Package 'sboost'

July 23, 2025

Type Package

Title Machine Learning with AdaBoost on Decision Stumps

Version 0.1.2

Description Creates classifier for binary outcomes using Adaptive Boosting (AdaBoost) algorithm on decision stumps with a fast C++ implementation. For a description of AdaBoost, see Freund and Schapire (1997) <doi:10.1006/jcss.1997.1504>. This type of classifier is nonlinear, but easy to interpret and visualize. Feature vectors may be a combination of continuous (numeric) and categorical (string, factor) elements. Methods for classifier assessment, predictions, and cross-validation also included.

License MIT + file LICENSE

URL https://github.com/jadonwagstaff/sboost

BugReports https://github.com/jadonwagstaff/sboost/issues

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0)

LinkingTo Rcpp (>= 0.12.17)

Imports dplyr (>= 0.7.6), rlang (>= 0.2.1), Rcpp (>= 0.12.17), stats (>= 3.4)

RoxygenNote 7.1.2

Suggests testthat

NeedsCompilation yes

Author Jadon Wagstaff [aut, cre]

Maintainer Jadon Wagstaff <jadonw@gmail.com>

Repository CRAN

Date/Publication 2022-05-26 13:10:02 UTC

Contents

assess 2
malware 3
mushrooms 4
predict.sboost_classifier 5
sboost 6
validate 7

Index **10**

`assess` *sboost Assessment Function*

Description

Assesses how well an sboost classifier classifies the data.

Usage

```
assess(object, features, outcomes, include_scores = FALSE)
```

Arguments

`object`	*sboost_classifier* S3 object output from sboost.
`features`	feature set data.frame.
`outcomes`	outcomes corresponding to the features.
`include_scores`	if true feature_scores are included in output.

Value

An *sboost_assessment* S3 object containing:

performance Last row of cumulative statistics (i.e. when all stumps are included in assessment).

cumulative_statistics *stump* - the index of the last decision stump added to the assessment.
true_positive - number of true positive predictions.
false_negative - number of false negative predictions.
true_negative - number of true negative predictions.
false_positive - number of false positive predictions.
prevalence - true positive / total.
accuracy - correct predictions / total.
sensitivity - correct predicted positive / true positive.
specificity - correct predicted negative / true negative.
ppv - correct predicted positive / predicted positive.
npv - correct predicted negative / predicted negative.
f1 - harmonic mean of sensitivity and ppv.

feature_scores If include_scores is TRUE, for each feature in the classifier lists scores for each row in the feature set.

classifier sboost *sboost_classifier* object used for assessment.

outcomes Shows which outcome was considered as positive and which negative.

call Shows the parameters that were used for assessment.

See Also

sboost documentation.

Examples

```
# malware
malware_classifier <- sboost(malware[-1], malware[1], iterations = 5, positive = 1)
assess(malware_classifier, malware[-1], malware[1])

# mushrooms
mushroom_classifier <- sboost(mushrooms[-1], mushrooms[1], iterations = 5, positive = "p")
assess(mushroom_classifier, mushrooms[-1], mushrooms[1])
```

---

malware *Malware System Calls*

Description

System call data for apps identified as malware and not malware.

Usage

```
malware
```

Format

A data frame with 7597 rows and 361 variables: *outcomes* 1 if malware, 0 if not. *X1... X360* system calls.

Details

Experimental data generated in this research paper:

M. Dimjašević, S. Atzeni, I. Ugrina, and Z. Rakamarić, "Evaluation of Android Malware Detection Based on System Calls," in Proceedings of the International Workshop on Security and Privacy Analytics (IWSPA), 2016.

Data used for kaggle competition: https://www.kaggle.com/c/ml-fall2016-android-malware

Source

https://zenodo.org/record/154737#.WtoA1IjwaUl

`mushrooms` *Mushroom Classification*

Description

A classic machine learning data set describing hypothetical samples from the Agaricus and Lepiota family.

Usage

```
mushrooms
```

Format

A data frame with 7597 rows and 361 variables:

outcomes p=poisonous, e=edible

cap_shape bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s

cap_surface fibrous=f, grooves=g, scaly=y, smooth=s

cap_color brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y

bruises bruises=t, no=f

odor almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s

gill_attachment attached=a, descending=d, free=f, notched=n

gill_spacing close=c, crowded=w, distant=d

gill_size broad=b, narrow=n

gill_color black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y

stalk_shape enlarging=e, tapering=t

stalk_root bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?

stalk_surface_above_ring fibrous=f, scaly=y, silky=k, smooth=s

stalk_surface_below_ring fibrous=f, scaly=y, silky=k, smooth=s

stalk_color_above_ring brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y

stalk_color_below_ring brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y

veil_type partial=p, universal=u

veil_color brown=n, orange=o, white=w, yellow=y

ring_number none=n, one=o, two=t

ring_type cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z

spore_print_color black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y

population abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y

habitat grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

Details

Data gathered from:

Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf

Source

https://archive.ics.uci.edu/ml/datasets/mushroom

---

predict.sboost_classifier

Make predictions for a feature set based on an sboost classifier.

Description

Make predictions for a feature set based on an sboost classifier.

Usage

```
## S3 method for class 'sboost_classifier'
predict(object, features, scores = FALSE, ...)
```

Arguments

object	*sboost_classifier* S3 object output from sboost.
features	feature set data.frame.
scores	if true, raw scores generated; if false, predictions are generated.
...	further arguments passed to or from other methods.

Value

Predictions in the form of a vector, or scores in the form of a vector. The index of the vector aligns the predictions or scores with the rows of the features. Scores represent the sum of all votes for the positive outcome minus the sum of all votes for the negative outcome.

See Also

sboost documentation.

Examples

```
# malware
malware_classifier <- sboost(malware[-1], malware[1], iterations = 5, positive = 1)
predict(malware_classifier, malware[-1], scores = TRUE)
predict(malware_classifier, malware[-1])

# mushrooms
mushroom_classifier <- sboost(mushrooms[-1], mushrooms[1], iterations = 5, positive = "p")
predict(mushroom_classifier, mushrooms[-1], scores = TRUE)
predict(mushroom_classifier, mushrooms[-1])
```

`sboost` *sboost Learning Algorithm*

Description

A machine learning algorithm using AdaBoost on decision stumps.

Usage

```
sboost(features, outcomes, iterations = 1, positive = NULL, verbose = FALSE)
```

Arguments

`features`	feature set data.frame.
`outcomes`	outcomes corresponding to the features.
`iterations`	number of boosts.
`positive`	the positive outcome to test for; if NULL, the first outcome in alphabetical (or numerical) order will be chosen.
`verbose`	If true, progress bar will be displayed in console.

Details

Factors and characters are treated as categorical features. Missing values are supported.

See https://jadonwagstaff.github.io/projects/sboost.html for a description of the algorithm.

For original paper describing AdaBoost see:

Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119-139 (1997)

Value

An *sboost_classifier* S3 object containing:

classifier *stump* - the index of the decision stump
feature - name of the column that this stump splits on.
vote - the weight that this stump has on the final classifier.
orientation - shows how outcomes are split. If *feature* is numeric shows split orientation, if *feature* value is less than *split* then vote is cast in favor of left side outcome, otherwise the vote is cast for the right side outcome. If *feature* is categorical, vote is cast for the left side outcome if *feature* value is found in *left_categories*, otherwise vote is cast for right side outcome.
split - if *feature* is numeric, the value where the decision stump splits the outcomes; otherwise, NA.
left_categories - if *feature* is categorical, shows the *feature* values that sway the vote to the left side outcome on the *orientation* split; otherwise, NA.

outcomes Shows which outcome was considered as positive and which negative.

training *stumps* - how many decision stumps were trained.
features - how many features the training set contained.
instances - how many instances or rows the training set contained.
positive_prevalence - what fraction of the training instances were positive.

call Shows the parameters that were used to build the classifier.

See Also

`predict.sboost_classifier` - to get predictions from the classifier.

`assess` - to evaluate the performance of the classifier.

`validate` - to perform cross validation for the classifier training.

Examples

```
# malware
malware_classifier <- sboost(malware[-1], malware[1], iterations = 5, positive = 1)
malware_classifier
malware_classifier$classifier

# mushrooms
mushroom_classifier <- sboost(mushrooms[-1], mushrooms[1], iterations = 5, positive = "p")
mushroom_classifier
mushroom_classifier$classifier
```

`validate` *sboost Validation Function*

Description

A k-fold cross validation algorithm for sboost.

Usage

```
validate(
  features,
  outcomes,
  iterations = 1,
  k_fold = 6,
  positive = NULL,
  verbose = FALSE
)
```

Arguments

features	feature set data.frame.
outcomes	outcomes corresponding to the features.
iterations	number of boosts.
k_fold	number of cross-validation subsets.
positive	is the positive outcome to test for; if NULL, the first in alphabetical order will be chosen
verbose	If true, progress bars will be displayed in console.

Value

An *sboost_validation* S3 object containing:

performance Final performance statistics for all stumps.

training_summary_statistics Mean and standard deviations for test statistics generated by assess cumulative statistics for each of the training sets.

testing_summary_statistics Mean and standard deviations for test statistics generated by assess cumulative statistics for each of the testing sets.

training_statistics sboost *sboost_assessment* cumulative statistics objects used to generate training_statistics.

testing_statistics sboost *sboost_assessment* cumulative statistics objects used to generate testing_statistics.

classifier_list sboost *sboost_classifier* objects created from training sets.

outcomes Shows which outcome was considered as positive and which negative.

k_fold number of testing and training sets used in the validation.

call Shows the parameters that were used for validation.

See Also

sboost documentation.

Examples

```
# malware
validate(malware[-1], malware[1], iterations = 5, k_fold = 3, positive = 1)

# mushrooms
validate(mushrooms[-1], mushrooms[1], iterations = 5, k_fold = 3, positive = "p")
```

Index

∗ **datasets**
malware, 3
mushrooms, 4

assess, 2, *7*, *8*

malware, 3
mushrooms, 4

predict.sboost_classifier, 5, *7*

sboost, *3*, *5*, 6, *8*

validate, *7*, 7