Bagging and Random Forests

CSci 5525: Machine Learning

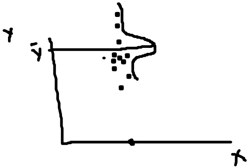
Instructor: Nicholas Johnson

November 17, 2020

Announcements

- Project progress report due today (11:59 PM CST)
- Exam 2 coming up $<$ 1 week (Monday Nov 23, due 48 hours later)
  - Covers lectures 11 (Deep Learning I) - 21 (tentatively PCA)

Bias-Variance Revisited



$y|x$

Expected test error:

$$E[(f_D(x) - y)^2] \rightarrow 0$$

$$= \underbrace{E_D\left[(f_D(x) - \bar{f}(x))^2\right]}_{variance} + \underbrace{E\left[(\bar{y}(x) - y)^2\right]}_{\text{noise}} + \underbrace{E\left[\left(\bar{f}(x) - \bar{y}(x)\right)^2\right]}_{\text{bias}^2}$$

We introduce a method for reducing the variance term

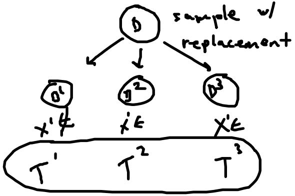
$$f_D(x) \rightarrow \bar{f}(x) \qquad E\left[f_{D^1}, f_{D^2}, f_{D^3}\right] \approx \bar{f}(x)$$

Bagging

$D \sim P^m$

$|D^i| = |D|$

- Bootstrap Sampling
  - Consider training set D with m training examples
  - Create D^i by drawing m examples with replacement
  - In expectation, D^i will leave out a fraction of examples
  - Each D^i will approximate the distribution underlying D
- Bagging = Bootstrap Aggregating
- Overview of algorithm is as follows:
  - Create k bootstrap samples $D^1, \ldots, D^k$ k^{th} dataset
  - Train a distinct classifier on each D^i
  - Classify new instance by majority voting



Bagging for Regression

- In regression, y is real valued
  - For a random dataset D, get regressor $f_D(x)$
  - Bagging uses expectation for regression
  - The aggregate prediction $f_A(x) = E_D[f_D(x)]$
- Expected error of individual regressors is greater than the error of the aggregate prediction

$$E_D[(y - f_D(x))^2] \geq (y - \underbrace{f_A(x)}_{\text{ensemble model}})^2$$

  - Jensen's inequality: $E[\phi(X)] \geq \phi(E[X])$
  - Similar argument can be constructed for classification
- Bagging works well with unstable predictors
  - Depends on how much $f_D(x)$ changes with D
  - Bagging lowers variance for highly volatile predictors
  - Smoother response from aggregate
  - Example: Decision trees

Bagging for Classification

- Majority vote as a way to combine classifiers
- Do vote proportions estimate the posterior class probabilities?
  - Answer: No
  - Let $P(C_1|x) = 0.75$ and each classifier predicts class 1. Then by voting proportions $\hat{P}(C_1|x) = 1.0$ which is different.
- Classifiers may estimate class probabilities
  - Bagging using majority vote
  - Bagging by averaging the class probabilities

Bagging Results I

- Simulated data: Sample size of $N = 30$, 2 classes, 5 features
- Samples from a Gaussian with pairwise correlation 0.95
- Response generated according to

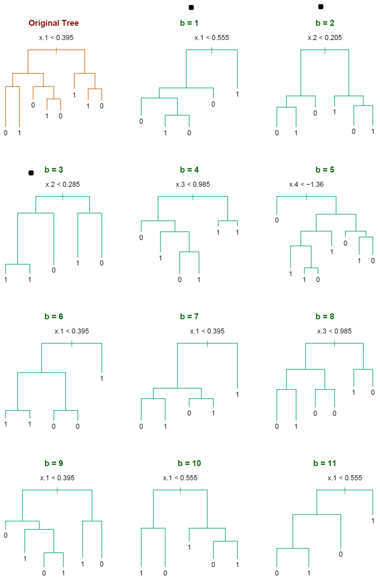
$$P(Y = 1|x_1 \leq 0.5) = 0.2 \,, \qquad P(Y = 1|x_1 > 0.5) = 0.8$$

$P(Y=0 \mid x_1 \leq 0.5) = 0.8$ $\qquad$ $P(Y=0 \mid x_1 > 0.5) = 0.2$

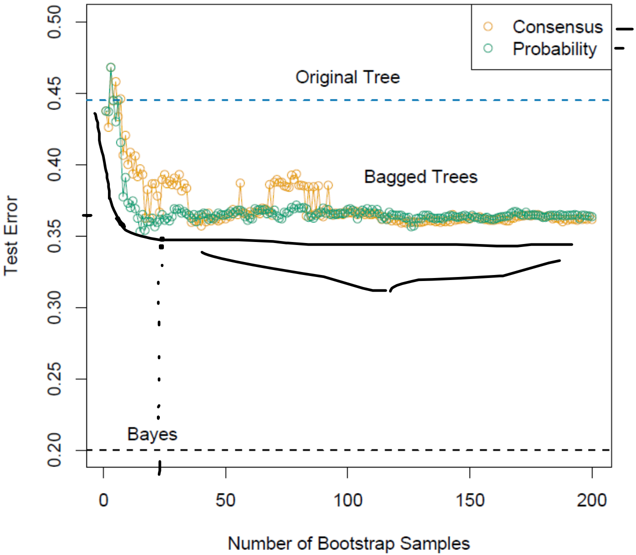
- Two classifiers
  - Decision tree for original training data
  - Bagged tree with 200 bootstrap samples

datasets

Bagging Results I

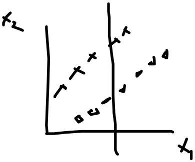


Bagging Results I

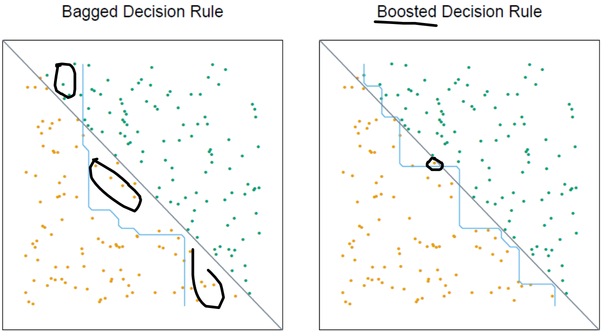


Bagging Results II

- Simulated data: Sample size of $N = 100$, 2 classes, 2 features
- Each classifier is a decision rule or stump
  - Single axis-oriented split along x_1 or x_2
  - Pick the one which gives lower training error
- Bagging a weak classifier can make it worse
- Two approaches
  - Bagging decision rule
  - Boosted decision rule



Bagging results II



Random Forests (RF)

dataset

- Builds a forest of decision trees
- Create k bootstrap samples $D^1, \ldots, D^k$
- Learn an un-pruned decision tree on each sample
- Learning: At each internal node
  - Randomly select $m < d$ features
  - Determine the best split using only these features
- Prediction: Use output from all trees in the forest
  - Classification: Majority vote
  - Regression: Average of responses

Random Forest: Regression or Classification

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Random Forests: Properties

- Parameter: Size m of the feature subset
  - RFs are not too sensitive to the value of m
  - Common choice for classification: $m = \sqrt{d}$ → # original feature
  - Common choice for regression: $m = \frac{d}{3}$
- Randomness of feature selection reduces variance
- Reduced feature set leads to faster algorithms
- Bagging of decision trees is a special case: $m = d$

Variance Reduction: "Decorrelated" Trees

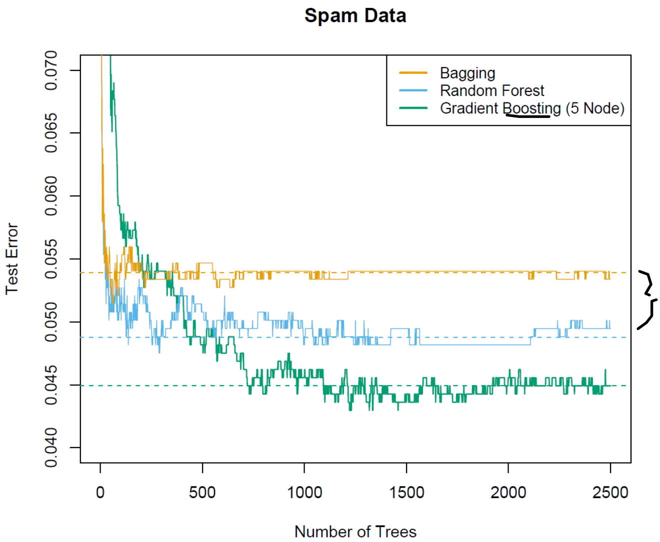
- Warm-up: k i.i.d. random variables z_i, variance σ^2
  - Variance of $\frac{1}{k}\sum_{i=1}^{k} z_i$ is $\frac{1}{k}\sigma^2$
- Random forest trees: identically distributed, not independent
  - If (z_i, z_j) have positive pairwise correlation ρ, variance is

$$\rho\sigma^2 + \frac{1-\rho}{k}\sigma^2$$

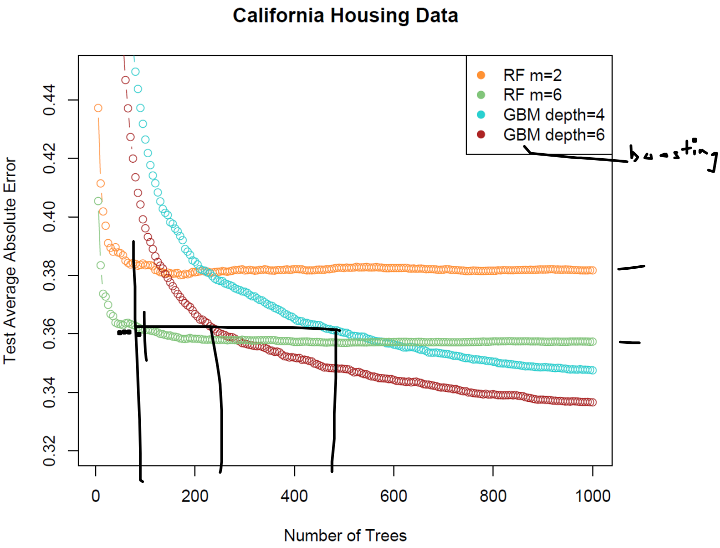
$m < d$

- Random forest variance reduction
  - As k (number of trees) increase, second term decreases
  - As m (number of random features) decrease, ρ decreases
  - "Decorrelated" trees help in reducing variance

Results: Spam Data



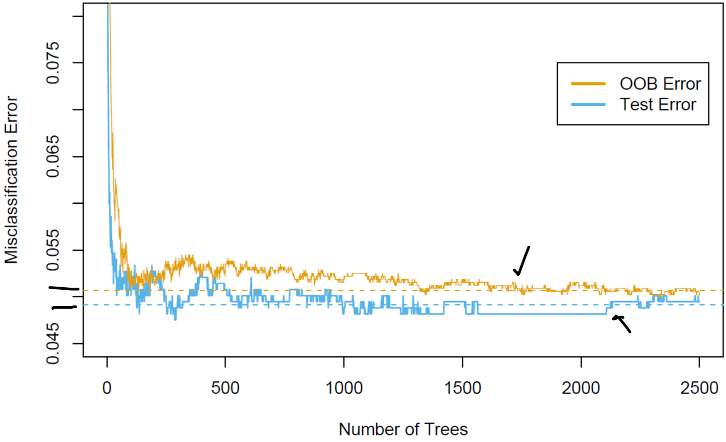
Results: Housing Data



Out-of-Bag (OOB) Samples

- Bagging provides out-of-bag error
    - Unbiased estimate of test error
- Each $(\mathbf{x}_i, y_i)$ not in dataset D_j can be viewed as test data
- Prediction on point $(\mathbf{x}_i, y_i)$
    - Consider trees which did not contain $(\mathbf{x}_i, y_i)$
    - Regression: Average their prediction on $(\mathbf{x}_i, y_i)$
    - Classification: Take majority vote on $(\mathbf{x}_i, y_i)$
- OOB error on the entire training set
    - Alternative to 10-fold cross-validation
    - Can be used to decide termination
    - Bagging provides estimate of test error without using test set

OOB Samples: Error Estimate



Random Forests: Pros and Cons

- Advantages
  - Low variance due to ensemble of decorrelated trees
  - Natural and fast to parallelize
  - No pruning required in order to generalize well
  - Better prediction accuracy than single decision trees
  - About the same accuracy as SVMs, LR, NN, etc.
  - Have few parameters to tweak
  - Cross validation is unnecessary
- Disadvantages
  - Loss of interpretability

Boosting and Random Forests

- Both are powerful methods with high accuracy
- Both are widely used in practice
- Boosting
  - Grows the model sequentially
  - Base classifier can be anything
  - Each base classifier is a weak learner
  - Weighted combination
  - Additive model, gradient boosting perspective
- Random Forests
  - Grow trees in parallel
  - Base models are trees
  - Some of the trees can be bad
  - Equal weights on all trees
  - Model averaging perspective