BENEFITS OF CREATING ENSEMBLES OF CLASSIFIERS

Dean Abbott - Abbott Analytics

Introduction

In recent years, there has been an explosion of papers in the data mining community discussing how to combine models or model predictions, and the reduction in model error that results. By combining predictions, more robust and accurate models nearly always improve without the need for the high-degree of fine tuning required for single-model solutions. Typically, the models for the combination process are drawn from the same algorithm family (decision trees), though this need not be the case.

In recent years, there have been many terms used to describe the process of combining models. The most popular methods today, and the methods available in many data mining software packages, are "Bagging" and "Boosting." Breiman [1] introduced Bagging, which combines outputs from decision tree models generated from bootstrap samples (with replacement) of a training data set. Models are combined by simple voting of the individual model output predictions. Fruend and Shapire [2] introduced Boosting, an iterative process of weighting more heavily cases classified incorrectly by decision tree models, and then combining all the models generated during the process.

ARCing by Breiman [3] is a form of boosting that, like boosting weighs incorrectly classified cases more heavily, but instead of the Fruend and Shapire formula for weighting, weighted random samples are drawn from the training data. For neural networks, Wolpert [4] used regression to combine neural network models, calling the process stacking. The improvements in model accuracy have been so significant, Friedman, Hastie, and Tibsharani [5] wrote that boosting "is one of the most important recent developments in classification methodology."

Terms used to describe these algorithms abound. Elder and Pregibon [6] used the term Blending to describe "the ancient statistical adage that 'in many counselors there is safety'". Elder later called this technique, particularly applied to combining models from different classifier algorithm families, Bundling [7]. The same concept has been described as Ensemble of Classifiers by Dietterich [8], Committee of Experts by Steinberg [9], and Perturb and Combine (P&C) by Breiman [3]. The concept is actually quite simple: train several models from the same data set, or from samples of the same data set, and combine the output predictions, typically by voting for classification problems and averaging output values for estimation problems.

It seems that producing relatively uncorrelated output predictions in the models to be combined is necessary to reduce error rates. If output predictions are highly correlated, little reduction in error is even possible as the "committee of experts" have no diversity to draw from, and therefore no means to overcome erroneous predictions. Structural stability of the classifiers to be combined is often described as a prerequisite for Reduction in error, though this is not necessarily the case as will be described shortly.

Decision trees make good candidates for combining because they are structurally unstable classifiers, and produce diversity in classifier decision boundaries. In other words, small perturbations in training data set can result in very different model structures and splits. Intuitively, this makes sense because trees use "stair-step" decision boundaries produced by if-then rules. Combining decision tree models has the positive benefit of smoothing these decision regions, thus improving the robustness of the classifier. Boosting and bagging are almost used exclusively with decision trees in the research and software products today.

Combining can be used for other algorithms, including neural networks and polynomial networks, as long as sufficient model diversity can be produced. Neural network model diversity can be achieved in many ways, such as by changing initial weight values, learning rates, momentum, the number of nodes, and the number of hidden layers. On the other hand, algorithms such as linear regression are not easily improved through combining models because they produce very stable and robust models; typically, changing data sets used for training does not change the resulting regression model coefficients very much. Polynomial networks have considerable structural instability, as different data sets can produce significantly different models. However, the apparent structural instability is deceiving: though many differences are seen in model structure, the differences are often different ways to achieve the same solution. Therefore, the output predictions are not very diverse.

While the reasons combining models works so well are not fully understood, there is ample evidence that improvements over single models are the norm rather than the exception. Breiman [3] demonstrates bagging and arcing improving single CART models on 11 machine-learning data sets in every case. Additionally, he documents that arcing, using no special data preprocessing or classifier manipulation (just read the data and create the model), often achieves the performance of hand-crafted classifiers that were tailored specifically for the data.

While the methods described here combine models from within an algorithm family (decision trees, for example), there is no reason combinations cannot be formed across algorithm families as well.

Examples of Combining Models

An example is shown below to demonstrate combining models across algorithm families. A full description is in a paper to be presented at The Second International Conference on Information Fusion [10]. Models from six algorithm families were trained for a data set. Dozens to hundreds of models were trained for each algorithm type, with only the best within each retained for combining, resulting in six models in the combination. All two- three-, four-, five-, and six-way combinations were created and results for each were tabulated.

Models were combined using a simple voting mechanism, with each algorithm model having one vote. To break ties, however, a slight weighting factor was used: model weights were created so that the models that performed best during training were given slightly more weight than others. The weight values ranged from 1.0 to 1.3. The six algorithms used were neural networks, decision trees, k-nearest neighbor, Gaussian mixture models, radial basis functions, and nearest cluster models.

One of the two data sets included in the paper was the glass data from the UCI machine learning data repository [11]. The data set had 150 training cases, 64 testing cases, 9 inputs, and 6 output classes. Training data refers to the cases that were used to find model weights and parameters. Testing data was used to check the training results on independent data, and was used ultimately to select which model would be selected from those trained. Training and testing data split randomly, with 70% of the data used for training, 30% for testing. No testing data was used for the glass data set because so few examples were available; models were trained and pruned to reduce the risk of overfitting the data.

A third and separate data set, the evaluation data set, was used to report all results shown in this paper. The evaluation data was not used during the model selection process - only to score the individual and combined models so that bias would not be introduced. The glass data was split in such as way as to retain the relative class representation in both the training and evaluation data sets.

Results on the evaluation data are shown below. The number of models is the number of models in the combination. The second column shows the number of combinations created. For example, with 2 models in the combination, there were 15 pairwise combinations represented (neural network and tree, neural network and nearest neighbor, etc.). The next two columns contain the maximum and minimum percent classification error (PCE) on the evaluation data set. Average simple finds the mean PCE for the number of model combinations. The last column shows the standard deviation of PCE for the combinations.

# Models in Combination	# Model Combos	Max Error	Min Error	Average	Std
1	6	37.5%	28.1%	31.5%	3.8%
2	15	34.4%	28.1%	30.5%	2.4%
3	20	29.7%	23.4%	27.0%	2.0%
4	15	29.7%	23.4%	27.1%	1.5%
5	6	28.1%	25.0%	26.8%	1.2%
6	1	25.0%	25.0%	25.0%	0.0%

It is interesting to note that the trend was to decrease the average error, though the largest decrease occurred between 2 and 3 models in the combination, and the error standard deviation also decreases. Interestingly, the best models were among the 3- and 4-way combinations, not the 6-way combination. What combining models provides here is risk reduction, not necessarily the best model performance.

In summary, combining models improved model accuracy on average as the number of models in the combination increased. However, determining which individual models combine best from training results only is difficult and there is no clear trend. Simply selecting the best individual models does not necessarily lead to a decrease in classification error.

References

[1] L. Breiman, "Bagging predictors", Machine Learning 24: 123-140, 1996.

[2] Y. Freund, and R.E. Shapire, "Experiments with a new boosting algorithm", In Proceedings of the Thirteenth International Conference on Machine Learning, July 1996.

[3] Breiman, L. 1996. "Arcing Classifiers", Technical Report, July 1996.

[4] D.H. Wolpert, Stacked generalization, Neural Networks 5:241-259, 1992.

[5] J. Friedman, T. Hastie, and R. Tibsharani, "Additive Logistic Regression: A Statistical View of Boosting", Technical Report, Stanford University, 1998.

[6] Elder, J.F., and Pregibon, D. 1995. A Statistical Perspective on Knowledge Discovery in Databases. Advances in Knowledge Discovery and Data Mining. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Editors. AAAI/MIT Press.

[7] Elder, J.F. IV, D.W. Abbott, Fusing Diverse Algorithms,. 29th Symposium on the Interface, Houston, TX, May 14-17, 1997.

[8] Dietterich, T. 1997. Machine-Learning Research: Four Current Directions. AI Magazine. 18(4): 97-136.

[9] D. Steinberg, CART Users Manual, Salford Systems. 1997.

[10] Abbott, D.W., "Combining Models to Improve Classifier Accuracy and Robustness", To be presented at The Second International Conference on Information Fusion , San Jose, CA, July 6, 1999.

[11] http://www.ics.uci.edu/~mlearn/MLRepository.html

DEAN ABBOTT has been applying data mining algorithms for more than 12 years for diverse areas as missile guidance, underwater signal classification, optical character recognition, automatic target recognition, cervical cancer detection, stock portfolio optimization, direct mail, and credit card fraud detection.

He has been an instructor for data mining courses and tutorials, and has presented papers on the application of data mining techniques at academic and department of defense conferences. He has also evaluated data mining tools and written on their relative strengths and weaknesses. Publications and presentations can be found at http://www.abbottanalytics.com/.