Package ‘xgrove’

January 10, 2025

Title Explanation Groves

Version 0.1-15

Date 2025-01-04

Maintainer Gero Szepannek <gero.szepannek@web.de>

Description Compute surrogate explanation groves for predictive machine learning models and analyze complexity vs. explanatory power of an explanation according to Szepannek, G. and von Holt, B. (2023) <doi:10.1007/s41237-023-00205-2>.

License GPL (>= 2)

Encoding UTF-8

Imports gbm, dplyr, rpart, rpart.plot, ggplot2

Suggests pdp, randomForest

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Gero Szepannek [aut, cre] (<https://orcid.org/0000-0001-8456-1283>)

Repository CRAN

Date/Publication 2025-01-10 09:00:01 UTC

Contents

plot.sgtree *Plot surrogate tree statistics*

Description

Plot statistics of surrogate trees to analyze complexity vs. explanatory power.

Usage

```
## S3 method for class 'sgtree'
plot(x, abs = "rules", ord = "upsilon", ...)
```

Arguments

x	An object of class sgtree.
abs	Name of the measure to be plotted on the x-axis, either "trees", "rules", "upsilon" or "cor".
ord	Name of the measure to be plotted on the y-axis, either "trees", "rules", "upsilon" or "cor".
...	Further arguments passed to plot.

Value

No return value.

Author(s)

<gero.szepannek@web.de>

Examples

```
library(randomForest)
library(pdp)
data(boston)
set.seed(42)
rf <- randomForest(cmedv ~ ., data = boston)
data <- boston[,-3] # remove target variable
ntrees <- c(4,8,16,32,64,128)
xg <- xgrove(rf, data, ntrees)
xg
plot(xg)
```

plot.xgrove — *Plot surrogate grove statistics*

Description

Plot statistics of surrogate groves to analyze complexity vs. explanatory power.

Usage

```
## S3 method for class 'xgrove'
plot(x, n.trees = NULL, abs = "rules", ord = "upsilon", ...)
```

Arguments

`x`	An object of class `xgrove`.
`n.trees`	Number of trees in case the effects of a grove should be visualized and `abs` and `ord` are ignored. If `NULL` a screeplot of complexity vs explanation is shown for `abs` vs. `ord`.
`abs`	Name of the measure to be plotted on the x-axis, either `"trees"`, `"rules"`, `"upsilon"` or `"cor"`.
`ord`	Name of the measure to be plotted on the y-axis, either `"trees"`, `"rules"`, `"upsilon"` or `"cor"`.
`...`	Further arguments passed to `plot`.

Value

No return value.

Author(s)

<gero.szepannek@web.de>

Examples

```
library(randomForest)
library(pdp)
data(boston)
set.seed(42)
rf <- randomForest(cmedv ~ ., data = boston)
data <- boston[,-3] # remove target variable
ntrees <- c(4,8,16,32,64,128)
xg <- xgrove(rf, data, ntrees)
xg
plot(xg)

# alternatively, visualize weights for the grove of size 8:
plot(xg, n.trees = 8)
```

sgtree *Surrogate trees*

Description

Compute surrogate trees of different depth to explain predictive machine learning model and analyze complexity vs. explanatory power.

Usage

```
sgtree(model, data, maxdeps = 1:8, cparam = 0, pfun = NULL, ...)
```

Arguments

`model`	A model with corresponding predict function that returns numeric values.
`data`	Data that must not (!) contain the target variable.
`maxdeps`	Sequence of integers: Maximum depth of the trees.
`cparam`	Complexity parameter for growing the trees.
`pfun`	Optional predict function `function(model, data)` returning a real number. Default is the `predict()` method of the `model`.
`...`	Further arguments to be passed to `rpart.control` or the `predict()` method of the `model`.

Details

A surrogate grove is trained via gradient boosting using `rpart` on `data` with the predictions of using of the `model` as target variable. Note that `data` must not contain the original target variable!

Value

List of the results:

`explanation`	Matrix containing tree sizes, rules, explainability Υ and the correlation between the predictions of the explanation and the true model.
`rules`	List of rules for each tree.
`model`	List of the `rpart` models.

Author(s)

<gero.szepannek@web.de>

References

- Szepannek, G. and Laabs, B.H. (2023): Can't see the forest for the trees – analyzing groves to explain random forests, Behaviormetrika, submitted.
- Szepannek, G. and Luebke, K.(2023): How much do we see? On the explainability of partial dependence plots for credit risk scoring, Argumenta Oeconomica 50, DOI: 10.15611/aoe.2023.1.07.

Examples

```
library(randomForest)
library(pdp)
data(boston)
set.seed(42)
rf    <- randomForest(cmedv ~ ., data = boston)
data  <- boston[,-3] # remove target variable
maxds <- 1:7
st    <- sgtree(rf, data, maxds)
st
# rules for tree of depth 3
st$rules[["3"]]
# plot tree of depth 3
rpart.plot::rpart.plot(st$model[["3"]])
```

---

upsilon *Explainability*

Description

Compute explainability given predicted data of the model and an explainer.

Usage

```
upsilon(porig, pexp)
```

Arguments

`porig`	An object of class `xgrove`.
`pexp`	Name of the measure to be plotted on the x-axis, either `"trees"`, `"rules"`, `"upsilon"` or `"cor"`.

Value

Numeric explainability upsilon.

Author(s)

`<gero.szepannek@web.de>`

References

- Szepannek, G. and Luebke, K.(2023): How much do we see? On the explainability of partial dependence plots for credit risk scoring, Argumenta Oeconomica 50, DOI: 10.15611/aoe.2023.1.07.

Examples

```
library(randomForest)
library(pdp)
data(boston)
set.seed(42)
# Compute original model
rf <- randomForest(cmedv ~ ., data = boston)
data <- boston[,-3] # remove target variable
# Compute predictions
porig <- predict(rf, data)

# Compute surrogate grove
xg <- xgrove(rf, data)
pexp <- predict(xg$model, data, n.trees = 16)
upsilon(porig, pexp)
```

`xgrove` *Explanation groves*

Description

Compute surrogate groves to explain predictive machine learning model and analyze complexity vs. explanatory power.

Usage

```
xgrove(
  model,
  data,
  ntrees = c(4, 8, 16, 32, 64, 128),
  pfun = NULL,
  remove.target = T,
  shrink = 1,
  b.frac = 1,
  seed = 42,
  ...
)
```

Arguments

`model`	A model with corresponding predict function that returns numeric values.
`data`	Training data.
`ntrees`	Sequence of integers: number of boosting trees for rule extraction.
`pfun`	Optional predict function `function(model, data)` returning a real number. Default is the `predict()` method of the `model`.

remove.target	Logical. If TRUE the name of the target variable is identified from terms(model) and automatically removed if this variable is still in data.
shrink	Sets the shrinkage argument for the internal call of gbm. As the model usually has a deterministic response the default is 1 different to the default of gbm applied train a model based on data.
b.frac	Sets the bag.fraction argument for the internal call of gbm. As the model usually has a deterministic response the default is 1 different to the default of gbm applied train a model based on data.
seed	Seed for the random number generator to ensure reproducible results (e.g. for the default bag.fraction < 1 in boosting).
...	Further arguments to be passed to gbm or the predict() method of the model.

Details

A surrogate grove is trained via gradient boosting using gbm on data with the predictions of using of the model as target variable. Note that data must not contain the original target variable! The boosting model is trained using stumps of depth 1. The resulting interpretation is extracted from pretty.gbm.tree. The column upper_bound_left of the rules and the groves value of the output object contains the split point for numeric variables denoting the uppoer bound of the left branch. Correspondingly, the levels_left column contains the levels of factor variables assigned to the left branch. The rule weights of the branches are given in the rightmost columns. The prediction of the grove is obtained as the sum of the assigned weights over all rows. Note that the training data must not contain the target variable. It can be either removed manually or will be removed automatically from data if the argument remove.target == TRUE.

Value

List of the results:

explanation	Matrix containing tree sizes, rules, explainability Υ and the correlation between the predictions of the explanation and the true model.
rules	Summary of the explanation grove: Rules with identical splits are aggegated. For numeric variables any splits are merged if they lead to identical parititions of the training data.
groves	Rules of the explanation grove.
model	gbm model.

Author(s)

<gero.szepannek@web.de>

References

- Szepannek, G. and von Holt, B.H. (2023): Can't see the forest for the trees – analyzing groves to explain random forests, Behaviormetrika, DOI: 10.1007/s41237-023-00205-2.
- Szepannek, G. and Luebke, K.(2023): How much do we see? On the explainability of partial dependence plots for credit risk scoring, Argumenta Oeconomica 50, DOI: 10.15611/aoe.2023.1.07.

Examples

```
library(randomForest)
library(pdp)
data(boston)
set.seed(42)
rf <- randomForest(cmedv ~ ., data = boston)
data <- boston[,-3] # remove target variable
ntrees <- c(4,8,16,32,64,128)
xg <- xgrove(rf, data, ntrees)
xg
plot(xg)

# Example of a classification problem using the iris data.
# A predict function has to be defined, here for the posterior probabilities of the class Virginica.
data(iris)
set.seed(42)
rf    <- randomForest(Species ~ ., data = iris)
data  <- iris[,-5] # remove target variable

pf <- function(model, data){
  predict(model, data, type = "prob")[,3]
  }

xgrove(rf, data, pfun = pf)
```

Index