Lecture 17: Bagging and Random Forest

Statistical Learning and Data Mining

Xingye Qiao
Department of Mathematical Sciences
Binghamton University

E-mail: qiao@math.binghamton.edu
Read: ELSII Chs. 8.7 & 15, and ISLR 8.2

Outline

The next section would be ......

Bagging

- Bootstrap = sampling with replacement.
- In addition to statistical inference, bootstrapped samples can also be utilized to improve existing classification or regression methods.
- Bagging = Bootstrap aggregating.
- Proposed by Leo Breiman in 1994 to improve the classification by combining classification results of many randomly generated training sets

Bagging classifier

- Suppose a method provides a prediction $\widehat{f}(\boldsymbol{x})$ at input $\boldsymbol{x}$ based on sample $\boldsymbol{X}$
- The bagging prediction is

$$\widehat{f}_B(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}_b(\boldsymbol{x})$$

where $\widehat{f}_b$ is the prediction based on a bootstrapped sample $\boldsymbol{X}_b^*$

- If we view $\widehat{f}(\boldsymbol{x})$ as an estimator, then we would be interested in $\mathrm{E}_F(\widehat{f}(\boldsymbol{x}))$, which is a function of the distribution F
- In this case, the bagging classifier $\widehat{f}_B(\boldsymbol{x})$ is nothing but an approximation to the ideal bootstrap estimate $\mathrm{E}_{F_n}(\widehat{f}^*(\boldsymbol{x}))$ where F_n denotes the empirical distribution.

Why Bagging Works?

$$\widehat{f}_B(\boldsymbol{x}) \approx E_{F_n}[\widehat{f}(\boldsymbol{x})] \approx E_F[\widehat{f}(\boldsymbol{x})]$$

- It is well known that

$$MSE = Variance + Bias^2$$

  In the current setting, this is

$$E_F[\widehat{f}(\boldsymbol{x}) - y]^2 = E_F[\widehat{f}(\boldsymbol{x}) - \mathrm{E}_F(\widehat{f}(\boldsymbol{x}))]^2 + (\mathrm{E}_F(\widehat{f}(\boldsymbol{x})) - y)^2$$

- By using the bagging estimate $\widehat{f}_B(\boldsymbol{x})$ in lieu of $\widehat{f}(\boldsymbol{x})$, we hope to make the variance part $E_F[\widehat{f}(\boldsymbol{x}) - \mathrm{E}_F(\widehat{f}(\boldsymbol{x}))]^2$ almost zero.
- Hence bagging improves MSE by reducing the variance.

Bagging for Classification

Initially invented to solve classification problem. Two approaches:

- If the basic classifier itself is a plug-in classifier, which works by first estimating $\eta_j(\boldsymbol{x})$ by $\widehat{\eta}_j(\boldsymbol{x})$ and then use classification rule

$$\widehat{\phi}(\boldsymbol{x}) := \operatorname*{argmax}_{j=1,\ldots,K} \widehat{\eta}_j(\boldsymbol{x}),$$

  then we can average the bootstrap estimates $\eta_j(\boldsymbol{x})$ to obtain

$$\widehat{\eta}_{j,B}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} \widehat{\eta}_{j,b}(\boldsymbol{x})$$

  then use

$$\widehat{\phi}_B(\boldsymbol{x}) = \operatorname*{argmax}_{j} \widehat{\eta}_{j,B}(\boldsymbol{x})$$

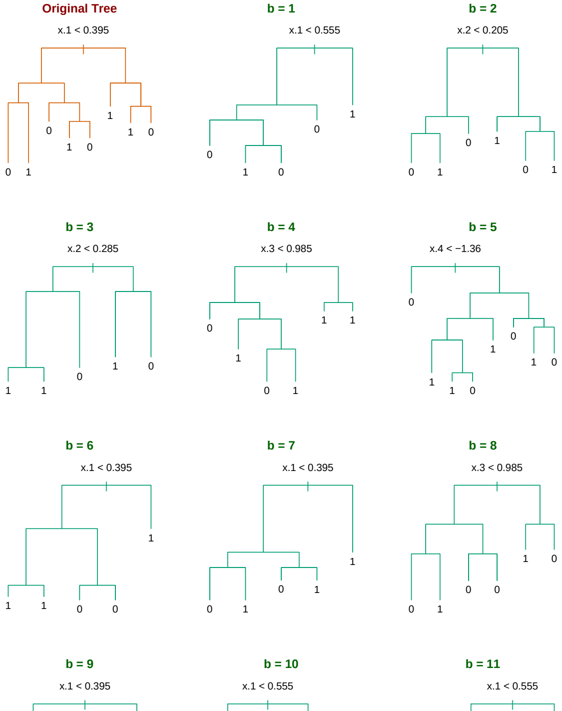
- However, except for a few classifiers such as kNN, most approaches do not work like that. Often, a classifier will report a prediction for $\boldsymbol{x}$ without reporting the estimate for $\eta(\boldsymbol{x})$. For example, SVM, CART, etc.
- In this case, use a majority voting scheme: Let

$$V_{j,B}(\boldsymbol{x}) := \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}_{\{\widehat{\phi}_b(\boldsymbol{x})=j\}}$$

  then we just use

$$\widehat{\phi}_B(\boldsymbol{x}) := \underset{j}{\operatorname{argmax}}\, V_{j,B}(\boldsymbol{x})$$

- In other words, we let the B classifiers to cast votes and choose the class with the most votes.


Original Tree
x.1 < 0.395
b = 1
x.1 < 0.555
b = 2
x.2 < 0.205
b = 3
x.2 < 0.285
b = 4
x.3 < 0.985
b = 5
x.4 < −1.36
b = 6
x.1 < 0.395
b = 7
x.1 < 0.395
b = 8
x.3 < 0.985
b = 9
x.1 < 0.395
b = 10
x.1 < 0.555
b = 11
x.1 < 0.555

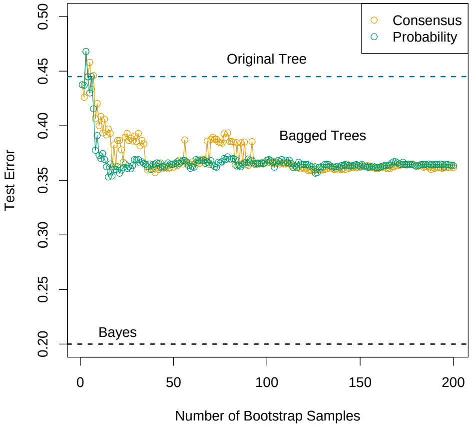


FIGURE 8.10. *Error curves for the bagging example of Figure 8.9. Shown is the test error of the original tree and bagged trees as a function of the number of bootstrap samples. The orange points correspond to the consensus vote, while the green points average the probabilities.*

- Unfortunately, the MSE argument of "how bagging works" does not apply for classification problem.
- MSE uses squared loss. Classification uses 0-1 loss where the decomposition to variance and bias^2 does not exist.
- "Wisdom of the crowd" argument:

> In binary classification, if there are B independent voters and each would classify $Y = 1$ correctly with probability $1 - e$ and the misclassification rate $e < 0.5$, then $Z = \sum_{b=1}^{B} \widehat{\phi}_b(\boldsymbol{x}) \sim Binomial(B, 1 - e)$ and $\frac{1}{B}\sum_{b=1}^{B} \widehat{\phi}_b(\boldsymbol{x}) \xrightarrow{p} 1 - e > 0.5$. Hence, the misclassification rate of the bagged classifier is $P(\frac{1}{B}\sum_{b=1}^{B} \widehat{\phi}_b(\boldsymbol{x}) < 0.5) \to 0$.

A catch: the bootstrap classifiers are not independent at all.

Random forest, however, adds in additional randomness to the bootstrap classifiers which makes them less independent.

What bagging cannot do?

- Bagging does not improve the estimator if it is only a linear function of the data (WHY??). It would only have some effect if the estimator is of a nonlinear nature of the data.
- Bagging does not improve an estimator/classifier if it is already very stable; may even worsen it.
- Bagging does not improve a classifier if it is a bad classifier (generalization error > 0.5 in binary classification case; worse than random guessing)
- Bagging in some sense expands the basis of the model space. But examples show that it has not done enough. In contrast, boosting can do a good job in terms of expending model space.

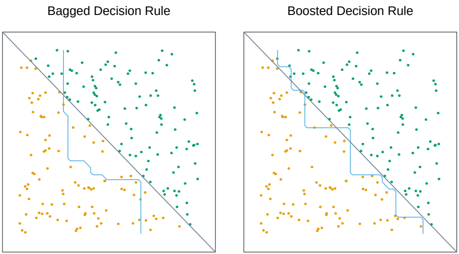


FIGURE 8.12. *Data with two features and two classes, separated by a linear boundary. (Left panel:) Decision boundary estimated from bagging the decision rule from a single split, axis-oriented classifier. (Right panel:) Decision boundary from boosting the decision rule of the same classifier. The test error rates are* 0.166, *and* 0.065, *respectively. Boosting is described in Chapter 10.*

Each basic classifier is just a stump (tree with only one split). The basic classifier is too weak and the model space needs to be expended much aggressively. Next section: introduce Random Forest.

The next section would be . . . . . .

Bagging in a different view

- In Bagging, we generate bootstrap sample and re-do classification, and aggregate the predictions by majority voting.
- The bootstrap classifiers are not independent and the average effect to reduce variance may not be clear.
- Random Forest: an application of bagging to aggregate bootstrap classification trees, but with some additional randomness due to random subset of variables in tree growing.
- Since trees are known to be noisy (large variance), bagging should bring a lot of benefit to trees.
- But these trees are identically distributed but not independent.

Random Forest

1. For $b = 1$ to B
   a. Draw a bootstrap sample of size N from $\boldsymbol{X}$, namely $\boldsymbol{X}_b^*$
   b. Grow a tree T_b based on $\boldsymbol{X}_b^*$ in the common way, except that at each node, find the best split among m random selected subset of all p variables.
   c. Stop growing when a node has reached to a pre-set minimal node size.
2. Report the forest: $\{T_b\}_1^B$

Regression: $\widehat{f}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} T_b(\boldsymbol{x})$
Classification: $\widehat{\phi}(\boldsymbol{x}) = \text{majority vote}\{T_b(\boldsymbol{x})\}$

Note that the set of m random selected variables may be different at different nodes, even within the same tree.

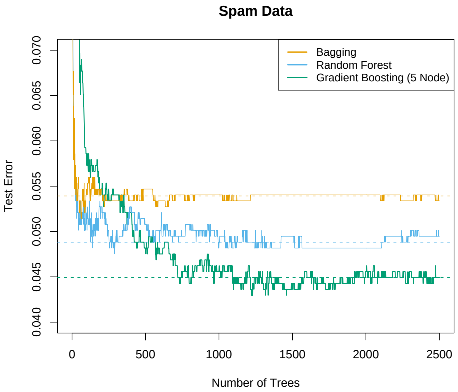


FIGURE 15.1. *Bagging, random forest, and gradient boosting, applied to the spam data. For boosting,* 5*-node trees were used, and the number of trees were chosen by* 10*-fold cross-validation (*2500 *trees). Each "step" in the figure corresponds to a change in a single misclassification (in a test set of* 1536*).*

Out-of-Bag Sample

- For each observation, there must be a set of trees whose bootstrap samples do not include it.
- For each bootstrap, there must be a set of observations not selected in the bootstrap sample. – This set of observations is called the Out-of-Bag Sample

OOB Errors

OOB for the whole forest:

- The OOB Error is defined to be the average of misclassification over all the n observations, where each observation is classified not by the whole forest, but by the sub-forest which includes those trees whose bootstrap samples do not include this observation.
- Similar to leave-one-out cross validation.
  - 2 observations do not share classifier, unlike in 5-fold cv.
  - But, the classifier for each observation is also a random forest
- Unlike the cross validation, there is no additional computational burden. The OOB error is obtained along the way of generating the random forest.

We can also calculate the OOB for a single (bootstrap) tree:

- The misclassification rate of the (bootstrap) tree when it is applied to a OOB sample.

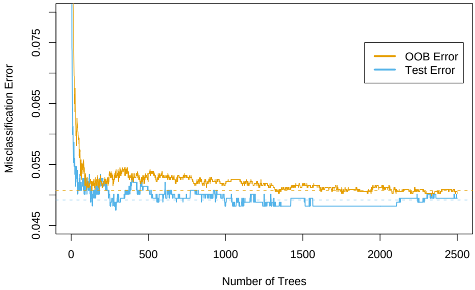


FIGURE 15.4. OOB *error computed on the* `spam` *training data, compared to the test error computed on the test set.*

Variance Importance

- Goal: provide a ranked list of variables representing their relative importance in classification.
- Two versions
  - Permutation based
  - Gini index (or impurity) based

Permutation based variance importance

1. The OOB error for the bth tree in the forest is $E_b(OOB)$
2. For each $j = 1, \dots, p$, the jth variable in the OOB sample can be randomized (randomly exchange the values on the jth variable). Call this permuted OOB sample OOB_j. A new OOB error for OOB_j can be calculated for the bth tree, denoted as $E_b(OOB_j)$
3. The difference is aggregated over all B trees for each j:

$$VI(j) := \sum_{b=1}^{B} [E_b(OOB_j) - E_b(OOB)]$$

The large VI, the more important the variable is.

- The permutation on variable j works by voiding the effect of variable j
- Note that it is a little different from removing the variable j from the data and regrowing the tree again!
- Note that the tree is static; we only try a different testing data. Hence there is not much computational burden.
- If a variable j is important, then $E_b(OOB_j) \gg E_b(OOB)$. Otherwise, $E_b(OOB_j) = E_b(OOB)$.

Gini index (or impurity) based variance importance

- Recall that when growing a tree, the split at a variable makes the Gini index on the mother node reduce to the weighted sum of the Gini indices on the two daughter nodes.
- We aggregate such reduction for each variable j over all the B trees
- Large VI = large reduction overall = helped to make many trees reduce impurity = being important.
- It is possible that a variable j does not appear in a single tree; but over the whole forest with many trees, the chance that it is not selected by any tree is very small.

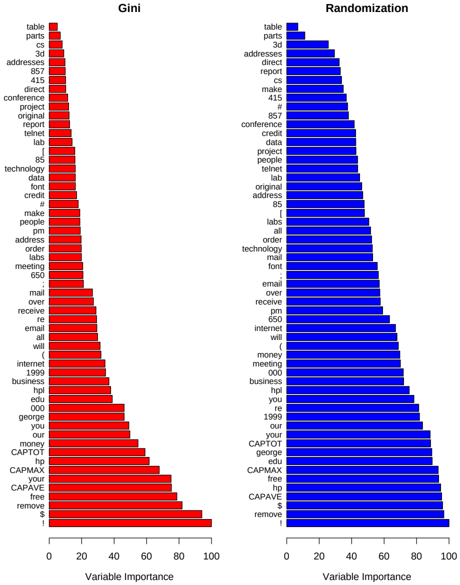


FIGURE 15.5. *Variable importance plots for a classi-*

Remarks

- Choice of m: classification – $\sqrt{p}$; regression – $p/3$
- Minimal node size: classification – 1; regression – 5
- Random forest is an improved version of bagging trees.
- Friedman and Hall showed that subsampling (without replacement) with $N/2$ is approximately equivalent to bootstrap, while smaller fraction of N may reduce the variance even further.
- In R, package `randomForest`.

Random forests cannot overfit?

- Increasing B does not cause the random forest sequence to overfit
- But the limit of the sequence ($B \to \infty$) can overfit the data
- Some people reported small gains in performance by controlling the depths of the individual trees grown in random forests. However, using full-grown trees seldom costs much, and results in one less tuning parameter.

Random forest and kNN

The random forest classifier has much in common with a weighted version of k-nearest neighbor classifier

- The tree-growing algorithm finds an ‘optimal’ path to that observation, choosing the most informative predictors from those at its disposal.
- The averaging process assigns weights to these training responses, which ultimately vote for the prediction.
- Via RF, those observations close to the target point get assigned weights, which form the classification decision