Package ‘edarf’

March 6, 2017

Version 1.1.1
Date 2017-03-05
Title Exploratory Data Analysis using Random Forests
Description Functions useful for exploratory data analysis using random forests which can be used to compute multivariate partial dependence, observation, class, and variable-wise marginal and joint permutation importance as well as observation-specific measures of distance (supervised or unsupervised). All of the aforementioned functions are accompanied by ‘ggplot2’ plotting functions.

Author Zachary M. Jones <zmj@zmjones.com> and Fridolin Linder <fridolin.linder@gmail.com>

Maintainer Zachary M. Jones <zmj@zmjones.com>
License MIT + file LICENSE
Depends R (>= 2.10)
Imports data.table, ggplot2, mmpf
Suggests party, randomForest, randomForestSRC, ranger, testthat, rmarkdown, knitr
LazyData true

BugReports https://github.com/zmjones/edarf
RoxygenNote 6.0.1
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2017-03-06 08:28:57

R topics documented:

extract_proximity .................................................. 2
partial_dependence ................................................ 3
plot_imp ............................................................ 4
extract_proximity

Methods to extract proximity matrices from random forests

Description

Extracts proximity matrices from random forest objects from the party, randomForest, randomForestSRC, or ranger packages

Usage

`extract_proximity(fit, newdata)`

Arguments

- `fit` : object of class 'RandomForest', 'randomForest', 'rfsrc', or 'ranger'
- `newdata` : new data with the same columns as the data used for fit
- `...` : arguments to be passed to `extract_proximity`

Value

an n x n matrix where position i, j gives the proportion of times observation i and j are in the same terminal node across all trees

See Also

- `plot_prox` for plotting principal components of proximity matrices.

Examples

```r
library(randomForest)

fit = randomForest(hp ~ ., mtcars, proximity = TRUE)
exttract_proximity(fit)

fit = randomForest(Species ~ ., iris, proximity = TRUE)
extrant_proximity(fit)
```
Partial dependence using random forests

Description

Calculates the partial dependence of the response on an arbitrary dimensional set of predictors from a fitted random forest object from the party, randomForest, randomForestSRC, or ranger packages.

Usage

partial_dependence(fit, vars, n, interaction, uniform, data, ...)

Arguments

fit object of class 'RandomForest', 'randomForest', 'rfsrc', or 'ranger'
vars a character vector of the predictors of interest
n two dimensional integer vector giving the resolution of the grid. the first element gives the grid on vars and the second on the other columns, which are subsampled.
interaction logical, if 'vars' is a vector, does this specify an interaction or a list of bivariate partial dependence
uniform logical, indicates whether a uniform or random grid is to be construct partial dependence calculation
data the data.frame used to fit the model, only needed for 'randomForest'
... additional arguments to be passed to marginalPrediction

Value

da.data.frame with the partial dependence of 'vars' if 'vars' has length = 1 then the output will be a data.frame with a column for the predicted value at each value of 'vars', averaged over the values of all other predictors. if 'vars' has length > 1 and interaction is true or false then the output will be a data.frame with a column for each element of 'vars' and the predicted value for each combination.

References


See Also

plot_pd for plotting partial_dependence.
plot_imp

Plot variable importance from random forests

Description
Plot variable importance from random forests

Usage
plot_imp(imp, sort = "decreasing")

Arguments

imp object of class c("importance", "data.frame") as returned by variable_importance

sort character indicating if sorting of the output is to be done. can be "ascending", or "descending."

Value
a ggplot2 object

Examples
library(randomForest)
data(iris)
fit = randomForest(Species ~ ., iris)
imp = variable_importance(fit, nperm = 2, data = iris)
plot_imp(imp)

Examples
library(randomForest)
data(iris)
fit = randomForest(Species ~ ., iris)
imp = variable_importance(fit, nperm = 2, data = iris)
plot_imp(imp)
plot_pd

Plot partial dependence from random forests

Description

Plot partial dependence from random forests

Usage

plot_pd(pd, facet = NULL)

Arguments

pd

object of class c("pd", "data.frame") as returned by partial_dependence

facet

a character vector indicating the variable that should be used to facet on if interaction is plotted. If not specified the variable with less unique values is chosen.

Value

a ggplot2 object

Examples

library(randomForest)
library(edarf)
data(iris)
fit = randomForest(Species ~ ., iris)
pd = partial_dependence(fit, "Petal.Width", data = iris)
plot_pd(pd)

plot_pred

Plot predicted versus observed values

Description

Plot predicted versus observed values

Usage

plot_pred(predicted, observed, perfect_line = TRUE, outlier_idx = NULL, labs = NULL, xlab = "Observed", ylab = "Predicted", title = "")
**Arguments**

- **predicted**: numeric vector of predictions
- **observed**: numeric vector of observations
- **perfect_line**: logical whether to plot a blue 45 degree line on which perfect predictions would fall
- **outlier_idx**: integer indices of outliers to be labelled between the predicted and observed value pairs are labeled an outlier
- **labs**: character labels for points, applied to a subset determined by the 'outlier_criterion'
- **xlab**: character label for the x-axis, defaults to "Observed"
- **ylab**: character label for the y-axis, defaults to "Predicted"
- **title**: character title defaults to ""

**Value**

a ggplot object

**Examples**

```r
library(randomForest)
library(edarf)
fit = randomForest(hp ~ ., mtcars)
pred = predict(fit, newdata = mtcars)
plot_pred(pred, mtcars$hp,
  outlier_idx = which(abs(pred - mtcars$hp) > .5 * sd(mtcars$hp)),
  labs = row.names(mtcars))
```

---

**plot_prox**

*Plot principle components of the proximity matrix*

**Description**

Plot principle components of the proximity matrix

**Usage**

```
plot_prox(pca, dims = 1:2, labels = NULL, alpha = 1, alpha_label = NULL,
  color = "black", color_label = NULL, shape = "1", shape_label = NULL,
  size = 2, size_label = NULL, xlab = NULL, ylab = NULL, title = "")
```
plot_prox

Arguments

pca  a prcomp object, pca of an n x n matrix giving the proportion of times across all trees that observation i,j are in the same terminal node

dims  integer vector of length 2 giving indices for the dimensions of pca to be plotted

labels  length n character vector giving observation labels

alpha  optional continuous vector of length n make points/labels transparent or a numeric of length 1 giving the alpha of all points/labels

alpha_label  character legend title if alpha parameter used

color  optional discrete vector of length n which colors the points/labels or a character vector giving the color of all points/labels

color_label  character legend title if color parameter is used

shape  optional discrete vector of length n which shapes points (not applicable if labels used) or a character vector of length 1 which gives the shape of all points

shape_label  character legend title if shape parameter is used

size  optional continuous vector of length n which sizes points or labels or a numeric of length 1 which gives the sizes of all the points

size_label  character legend title if size parameter used

xlab  character x-axis label

ylab  character y-axis label

title  character plot title

Value

a ggplot object

References

https://github.com/vqv/ggbiplot

Gabriel, "The biplot graphic display of matrices with application to principal component analysis," Biometrika, 1971

Examples

library(randomForest)

fit = randomForest(hp ~ ., mtcars, proximity = TRUE)
prox = extract_proximity(fit)
pca = prcomp(prox, scale = TRUE)
plot_prox(pca, labels = row.names(mtcars))

fit = randomForest(Species ~ ., iris, proximity = TRUE)
prox = extract_proximity(fit)
pca = prcomp(prox, scale = TRUE)
plot_prox(pca, color = iris$Species, color_label = "Species", size = 2)
variable_importance

Variable importance using random forests

Description
Computes local or aggregate variable importance for a set of predictors from a fitted random forest object from the party, randomForest, randomForestSRC, or ranger package.

Usage
variable_importance(fit, vars, interaction, nperm, data, ...)

Arguments
- fit: object of class ’RandomForest’, ’randomForest’, ’rfsr’, or ’ranger’
- vars: character, variables to find the importance of
- interaction: logical, compute the joint and additive importance for observations (type = “local”) or variables type = “aggregate”
- nperm: positive integer giving the number of times to permute the indicated variables (default 10)
- data: optional (unless using randomForest) data.frame with which to calculate importance
- ...: additional arguments to be passed to permutationImportance.

Value
a named list of vars with the return from permutationImportance for each.

References

See Also
plot_imp for plotting the results of variable_importance.

Examples
library(randomForest)
data(iris)
fit = randomForest(Species ~ ., iris)
variable_importance(fit, nperm = 2, data = iris)
Index

extract_proximity, 2

partial_dependence, 3, 5
plot_imp, 4, 8
plot_pd, 3, 5
plot_pred, 5
plot_prox, 2, 6

variable_importance, 4, 8