Groves of Trees

Daria Sorokina, Rich Caruana, and Mirek Riedewald

Department of Computer Science, Cornell University, Ithaca, NY, USA
{daria,caruana,mirek}@cs.cornell.edu

Topic: learning algorithms
Preference: oral

We present Groves of trees — a new method for building an additive ensemble of trees. Groves are based on additive models enhanced by techniques of gradient boosting and bagging as well combined with a new algorithm for training additive models. Like bagging (but unlike boosting), the ensemble can use large trees and does not suffer from overfitting. Like gradient boosting, it can be adapted to optimize an arbitrary loss function. Like backfitted additive models, it makes use of an additive structure of the response. As a result, it significantly outperforms other ensembles on regression problems, and on classification problems is consistently one of the top performing methods.

The components of the Groves training algorithm are:

Backfitting. A single Grove of trees is an additive model, where every additive component is a regression tree. We use a variant of backfitting algorithm [1] to train such a model. In the classical backfitting each tree should be trained on the residuals (true response minus the sum of the predictions) of all the models currently in the ensemble. Once all trees in the model are trained on the first round, backfitting discards and retrains trees one at a time until the model converges to a stable state. Retraining the trees helps to detect and fit the actual additive components of the response function and often results in a better fit.

Gradient descent. Gradient descent in the function space was introduced in the gradient boosting algorithm [2]. Gradient boosting is a family of ensemble methods trained as stagewise forward infinite additive models. In this framework, training the next model (usually a regression tree) in the ensemble is a gradient descent step in a function space minimizing a given loss function over the training data. This is achieved by training the tree on the "pseudo-residuals" — values of the gradient on the training set points and recalculating predictions in the leaves of such trees.

We adopt this theoretical framework and replace training on residuals in our backfitted models by training on "pseudo-residuals" defined by the gradient descent; we also add the appropriate recalculation of predictions in the leaves.

Different loss functions result in different algorithms. Minimizing least squares loss results in a regression algorithm with continuous predictions, while minimizing negative binomial log-likelihood results in a binary classification algorithm trained to predict probabilities of positive or negative events.

Bagging. As with single decision trees, a single Grove tends to overfit to the training set when the trees are large. Such models show large variance and benefit significantly from bagging, the well-known procedure for improving model performance by reducing variance [3]. On each iteration of bagging, we draw a bootstrap sample from the training set, and train the full model (in our case a Grove) from that sample. After repeating this procedure a number of times, we end up with an ensemble of models. The final prediction of the ensemble on each test data point is an average of the predictions of all models.

Dynamic programming training. We have developed an outer loop extension to the iterative backfitting training of a Grove. To ensure that more complex models perform at least as well as simpler models, training a Grove begins with training a single small tree even if the final Grove will consist of several large trees. Then the tree size and number of trees are gradually increased in stages. Which type of increase happens in each particular stage is determined by testing both possibilities on the out-of-bag data and greedily choosing whichever is better. Training on pseudo-residuals and backfitting are still performed at the core of the algorithm regardless of this outer loop that changes the size and number of trees in the Grove.

Experiments. We have compared the classification version of the algorithm with other methods that were studied in a recent extensive comparison of classification algorithms. Our results show that on average Groves outperforms the other learning methods from that study.

We have also compared regression version of the Groves with gradient boosting and bagged trees on a number of regression data sets. The results showed that regression Groves outperformed these other methods on every data set. The improvement over other algorithms was especially significant on highly non-linear and not very noisy data.

References

1. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
2. Friedman, J.: Greedy Function Approximation: a Gradient Boosting Machine. Annals of Statistics **29** (2001) 1189 – 1232
3. Breiman, L.: Bagging Predictors. Machine Learning **24** (1996) 123–140