Package: PPforest (via r-universe)

September 5, 2024

Type Package

Version 0.1.3

Title Projection Pursuit Classification Forest

Author Natalia da Silva, Eun-Kyung Lee, Di Cook

Maintainer Natalia da Silva <natalia.dasilva@fcea.edu.uy>

Description Implements projection pursuit forest algorithm for
supervised classification.

License GPL (>= 2)

URL https://github.com/natydasilva/PPforest

LazyData yes

Depends R (>= 3.2.0)

Imports Rcpp (>= 0.12.7), magrittr, plyr, dplyr (>= 0.7.5), tidyr,
doParallel, tibble

Suggests knitr, gridExtra, GGally, ggplot2, RColorBrewer, roxygen2 (>=
3.0.0), PPtreeViz, rmarkdown

VignetteBuilder knitr

LinkingTo Rcpp,RcppArmadillo

Encoding UTF-8

RoxygenNote 7.3.2

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

RemoteUrl https://github.com/natydasilva/ppforest

RemoteRef HEAD

RemoteSha 6e37d833bf1552f6c75b9294d9541df958e461a0

Contents

baggtree 2
crab 3
fishcatch 4

glass 5
image 5
leukemia 6
lymphoma 7
NCI60 8
node_data 8
olive 9
parkinson 10
permute_importance 11
PPclassify2 11
PPforest 12
ppf_avg_imp 14
ppf_global_imp 15
PPtree_split 16
print.PPforest 17
ternary_str 17
trees_pred 18
wine 19

Index **20**

`baggtree` *For each bootstrap sample grow a projection pursuit tree (PPtree object).*

Description

For each bootstrap sample grow a projection pursuit tree (PPtree object).

Usage

```
baggtree(
  data,
  class,
  m = 500,
  PPmethod = "LDA",
  lambda = 0.1,
  size.p = 1,
  parallel = FALSE,
  cores = 2
)
```

Arguments

`data`	Data frame with the complete data set.
`class`	A character with the name of the class variable.

`m` is the number of bootstrap replicates, this corresponds with the number of trees to grow. To ensure that each observation is predicted a few times we have to select this number no too small. `m = 500` is by default.

`PPmethod` is the projection pursuit index to be optimized, options LDA or PDA, by default it is LDA.

`lambda` a parameter for PDA index

`size.p` proportion of random sample variables in each split.

`parallel` logical condition, if it is TRUE then parallelize the function

`cores` number of cores used in the parallelization

Value

data frame with trees_pp output for all the bootstraps samples.

Examples

```
#crab data set
crab.trees <- baggtree(data = crab, class = 'Type',
m =  200, PPmethod = 'LDA', lambda = .1, size.p = 0.5 , parallel = TRUE, cores = 2)
str(crab.trees, max.level = 1)
```

`crab` *Astralian crabs*

Description

Measurements on rock crabs of the genus Leptograpsus. The data set contains 200 observations from two species of crab (blue and orange), there are 50 specimens of each sex of each species, collected on site at Fremantle, Western Australia.

Type is the class variable and has 4 classes with the combinations of specie and sex (BlueMale, BlueFemale, OrangeMale and OrangeFemale).

FL the size of the frontal lobe length, in mm

RW rear width, in mm

CL length of midline of the carapace, in mm

CW maximum width of carapace, in mm

BD depth of the body; for females, measured after displacement of the abdomen, in mm

Usage

```
data(crab)
```

Format

A data frame with 200 rows and 6 variables

Source

Campbell, N. A. & Mahon, R. J. (1974), A Multivariate Study of Variation in Two Species of Rock Crab of genus Leptograpsus, Australian Journal of Zoology 22(3), 417 - 425.

fishcatch *Fish catch data set*

Description

There are 159 fishes of 7 species are caught and measured. Altogether there are 7 variables. All the fishes are caught from the same lake(Laengelmavesi) near Tampere in Finland.

Type has 7 fish classes, with 35 cases of Bream, 11 cases of Parkki, 56 cases of Perch 17 cases of Pike, 20 cases of Roach, 14 cases of Smelt and 6 cases of Whitewish.

weight Weight of the fish (in grams)

length1 Length from the nose to the beginning of the tail (in cm)

length2 Length from the nose to the notch of the tail (in cm)

length3 Length from the nose to the end of the tail (in cm)

height Maximal height as % of Length3

width Maximal width as % of Length3

Usage

```
data(fishcatch)
```

Format

A data frame with 159 rows and 7 variables

Source

[http://www.amstat.org/publications/jse/jse_data_archive.htm](fishcatch)

`glass` *Glass data set*

Description

Contains measurements 214 observations of 6 types of glass; defined in terms of their oxide content.

Type has 6 types of glasses

X1 refractive index

X2 Sodium (unit measurement: weight percent in corresponding oxide).

X3 Magnesium

X4 Aluminum

X5 Silicon

X6 Potassium

X7 Calcium

X8 Barium

X9 Iron

Usage

```
data(glass)
```

Format

A data frame with 214 rows and 10 variables

`image` *The image data set*

Description

contains 2310 observations of instances from 7 outdoor images

Type has 7 types of outdoor images, brickface, cement, foliage, grass, path, sky, and window.

X1 the column of the center pixel of the region

X2 the row of the center pixel of the region.

X3 the number of pixels in a region = 9.

X4 the results of a line extraction algorithm that counts how many lines of length 5 (any orientation) with low contrast, less than or equal to 5, go through the region.

X5 measure the contrast of horizontally adjacent pixels in the region. There are 6, the mean and standard deviation are given. This attribute is used as a vertical edge detector.

X6 X5 sd

X7 measures the contrast of vertically adjacent pixels. Used for horizontal line detection.

X8 sd X7

X9 the average over the region of (R + G + B)/3

X10 the average over the region of the R value.

X11 the average over the region of the B value.

X12 the average over the region of the G value.

X13 measure the excess red: (2R - (G + B))

X14 measure the excess blue: (2B - (G + R))

X15 measure the excess green: (2G - (R + B))

X16 3-d nonlinear transformation of RGB. (Algorithm can be found in Foley and VanDam, Fundamentals of Interactive Computer Graphics)

X17 mean of X16

X18 hue mean

Usage

```
data(image)
```

Format

A data frame contains 2310 observations and 19 variables

`leukemia` *Leukemia data set This dataset comes from a study of gene expression in two types of acute leukemias, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Gene expression levels were measured using Affymetrix high density oligonucleotide arrays containing 6817 human genes. A data set containing 72 observations from 3 leukemia types classes.*

Type *has 3 classes with 38 cases of B-cell ALL, 25 cases of AML and 9 cases of T-cell ALL.*

Gene1 to Gen 40 *gene expression levels*

Description

Leukemia data set This dataset comes from a study of gene expression in two types of acute leukemias, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Gene expression levels were measured using Affymetrix high density oligonucleotide arrays containing 6817 human genes. A data set containing 72 observations from 3 leukemia types classes.

Type has 3 classes with 38 cases of B-cell ALL, 25 cases of AML and 9 cases of T-cell ALL.

Gene1 to Gen 40 gene expression levels

Usage

```
data(leukemia)
```

Format

A data frame with 72 rows and 41 variables

Source

Dudoit, S., Fridlyand, J. and Speed, T. P. (2002). Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. Journal of the American statistical Association 97 77-87.

`lymphoma` *Lymphoma data set*

Description

Gene expression in the three most prevalent adult lymphoid malignancies: B-cell chronic lymphocytic leukemia (B-CLL), follicular lymphoma (FL), and diffuse large B-cell lym- phoma (DLBCL). Gene expression levels were measured using a specialized cDNA microarray, the Lymphochip, containing genes that are preferentially expressed in lymphoid cells or that are of known immunologic or oncologic importance. This data set contain 80 observations from 3 lymphoma types.

Type Class variable has 3 classes with 29 cases of B-cell ALL (B-CLL), 42 cases of diffuse large B-cell lymphoma (DLBCL) and 9 cases of follicular lymphoma (FL).

Gene1 to Gen 50 gene expression

Usage

```
data(lymphoma)
```

Format

A data frame with 80 rows and 51 variables

Source

Dudoit, S., Fridlyand, J. and Speed, T. P. (2002). Comparison of Discrimination Methods for the Classification of Tumors Using Gene Ex- pression Data. Journal of the American statistical Association 97 77-87.

NCI60 *NCI60 data set*

Description

cDNA microarrays were used to examine the variation in gene expression among the 60 cell lines. The cell lines are derived from tumors with different sites of origin. This data set contain 61 observations and 30 feature variables from 8 different tissue types.

Type has 8 different tissue types, 9 cases of breast, 5 cases of central nervous system (CNS), 7 cases pf colon, 8 cases of leukemia, 8 cases of melanoma, 9 cases of non-small-cell lung carcinoma (NSCLC), 6 cases of ovarian and 9 cases of renal.

Gene1 to Gen 30 gene expression information

Usage

```
data(NCI60)
```

Format

A data frame with 61 rows and 31 variables

Source

Dudoit, S., Fridlyand, J. and Speed, T. P. (2002). Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. Journal of the American statistical Association 97 77-87.

node_data *Data structure with the projected and boundary by node and class.*

Description

Data structure with the projected and boundary by node and class.

Usage

```
node_data(ppf, tr, Rule = 1)
```

Arguments

ppf	is a PPforest object
tr	numerical value to identify a tree
Rule	split rule 1:mean of two group means, 2:weighted mean, 3: mean of max(left group) and min(right group), 4: weighted mean of max(left group) and min(right group)

Value

Data frame with projected data for each class and node id and the boundaries

Examples

```
#crab data set with all the observations used as training

pprf.crab <- PPforest(data = crab, std =TRUE, class = 'Type',
 size.tr = 1, m = 200, size.p = .5, PPmethod = 'LDA')
node_data(ppf = pprf.crab, tr = 1)
```

olive *The olive data set*

Description

contains 572 observations and 10 variables

Region Three super-classes of Italy: North, South and the island of Sardinia

area Nine collection areas: three from North, four from South and 2 from Sardinia

palmitic fatty acids percent x 100

palmitoleic fatty acids percent x 100

stearic fatty acids percent x 100

oleic fatty acids percent x 100

linoleic fatty acids percent x 100

linolenic fatty acids percent x 100

arachidic fatty acids percent x 100

eicosenoic fatty acids percent x 100

Usage

```
data(olive)
```

Format

A data frame contains 573 observations and 10 variables parkinson *Parkinson data set*

Description

A data set containing 195 observations from 2 parkinson types.

Type Class variable has 2 classes, there are 48 cases of healthy people and 147 cases with Parkinson. The feature variables are biomedical voice measures.

X1 Average vocal fundamental frequency

X2 Maximum vocal fundamental frequency

X3 Minimum vocal fundamental frequency

X4 MDVP:Jitter(%) measures of variation in fundamental frequency

X5 MDVP:Jitter(Abs) measures of variation in fundamental frequency

X6 MDVP:RAP measures of variation in fundamental frequency

X7 MDVP:PPQ measures of variation in fundamental frequency

X8 Jitter:DDP measures of variation in fundamental frequency

X9 MDVP:Shimmer measures of variation in amplitude

X10 MDVP:Shimmer(dB) measures of variation in amplitude

X11 Shimmer:APQ3 measures of variation in amplitude

X12 Shimmer:APQ5 measures of variation in amplitude

X13 MDVP:APQ measures of variation in amplitude

X14 Shimmer:DDA measures of variation in amplitude

X15 NHR measures of ratio of noise to tonal components in the voice

X16 HNR measures of ratio of noise to tonal components in the voice

X17 RPDE nonlinear dynamical complexity measures

X18 D2 nonlinear dynamical complexity measures

X19 DFA - Signal fractal scaling exponent

X20 spread1 Nonlinear measures of fundamental frequency variation

X21 spread2 Nonlinear measures of fundamental frequency variation

X22 PPE Nonlinear measures of fundamental frequency variation

Usage

```
data(parkinson)
```

Format

A data frame with 195 rows and 23 variables

Source

[https://archive.ics.uci.edu/ml/datasets/Parkinsons](Parkinson)

permute_importance *Obtain the permuted importance variable measure*

Description

Obtain the permuted importance variable measure

Usage

```
permute_importance(ppf)
```

Arguments

ppf	is a PPforest object

Value

A data frame with permuted importance measures, imp is the permuted importance measure defined in Brieman paper, imp2 is the permuted importance measure defined in randomForest package, the standard deviation (sd.im and sd.imp2) for each measure is computed and the also the standardized mesure.

Examples

```
pprf.crab <- PPforest(data = crab, class = 'Type',
std = TRUE, size.tr = 1, m = 100, size.p = .4, PPmethod = 'LDA', parallel = TRUE, core = 2)
permute_importance(ppf = pprf.crab)
```

PPclassify2 *Predict class for the test set and calculate prediction error after finding the PPtree structure, .*

Description

Predict class for the test set and calculate prediction error after finding the PPtree structure, .

Usage

```
PPclassify2( Tree.result, test.data = NULL, Rule = 1, true.class = NULL)
```

Arguments

`Tree.result` the result of PP.Tree

`test.data` the test dataset

`Rule` split rule 1:mean of two group means, 2:weighted mean, 3: mean of max(left group) and min(right group), 4: weighted mean of max(left group) and min(right group)

`true.class` true class of test dataset if available

Value

predict.class predicted class

predict.error prediction error

References

Lee, YD, Cook, D., Park JW, and Lee, EK(2013) PPtree: Projection pursuit classification tree, Electronic Journal of Statistics, 7:1369-1386.

Examples

```
#crab data set

Tree.crab <- PPtree_split('Type~.', data = crab, PPmethod = 'LDA', size.p = 0.5)
Tree.crab


PPclassify2(Tree.crab)
```

PPforest *Projection Pursuit Random Forest*

Description

`PPforest` implements a random forest using projection pursuit trees algorithm (based on PPtreeViz package).

Usage

```
PPforest(data, class, std = TRUE, size.tr, m, PPmethod, size.p,
 lambda = .1, parallel = FALSE, cores = 2, rule = 1)
```

Arguments

`data` Data frame with the complete data set.

`class` A character with the name of the class variable.

`std` if TRUE standardize the data set, needed to compute global importance measure.

`size.tr` is the size proportion of the training if we want to split the data in training and test.

`m` is the number of bootstrap replicates, this corresponds with the number of trees to grow. To ensure that each observation is predicted a few times we have to select this number no too small. `m = 500` is by default.

`PPmethod` is the projection pursuit index to optimize in each classification tree. The options are `LDA` and `PDA`, linear discriminant and penalized linear discriminant. By default it is `LDA`.

`size.p` proportion of variables randomly sampled in each split.

`lambda` penalty parameter in PDA index and is between 0 to 1 . If `lambda = 0`, no penalty parameter is added and the PDA index is the same as LDA index. If `lambda = 1` all variables are treated as uncorrelated. The default value is `lambda = 0.1`.

`parallel` logical condition, if it is TRUE then parallelize the function

`cores` number of cores used in the parallelization

`rule` split rule 1: mean of two group means 2: weighted mean of two group means - weight with group size 3: weighted mean of two group means - weight with group sd 4: weighted mean of two group means - weight with group se 5: mean of two group medians 6: weighted mean of two group medians - weight with group size 7: weighted mean of two group median - weight with group IQR 8: weighted mean of two group median - weight with group IQR and size

Value

An object of class `PPforest` with components.

`prediction.training` predicted values for training data set.

`training.error` error of the training data set.

`prediction.test` predicted values for the test data set if `testap = TRUE`(default).

`error.test` error of the test data set if `testap = TRUE`(default).

`oob.error.forest` out of bag error in the forest.

`oob.error.tree` out of bag error for each tree in the forest.

`boot.samp` information of bootrap samples.

`output.trees` output from a `trees_pp` for each bootrap sample.

`proximity` Proximity matrix, if two cases are classified in the same terminal node then the proximity matrix is increased by one in `PPforest` there are one terminal node per class.

`votes` a matrix with one row for each input data point and one column for each class, giving the fraction of (OOB) votes from the `PPforest`.

`n.tree` number of trees grown in `PPforest`.

`n.var` number of predictor variables selected to use for spliting at each node.

`type` classification.

`confusion` confusion matrix of the prediction (based on OOB data).

`call` the original call to `PPforest`.

`train` is the training data based on `size.tr` sample proportion

`test` is the test data based on `1-size.tr` sample proportion

References

Natalia da Silva, Dianne Cook & Eun-Kyung Lee (2021) A Projection Pursuit Forest Algorithm for Supervised Classification, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2020.1870480

Examples

```
#crab example with all the observations used as training

pprf.crab <- PPforest(data = crab, class = 'Type',
 std = FALSE, size.tr = 1, m = 200, size.p = .5,
 PPmethod = 'LDA' , parallel = TRUE, cores = 2, rule=1)
pprf.crab
```

`ppf_avg_imp` *Global importance measure for a PPforest object as the average IMP PPtree measure over all the trees in the forest*

Description

Global importance measure for a PPforest object as the average IMP PPtree measure over all the trees in the forest

Usage

```
ppf_avg_imp(ppf, class)
```

Arguments

`ppf` is a PPforest object

`class` A character with the name of the class variable.

Value

Data frame with the global importance measure

Examples

```
#crab data set with all the observations used as training

pprf.crab <- PPforest(data = crab, std =TRUE, class = 'Type',
 size.tr = 1, m = 100, size.p = .5, PPmethod = 'LDA')
 ppf_avg_imp(pprf.crab, 'Type')
```

---

ppf_global_imp *Global importance measure for a PPforest object*

Description

Global importance measure for a PPforest object

Usage

```
ppf_global_imp(data, class, ppf)
```

Arguments

data	Data frame with the complete data set.
class	A character with the name of the class variable.
ppf	is a PPforest object

Value

Data frame with the global importance measure

Examples

```
#crab data set with all the observations used as training

pprf.crab <- PPforest(data = crab, std = TRUE, class = 'Type',
 size.tr = 1, m = 200, size.p = .5, PPmethod = 'LDA', parallel = TRUE, cores = 2)

ppf_global_imp(data = crab, class = 'Type', pprf.crab)
```

PPtree_split *Projection pursuit classification tree with random variable selection in each split*

Description

Find tree structure using various projection pursuit indices of classification in each split.

Usage

```
PPtree_split(form, data, PPmethod='LDA',
size.p=1,  lambda = 0.1,...)
```

Arguments

form	A character with the name of the class variable.
data	Data frame with the complete data set.
PPmethod	index to use for projection pursuit: 'LDA', 'PDA'
size.p	proportion of variables randomly sampled in each split, default is 1, returns a PPtree.
lambda	penalty parameter in PDA index and is between 0 to 1 . If lambda = 0, no penalty parameter is added and the PDA index is the same as LDA index. If lambda = 1 all variables are treated as uncorrelated. The default value is lambda = 0.1.
...	arguments to be passed to methods

Value

An object of class PPtreeclass with components

Tree.Struct	Tree structure of projection pursuit classification tree
projbest.node	1-dim optimal projections of each split node
splitCutoff.node	cutoff values of each split node
origclass	original class
origdata	original data

References

Lee, YD, Cook, D., Park JW, and Lee, EK (2013) PPtree: Projection pursuit classification tree, Electronic Journal of Statistics, 7:1369-1386.

Examples

```
#crab data set

Tree.crab <- PPtree_split('Type~.', data = crab, PPmethod = 'LDA', size.p = 0.5)
Tree.crab
```

---

print.PPforest *Print PPforest object*

Description

Print PPforest object

Usage

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

Arguments

x	is a PPforest class object
...	additional parameter

Value

printed results for PPforest object

---

ternary_str *Data structure with the projected and boundary by node and class.*

Description

Data structure with the projected and boundary by node and class.

Usage

```
ternary_str(ppf, id, sp, dx, dy)
```

Arguments

ppf	is a PPforest object
id	is a vector with the selected projection directions
sp	is the simplex dimensions, if k is the number of classes sp = k - 1
dx	first direction included in id
dy	second direction included in id

Value

Data frame needed to visualize a ternary plot

Examples

```
#crab data set with all the observations used as training
pprf.crab <- PPforest(data = crab, std =TRUE, class = "Type",
 size.tr = 1, m = 100, size.p = .5, PPmethod = 'LDA')
 require(dplyr)
pl_ter <- function(dat, dx, dy ){
  p1  <- dat[[1]] %>% dplyr::filter(pair %in% paste(dx, dy, sep = "-") ) %>%
    dplyr::select(Class, x, y) %>%
    ggplot2::ggplot(ggplot2::aes(x, y, color = Class)) +
    ggplot2::geom_segment(data = dat[[2]], ggplot2::aes(x = x1, xend = x2,
                                               y = y1, yend = y2), color = "black" ) +
    ggplot2::geom_point(size = I(3), alpha = .5) +
    ggplot2::labs(y = " ",  x = " ") +
    ggplot2::theme(legend.position = "none", aspect.ratio = 1) +
    ggplot2::scale_colour_brewer(type = "qual", palette = "Dark2") +
    ggplot2::labs(x = paste0("T", dx, " "), y = paste0("T", dy, " ")) +
    ggplot2::theme(aspect.ratio = 1)
  p1
}
#ternary plot in tree different selected dierections
 pl_ter(ternary_str(pprf.crab, id = c(1, 2, 3), sp = 3, dx = 1, dy = 2), 1, 2 )
```

trees_pred *Obtain predicted class for new data from baggtree function or PPforest*

Description

Obtain predicted class for new data from baggtree function or PPforest

Usage

```
trees_pred(object, xnew, parallel = FALSE, cores = 2, rule = 1)
```

Arguments

`object` Projection pursuit classification forest structure from PPforest or baggtree

`xnew` data frame with explicative variables used to get new predicted values.

`parallel` logical condition, if it is TRUE then parallelize the function

`cores` number of cores used in the parallelization `rule` split rule 1: mean of two group means 2: weighted mean of two group means - weight with group size 3: weighted mean of two group means - weight with group sd 4: weighted mean of two group means - weight with group se 5: mean of two group medians 6: weighted mean of two group medians - weight with group size 7: weighted mean of two group median - weight with group IQR 8: weighted mean of two group median - weight with group IQR and size

Value

predicted values from PPforest or baggtree

Examples

```
## Not run:
crab.trees <- baggtree(data = crab, class = 'Type',
m =  200, PPmethod = 'LDA', lambda = .1, size.p = 0.4 )

pr <- trees_pred(  crab.trees,xnew = crab[, -1], parallel= FALSE, cores = 2)

pprf.crab <- PPforest(data = crab, class = 'Type',
 std = FALSE, size.tr = 2/3, m = 100, size.p = .4, PPmethod = 'LDA', parallel = TRUE )

trees_pred(pprf.crab, xnew = pprf.crab$test ,parallel = TRUE)

## End(Not run)
```

`wine` *Wine data set*

Description

A data set containing 178 observations from 3 wine grown cultivares in Italy.

Type Class variable has 3 classes that are 3 different wine grown cultivares in Italy.

X1 to X13 Check vbles

Usage

```
data(wine)
```

Format

A data frame with 178 rows and 14 variables

Index

∗ **datasets**
- crab, 3
- fishcatch, 4
- glass, 5
- image, 5
- leukemia, 6
- lymphoma, 7
- NCI60, 8
- olive, 9
- parkinson, 10
- wine, 19

∗ **tree**
- PPclassify2, 11
- PPtree_split, 16

baggtree, 2

crab, 3

fishcatch, 4

glass, 5

image, 5

leukemia, 6
lymphoma, 7

NCI60, 8
node_data, 8

olive, 9

parkinson, 10
permute_importance, 11
PPclassify2, 11
ppf_avg_imp, 14
ppf_global_imp, 15
PPforest, 12
PPtree_split, 16
print.PPforest, 17

ternary_str, 17
trees_pred, 18

wine, 19