IV SEMINÁRIO INTERNACIONAL DE ESTATÍSTICA COM R
R & PYTHON E AS TENDÊNCIAS DE COLABORAÇÃO
NITERÓI, 21 A 23 DE MAIO DE 2019

SHINY ADABOOSTING: AN INTERACTIVE DASHBOARD TO ADAPTIVE BOOSTING ALGORITHM

Mateus Maia Marques[1]

Anderson Ara[2]

Abstract

Boosting methods are becoming more and more popular due their outstanding performance when compared with some others statistical learning techniques. The Adaptive Boosting, or simply AdaBoost, was one of the first boosting techinques developed, and consists, generally, in a linear combination of weak models (models that peform slightly better than a random guess) to built a strong classifier. The main purpose of this article was to built an interactive application, using the Shiny R package, that possibilites the user to apply the AdaBoost model to some datasets and observe the behaviour of the model performance concerning parameter's varation, base learners, presence of noise and others aspects as evaluating aspects as overfitting, accuracy and computational time.

Palavras-chave: boosting, shiny, machine learning, data modeling, data visualization.

Introduction

The Adaptative Boosting algorithm (AdaBoost), developed by Freund and Schapire (1995), has been showed to be a great statistical model that can be outperform a lot of others statistical learning algorithms. Like all others ensemble methods, the AdaBoosting is built by the combination of several models that vote to classify and predict an observation. In AdaBoosting, these classifiers are modeled sequentially and each new model it's weighted considering the capacity to predict correctly the previous missclassified observations. Generally, the base models are the decision-tree algorithm (C4.5) (Quilan,1993), however any other weak leaner can be used in AdaBoost. AdaBoost has some interesting characteristcs that helped this technique to consolidate a strong statistical learning algorithm as his resistance to overfitting (Schapire, 2013), and the flexiblity to diferente types of data.

Kearns and Valiant (1988) were the one who start to awnser the question of whether a weak leaner model that perform slightly better than random guessing can be boosted into a accurate and better learning algorithm. Schapire came up with the first provable boosting algorithm in 1989. A years latters Freund (1995) developed a much more efficient boosting

[1] Universidade Federal da Bahia (UFBA), mateusmaia11@gmail.com

[2] Universidade Federal da Bahia (UFBA), anderson.ara@ufba.br

algorithm. The firsts experiments with these early algorithms of boosting were accomplished by Drucker, Schapire and Simard (1993) on a Optical Character Recognition Task. Viola and Jones (2001), proposed an AdaBoost based faced detection framework that is capable of processing images exceptionally rapidly and achiving high detection rates. One of the most frequent use of AdaBoosting and Boosting techniques are also in the areas of text filtering and classification (Lee, et.al, 2011). Besides the above applications, AdaBoost and its variations algorithms are also widely used in speech recognition (Saon, 2012), object detection (Chen, 2011), vehicle detection (Rios-Cabrera, 2011), and so on.

Given the importance of this ensemble method, this article purpose a interactive dashboard, fully-built in R language, that can allow the user to apply the AdaBoost to differents techniques, changing the each parameter from the model, and also viewing all

steps that the algorithm can perform in order to give a clear explanation about the behavior of this type of model.

Objective

This work aims the construction of a interactive plattaform that permits the user to run the AdaBoost to diferentes datasets, and evalueate the algorithm in diferentes aspects.

Material and Methodology

Essentially, as said before, boosting consists of repeatedly using a base weak learning algorithm, on differently weighted versions of the training data, yielding a sequence of weak classifiers that are combined in a addiction function. The weighting of each model depends on the accuracy of the previous, in order to increase the importance of classify correctly wrong predicted observations from the last model. The ensemble prediction function of AdaBoost $H: X \rightarrow \{-1,1\}$ is given by

$$H(\boldsymbol{x}) = sign\left(\sum_{m=1}^{M} \alpha_m \, h_m(\boldsymbol{x})\right) \qquad (1)$$

where $\boldsymbol{\alpha_1}, \ldots, \boldsymbol{\alpha_M}$ is a set of weights from the respective $\boldsymbol{h_1}, \ldots, \boldsymbol{h_M}$ set of models.

To build this model, we followed the pseudo-code below, using the base models $\boldsymbol{h_i}$ as stump models (decision trees with justo ne Split node) and fully grow trees.

The pseudo code outline is:

- Given: (x_1, y_1),..., (x_n, y_n), where $x_i \in X$, $y_i \in \{-1,1\}$,

- Initialize : $D_1(i) = \frac{1}{n}$ for $i = 1, \dots, n$
- For $m = 1, \dots, M$
  - Train weak learner learner using Distribution D_m
  - Get the hypothesis $h_m: X \to \{-1,1\}$
  - Aim: Select h_m with low weighted error.

$$\epsilon_m = Pr_{i \sim D_t}[h_m(x_i) \neq y_i]$$

  - Choose $\boldsymbol{\alpha_m} = \frac{1}{2} ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$
  - Update for $i = 1, \dots, n$

$$D_{m+1}(i) = \frac{D_m(i)\exp(-\boldsymbol{\alpha}_m y_i h_m)}{Z_m}$$

Where Z_m is a normalization factor.

Then the output will be given by Equation (1).

The datasets used in the Shiny dashboard were generated artifically trying to simulate different behaviours ir order to show the flexibilty from AdaBoosting. There are three types: Circles, Moons, and Spirals. Each dataset contains two explanatory variables x_1, x_2 to enable a clear visualization of the observations, and each instance has the response variable $y \in \{-1,1\}$ indicating the class. All databases are balanced. Also, it's presented a noisy version of each data to show the robustness from AdaBoost (Wyner, 2017).

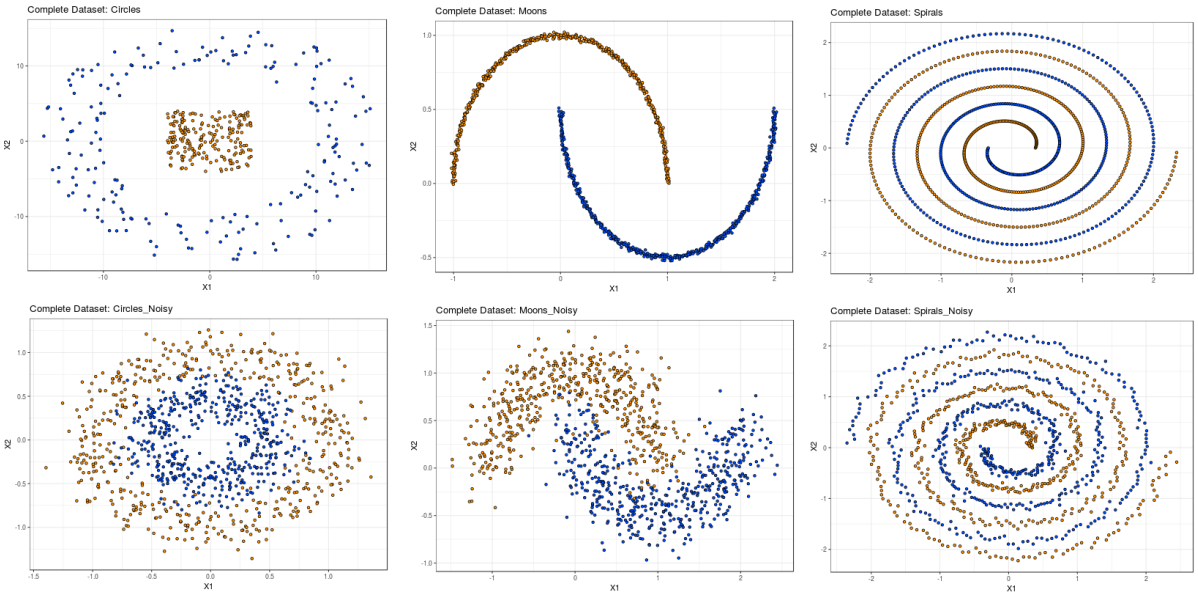

Figure 1 – All Datasets available in Shiny Application.

Authors: MAIA, M. and ARA, A., 2019

Once selected the database, it's possible to configure the parameters to run the AdaBoost model like:

- The proportion of training set and the test of classication model.
- The number of classifiers that AdaBoost will use (represented by M in AdaBoost function).
- The type of model, Stump Trees or Complete Trees, that will be used.

Also, it's possible to generate a animation from the model's construction.

Results and Discussion

The entire results it's given by the ShinyDashboard that can be acessed at https://mateusmaia.shinyapps.io/adaboosting/. The initial panel of application presentes an overviw about the AdaBoosting and a "How-to" for the user confortably peform the modelling. At the side bar, it's posible to set, and define the parameters from the model.

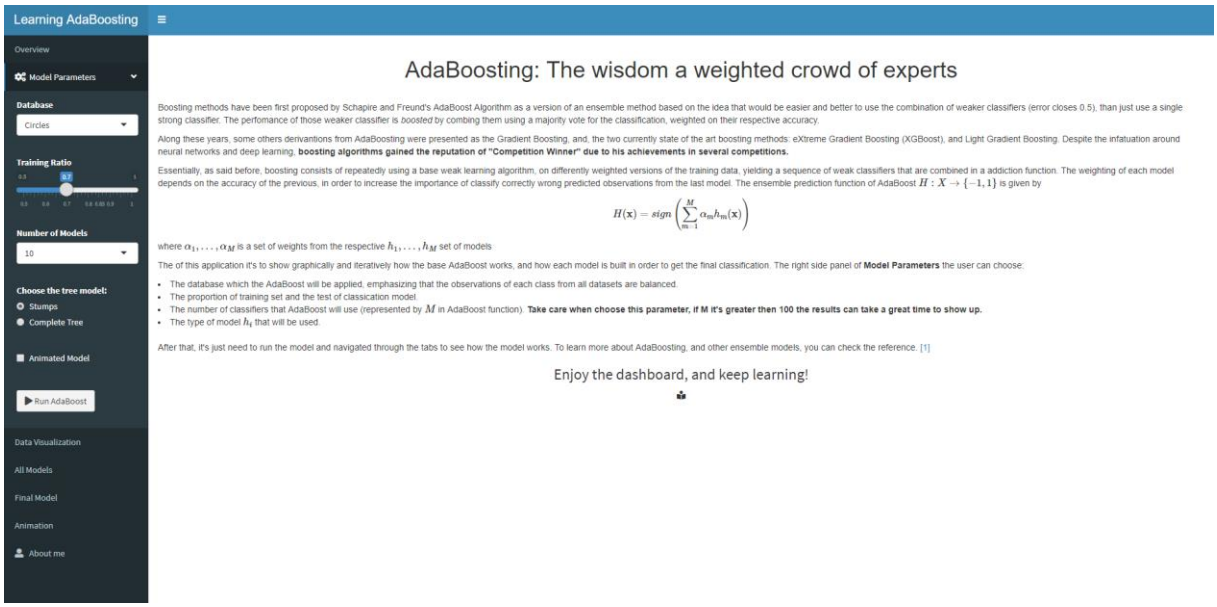

Figure 2 – Main Panel and Model Parameters from Shiny AdaBoost.

Authors: MAIA, M. and ARA, A., 2019

After setting the parameters and run the models, the user can navigate through the other tabs, where it's possible see the behaviour and other characteristics from results. The first tab only displays the configuration from the database selected, the training set and the

test set. It's interesting to visualize the each one of those sets to understand better which observations are used to calculate and predict a model.

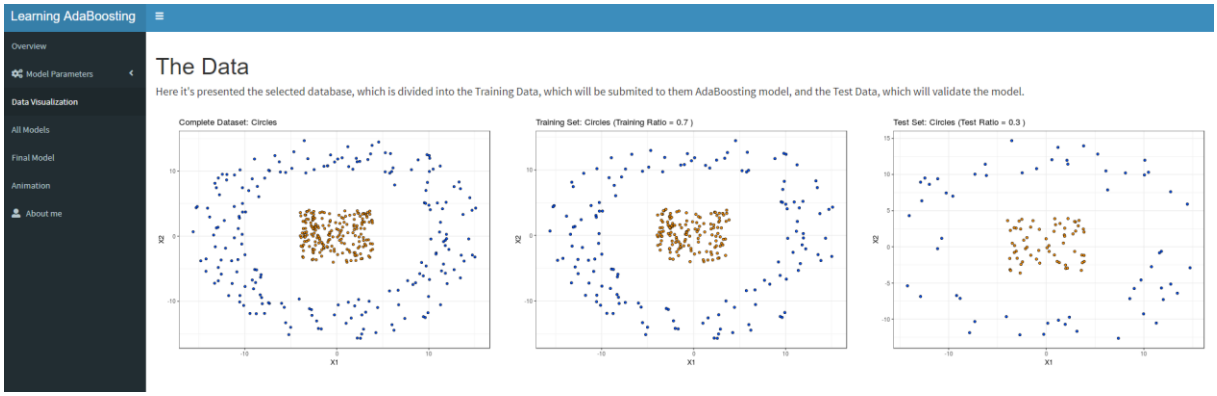

Figure 3 – Data Visualization Panel from Complete Data, Train Data, Test Data.

Authors: MAIA, M. and ARA, A., 2019

The third panel presents all models that were created to build the final. Is importante to emphasize that each plot shows the weight adressed to the observations in each step, the decision boundary and the voting power α_m from each model. So, this tab enable the user to observe beyond the final model, that's is commonly generated from others AdaBoost packages, and possibility to understand how the final result is obtained.

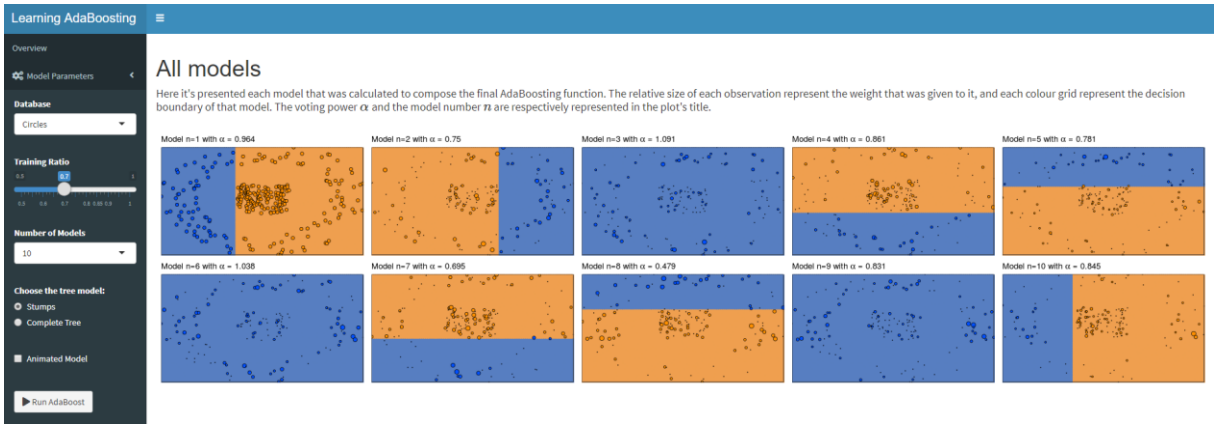

Figure 4 – All models panel, presente the model and their parameters utilized in each step.

Authors: MAIA, M. and ARA, A., 2019

The "Final Model" panel shows the model achieved by the combination from all models, represented by Equation (1), where the first plot presentes the test observations predictions and the decision boundary from the AdaBoost. Also, is presented a plot of the

error rate by the numbers of models used, which can evidence the relation of the greater the model's numbers the smaller the training and error rate, moreover is showed the resistance to overfitting by the stabilization of test error from a determined number of models. The accuracy and elapsed model time are given to use as comparisson metrics between models.

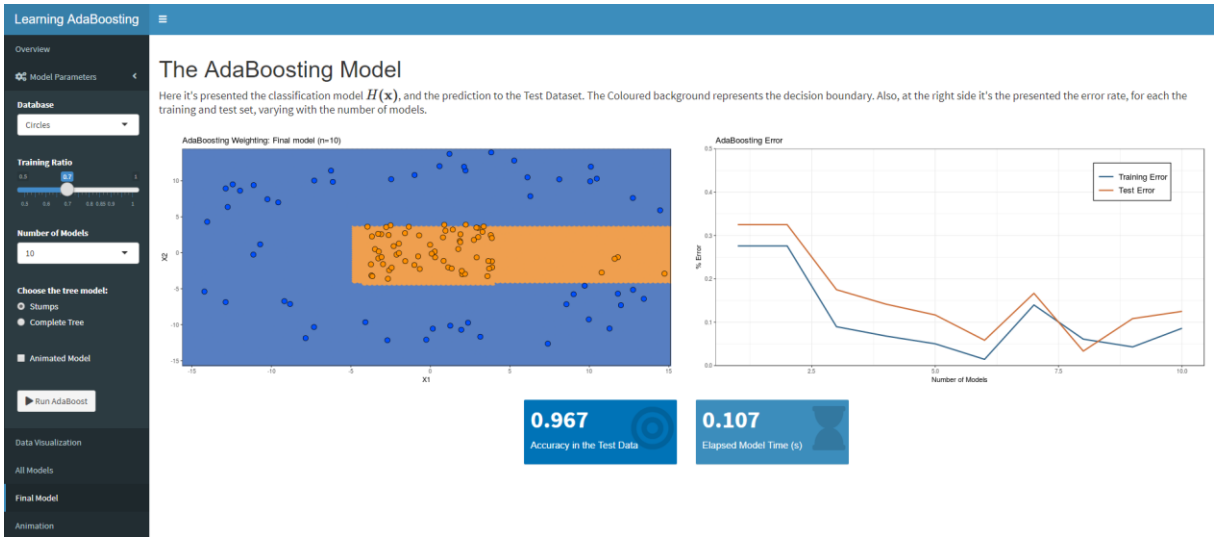

Figure 5 – Final Model Panel with the predictions from the model $H(\boldsymbol{x})$, the error plot and the model evaluation metrics.

Authors: MAIA, M. and ARA, A., 2019

The last panel displays an animated version from all models built to show dinamically how the AdaBoost behaves. Besides that, an animated error plot is also displayed.

Conclusion

The AdaBost can be defined as a powerful ensemble classifier formed by successvely reffiting a weak classifier to different weighted realizations of a data set. In this article we proposed a complete interactive Shiny Dashboard to apply the algorithm to different datasets varying diferentes parameters from the model as the number of models used, as well the type of base learner utilized. The results provided a clear visualization from each step that AdaBoost uses to build the final classifier, moreover spell out the decision boundary generated by the model, the predicitions that were made, the error rates evaluation per number of model, and others evaluation metrics. Regarding future works, the developed Shiny Dashbord may be customizable with others base weak learners as logistic regression (Friedman,2000) and also recently others variations from boosting algorithms as XGBoosting (Chen,2016).

References

CHEN, Tianqi et al. **Xgboost: extreme gradient boosting**. R package version 0.4-2, p. 1-4, 2015.

CHEN, Shi et al. **Boosting part-sense multi-feature learners toward effective object detection**. Computer Vision and Image Understanding, v. 115, n. 3, p. 364-374, 2011.

DRUCKER, Harris; SCHAPIRE, Robert; SIMARD, Patrice. **Boosting performance in neural networks.** In: Advances in Pattern Recognition Systems using Neural Network Technologies. 1993. p. 61-75

FREUND, Yoav. **Boosting a weak learning algorithm by majority**. Information and computation, v. 121, n. 2, p. 256-285, 1995.

FREUND, Yoav; SCHAPIRE, Robert E. **A decision-theoretic generalization of on-line learning and an application to boosting.** Journal of computer and system sciences, v. 55, n. 1, p. 119-139, 1997.

LEISCH, Friedrich; DIMITRIADOU, **mlbench: Machine Learning Benchmark Problems**. R package version 2.1 -1., 2010.

FRIEDMAN, Jerome et al. **Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors).** The annals of statistics, v. 28, n. 2, p. 337-407, 2000.

WICKHAM, Hadley (2017). **tidyverse: Easily Install and Load the 'Tidyverse'**. R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

OOMS, Jeroen (2018). **gifski: Highest Quality GIF Encoder**. R package version 0.8.6.https://CRAN.R-project.org/package=gifski

KEARNS, Michael; VALIANT, **Leslie. Cryptographic limitations on learning Boolean formulae and finite aut omata**. Journal of the ACM (JACM), v. 41, n. 1, p. 67-95, 1994.

KEARNS, M. **Learning Boolean formulae or finite automata is as hard as factoring**. *Technical Report TR-14-88 Harvard University Aikem Computation Laboratory*.,1988.

SALZBERG, Steven L. **C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc.**, 1993. Machine Learning, v. 16, n. 3, p. 235-240, 199.

SAON, George; SOLTAU, Hagen**. Boosting systems for large vocabulary continuous speech recognition**. **Speech communication**, v. 54, n. 2, p. 212-218, 2012.

SCHAPIRE, Robert E. **Explaining adaboost. In: Empirical inference**. Springer, Berlin, Heidelberg, 2013. p. 37-52

SCHAPIRE, Robert E. **The strength of weak learnability**. Machine learning, v. 5, n. 2, p. 197-227, 1990.

URBANEK, Simon **png: Read and write PNG images**. R package version 0.1-7. https://CRAN.R-project.org/package=png, 2013.

RIOS-CABRERA, Reyes; TUYTELAARS, Tinne; VAN GOOL, Luc**. Efficient multi-camera vehicle detection, tra cking, and identification in a tunnel surveillance application.** Computer Vision and Image Understanding, v. 1 16, n. 6, p. 742-753, 2012.

THERNEAU, Terry ; ATKINSON, Beth **rpart: Recursive Partitioning and Regression Trees**. R package versio n 4.1-13. https://CRAN.R-project.org/package=rpart, 2018.

PEDERSEN, Thomas Lin; ROBINSON, David **gganimate: A Grammar of Animated Graphics.** R package versi on 1.0.3.https://CRAN.R-project.org/package=gganimate, 2019.

VIOLA, Paul et al. **Rapid object detection using a boosted cascade of simple features**. CVPR (1), v. 1, p. 511-518, 2001.

CHANG, Winston; CHENG, Joe ; ALLAIRE JJ; XIE, Yihui ,MCPHERSON, Jonathan **shiny: Web Application Fr amework for R.** R package version 1.2.0. https://CRAN.R-project.org/package=shiny, 2018.

CHANG, Winston; RIBEIRO, Barbara ,**shinydashboard: Create Dashboards with 'Shiny'.** R package version 0. 7.1.https://CRAN.R-project.org/package=shinydashboard ,2018.

WYNER, Abraham J. et al. **Explaining the success of adaboost and random forests as interpolating classifiers.** The Journal of Machine Learning Research, v. 18, n. 1, p. 1558-1590, 2017

Attatchment

How the code was too extense, he can be accessed through the link http://rpubs.com/mateusmaia/shinyadaboosting. If this isn't work it please contact the authors.