Tree Model Notes (WIP)

Ryan Gehring

Contents

Theory

Historical Context

Tree models came out in the 90's and were part of the nonparametric supervised learning algorithm boom in the west. Definitely from the machine learning family of algorithms, tree models in particular seem to be based on lots of good ideas of things to do, but lack formal justification for many of the techniques. As a result, the asymptotics were worked out much later. As early nonparametric models, most of the magic of tree models is just taking averages over subsets of the data and examining how good the prediction got—mathematics are sort of an afterthought. I will accordingly keep these notes brief and just describe a few classes of tree models and summarize the techniques used.

Classification and Regression Trees

CART models see a lot of a application for simple business analysis, as the produced regression tree is usually human-interpretable, tweakable, and interesting. These models are binary trees, with leaves representing model prediction (usually an average of the response variable in the bucket) and nodes representing *splitting criteria*, usually a less-than-or-equal condition on a predictive variable. The model is scored for any particular example by iterating through the tree until a leaf is reached, following the appropriate splitting criteria—taking a left child if true, else taking the right.

Basic CART models are trained by greedy optimization. New splits are generated based on the segregation of the data which reduces SSE prediction error the most across the dataset. The algorithm continues until prediction error falls below some threshhold or the number of splits reaches a cap. Extensions to CART allow for fitting more complex models at the leaves, trying less greedy fitting approaches such as 'Texas two-step" which examines splits two levels deep, or trying bootstrapping or other sampling techniques to be discussed in the next section.

Random Forests

Random forests see a lot of use in industry as quick and dirty models. They are nonparametric, so you can be much less rigorous about feature selection, and out of the box let you know about what you can expect to get in terms of prediction accuracy with your data source. They are sometimes criticized for being 'black boxy,' and somewhat slow to score, making them less popular for production models, but serve quite a useful purpose in R&D and business analysis.

Random Forest models are ensembles of tree models trained on several samples from the original dataset, but sampled with replacement, a sampling technique known as bootstrapping. The idea has biological motivation - the idea being that sampling with replacement results in about one third of each new dataset being a duplicate of a random row enforces some variability and regularization into the datasets, preventing overfitting sort of like mutations in genetics leading to genetic diversity. The trained tree models' predictions are averaged together to give the final model prediction, an ensemble technique known as bagging. The models are trained just like CART models, except that at splits only a random subset of variables is allowed to be considered, rather than finding the greedy optimum. Improvements could include the same extensions to CART, as well as different model-voting schemes.

Stochastic Gradient Boosting

This tree model greedily minimizes a loss function via weighted sum of several small "base learner" models, often CART models of restricted depth. Models are trained incrementally on the residuals of the partially trained, previous components of the models. IE, I fit base learner one, score the model, feed the residuals in as new Y values to a second base learner, repeat. After each base learner is trained, the weight on the model in the global sum is greedily optimized. Similar to random forests in their pros and cons— another

Theory continued on next page. . .

black boxy nonparametric model that achieves great prediction accuracy compared to most algorithms right out of the gate.