Package ‘blkbox’

August 29, 2016

Type Package

Title Data Exploration with Multiple Machine Learning Algorithms

Version 1.0

Date 2016-08-05

Author Zachary Davies, Boris Guennewig

Maintainer Boris Guennewig <b.guennewig@garvan.org.au>

Description Allows data to be processed by multiple machine learning algorithms at the same time, enables feature selection of data by single a algorithm or combinations of multiple. Easy to use tool for k-fold cross validation and nested cross validation.

License GPL (>= 2)

LazyData TRUE

Depends R (>= 3.0.0), methods

Imports dplyr, plyr, tidyr, magrittr, caret, ggplot2, glmnet, bartMachine, reshape2, randomForest, kknn, pamr, nnet, party, rJava, e1071, pROC, stringr, xgboost, parallel, knitr, rmarkdown, shiny, shinyjs, reshape, gtools, tibble

Suggests bigrf

RoxygenNote 5.0.1

NeedsCompilation no

VignetteBuilder knitr

Additional_repositories http://zacdav.github.io/drat/

Repository CRAN

Date/Publication 2016-08-08 02:37:07

R topics documented:

`blkbox` *Train and Test datasets.*

Description

This standard function will allow multiple machine learning algorithms to be utilized on the same data to determine, which algorithm may be the most appropriate.

Usage

```
blkbox(data, labels, holdout, holdout.labels, ntrees, mTry, Kernel, Gamma,
  exclude, max.depth, xgtype = "binary:logistic", seed)
```

Arguments

`data` Data partitioned by into a list or a data frame of training data where the features correspond to columns and the samples are rows. As data size increases the memory required and run time of some algorithms may compound exponentially.

`labels` a character or numeric vector that contains the training class identifiers for the samples in the data frame. Must appear in the same order. Does not need to be specified if using a partitoned data list.

`holdout` a data frame of holdout of testing data where the features correspond to columns and the samples are the rows. Does not need to be specified if using a partitoned data list.

`holdout.labels` a character or numeric vector that contains the holdout or testing class identifiers for the samples in the holdout data frame. Does not need to be specified if using a partitoned data list.

`ntrees` The number of trees used in the ensemble based learners (randomforest, bigrf, party, bartmachine). default = 500.

`mTry` The number of features sampled at each node in the trees of ensemble based learners (randomforest, bigrf, party, bartmachine). default = sqrt(number of features).

`Kernel` The type of kernel used in the support vector machine algorithm (linear, radial, sigmoid, polynomial). default = "linear".

`Gamma` dvanced parameter, defines the distance of which a single training example reaches. Low gamma will produce a SVM with softer boundaries, as Gamma increases the boundaries will eventually become restricted to their singular support vector. default is 1/(ncol - 1).

`exclude` removes certain algorithms from analysis - to exclude random forest which you would set exclude = "randomforest". The algorithms each have their own numeric identifier. randomforest = "randomforest", knn = "kknn", bartmachine = "bartmachine", party = "party", glmnet = "GLM", pam = "PamR, nnet = "nnet", svm = "SVM", xgboost = "xgboost".

`max.depth` the maximum depth of the tree in xgboost model, default is sqrt(ncol(data)).

`xgtype` either "binary:logistic" or "reg:linear" for logistic regression or linear regression respectively.

`seed` Sets the seed for the bartMachine model.

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
my_data <- iris[1:100, 1:4]
my_labels <- as.character(iris[1:100, 5])
my_partition = Partition(data = my_data, labels = my_labels)
model_1 <- blkbox(data = my_partition)
```

`blkboxCV` *k-fold cross validation with blkbox.*

Description

A function that builds upon the blkbox function and performs k-fold cross validation and then provides votes for each fold as well as the importance of each feature in the models.

Usage

```
blkboxCV(data, labels, folds = 10, seed, ntrees, mTry, repeats = 1, Kernel,
  Gamma, max.depth, xgtype = "binary:logistic", exclude = c(0),
  Method = "GLM", AUC = "NA")
```

Arguments

`data` A data.frame where the columns correspond to features and the rows are samples. The dataframe will be shuffled and split into k folds for downstream analysis.

`labels` A character or numeric vector of the class identifiers that each sample belongs.

`folds` The number of times the data set will be subsectioned (number of samples / k, if modulo exists the groups will be as close to the same size as possible). Each data subsection will be used as a holdout portion. default = 10.

`seed` A numeric value. defaults to a randomly generated set of seeds that are output when run starts.

`ntrees` The number of trees used in the ensemble based learners (randomforest, bigrf, party, bartmachine). default = 500.

`mTry` The number of features sampled at each node in the trees of ensemble based learners (randomforest, bigrf, party, bartmachine). default = sqrt(number of features).

`repeats` repeat the cross validation process. default = 1.

`Kernel` The type of kernel used in the support vector machine algorithm (linear, radial, sigmoid, polynomial). default = "linear".

`Gamma` Advanced parameter, defines the distance of which a single training example reaches. Low gamma will produce a SVM with softer boundaries, as Gamma increases the boundaries will eventually become restricted to their singular support vector. default is 1/(ncol - 1).

`max.depth` the maximum depth of the tree in xgboost model, default is sqrt(ncol(data)).

`xgtype` either "binary:logistic" or "reg:linear" for logistic regression or linear regression respectively.

`exclude` removes certain algorithms from analysis - to exclude random forest which you would set exclude = "randomforest". The algorithms each have their own numeric identifier. randomforest = "randomforest", knn = "kknn", bartmachine = "bartmachine", party = "party", glmnet = "GLM", pam = "PamR, nnet = "nnet", svm = "SVM", xgboost = "xgboost".

`Method` The algorithm used to feature select the data. Uses the feature importance from the algorithms to rank and remove anything below the AUC threshold. Default is "GLM".

`AUC` Area under the curve selection measure. The relative importance of features is calculated and then ranked. The features responsible for the most importance are therefore desired, the AUC value is the percentile in which to keep features above. 0.5 keeps the highest ranked features responsible for 50 percent of the cumulative importance. Default is NA which means feature are not selected at after CV. Will default to 1.0 if Method is "xgboost".

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
model_2 <- blkboxCV(data = my_data, labels = my_labels)
```

blkboxNCV *Nested cross fold validation with blkbox.*

Description

A function that builds upon the blkbox and blkboxNCV function and performs nested k-fold cross validation and then provides votes for each fold as well as the importance of each feature in the models. Provides feature importance tables and details for each inner and outerfold run.

Usage

```
blkboxNCV(data, labels, outerfolds = 5, innerfolds = 5, ntrees, mTry,
  Kernel, Gamma, max.depth, xgtype = "binary:logistic", exclude = c(0),
  inn.exclude, Method = "GLM", AUC = 0.5, metric = c("ERR", "AUROC",
  "ACC", "MCC", "F-1"), seed)
```

Arguments

`data` A data.frame where the columns correspond to features and the rows are samples. The dataframe will be shuffled and split into k folds for downstream analysis.

`labels` A character or numeric vector of the class identifiers that each sample belongs.

`outerfolds` The number of folds that will be in the first k-fold loop, this determines the number of holdouts. Default is 5.

`innerfolds` The number of folds that occur in the internal feature selection cross fold validation before testing on the corresponding holdout. Default is 5.

`ntrees` The number of trees used in the ensemble based learners (randomforest, bigrf, party, bartmachine). default = 500.

`mTry` The number of features sampled at each node in the trees of ensemble based learners (randomforest, bigrf, party, bartmachine). default = sqrt(number of features).

`Kernel` The type of kernel used in the support vector machine algorithm (linear, radial, sigmoid, polynomial). default = "linear".

`Gamma` Advanced parameter, defines the distance of which a single training example reaches. Low gamma will produce a SVM with softer boundaries, as Gamma increases the boundaries will eventually become restricted to their singular support vector. default is 1/(ncol - 1).

`max.depth` the maximum depth of the tree in xgboost model, default is sqrt(ncol(data)).

`xgtype` either "binary:logistic" or "reg:linear" for logistic regression or linear regression respectively.

`exclude` removes certain algorithms from analysis - to exclude random forest which you would set exclude = "randomforest". The algorithms each have their own numeric identifier. randomforest = "randomforest", knn = "kknn", bartmachine = "bartmachine", party = "party", glmnet = "GLM", pam = "PamR, nnet = "nnet", svm = "SVM", xgboost = "xgboost".

`inn.exclude` removes certain algorithms from after feature selection analysis. similar to 'exclude'. Defaults to exclude all but Method.

`Method` The algorithm used to feature select the data. Uses the feature importance from the algorithms to rank and remove anything below the AUC threshold. Defaults to "GLM", therefore the inner folds will use "GLM" only unless specified otherwise.

`AUC` Area under the curve selection measure. The relative importance of features is calculated and then ranked. The features responsible for the most importance are therefore desired, the AUC value is the percentile in which to keep features above. 0.5 keeps the highest ranked features responsible for 50 percent of the cumulative importance. default = 0.5. Will Change to 1.0 default when Method = "xgboost".

`metric` A character string to determine which performance metric will be passed on to the Performance() function. Refer to Performance() documentation. default = c("ERR", "AUROC", "ACC", "MCC", "F-1")

`seed` A single numeric value that will determine all subsequent seeds set in NCV.

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
blkboxNCV(data = my_data,
        labels = my_labels,
        Method = "randomforest",
        AUC = 0.9)
```

---

blkboxROC *ROC plots for blkbox*

Description

will plot ROC curves for output from Performance function if "AUROC" was specified.

Usage

```
blkboxROC(results, title = "ROC")
```

Arguments

`results` The output of blkbox Performance that had "AUROC" as one of the specified metrics.

`title` The title of the plot. Default is "ROC".

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
# model_1 can be any blkbox or blkboxCV model
perf = Performance(model_1)
# Standard ROC curve
blkboxROC(perf)
```

`blkboxUI` *blkbox User Interface*

Description

Invokes the shiny interface for blkbox.

Usage

```
blkboxUI()
```

Author(s)

Zachary Davies

Examples

```
blkboxUI()
```

`cv.plot` *Crossfold Validation Performance Plot.*

Description

Compares the performance of each algorithm in a boxplot OR barplot. Each holdout will contribute at least one data point to each algorithm.

Usage

```
cv.plot(obj, metric = "AUROC", y_ranges = c(0, 1), title = "",
  type = "boxplot")
```

Arguments

`obj` An object produced by the blkboxCV function.

`metric` Which metric you wish to plot. Area under the Receiver operating curve = "AUROC", Accuracy = "ACC", Error rate = "ERR", Matthews correlation coefficient = "MCC", F-1 score = "F-1". default = c("AUROC")

`y_ranges` is the y axis limits for the plot, defaults to c(0,1). Must be a numeric vector with two entries. Invalid for barplots.

`title` the title to be adhered to the plot. Default is no title.

`type` The plot can be either a barplot or boxplot. For the barplot the consensus performance is used, for a boxplot consensus is false. If only one performance measure is found for each algorithm then it will be forced to a barplot. default = "boxplot", unless data is unsupported.

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
cv.plot(blkboxCV(...), metric = "AUROC", title = "Example Performance Barplot")
```

`ncv.plot` *Nested Crossfold Validation Performance Plot.*

Description

Compares the performance of each algorithm in a boxplot. Each holdout will contribute at least one data point to each algorithms boxplot.

Usage

```
ncv.plot(obj, metric, y_ranges, title)
```

Arguments

`obj` An object produced by the blkboxCV function.

`metric` Which metric you wish to plot, can only plot those specified to the blkboxNCV function at time of running. Area under the Receiver operating curve = "AUROC", Accuracy = "ACC", Error rate = "ERR", Matthews correlation coefficient = "MCC", F-1 score = "F-1". Default is the first metric specified to your NCV arguments vector.

`y_ranges` is the y axis limits for the plot, defaults to c(0,1). Must be a numeric vector with two entries.

`title` the title to be adhered to the plot. Default is no title.

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
ncv.plot(blkboxNCV(...), metric = "AUROC", title = "NCV Performance Boxplot")
```

Partition *blkbox paritioning*

Description

Prepares data for standard training and testing, data will be split into training and holdout set and output in a list which can be directly supplied to blkbox.

Usage

```
Partition(data, labels, size, seed)
```

Arguments

`data`	A data.frame of the data. Rows represent samples and columns features.
`labels`	The labels corresponding to the data, order must match with order of rows in data.
`size`	determines the size of the holdout data, must be a numeric value between 0 and 1 that. Default is 0.8.
`seed`	Determines the seed used to randomly sample the data by row.

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
# Partitioning Data
my_partition = Partition(data = my_data,
                         labels = my_labels)
# Creating a Training & Testing Model
model_1 <- blkbox(data = my_partition)
```

Performance *blkbox Performance.*

Description

Determines the performance of each model within the blkbox or blkboxCV output. Can choose from a range of performance metrics.

Usage

```
Performance(object, metric = "AUROC", consensus = FALSE)
```

Arguments

`object`	the blkboxCV or blkbox output
`metric`	Which metric will be used for performance. Area under the Receiver operating curve = "AUROC", Accuracy = "ACC", Error rate = "ERR", Matthews correlation coefficient = "MCC", F-1 score = "F-1". default = "AUROC".
`consensus`	if the process was repeated it will calculate the consensus vote for each sample across the repititons before then calculating the performance across all samples. Default is False.

Author(s)

Zachary Davies, Boris Guennewig

Examples

```
Performance(blkbox(...), metric = "AUROC")
Performance(blkboxCV(...), metric = "ERR")
```

Index