  - trim, 7
  - visTree, 8

∗ **matrix**
  - plot_minmax, 5

∗ **pathway**
  - l_node, 3
  - makeTransparent, 3
  - minmax_mat, 4
  - path_node, 4
  - plot_minmax, 5
  - ptree_criteria, 6
  - ptree_left, 6
  - ptree_right, 7
  - ptree_y, 7
  - trim, 7
  - visTree, 8

∗ **tree**
  - l_node, 3
  - makeTransparent, 3
  - minmax_mat, 4
  - path_node, 4
  - plot_minmax, 5
  - ptree_criteria, 6
  - ptree_left, 6
  - ptree_right, 7
  - ptree_y, 7
  - trim, 7
  - visTree, 8

∗ **visualization**
  - minmax_mat, 4
  - visTree, 8

blsdata, 2

l_node, 3

makeTransparent, 3
minmax_mat, 4

path_node, 4
plot_minmax, 5
ptree_criteria, 6
ptree_left, 6
ptree_right, 7
ptree_y, 7

trim, 7

visTree, 8