CS-BIGS 5(1): 12-16
http://www.bentley.edu/centers/sbigs/olinsky.pdf
© 2012 CS-BIGS

Assessing Gradient Boosting in the Reduction of Misclassification Error in the Prediction of Success for Actuarial Majors

Alan Olinsky
Bryant University, USA

Kristin Kennedy
Bryant University, USA

Bonnie Brayton Kennedy
Salve Regina University USA

This paper provides a relatively new technique for predicting the retention of students in an actuarial mathematics program. The authors utilize data from a previous research study. In that study, logistic regression, classification trees, and neural networks were compared. The neural networks (with prior imputation of missing data) and classification trees (with no imputation required) were most accurate. However, in this paper, we examine the use of gradient boosting to improve the accuracy of classification trees. We focus on trees since they generate transparent rules that are easily interpretable, especially by non-statisticians. Gradient boosting is an enhancement that is applied specifically to decision trees, and we show that it does, at least in this study, improve the classification accuracy of our default tree. The exposition is accessible to readers with an intermediate level of statistics.

Keywords: *Logistic Regression, Data Mining, Neural Nets, Decision Trees, Gradient Boosting*

Introduction

This paper analyzes data pertaining to the retention of students in an actuarial mathematics program; the actual data were related to 328 Bryant University graduating students who had declared Actuarial Mathematics (AM) as incoming freshmen and remained in the major. In a previous paper (Schumaker et al., 2010), the data were reviewed using data mining techniques, such as logistic regression, neural networks, and decision trees. The original purpose of analyzing the data was to investigate the likelihood that incoming college freshmen, who declared AM as their major, actually did graduate in AM.

A logistic regression, decision tree, and neural network were previously used to predict the successful completion of the program based on the following variables: Math SAT (MSAT), verbal SAT (VSAT), percentile rank in the students' high school graduating class (RANK), his/her percentage score on a college mathematics placement exam (PMT), administered before classes begin in freshman year, and gender (GENDER).

This paper re-analyzes the data by using gradient boosting with decision trees in an attempt to further reduce the misclassification error obtained with these trees from the previous study (Schumaker et al., 2010). The data set is provided with the paper for others to conduct their own analyses.

Gradient boosting can be considered when trying to reduce several different types of errors, depending on the problem. In this study we examine the misclassification rate as the error rate, namely, the percent of predictions of the value of the target variable by the model that are incorrect.

Decision trees also give results that are transparent and clearly understood, compared to, for example, neural networks. In addition, decision trees can handle missing values whereas logistic regression can only be performed when cases with missing values are excluded from the analysis or imputation occurs first. Indeed, there were a fair number of missing values in the original data set. We examine whether the technique of gradient boosting, which was developed by Jerome Friedman (Friedman, 2001, 2002), might improve the fit of the model over traditional decision tree analysis. Specifically, we focus on the misclassification error.

Decision Tree Algorithms

Traditional decision trees use algorithms that search for an optimal partition of the data defined in terms of the values of a single target variable. The optimality criterion depends on how this target variable is distributed into the partition segments. The more similar the target values are within the segments, the greater the worth of the partition. Most partitioning algorithms further partition each segment by recursive partitioning. The partitions are then combined to create a predictive model, which is evaluated by goodness-of-fit statistics defined in terms of the target variable (Georges 2009).

Decision trees do not impute the data; the tree can still decide how the partition should be calculated, even with missing data. Logistic regression and neural networks either impute the data or delete that particular instance of data. Often the average data value is used. For example, if there are 5 variables and 1 variable is missing, all of the data for that instance will be lost in a logistic regression model.

Gradient Boosting

Gradient boosting is a method that is specifically applied to decision trees, and is intended to improve their results. As stated in Enterprise Miner's overview of the Gradient Boosting Node (Georges 2009):

> *"Gradient boosting is a boosting approach that resamples the data set several times to generate results that form a weighted average of the resampled data set. Tree boosting creates a series of decision trees which together form a single predictive model. A tree in the series is fit to the residual of the prediction from the earlier trees in the series. This residual is defined in terms of the derivative of a loss function. For squared error loss with an interval target the residual is simply the target value minus the predicted value. Each time the data are used to grow a tree and the accuracy of the tree is computed. The successive samples are adjusted to accommodate previously computed inaccuracies. Because each successive sample is weighted according to the classification accuracy of previous models, this approach is sometimes called stochastic gradient boosting. Boosting is defined for binary, nominal, and interval targets.*
>
> *Like decision trees, boosting makes no assumptions about the distribution of the data. For an interval input, the model only depends on the ranks of the values. For an interval target, the influence of an extreme value depends on the loss function. The Gradient Boosting node offers a Huber M-estimate loss which reduces the influence of extreme target values. Boosting is less prone to overfit the data than a single decision tree, and if a decision tree fits the data fairly well, then boosting often improves the fit."*

This process is similar to a bootstrapping technique in that many trees are generated. With each successive tree, it is hoped that gradient boosting will reduce the error. Errors can be defined in different ways. For this study, in which we were trying to correctly predict success in an actuarial program, we found that misclassification error was most appropriate. An advantage of stochastic gradient boosting is that it is not necessary to select predictor variables ahead of time. It is also not necessary to transform predictor variables. In addition, gradient boosting is resistant to outliers as the steepest gradient algorithm stresses points that are close to the correct classification.

It should be noted that gradient boosting is functionally similar to random forests since it creates a tree ensemble, and it also uses randomization during the creation of the trees. However, whereas a random forest builds the trees in parallel and these trees "vote" on the prediction, gradient boosting creates a series of trees in which the prediction receives incremental improvement by each tree in the series.

Software

All output and data analysis for this paper were generated using SAS Enterprise Miner (2009) software, Version 6.1 of the SAS System for Windows.

This software can be used as a standalone package on an individual machine or through On Demand, in which the user logs on to a cloud computer at SAS and runs the software on the server (http://www.sas.com/success/bryantuniversity.html). We have found SAS to be very reasonable in its pricing to academia and trainers' kits are provided free of charge to academics. Enterprise Miner includes modeling nodes for the decision trees and for gradient boosting used in this study.

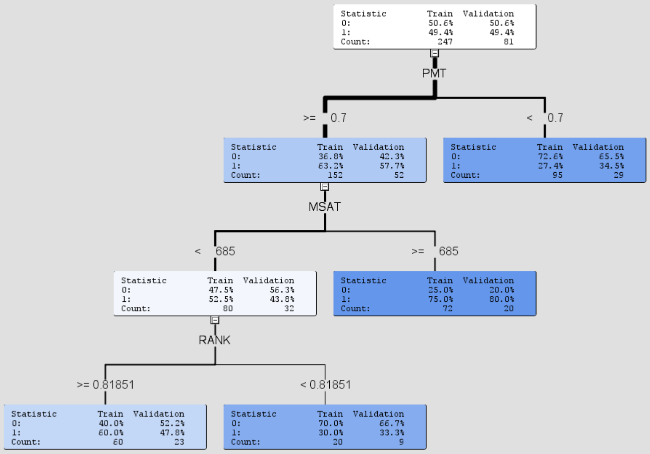
Model

This model includes the nodes from the previous study. Figure 1 is actually a process flow diagram that demonstrates the original model with the new enhancement of gradient boosting. It should be noted that the process flow begins with the dataset, then a data partition (to separate the data into training and validation sets), an impute node (to replace missing data for the subsequent regression and neural network nodes), the decision tree (using default settings) and gradient boosting nodes as well as other modeling tools. The results from the gradient boosting method are compared to those from the previous study (Schumaker et al. 2010).

Results

Beginning with a decision tree for this problem, it can be noted that the tree in Figure 2 is explanatory and easy to interpret. We can see that our placement exam is most important in determining the success of our actuarial students, followed by their Math SAT score and then finally by their rank in class in their senior year of high school.

To determine if the technique of gradient boosting improves the results obtained from the decision tree, we examine the results from the original decision tree. These results are portrayed in Table 1, in the column labeled Valid: Misclassification Rate. It can be noted that the misclassification rate for decision trees for the validation data is .358025. It is important to analyze the results from the validation set to make sure that the model is applicable to data sets other than the training set that was used to construct the original model. In clarifying this column then, this model misclassifies approximately 36% of the cases. On the other hand, it correctly classifies approximately 64% of the cases.



Statistic	Train	Validation
0:	50.6%	50.6%
1:	49.4%	49.4%
Count:	247	81

Statistic	Train	Validation
0:	36.8%	42.3%
1:	63.2%	57.7%
Count:	152	52

Statistic	Train	Validation
0:	72.6%	65.5%
1:	27.4%	34.5%
Count:	95	29

Statistic	Train	Validation
0:	47.5%	56.3%
1:	52.5%	43.8%
Count:	80	32

Statistic	Train	Validation
0:	25.0%	20.0%
1:	75.0%	80.0%
Count:	72	20

Statistic	Train	Validation
0:	40.0%	52.2%
1:	60.0%	47.8%
Count:	60	23

Statistic	Train	Validation
0:	70.0%	66.7%
1:	30.0%	33.3%
Count:	20	9

Figure 2. Decision Tree

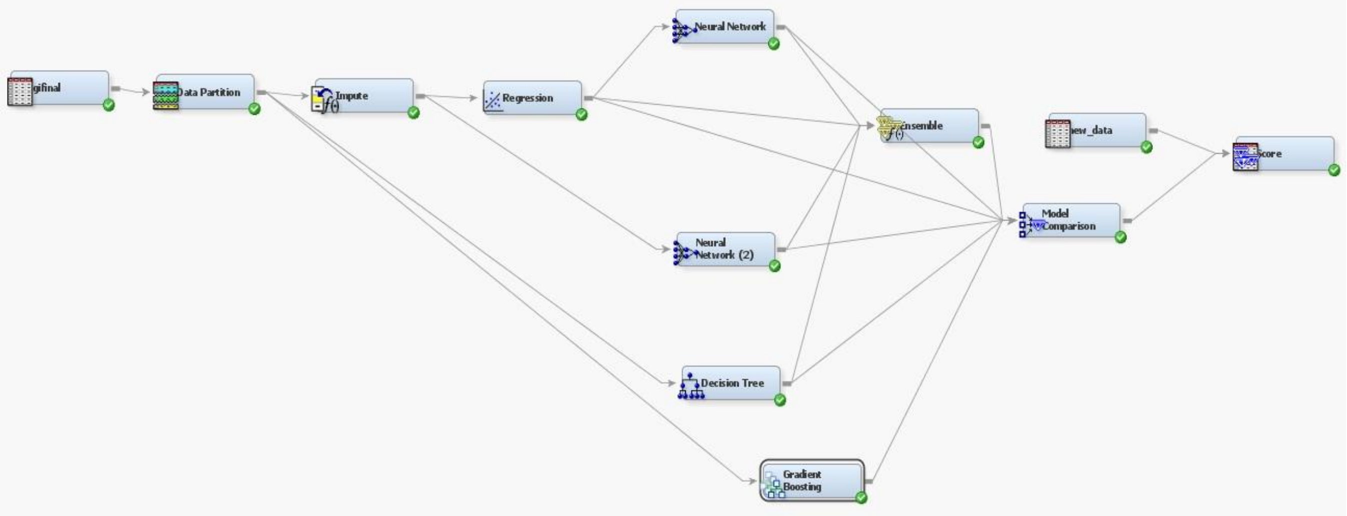
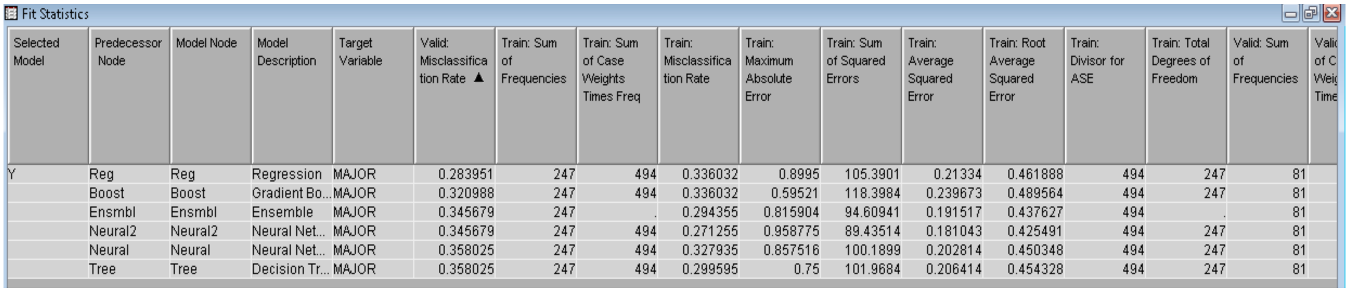


Figure 1. Complete Model in Enterprise Miner

Table 1. Fit Statistics of Alternative Methods



Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Valid: Misclassification Rate ▲	Train: Sum of Frequencies	Train: Sum of Case Weights Times Freq	Train: Misclassification Rate	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Root Average Squared Error	Train: Divisor for ASE	Train: Total Degrees of Freedom	Valid: Sum of Frequencies	Vali of C Weig Time
Y	Reg	Reg	Regression	MAJOR	0.283951	247	494	0.336032	0.8995	105.3901	0.21334	0.461888	494	247	81	
	Boost	Boost	Gradient Bo...	MAJOR	0.320988	247	494	0.336032	0.59521	118.3984	0.239673	0.489564	494	247	81	
	Ensmbl	Ensmbl	Ensemble	MAJOR	0.345679	247	.	0.294355	0.815904	94.60941	0.191517	0.437627	494	.	81	
	Neural2	Neural2	Neural Net...	MAJOR	0.345679	247	494	0.271255	0.958775	89.43514	0.181043	0.425491	494	247	81	
	Neural	Neural	Neural Net...	MAJOR	0.358025	247	494	0.327935	0.857516	100.1899	0.202814	0.450348	494	247	81	
	Tree	Tree	Decision Tr...	MAJOR	0.358025	247	494	0.299595	0.75	101.9684	0.206414	0.454328	494	247	81	

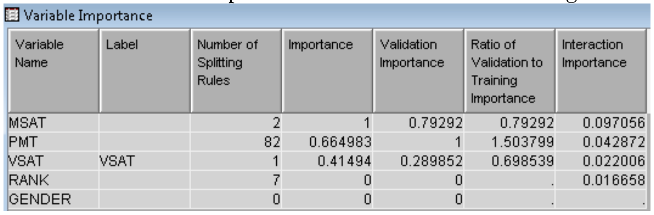
We can compare the results of gradient boosting to all the previously used methods. We notice that the logistic regression had the lowest misclassification error as seen in Table 1, under the results within the column Valid: Misclassification Rate. Gradient boosting came in second for returning the next lowest misclassification error, and gradient boosting was better than the decision tree by itself.

We observe a 4 percentage point improvement on the misclassification rate: 32% misclassification rate with gradient boosting versus 35.8% with a decision tree. It should also be noted that the logistic regression (28.4% misclassification rate) would not have done as well except for the imputation that occurred in a prior operation. Otherwise, the analysis would have suffered from a substantial loss of cases.

When performing gradient boosting on this data set, we thus obtained improved misclassification error results. Unfortunately, no tree is actually displayed when gradient boosting is used, because many trees are incorporated into the model. This is one possible disadvantage of gradient boosting. Nevertheless, we do obtain a Table of Variable Importance resulting from the gradient boosting approach. This is illustrated in Table 2.

Interestingly, variable importance now yields an ordering of MSAT, PMT, and VSAT and does not give much emphasis to RANK or GENDER. The tree prior to using gradient boosting had the ordering of PMT, MSAT, and RANK and did not give much emphasis to VSAT or GENDER.

Table 2. Variable Importance with Gradient Boosting



Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance	Interaction Importance
MSAT		2	1	0.79292	0.79292	0.097056
PMT		82	0.664983	1	1.503799	0.042872
VSAT	VSAT	1	0.41494	0.289852	0.698539	0.022006
RANK		7	0	0	.	0.016658
GENDER		0	0	0	.	.

Conclusions

Although logistic regression, neural networks and decision trees are popular methods for predicting a categorical variable, decision trees are probably a better choice because they generate transparent rules that are easily interpretable, especially by non-statisticians. We found that for this case gradient boosting made decision trees more accurate in terms of reducing the misclassification rate.

Although it may seem that the 4% improvement rate in misclassification error is fairly small, with 328 students, this result is both statistically significant at the .05 level and practically significant in that it could result in approximately 12 additional students being correctly placed in the actuarial major. This improvement rate would certainly yield a major impact in a larger dataset. Of course, the results from these predictive modeling techniques only offer a starting point in deciding the future of our math majors. It should also be mentioned that it is unfortunate that there is no graphical output from gradient boosting to match the output from an individual tree. Nevertheless, an improved model can better predict "success" in future scorings.

However, some caveats are in order. Note that the procedure for applying this method to classification problems requires that separate sequences of (boosted) trees be built for each category. Therefore, it is not wise to analyze categorical dependent variables with many classes as the computations performed may require an unreasonable amount of time. Trees themselves also have shortcomings. The description given by the relationship in the tree may not be the only accurate one. It might appear that certain inputs uniquely explain the variations in the target. However, a completely different set of inputs might give a different explanation that is just as accurate. This was a limited case, and we realize that extensive simulation should be conducted in order to make generalizations about gradient boosting enhancing the results of a decision tree. We hope that other studies will be carried out for other majors, especially at other universities, to verify our results.

References

Georges, J. 2009. Applied Analytics Using SAS Enterprise Miner 5.3: Course Notes; SAS Institute, Cary, N.C., USA.

Friedman, Jerome H. 2001. Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5):1189-1232.

Friedman, Jerome H. 2002. Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4):367-378.

Schumacher, P., Olinsky, A., Quinn, J. 2010. A Comparison of Logistic Regression, Neural Networks, and Classification Trees Predicting Success of Actuarial Students. Journal of Education for Business, 85(5):258-263.

SAS Enterprise Miner 2009. SAS Institute, Cary, N.C., USA.

Correspondence: aolinsky@bryant.edu