Introduction to Machine Learning

Session 2d: Boosting

Reto Wüest

Department of Political Science and International Relations
University of Geneva

Outline

Boosting

Boosting

- Like bagging, boosting is a general approach that can be applied to many machine learning methods for regression or classification.
- Recall that bagging creates multiple bootstrap training sets from the original training set, fits a separate tree to each bootstrap training set, and then combines all trees to create a single prediction.
- This means that each tree is built on a bootstrap sample, independent of the other trees.

Boosting

- In boosting, the trees are grown sequentially: each tree is grown using information from previously grown trees.
- Boosting does not involve bootstrap sampling. Instead, each tree is fit on a modified version of the original data set.

Boosting

Algorithm: Boosting for Regression Trees

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{1}$$
   (c) Update the residuals
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{2}$$
3. Output the boosted model
   $$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{3}$$

What Is the Idea Behind boosting?

- Unlike fitting a single large decision tree, which potentially overfits the data, boosting learns slowly.
- Given the current model, we fit a new decision tree to the residuals from that model (rather than the outcome Y).
- We then add the new decision tree into the fitted function in order to update the residuals.

What Is the Idea Behind boosting?

- Each of the trees can be rather small, with just a few terminal nodes, determined by parameter d.
- Fitting small trees to the residuals means that we slowly improve $\hat{f}$ in areas where it does not perform well.
- The shrinkage parameter λ slows the process even further, allowing more and different shaped trees to attack the residuals.

Tuning Parameters for Boosting

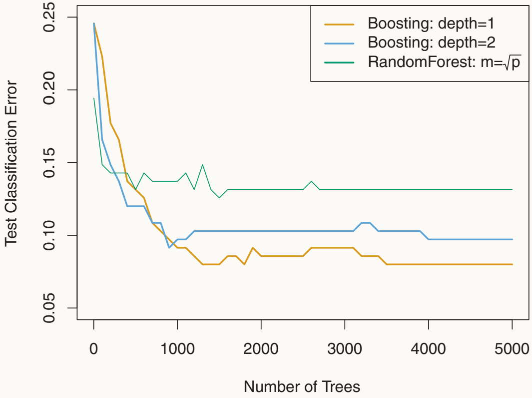
1. Number of trees B
   - Unlike bagging and random forests, boosting can overfit if B is too large.
   - Use CV to select B.
2. Shrinkage parameter λ
   - Controls the rate at which boosting learns.
   - A small positive number, typical values are 0.01 or 0.001.
   - Very small λ can require a very large value of B in order to achieve good performance.

Tuning Parameters for Boosting

3. Number of splits in each tree d
   - Controls the complexity of the boosted ensemble.
   - It is the interaction depth, since d splits can involve at most d variables.
   - Often $d = 1$ works well, in which case each tree is a stump (consisting of a single split).

Example: Gene Expression Data

Boosting and Random Forests Results for the Gene Expression Data



(Source: James et al. 2013, 324)

For the two boosted models, $\lambda = 0.01$. Note that the test error rate for a single tree is 24%.