CENTRALE LYON

STATISTICAL LEARNING

COURSE 5 - BAGGING-RANDOM FOREST AND BOSSTING

ECOLE CENTRALE DE LYON - BACHELOR 2ND YEAR
2024-2025

Trees are are not very robust. A small change in data can cause a large change in the final estimates tree

So trees have often low-bias but hight variance.

Aim : Propose procedure to reduce the variance. Bagging, Random forest and boosting are among the most popular methodologies to do this.

Bagging

- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a statistical learning method; we introduce it here because it is particularly useful and frequently used in the context of decision trees.
- Recall that given a set of n independent observations $Z_1; \cdots; Z_n$, each with variance σ^2, the variance of the mean $\bar{Z}$ of the observations is given by σ^2/n.
- Averaging a set of observations reduces variance. The problem : in practice we do not have access to multiple training sets.
-

Bagging -Regression Trees

- Use bootstrap, by taking repeated samples from the (single) training data set.
- generate B different bootstrapped training data sets.
- train the model (tree) on the bth bootstrapped training set in order to get $\hat{f}_b(x)$, the prediction at a point x.
- average all the predictions to obtain

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x)$$

This is called bagging.

Bagging -Classification Trees

In a classical tree the prediction at x is given by / If x belongs to the region R_m

$$\hat{G}(x) = argmax_k \#\{x_i = k, x_i \in R_m\}$$

- For classification trees with K classes :

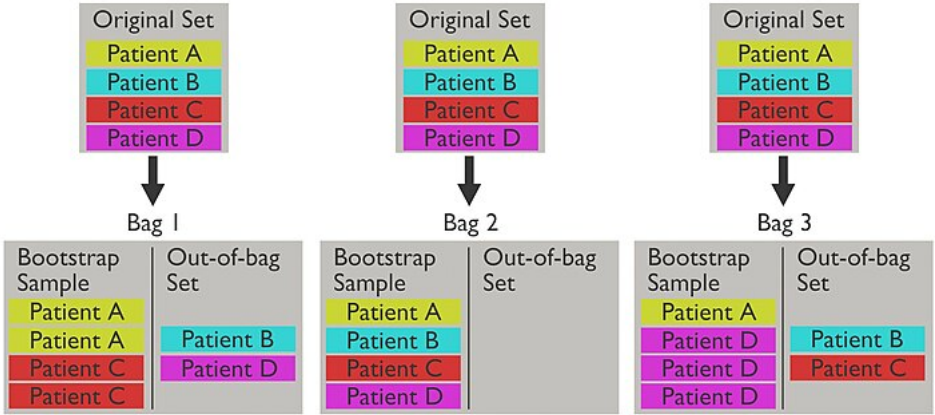
$$\hat{f}_{bag}(x) = (p_1(x), \cdots, p_K(x)$$

  where $p_k(x)$ is equal to the proportion of trees predicting class k at x.
- the bagger classifier is $\hat{G}_{bag}(x) = argmax_k \hat{f}_{bag}(x)$

Assessment of the model

To asses the model in bagging we can use two methods :

1. the test data and predict using the bagging predictor and compute the error (MSE in regression or classification error)
2. When bootstrap aggregating is performed, two independent sets are created. One set, the bootstrap sample, is the data chosen to be "in-the-bag" by sampling with replacement. The out-of-bag set is all data not chosen in the sampling process.


Original Set
Patient A
Patient B
Patient C
Patient D
Bag 1
Bootstrap Sample
Patient A
Patient A
Patient C
Patient C
Out-of-bag Set
Patient B
Patient D
Original Set
Patient A
Patient B
Patient C
Patient D
Bag 2
Bootstrap Sample
Patient A
Patient B
Patient C
Patient D
Out-of-bag Set
Original Set
Patient A
Patient B
Patient C
Patient D
Bag 3
Bootstrap Sample
Patient A
Patient D
Patient D
Patient D
Out-of-bag Set
Patient B
Patient C

Assessment of the model-II

Predict the response for the ith observation using each of the trees in which that observation was OOB. This will yield around $B/3$ predictions for the ith observation, which we average.

Random forest

- A drawback of the bagging procedure is that the different trees are correlated.
- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. This reduces the variance when we average the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.

Random forest Algorithm

1. For $b = 1$ to B
   - 1.1 Draw a bootstrap sample of size n for the training data
   - 1.2 Grow a random forest tree T_b by repeating the following steps for each terminal node until the minimum node size is reached
     - 1.2.1 Select m variables at random from the p features
     - 1.2.2 Pick the best variable/split-point among m
     - 1.2.3 Spilt the node in two daughter nodes
2. Output the ensemble of trees $T_{b\,b=1}^{B}$

Make prediction at x : Regession : $\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$

Classification : For classification trees with K classes :

$$\hat{f}_{rf}(x) = (p_1(x), \cdots, p_K(x)$$

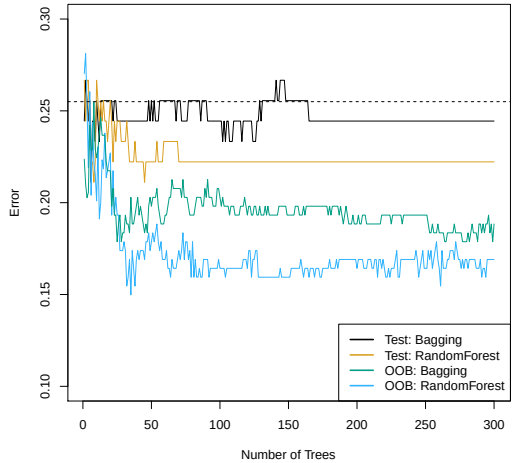
where $p_k(x)$ is equal to the proportion of trees predicting class k at x.

$$\hat{G}_{rf}(x) = argmax_k \hat{f}_{rf}(x)$$

Choice of $m \approx \sqrt{(p)}$ that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors

Example the heart data

Heart Data contains a binary outcome HD for 303 patients. There are 13 predictors (quantitative and qualitative)



Influence of *m*

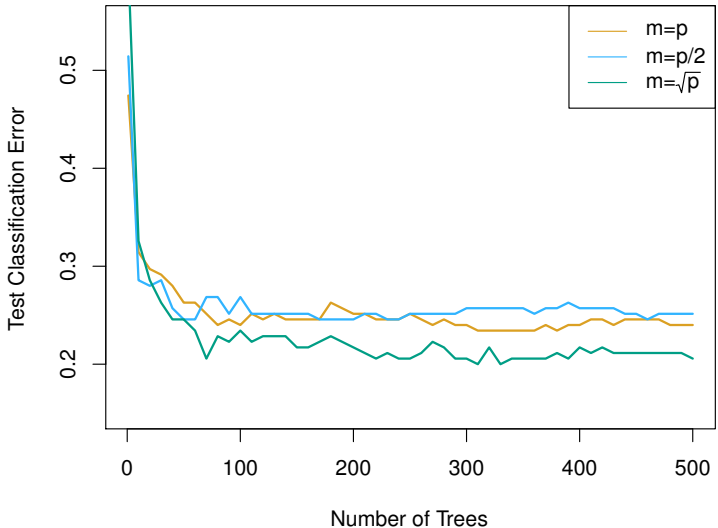


Figure: from An introduction to statistical learning Chap 8

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.
- 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 bootstrap data set, 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

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

$$\hat{f}(x) = \hat{f}(x) + \lambda \hat{f}_b(x)$$

   - 2.3 Update the residuals,

$$r_i = r_i - \lambda \hat{f}_b(xi)$$

3. Output the boosted model,

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

Tuning parameters for boosting

1. The number of trees B. Unlike bagging and random forests, boosting can overfi

   t if B is too large, although this overfitting tends to occur slowly if at all. We use cross-validation to select B.

2. The shrinkage parameter λ, a small positive number. This 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.

3. The number of splits d in each tree, which 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.

Example Gene expression data

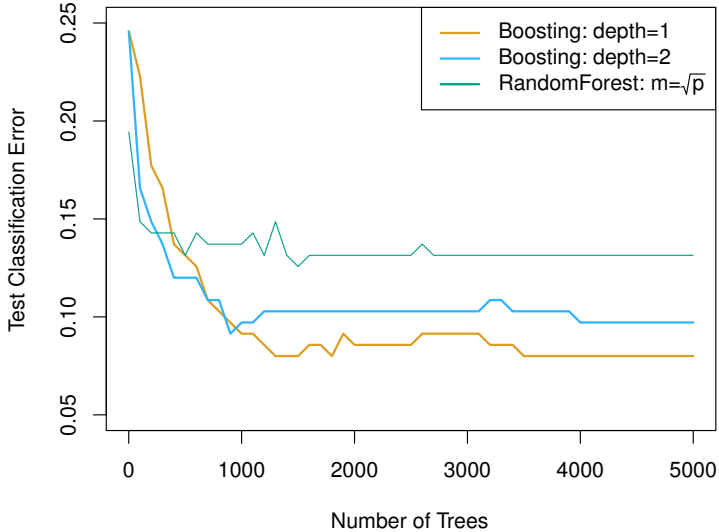


Figure: from An introduction to statistical learning Chap 8