Boosting

Instructor: Haoran LEI

Hunan University

Boosting

Like bagging, *boosting* is a general approach that can be applied to many statistical learning methods for regression or classification.

We only discuss boosting for decision trees.

Boosting

- Recall that bagging involves creating multiple copies of the original training data set using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model.
- Notably, each tree is built on a bootstrapped dataset, independent of the other trees.

Boosting works in a similar way, except that the trees are grown *sequentially:* each tree is grown using information from previously grown trees.

Boosting algorithm for regression trees

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

Boosting algorithm for regression trees (cont.)

3. Output the boosted model:

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

Unlike fitting a single large decision tree to the data, which amounts to *fitting the data hard* and *potentially overfitting*, the boosting approach instead **learns slowly**.

What is the idea behind this procedure?

- Given the current model, we fit a decision tree to the residuals from the model. We then add this new decision tree into the fitted function in order to update the residuals.
- Each of these trees can be rather small, with just a few terminal nodes, determined by the parameter d in the algorithm.
- By fitting small trees to the residuals, we slowly improve $\hat{f}$ in areas where it does not perform well. The shrinkage parameter λ slows the process down even further, allowing more and different shaped trees to **attack the residuals**.

Tuning parameters for boosting

1. **The number of trees B.**

   Unlike bagging and random forests, boosting can overfit if B is too large, although this overfitting tends to occur slowly. We use cross-validation to select B.

2. **The shrinkage parameter λ.**

   This is a small positive number, and it controls the rate at which boosting learns. Typical values are 0.01 or 0.001, and the right choice can depend on the problem. Very small λ can require using a very large value of B in order to achieve good performance.

Tuning parameters for boosting

3. **The number of splits d in each tree.**
   - It controls the complexity of the boosted ensemble.
   - Often $d = 1$ works well, in which case each tree is a stump, consisting of a single split and resulting in an additive model.
   - More generally d is the interaction depth, and controls the interaction order of the boosted model, since d splits can involve at most d variables.

Summary of trees

- Decision trees are simple and interpretable models for regression and classification.
- Bagging, random forests and boosting are good methods for improving the prediction accuracy of (single) trees. They work by growing many trees on the training data and then combining the predictions of the resulting ensemble of trees.
- The latter two methods— random forests and boosting— are among the state-of-the-art methods for supervised learning. However their results can be difficult to interpret.