
Boosting for Regression Using Regression Error Characteristic Curves

Aloísio Carlos de Pina

Gerson Zaverucha

Department of Systems Engineering and Computer Science, COPPE/PESC, Universidade Federal do Rio de Janeiro, C.P. 68511 - CEP. 21945-970, Rio de Janeiro, RJ, Brazil

LONG@COS.UFRJ.BR

GERSON@COS.UFRJ.BR

Abstract

Boosting is one of the most popular methods for constructing ensembles. The objective of this work is to present a boosting algorithm for regression based on the Regressor-Boosting algorithm, in which we propose the use of REC curves in order to select a good threshold value, so that only residuals greater than that value are considered as errors. The algorithm was empirically evaluated and its results were analyzed also by means of REC curves.

1. Introduction

An ensemble (Dietterich, 1998a) is a set of predictors whose individual decisions are combined in some way (typically by weighted or unweighted voting). Ensembles are often much more accurate than the individual predictors that make them up. Many methods for constructing ensembles have been developed (Caruana & Niculescu-Mizil, 2004; Dietterich, 1998a). Some methods are general and can be applied to any learning algorithm.

Boosting (Schapire, 1990) is one of the most popular methods for constructing ensembles. It produces multiple models by repeatedly altering the set of training examples given to an existing learner. Examples are given weights, and at each iteration a new hypothesis is learned and the examples are reweighted to focus the system on examples that the latest hypothesis gets wrong. The final predictor is constructed by a weighted vote of the individual models. Each predictor is weighted according to its accuracy. Boosting was developed by computational learning theorists to guarantee performance improvements for weak learners that only need to generate a hypothesis with an accuracy greater than $1/2$ (Freund & Shapire, 1996). In the past few years, several studies were carried through about boosting (Schapire, 2002).

In (Avnimelech & Intrator, 1999) it is presented a boosting algorithm for regression as a method for training an ensemble of predictors to optimize their collective

performance. The algorithm, called Regressor-Boosting algorithm, is based on a fundamental observation that often the mean squared error (MSE) of a predictor is significantly greater than the squared median of the error, due to a small number of big errors. By reducing the number of big errors, it is possible to reduce the MSE. The problem is to define a threshold from where a residual must be considered as a big error, since several factors are involved.

The objective of this work is to present a boosting algorithm for regression based on the Regressor-Boosting algorithm, in which we propose the use of REC curves (Bi & Bennett, 2003) in order to select a good threshold value, so that only residuals greater than that value are considered as errors. An extensive experimental evaluation was carried through with the objective of analyzing the performance of the new boosting algorithm. We have used in our tests the REC curves proposed by Bi and Bennett (2003). The REC curves facilitate visual comparison of regression functions and they are qualitatively invariant to choices of error metrics and scaling of the residual. Besides, the curve area provides a valid measure of the expected performance of the regression model and the information represented in REC curves can be used to guide the modeling process based on the goals of the modeler.

This paper is organized as follows. The next section has a brief review of the main characteristics of the Regressor-Boosting algorithm. In Section 3, a summary of REC curves is presented. Then, the boosting algorithm for regression using REC curves is presented in Section 4. In Section 5, an experimental evaluation of the new boosting approach is reported. Finally, in Section 6, the conclusions and the plans for future research are presented.

2. The Regressor-Boosting Algorithm

The essence of the algorithm is the construction of an ensemble of three estimators. They are trained on different input distributions and their outputs are combined in order to reduce the big error rate. An error is considered big if it is greater than a threshold value γ (it is then called big error with reference to γ).

Appearing in *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*.

The algorithm works as follow. The training data is split in three sets. The first set is used to train a first expert. The second set is used to build the training set for the second expert. This second training set contains the instances from the second set on which the first expert has a big error and a similar amount of instances on which it does not have a big error. The two experts are tested with the third set, and the training set of the third expert contains only those instances from the third set on which exactly one of the two first experts had a big error. The output of the ensemble is the median of the outputs of the three experts.

Instead of the majority vote of an ensemble used for classification tasks, the combination model used is the median of the three predictors. When the ensemble is used for prediction the worst of the three estimates for any sample point is irrelevant, because the median must be one of the other estimates. It was shown that the median is theoretically and empirically better than the average of the outputs (Avnimelech & Intrator, 1999) and may also be applied to more recent versions of the boosting algorithm.

3. Regression Error Characteristic Curves

Results achieved by Provost, Fawcett and Kohavi (1998) and indicate ROC analysis (Provost & Fawcett, 1997) as a superior methodology than the accuracy comparison in the evaluation of classification learning algorithms. But ROC curves are limited to classification problems. Regression Error Characteristic (REC) curves (Bi & Bennett, 2003) generalize ROC curves to regression with similar benefits. As in ROC curves, the graph should characterize the quality of the regression model for different levels of error tolerance.

The REC curve is a technique for evaluation and comparison of regression models that facilitates the visualization of the performance of many regression functions simultaneously in a single graph. An REC graph contains one or more monotonically increasing curves (REC curves) each corresponding to a regression model.

One can easily compare many regression functions by examining the relative position of their REC curves. The shape of the curve reveals additional information that can be used to guide modeling.

REC curves plot the error tolerance on the x-axis and the accuracy of a regression function on the y-axis. Accuracy is defined as the percentage of points predicted within the tolerance. A good regression function provides a REC curve that climbs rapidly towards the upper-left corner of the graph, in other words, the regression function achieves high accuracy with a low error tolerance.

In regression, the residual is the analogous concept to the classification error in classification. The residual is defined as the difference between the predicted value $f(x)$ and actual value y of response for any point (x, y) . It could

be the squared error $(y - f(x))^2$ or absolute deviation $|y - f(x)|$ depending on the error metric employed. Residuals must be greater than a tolerance e before they are considered as errors.

The area over the REC curve (AOC) is a biased estimate of the expected error for a regression model. It is a biased estimate because it always underestimates the actual expectation. If e is calculated using the absolute deviation (AD), then the AOC is close to the mean absolute deviation (MAD). If e is based on the squared error (SE), the AOC approaches the mean squared error (MSE). The evaluation of regression models using REC curves is qualitatively invariant to the choices of error metrics and scaling of the residual. The smaller the AOC is, better the regression function will be.

In order to adjust the REC curves in the REC graph, a null model is used to scale the REC graph. Reasonable regression approaches produce regression models that are better than the null model. The null model can be, for instance, the mean model: a constant function with the constant equal to the mean of the response of the training data.

An example of REC graph can be seen in Figure 1. The number between parentheses in the figure is the AOC value for each REC curve. A regression function dominates another one if its REC curve is always above the REC curve corresponding to the other function. In the figure, the regression function dominates the null model, as should be expected.

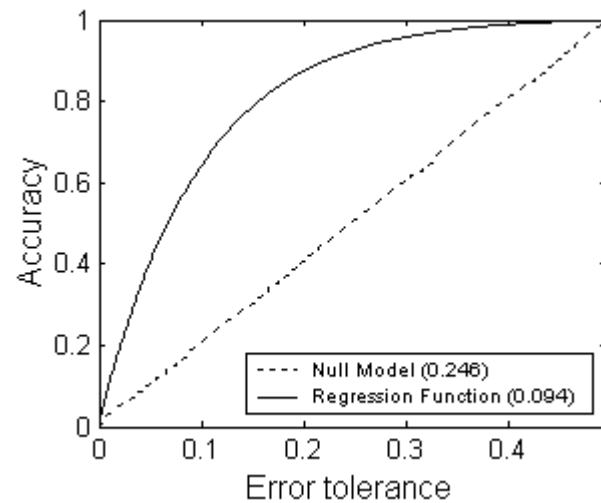


Figure 1. Example of REC graph.

4. Boosting for Regression Using REC Curves

The Regressor-Boosting Algorithm uses a threshold γ for big errors. In practice there are several considerations when choosing a value for γ , such as the size of the data set. The optimal γ is one for which the big errors are responsible for a significant part of the MSE, but the big error rate is low. Usually the sets on which the second and

third estimators are trained are more difficult and have a higher big error rate. In most cases, the choice of a good γ may require tuning.

We propose the use of REC curves in order to find a good value for γ . We select the value for the threshold γ as function of the AOC of the REC curve obtained for each expert. The big errors are then those residuals greater than $f(\text{AOC})$, where $f(\cdot)$ must be defined. We achieved good results by simply multiplying the AOC value by a scalar, adjusted by means of a validation set in an internal cross-validation (Kohavi, 1995; Mitchell, 1997).

Nevertheless, the best improvement provided by this approach is that we can access all the benefits of the REC curves described in Section 3, making possible a better evaluation of each part of the ensemble and facilitating the tuning. The algorithm of boosting for regression using REC curves is depicted in Algorithm 1.

Algorithm 1. Boosting for regression using REC curves.

- | |
|---|
| <ol style="list-style-type: none"> 1. Split the data set in training set and test set. 2. Split the training set to three sets: Set1, Set2 and Set3. 3. Train Expert1 on TrainingSet1 = Set1. 4. Test Expert1 on Set2. Construct the REC curve, compute the AOC and determine the threshold γ. Add to TrainingSet2 all instances on which Expert1 has a big error. Add to TrainingSet2 a similar amount of instances on which Expert1 does not have a big error. 5. Train Expert2 on TrainingSet2. 6. Test Expert1 and Expert2 on Set3. Construct the REC curves, compute the AOCs and determine the thresholds γ_1 and γ_2. Add to TrainingSet3 all instances on which exactly one of the experts (Expert1 or Expert2) has a big error (with reference to γ_1 for Expert1 and with reference to γ_2 for Expert2). 7. Train Expert3 on TrainingSet3. 8. The output of the ensemble for each instance of the test set is the median of the outputs of the three experts. |
|---|

5. Experimental Evaluation

An extensive experimental evaluation was carried through with the objective of analyzing the performance of the new boosting algorithm. This section describes the aspects of the experiments and shows their results.

In the experiments, 20 data sets were used in order to include several domains and difficulties. The data sets have been obtained from the UCI Machine Learning Repository (Blake & Merz, 2005), Delve repository (<http://www.cs.toronto.edu/~delve/>), MLnet Archive (<http://www.mlnet.org/>), Luís Torgo's Home Page (<http://www.niaad.liacc.up.pt/~ltorgo/>), StatLib data (<http://lib.stat.cmu.edu/>) and Brazilian utilities (Teixeira & Zaverucha, 2003). In Table 1 we present the used data sets and a summary of their main characteristics: the file name, the number of instances and the total number of attributes. The test method used in this research was the 10-fold cross-validation (Dietterich, 1998b).

Table 1. Data sets used in the experimental evaluation.

Data Set	File	Insts	Atts
Abalone	abalone	4177	9
Airplane Companies	stock	950	10
Auto Imports	autoPrice	159	16
Auto-Mpg	auto	398	8
Bank	bank8FM	4499	9
Basketball Points	basketball	96	5
Brazilian utilities	electricload	83	13
Breast Cancer	r_wpbc	194	33
California Housing	cal_housing	20640	9
Census	house_8L	22784	9
Computer Activity	cpu_small	8192	13
Elevators	elevators	8752	19
Housing	housing	506	14
Kinematics	kin8nm	8192	9
Machine-Cpu	machine	209	7
Pole Telecomm	pol	5000	49
Pollution	pollution	60	16
Pumadyn	puma8NH	4499	9
pwLinear	pwLinear	200	11
Triazines	triazines	186	61

The base learners used in the experimental tests of the Regressor-Boosting algorithm reported in (Avnimelech & Intrator, 1999) were Neural Networks (Bishop, 1995; Haykin, 1998). So, in order to compare our conclusions to those achieved by Avnimelech and Intrator, we also used Neural Networks in our tests.

The implementation of the Neural Networks (multi-layer perceptrons) was obtained from the WEKA System (<http://www.cs.waikato.ac.nz/~ml/weka/>). The network uses backpropagation to train (Rumelhart & McClelland, 1986). The hidden layer is composed by n nodes with sigmoid activation function, where n is equal to half of the number of attributes. The output is a single unthresholded linear unit. The learning rate is equal to 0.3. In order to avoid overfitting, the number of epochs to train was adjusted by means of an internal cross-validation with validation sets.

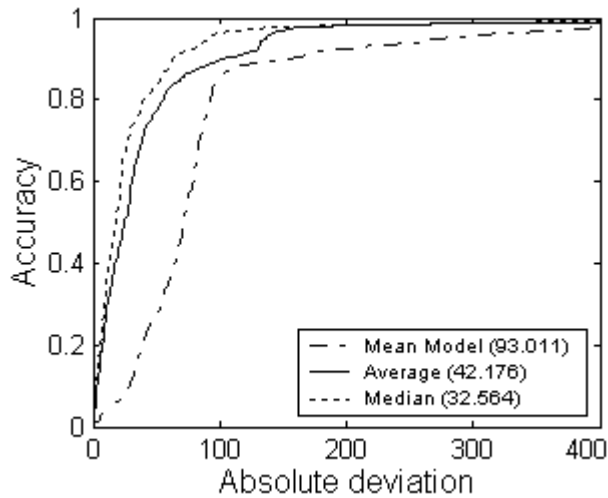


Figure 2. Comparison of median and average as methods for combining the outputs of the three experts for the Machine-Cpu data set.

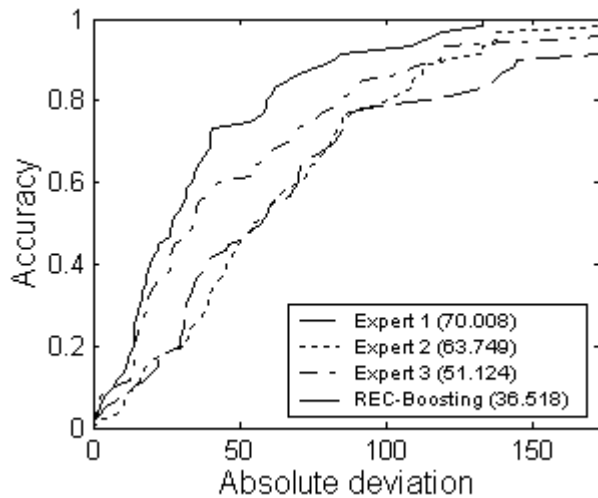


Figure 3. Comparison of the individual performances of the experts and the boosting for the Pollution data set.

We tested both methods for combining the outputs of the three experts: the median and the average. Some tests confirmed the results achieved by Avnimelech and Intrator (1999), in which the median showed to be the best approach. Figure 2 shows one of these cases. However, in the majority of tests the difference was negligible.

We verified that the performance of the boosting is better or as good as that of the best expert. Figure 3 illustrates this conclusion.

We have compared the Boosting for Regression using REC curves algorithm to the Regressor-Boosting algorithm, to a single Neural Network and to the mean model. The Neural Network was trained with the complete training set (the union of Set1, Set2 and Set3). The AOCs of the REC curves provided by each model for each data set are shown in Table 2. A summary of the results is presented in Table 3. The number of wins consists in to count the number of data sets where the Boosting for Regression using REC curves algorithm achieved a higher performance than a second model and to count the number of data sets where it achieved a lower performance. The number of significant wins works in the same way, but a win is counted only if it is statistically significant. In order to determine if the difference between the performances of two models is statistically significant, we have used the t-test at the 0.05 significance level. The last measure of performance considered is the score (Domingos, 1996), where all models are compared simultaneously: for each data set, each model received a number of points between 0 and 3 (the best model received 3, the second place received 2 and so on until 0). The final value for each model is the ratio between the sum of all points received and the maximum number of points possible. Considering all the measures of performance, the Boosting for Regression using REC curves outperformed the other models.

For some data sets, the improvement in the results provided by the new boosting approach is clearly noted, as can be seen, for instance, in Figure 4. Note that, although the REC curves are near, the curve corresponding to the Boosting for Regression using REC curves covers almost completely the others, thus the regression function generated by the ensemble dominates the other functions and therefore it is preferable.

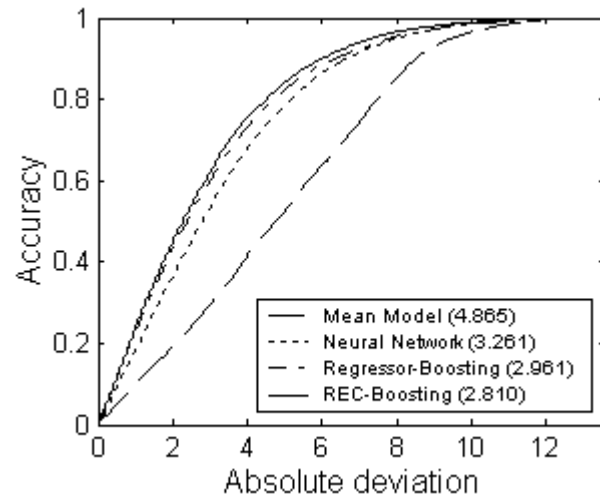


Figure 4. Comparison of the models for the Pumadyn data set.

Table 2. AOCs of the REC curves provided for each data set.

Data Set	Mean Model	Neural Network	Regressor-Boosting	REC-Boosting
Abalone	2.111	1.577	1.729	1.527
Airplane Companies	5.440	5.559	5.185	6.627
Auto Imports	4488.607	2886.862	2176.815	2095.494
Auto-Mpg	6.475	4.331	3.819	4.273
Bank	0.124	0.027	0.025	0.023
Basketball Points	0.083	0.094	0.080	0.091
Brazilian utilities	0.056	0.062	0.054	0.058
Breast Cancer	29.146	34.471	30.907	31.496
California Housing	91144.647	56490.332	49086.957	53271.561
Census	32384.742	34602.705	24861.507	21257.289
Computer Activity	10.443	2.407	2.638	2.916
Elevators	0.0044	0.0018	0.0018	0.0017
Housing	6.566	6.837	5.986	5.946
Kinematics	0.216	0.124	0.106	0.109
Machine-Cpu	93.011	44.546	44.544	32.564
Pole Telecomm	32.211	10.465	10.547	9.328
Pollution	48.050	43.981	54.374	36.518
Pumadyn	4.865	3.261	2.961	2.810
PwLinear	3.566	1.402	1.896	1.986
Triazines	0.116	0.116	0.123	0.110

Table 3. Summary of results.

Measure	Mean Model	Neural Network	Regressor-Boosting	REC-Boosting
Number of wins	16-4	17-3	11-9	-
Number of significant wins	14-0	7-1	5-2	-
Score	21.7	40.0	65.0	73.3

6. Conclusions and Future Works

We have presented here a boosting algorithm for regression that uses Regression Error Characteristic curves in order to define a good threshold for what we can consider as an error.

Experimental tests have demonstrated the efficacy and applicability of the approach. By analyzing the REC curves and three measures of performance, we could

verify that the Boosting for Regression using REC curves outperformed all models tested.

As a future work, we intend to apply this boosting algorithm with other regression algorithms in order to investigate their behaviors and verify if the performance of one of them can be improved to the point of be considered as a good choice over the others for most regression problems.

Acknowledgments

The authors are partially financially supported by the Brazilian Research Agency CNPq.

References

- Avnimelech, R., & Intrator, N. (1999). Boosting Regression Estimators. *Neural Computation*, *11*, 491–513.
- Bi, J., & Bennett, K. (2003). Regression Error Characteristic Curves. *Proceedings of the 20th International Conference on Machine Learning* (pp. 43–50). Washington, DC.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.
- Blake, C. L., & Merz, C. J. (2005). *UCI Repository of Machine Learning Databases*. Machine-readable data repository, Department of Information & Computer Science, University of California, Irvine. [http://www.ics.uci.edu/~mlearn/MLRepository.html]
- Caruana, R., & Niculescu-Mizil, A. (2004). An Empirical Evaluation of Supervised Learning for ROC Area. *Proceedings of the 1st Workshop on ROC Analysis in Artificial Intelligence* (pp. 1–8).
- Dietterich, T. G. (1998a). Machine Learning Research: Four Current Directions. *The AI Magazine*, *18*, 97–136.
- Dietterich, T. G. (1998b). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, *10*, 1895–1924.
- Domingos, P. (1996). Unifying Instance-Based and Rule-Based Induction. *Machine Learning*, *24*, 141–168.
- Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the 13th International Conference on Machine Learning* (pp. 148–156). Bari, Italy.
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (pp. 1137–1143). Morgan Kaufmann, San Mateo, CA.
- Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw-Hill.
- Provost, F., & Fawcett, T. (1997). Analysis and Visualization of Classifier Performance: Comparison Under Imprecise Class and Cost Distributions. *Proceedings of the International Conference on Knowledge Discovery and Data Mining* (pp. 43–48). Newport Beach, CA: AAAI Press.
- Provost, F., Fawcett, T., & Kohavi, R. (1998). The Case Against Accuracy Estimation for Comparing Classifiers. *Proceedings of the 15th International Conference on Machine Learning* (pp. 445–453). Morgan Kaufmann.
- Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing: exploration in the microstructure of cognition, 1 & 2*. Cambridge: MIT Press.
- Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.
- Schapire, R. E. (2002). The boosting approach to machine learning: An overview. *MSRI Workshop on Nonlinear Estimation and Classification*.
- Teixeira, M., & Zaverucha, G. (2003). Fuzzy Bayes and Fuzzy Markov Predictors. *Journal of Intelligent and Fuzzy Systems*, *13*, 155–165.