2.4 USING MULTIPLE MACHINE LEARNING TECHNIQUES TO IMPROVE THE CLASSIFICATION OF A STORM SET

David John Gagne II*
School of Meteorology
University of Oklahoma
Norman, Oklahoma
djgagne@ou.edu

Amy McGovern
School of Computer Science
University of Oklahoma
Norman, Oklahoma
amcgovern@ou.edu

ABSTRACT

Our approach to the AI contest is based on our research in classifying simulated and observed storms using decision trees (Gagne II and McGovern, 2008). Based on our results in that project, we developed a multi-algorithm ensemble approach that outperforms the baseline decision tree provided by the contest. Our approach uses a combination of decision trees, neural networks, logistic regressions, random forests, and boosting to generate the final model. We combine the models using multi-class linear regression and take a vote over 10 different regression models, each trained using cross validation. This final ensemble model outperforms the baseline decision tree as well as all of the individual models included in the ensemble.

1. INTRODUCTION

As part of a research project on tornado warning verification by storm type, a set of 2,425 storms was classified using a decision tree (Guillot et al., 2008). Since the project's primary focus was tornado warning verification and not storm classification, a very basic decision tree was used. When evaluated on a test set of storms, the decision tree achieved a True Skill Statistic (TSS), a measurement of the accuracy of the prediction that eliminates correct predictions found by random chance (Woodcock, 1976), of .583. In order to improve the accuracy of the classification of storms in the dataset, a contest was announced as part of the AMS Annual Meeting to develop an algorithm that would most improve on the TSS of the original decision tree. This paper describes our entry for the contest.

The various methods that we used in classifying this set of storms fall under the broad category of machine learning classification algorithms. The purpose of machine learning classification algorithms is to sort through various attributes, or descriptive aspects of a data set, and find relationships between those attributes that can be used to predict the value of one or more of the attributes in the data set. All of these algorithms require some form of a training dataset, which is often a dataset that has been already labeled with the correct classification. The model building is done from this data set. In order to evaluate the model, other sets of data, called test sets, are entered into the model so as to determine the accuracy of the model's classifications.

2. DATA

Each storm in the dataset is described by a set of twenty-three different attributes, the same ones used in Guillot et al. (2008). These include morphological characteristics such as aspect ratio, convective area, size, and latitudinal and longitudinal radius; positional characteristics such as latitude and longitude of centroid, speed, orientation to due north, motion east, and motion south; reflectivity-derived attributes; hail attributes such as Maximum Expected Size of Hail (MESH) and Probability Of Severe Hail (POSH); and measurements of low level shear. The storms in the dataset were divided into four types: isolated supercells, convective lines, pulse storms, and non-organized storms, as defined by (Guillot et al., 2008). For the competition, a training set of 1356 storms was provided along with a test set of 1069 storms with the storm type attribute removed.

3. METHODOLOGY

Our approach stemmed from the understanding that an ensemble approach to a machine learning problem will usually outperform

**Corresponding author address:* David John Gagne II, Univ. of Oklahoma, School of Meteorology, 120 David L. Boren Blvd., Norman, OK 73072;
E-mail: djgagne@ou.edu

any single method (Schapire, 1990 1999; Breiman, 1996). By learning a combination of models and having each one contribute a portion of the overall prediction, the ensemble magnifies each model's strengths. The process occurred in three major steps: training multiple models, using multi-class regression to train a combined model, and using an ensemble/bagging approach for evaluating the combined model across the ten-fold-cross-validation and making predictions for the test set.

For the first step of the process we trained multiple machine learning models using the provided training set. The algorithms for these models came from the Waikato Environment for Knowledge Analysis (WEKA), a suite of machine learning algorithms developed at the University of Waikato (Witten and Frank, 2005). This program was used primarily because it contained a very large number of machine learning algorithms, the dataset could be easily manipulated to be interpreted by the program, and the program was written in Java, so its models could simply be imported into another Java program for other uses. We used nine unique models from WEKA along with three sets of attributes for some of the models to bring the total to seventeen. The following models were used:

1. REP Tree: A quick decision tree generator that uses information gain and variance to determine its rules (Witten and Frank, 2005);
2. BF Tree: A decision tree that uses a best first method of determining its branches (Shi, 2007);
3. Logistic Model Tree: A decision tree with logistic regression models at its leaves (Landwehr, 2005);
4. Multilayer Perceptron: a neural network function that uses back-propagation to classify the data (Witten and Frank, 2005);
5. Logistic Model: a logistic regression fitted to the data (Le Cessie, 1992);
6. Random Forest: a method that creates a forest of random trees that is similar to boosting (Brieman, 2001);
7. Ada Boosting with Decision Stumps: creates a series of decision stumps and uses a weighted average of their results for predictions (Freund and Schapire, 1996);
8. Ada Boosting with REP Trees: uses Ada Boosting with REP Trees;
9. Ada Boosting with a Random Forest: uses Ada Boosting with a series of Random Forests

In addition, the Multilayer Perceptron, Logistic Model, Logistic Model Tree, and Random Forest

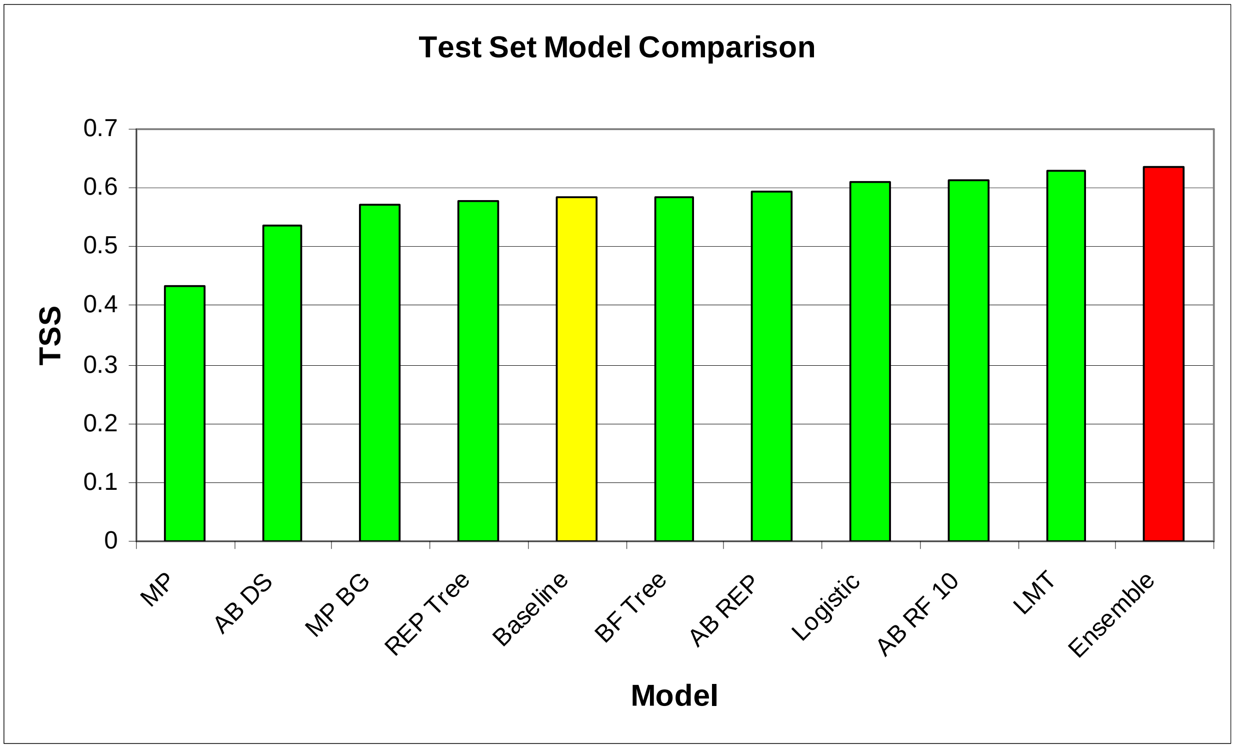

Figure 1. This graph compares the TSS of the individual models and the ensemble approach.

models were run with three different sets of attributes. One set had all the attributes given, one had only the attributes chosen by the REP Tree, and one had only the attributes chosen by the BF Tree. This was done in order to try to reduce over fitting since those models do not have a way of determining the gain of their given attributes. Ten-fold cross validation was used to train each model, and then the predictions from each model were used to build the ensemble model.

In the next step of the process, we combined the predictions of the seventeen total models using multi-class regression (Bishop, 2006). We learned the weights, *W,* for the function:

$$y(w,x) = W^{T}\Phi(x) \qquad (1)$$

The inputs to this function take the form of *x* but they can be transformed through the function ϕ. For this work, $\phi(x) = x$. The standard method for training the weights is:

$$W = \Phi^{+}T \qquad (2)$$

where Φ is a matrix with the individual ϕ for each model, $\oplus$ represents the pseudo-inverse of the matrix Φ, and T is a matrix of the predictions for each example.

For the third step of the process, we used each of the regression models learned from 10-fold cross validation to add an extra layer of ensemble learning. For each of these subsets of the original training set, we repeated step two of the process to create ten individual weighted regression functions and then evaluated the validation and test sets. This generated ten predictions for each instance of the training and validation sets. We used the ten predictions to create one overall prediction by taking the modal prediction of the ten. If most of the folds agree on a particular prediction, then that prediction is most likely the correct one. From this, a final set of prediction of the 10. This was the prediction that we submitted to the competition. Similar to Bagging (Brieman, 1996), the modal vote obtains a more robust estimate of the true prediction under the assumptions of ensemble learning stated above.

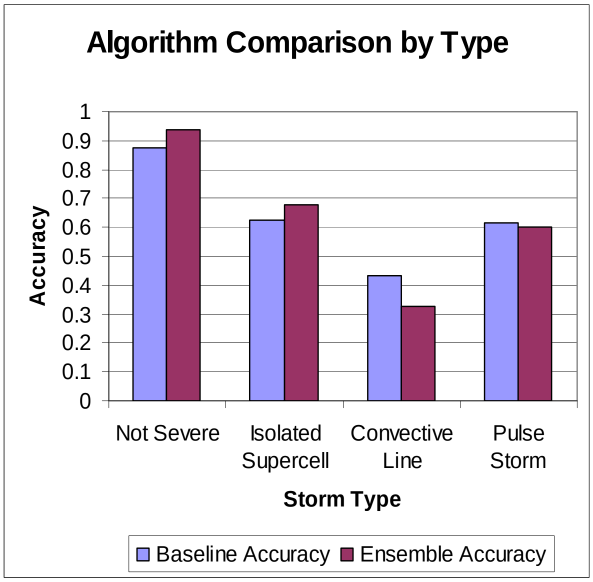

Figure 2. This graph compares the classification accuracy of the four different storm types in the dataset between the baseline decision tree model and the ensemble algorithm.

4. RESULTS

The TSS of the ensemble approach is improved over both the Guillot et al. decision tree and over the individual models. On the test set, the algorithm received a TSS of .637 in comparison to the baseline algorithm's TSS of .583. Figure 1 shows a comparison of accuracy between different models on the test set data. Between the different models there is little variation in TSS. On the test set, the Multilayer

	Non-severe	Supercell	Line	Pulse	Total	Accuracy
Non-severe	534	0	0	35	569	0.938489
Supercell	3	89	11	28	131	0.679389
Line	3	32	31	29	95	0.326316
Pulse	71	17	21	165	274	0.60219
Total	611	138	63	257	1069	
Accuracy	0.8739771	0.644928	0.492063	0.642023		0.766137
TSS	0.6369562					

Table 1. This is the confusion matrix for the ensemble approach. The rows compare the observed results and the columns compare the forecasted results.

Perceptron (.432), Ada Boost with Decision Stumps (.535), Multilayer Perceptron with selected attributes (.57), and the REP Tree (.578) performed worse than the baseline decision tree (.583). On the positive side, the ensemble algorithm submitted to the competition performed better every model.

When examining the accuracies of the specific storm types of our ensemble approach versus the original decision tree approach, some improvement can be found in two of the types, but in the others there was a slight decrease in accuracy. As Figure 2 shows, the accuracy of Not Severe storm classification increased from .875 to .938 and the accuracy of the Supercell classification increased from .626 to .679. However, the classification accuracy for Convective Lines decreased from .432 to .326, and the accuracy for Pulse Storms decreased from .613 to .602.

The confusion matrix in Table 1 shows the numerical results of the test set classification for our ensemble approach. It shows how many storms the algorithm predicted for each type in comparison to what each type was observed to be. It can be used to calculate a variety of statistics, including the accuracy, which was .766.

5. CONCLUSIONS

The work with multiple algorithms on the same dataset showed that incorporating multiple means to examine data into one system can improve the overall classification of the dataset. The ensemble model factors in the relative performance of each model to give more weight to the stronger models and less to the weaker ones, elevating the overall system to higher accuracies than most of the individual models can perform. Other boosting algorithms, such as the 20 tree random forest shown in the graph, had a slightly greater effect on the accuracy and TSS of the classification system, but do not result in major gains in the classification.

Another way to improve the classification of the dataset is to find other attributes that better describe the relationship between the different storm types. If the right attribute is found, it could greatly improve the accuracy of the classification system. We made multiple attempts with this project to find attributes that would fulfill this condition, but none of them proved to be better than the given attributes at classifying the storms. Two of our attempts were adding and multiplying attribute values together to create an index for each storm type, which did not result in any improvement in the classification. We also calculated probability distributions for each storm type and attribute and applied them to every instance in the dataset. It created a large number of new attributes, but the gain from these was still less than the gain from the original attributes. One other solution that was not possible with this project would be to reanalyze the raw data directly to find additional attributes from it and use those to improve the calculation. We are doing just that with a combined dataset of simulated storms and actual storms (Gagne et al., 2008) and have demonstrated that these additional attributes considerably improve the classification system.

ACKNOWLEDGEMENTS

We would like to thank the University of Oklahoma’s College of Engineering for their seed funding and the American Meteorological Society Artificial Intelligence Committee and Weather Decision Technologies for their sponsorship of the AI Competition and for providing funds to travel to the 2008 AMS Annual Meeting where the competition is being held. Special thanks go out to Valliappa Lakshmanan, Beth Ebert, and Sue Ellen Haupt for their coordination of the competition process and assistance with various matters. Special thanks also go to the University Of Oklahoma School Of Meteorology for providing additional travel funds for the 2008 AMS Annual Meeting.

REFERENCES

Brieman, L., 1996: Bagging Predictors. *Machine Learning*, **26**(2), 123-140.

Breiman, L., 2001: Random forests. *Machine Learning*, **45**, 5-32.

Bishop, C. M., 2006: *Pattern Recognition and Machine Learning*. Springer, 729 pp.

Freund, Y. and R. E. Schapire, 1996: Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, San Francisco, CA, 148-156.

Gagne II, D.J., A. McGovern, J. Brotzge, 2008: Automated classification of convective areas from radar reflectivity using decision trees. Preprints, *19th Conference on Probability and Statistics and Sixth Conference on Artificial Intelligence Applications to Environmental Science*, New Orleans, LA, Amer. Meteor. Soc., CD-ROM, J4.6.

Guillot, E., V. Lakshmanan, T. Smith, G. Stumpf, D. Burgess, and K. Elmore, 2008: Tornado and severe thunderstorm warning forecast skill and its relationship to storm type. Preprints, *24th Conference on IIPS*, New Orleans, LA, Amer. Meteor. Soc., CD-ROM, 4A.3.

Landwehr, N., M. Hall, and E. Frank, 2005: Logistic Model Trees. *Machine Learning*, **59**, 161-205.

le Cessie, S., J.C. van Houwelingen, 1992: Ridge estimators in logistic regression. *Applied Statistics*, **41**, 191-201

Schapire, R. E.,1990: The strength of weak learnability. *Machine Learning*, **5**(2), 197-227.

Schapire, R. E., 1999: Theoretical view of boosting and applications. In *Algorithmic Learning Theory: Proceedings of the 10th International Conference (ALT '99)*, pp 13-25, Springer-Verlag, Berlin.

Shi, H.: Best first decision tree learning. M.S. thesis, Department of Computer Science, University of Waikato, 24 pp.

Witten, I., Frank, E., 2005: *Data Mining: Practical machine learning tools and techniques*. 2nd edition. Morgan Kaufmann, 525 pp.

Woodcock, F., 1976: The evaluation of yes/no forecasts for scientific and administrative purposes. *Monthly Weather Review*, **104**, 1209-1214.