Ensemble Learning (2): Bagging and Random Forest

Yanrong Yang

RSFAS/CBE, Australian National University

18th October 2022

Contents of this week

Ensemble Learning methods

- Bagging (bootstrap aggregation)
  - a technique for reducing the variance of an estimated prediction function.
  - work especially well for high-variance, low-bias procedures, such as trees.
  - For regression, we simply fit the same regression tree many times to bootstrap-sampled versions of the training data, and average the result; for classification, a committee of trees each cast a vote for the predicted class.
- Random Forest
  - a substantial modification of bagging that builds a large collection of de-correlated trees, and then averages them.
  - Compared to bagging, random forest deduce variance further.
- Boosting (Week 7)
  - a committee of weak learners evolves over time, and the members cast a weighted vote.
  - large improvements for weak learners with low-variance, high-bias procedures, such as simple trees.

Motivation of Ensemble Learning

Consider a regression model

$$y = y_*(x) + \varepsilon. \tag{1}$$

The aim is to predict the response y when x is a test data, or equivalently, to estimate the unknown function $y_*(\cdot)$.
The population MSE for a prediction $\widehat{y}$ (at a test point x_0) is decomposed as follows.

$$\mathbb{E}\left(\widehat{y}(x_0) - y(x_0)\right)^2 = \underbrace{\left(\mathbb{E}\left[\widehat{y}(x_0)\right] - y_*(x_0)\right)^2}_{bias} + \underbrace{Var\left(\widehat{y}(x_0)\right)}_{variance} + \underbrace{Var(y(x_0))}_{Bayes\ \ error}$$

- bias: how wrong the expected prediction is (corresponds to underfitting)
- variance: the amount of variability in the predictions (corresponds to overfitting)
- bayes error: the inherent unpredictability of the model

Illustration: Large Bias and Low Variance

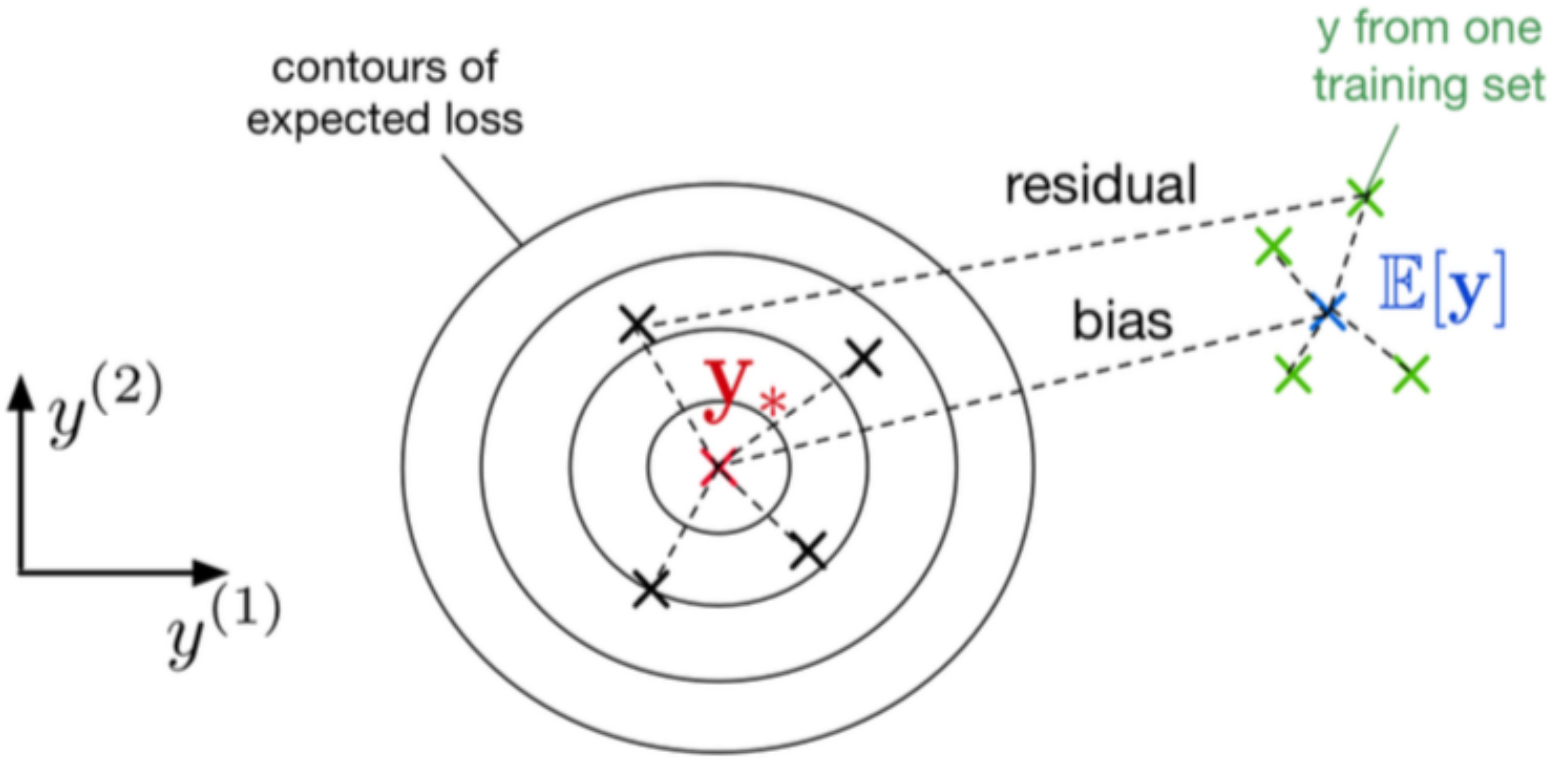
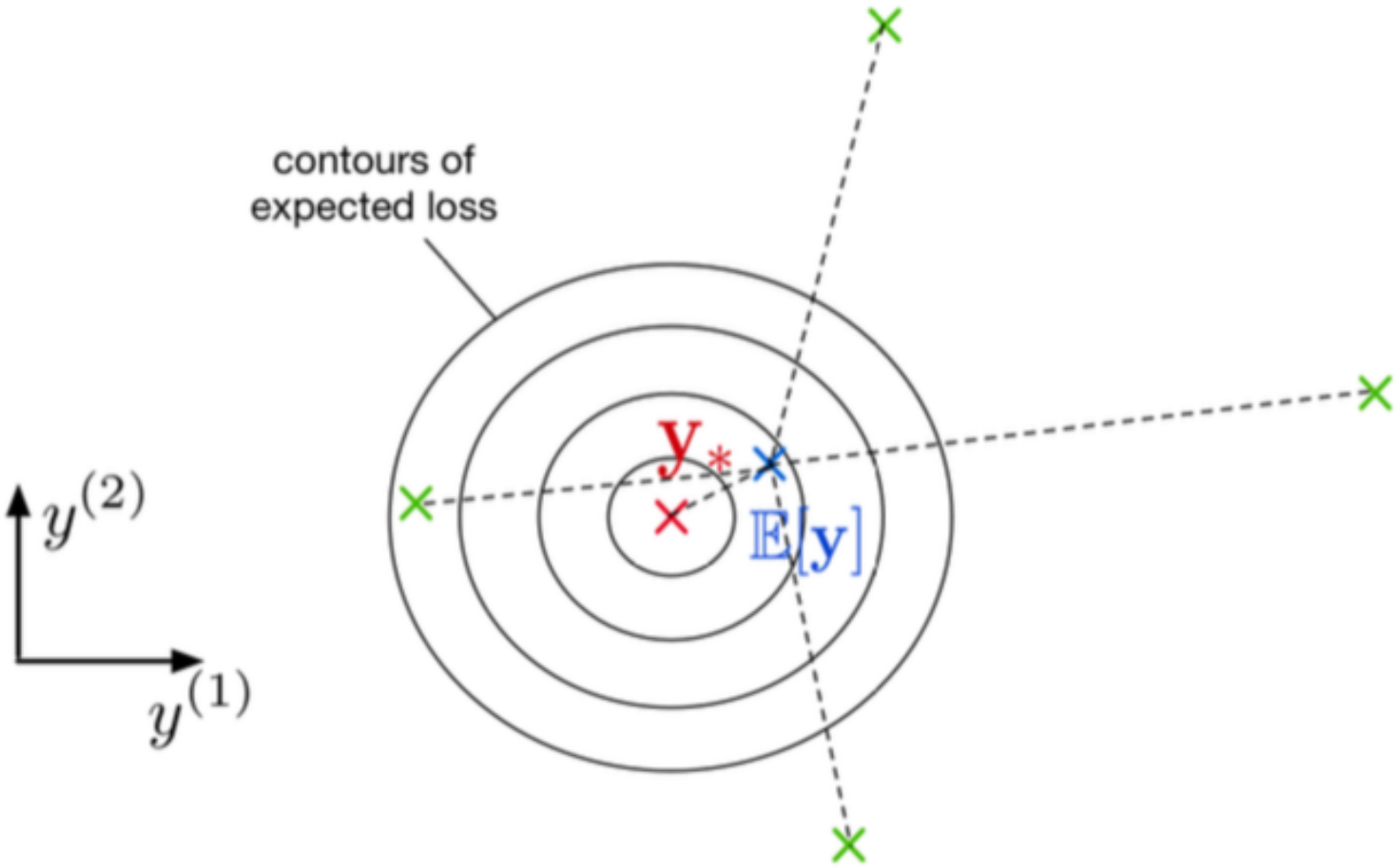
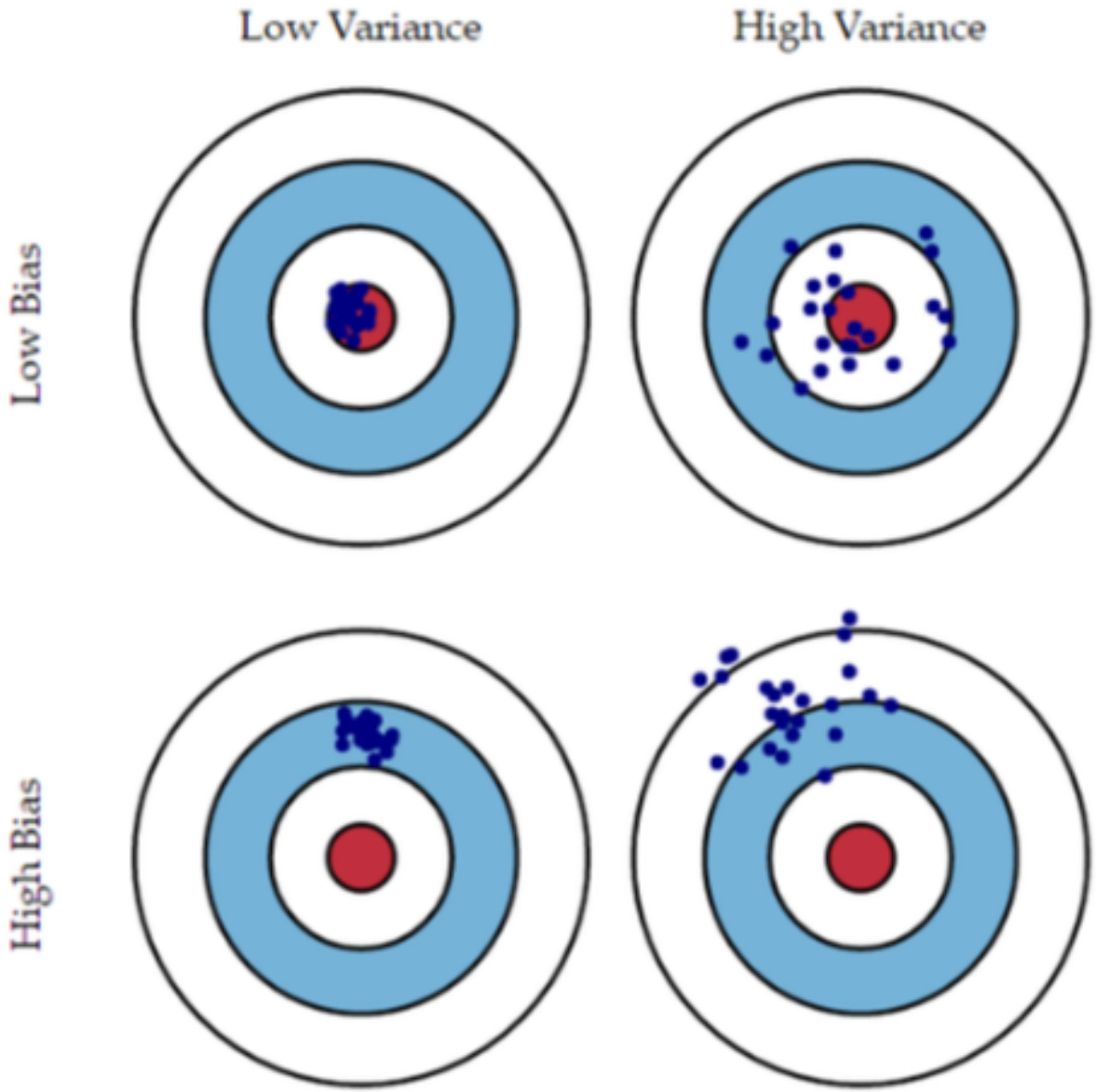


Illustration: Large Variance and Low Bias



Summary of Bias and Variance



Bagging

Motivation of Bagging

- Suppose we could somehow sample m independent training sets from p_{sample}.
- We could then compute the prediction y_i based on each one, and take the average $y = \frac{1}{m} \sum_{i=1}^{m} y_i$.
- How does this affect the three terms of the expected loss?
  - **Bayes error: unchanged**, since we have no control over it
  - **Bias: unchanged**, since the averaged prediction has the same expectation

$$\mathbb{E}[y] = \mathbb{E}\left[\frac{1}{m} \sum_{i=1}^{m} y_i\right] = \mathbb{E}[y_i]$$

  - **Variance: reduced**, since we're averaging over independent samples

$$\text{Var}[y] = \text{Var}\left[\frac{1}{m} \sum_{i=1}^{m} y_i\right] = \frac{1}{m^2} \sum_{i=1}^{m} \text{Var}[y_i] = \frac{1}{m} \text{Var}[y_i].$$

Bagging

Bagging (Bootstrap aggregation) averages the prediction over a collection of bootstrap samples, thereby reducing its variance.

- Fit Bootstrap Samples. Obtain bootstrap sample $\mathbf{Z}^{*b} = \{(x_1^*, y_1^*), (x_2^*, y_2^*), \ldots, (x_N^*, y_N^*)\}, b = 1, 2, \ldots, B$ from the training data $\mathbf{Z} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$. Fit the model with each bootstrap sample $\mathbf{Z}^{*b}$ and get the prediction $\widehat{f}^{*b}(x)$.
- Prediction.
  - Regression.

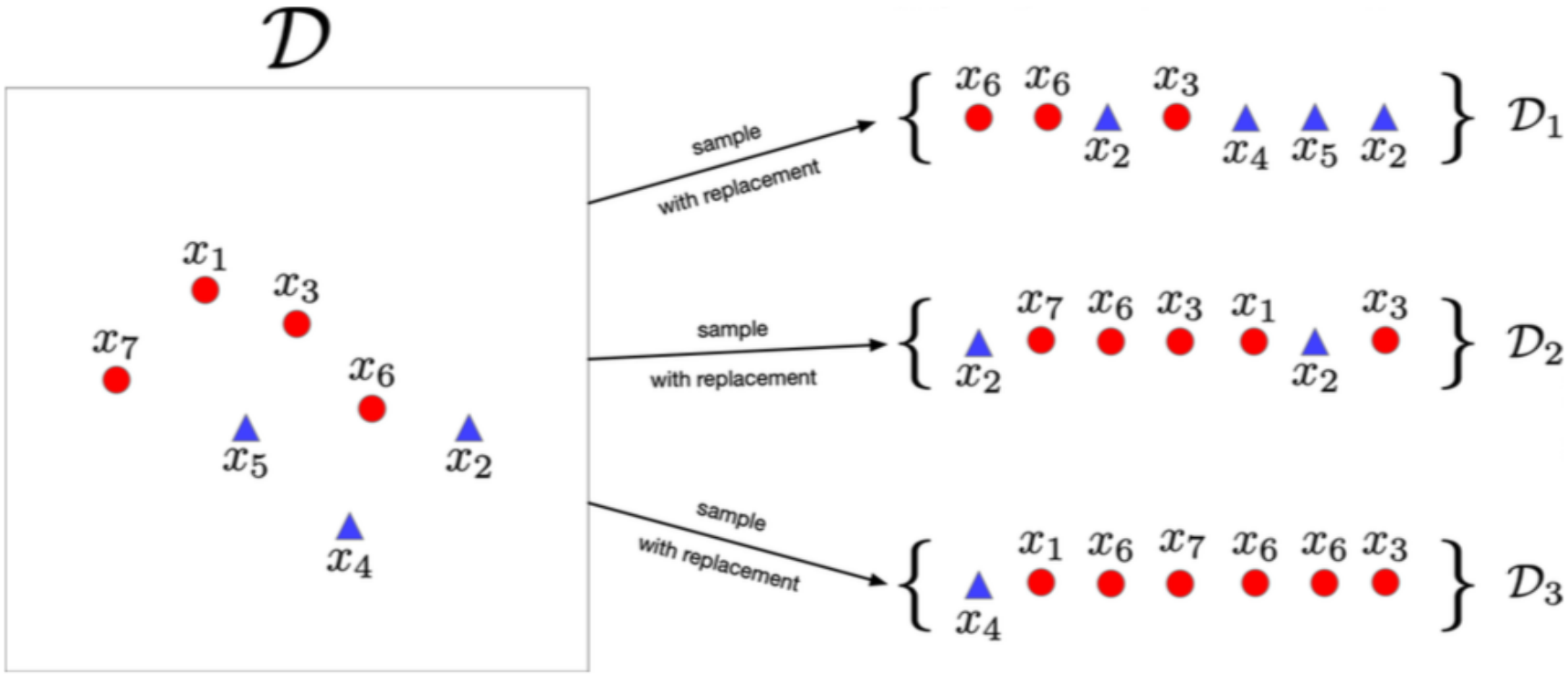
$$\widehat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}^{*b}(x). \tag{2}$$

  - Classification.

$$\widehat{f}_{bag}(x) = \arg\min_k p_k(x), \tag{3}$$

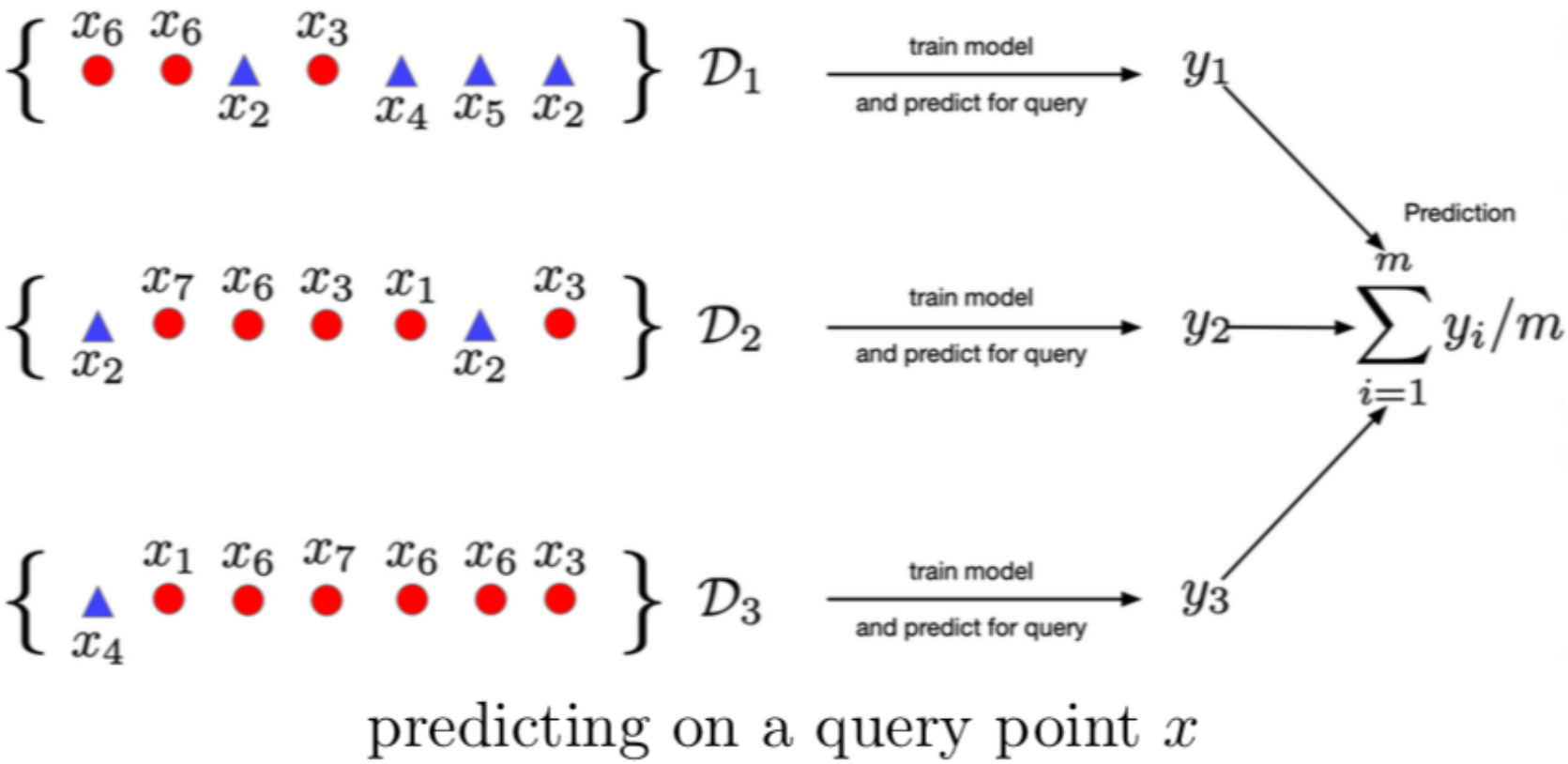
where $p_k(x) = \sum_{b=1}^{B} I(\widehat{f}^{*b}(x) = k)$.

Illustration of Bagging Procedure (1)



in this example $n = 7$, $m = 3$

Illustration of Bagging Procedure (2)

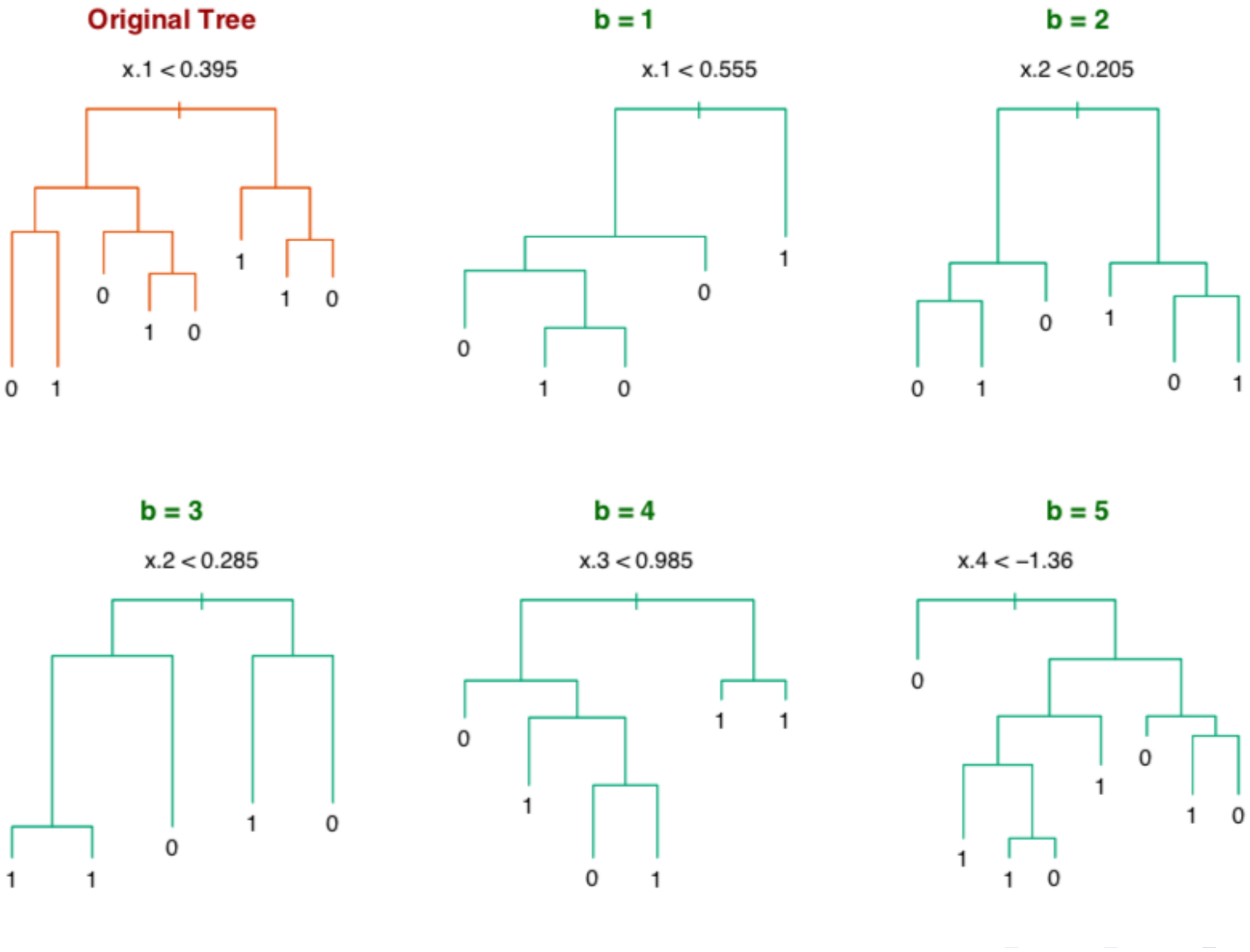


Example: Simulated Data

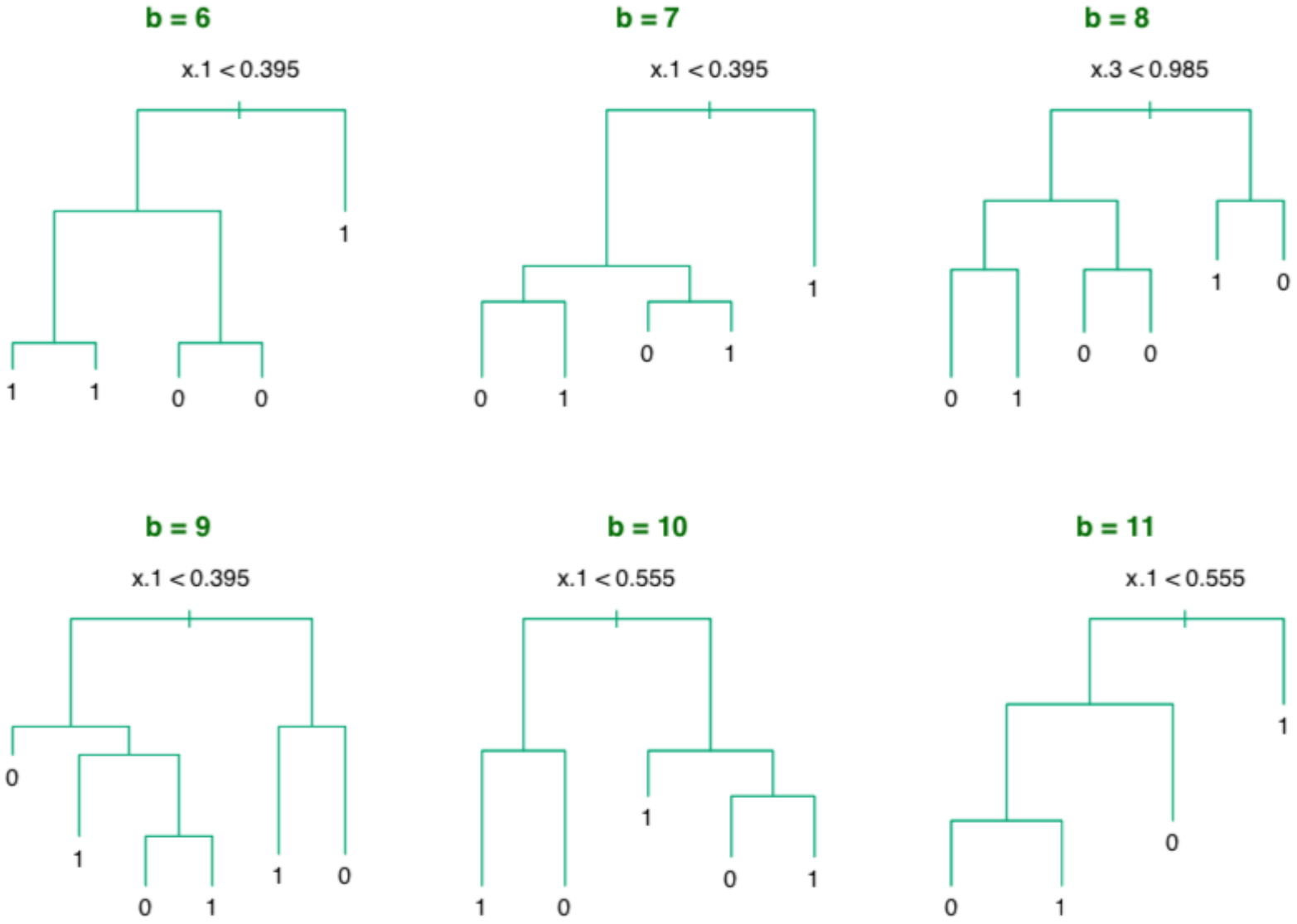
Consider a classification problem with the following data generating process.

- Predictors: totally 5 predictors, each having a standard Gaussian distribution with pairwise correlation 0.95.
- Response: taking two values 1 or 0. The distribution of the response is generated according to $Pr(Y = 1|x_1 \leq 0.5) = 0.2$ and $Pr(Y = 1|x_1 > 0.5) = 0.8$.
- Sample size $N = 30$.
- The number of bootstrap samples $B = 200$.

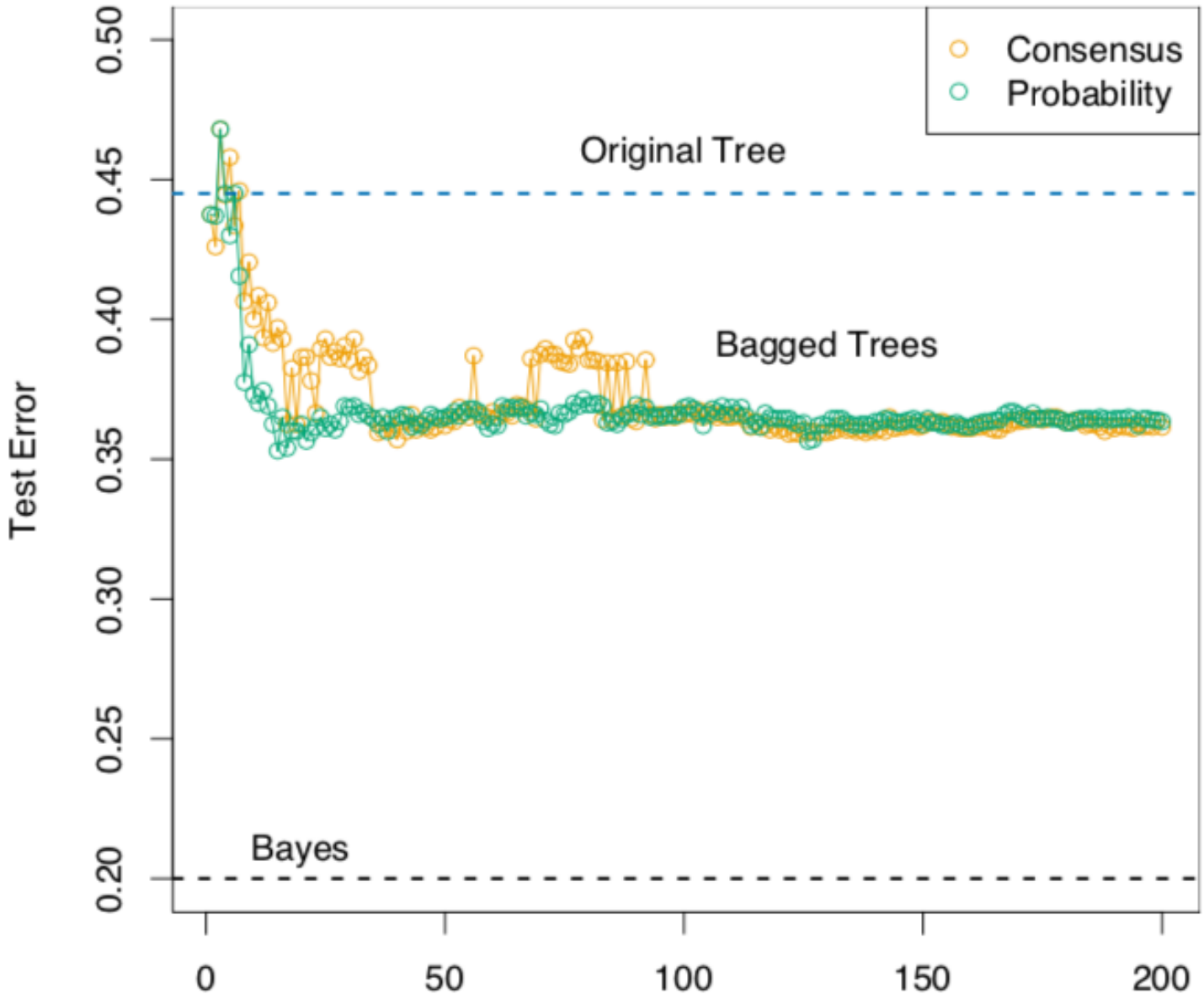
Example: Bagging Trees



Example: Bagging Trees



Example: Test Error for Bagging



Comments on Bagging

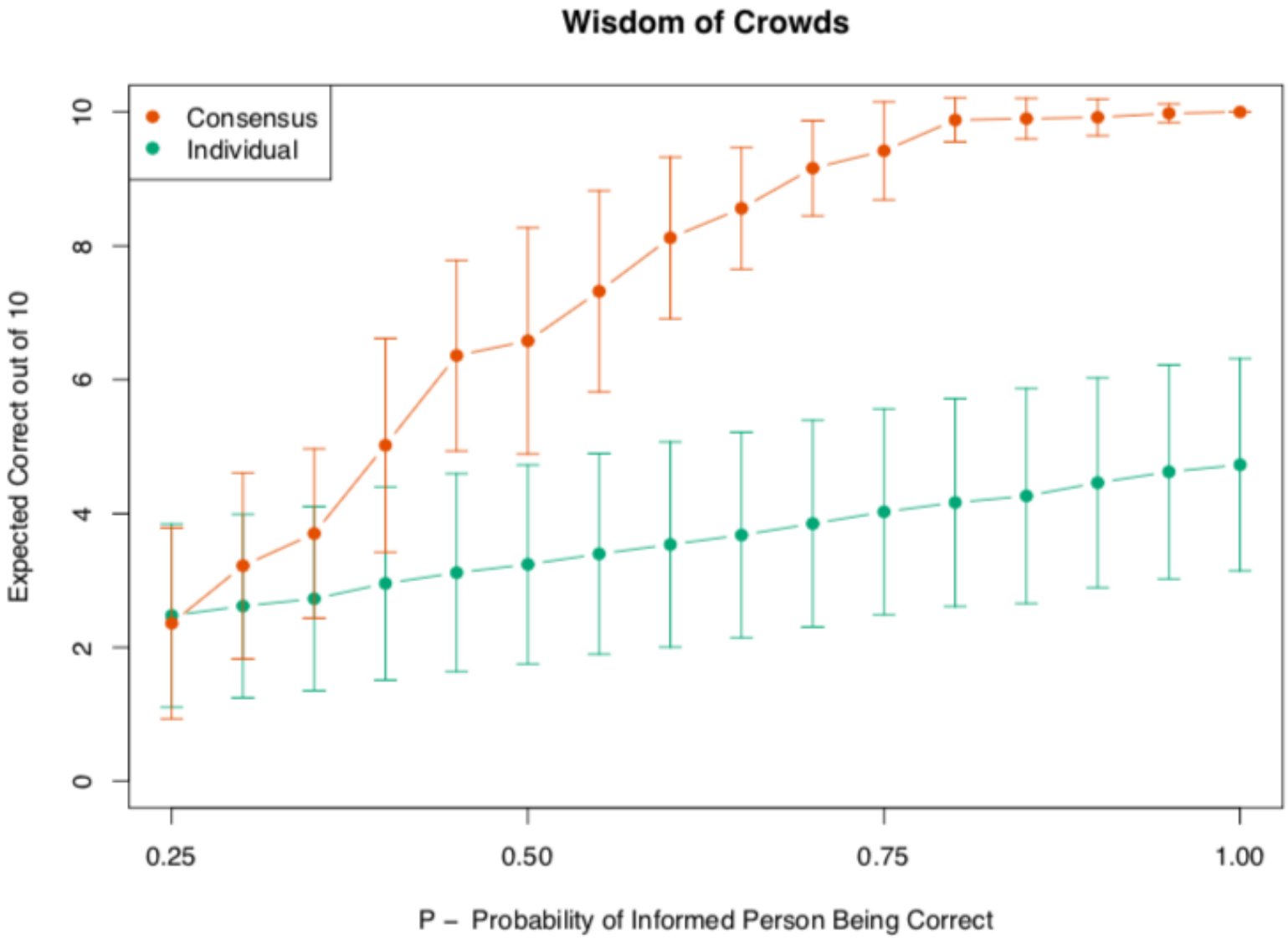
- In terms of bias and variance decomposition, bagging for regression problems will decrease the variance (under some conditions) and will not increase the bias. So bagging for regression is often a good statistical technique.
- Applying bagging on classification problems is more complicated. The bias and variance decomposition does not hold for classification under $0-1$ loss, because of the non-additivity of bias and variance. Bagging a good classifier can make it better, but bagging a bad classifier can make it worse.

Advantage of Bagging

- Example 1: Suppose $Y = 1$ for all x, and the classifier $\widehat{G}(x)$ predicts $Y = 1$ (for all x) with probability 0.4 and predicts $Y = 0$ (for all x) with probability 0.6. Then the misclassification error $\widehat{G}(x)$ is 0.6 but that of the bagged classifier is 1.
- Example 2: Let the Bayes optimal decision at x be $G(x) = 1$ in a two-calss example. Suppose each of the weak learners G_b^* have an error-rate $e < 0.5$, and let $S_1(x) = \sum_{b=1}^{B} I(G_b^*(x) = 1)$ be the consensus vote for class 1. Since the weak learners are assumed to be independent, $S_1(x)$ $Bin(B, 1 - e)$, and $Pr(S_1 > \frac{B}{2}) \to 1$ as B gets large.

This phenomenon has been popularized outside of statistics as the "Wisdom of Crowds" - the collective knowledge of a diverse and independent body of people typically exceeds the knowledge of any single individual, and can be harnessed by voting.

Example: Wisdom of Crowds

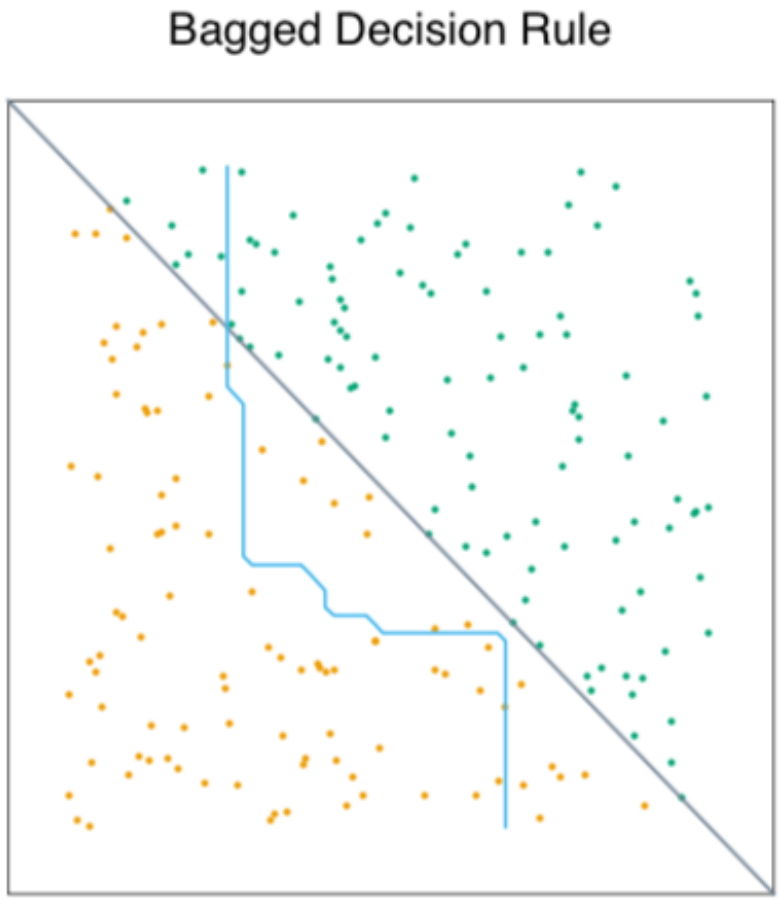


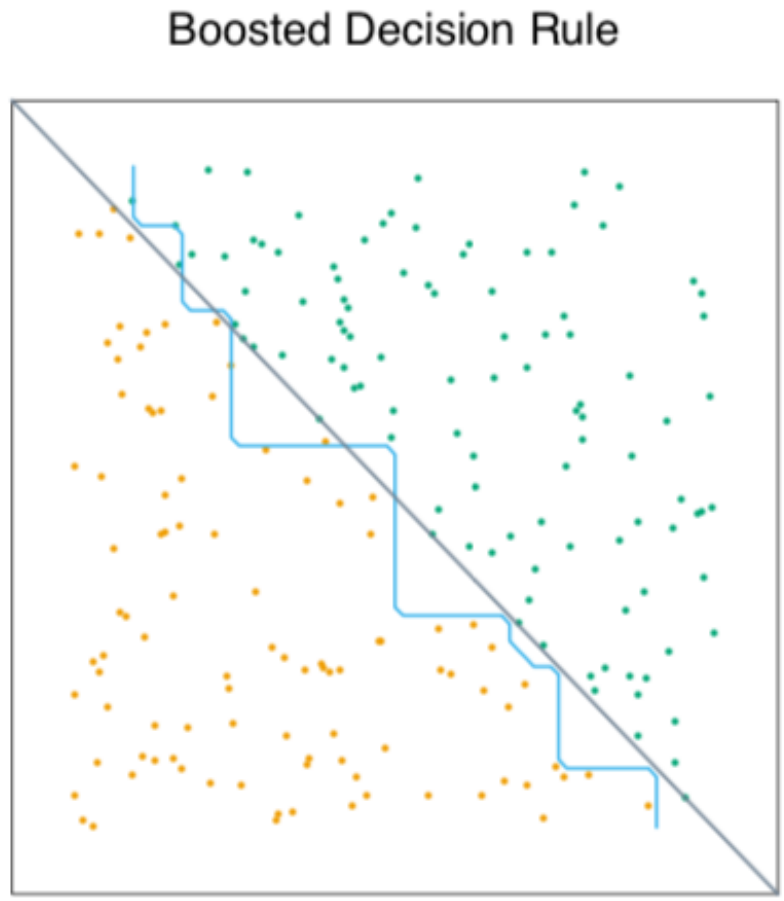
Bagging vs Boosting

Bagging is helpless under some situations. For example,

- The 100 data points shown have two features and two classes, separated by the gray linear boundary $x_1 + x_2 = 1$.
- We choose as our classifier $\widehat{G}(x)$ a single axis-oriented split, choosing the split along either x_1 or x_2 that produces the largest decrease in training misclassification error.

Example: Bagging vs Boosting





Random Forest

Motivation of Random Forest

- An averaging of B i.i.d random variables, each with variance σ^2, has variance $\frac{1}{B}\sigma^2$.
- If the variables are i.d but dependent with positive pairwise correlation ρ, the variance of the average is

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2. \tag{4}$$

  As B increases, the second term disappears, but the first remains.
- Hence the size of the correlation of pairs of bagged trees limits the benefits of averaging.

Idea of Random Forest

The idea in random forests is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much.

- This is achieved in the tree-growing process through random selection of the input variables.
- Specifically, when growing a tree on a bootstrapped dataset, before each split, select $m \leq p$ of the input variables at random as candidates for splitting.

Algorithm of Random Forest

Algorithm 15.1 *Random Forest for Regression or Classification.*

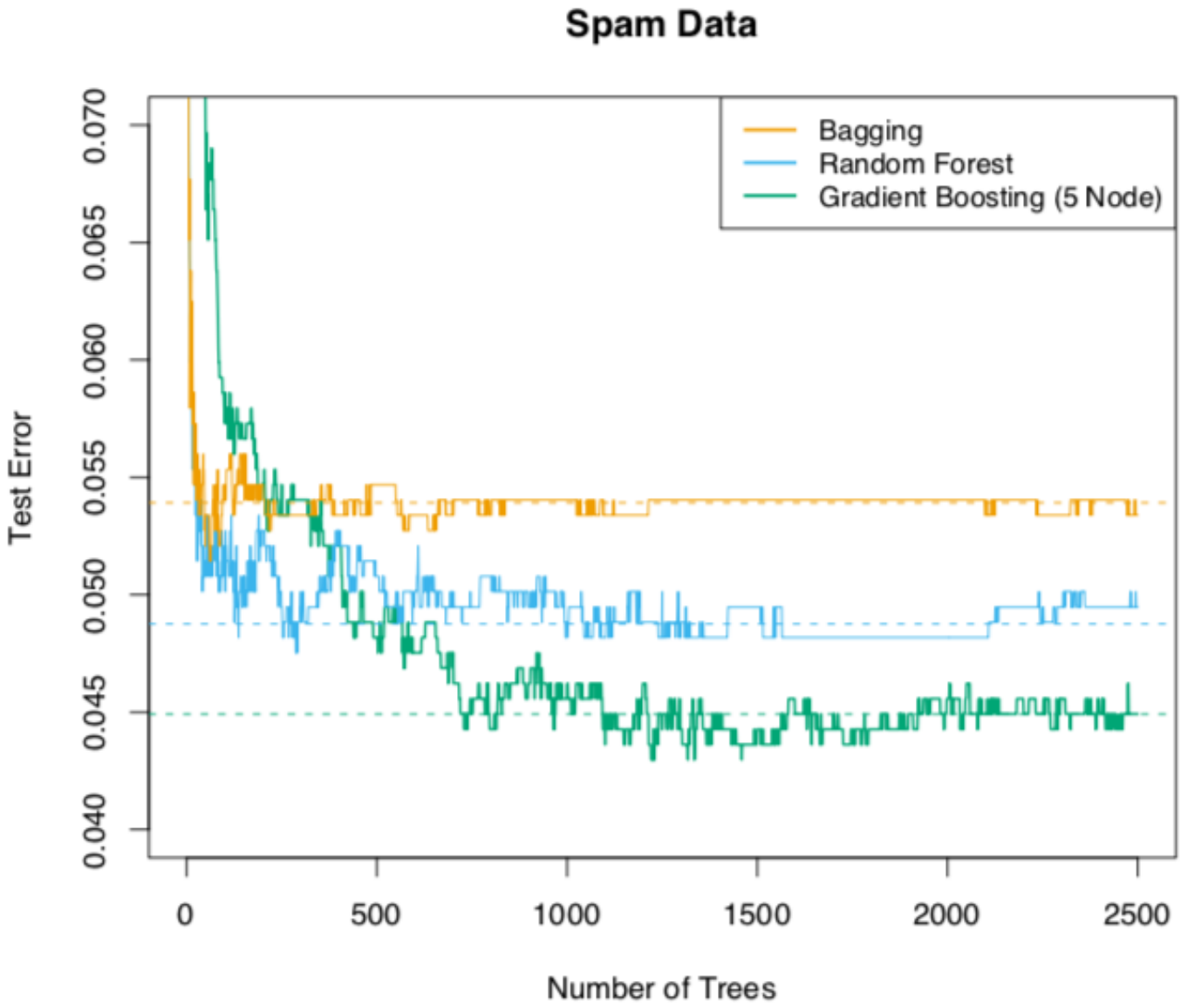
1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. 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 $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Example: Comparision



Comments on Random Forest

- For classification, the default value for m is $\sqrt{p}$ and the minimum node size is one.
- For regression, the default value for m is $\frac{p}{3}$ and the minimum node size is five.

In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters. In empirical analysis on California Housing Data, the $m = 6$ performs much better than the default value $\frac{8}{3} \asymp 2$.

Out of Bag Error

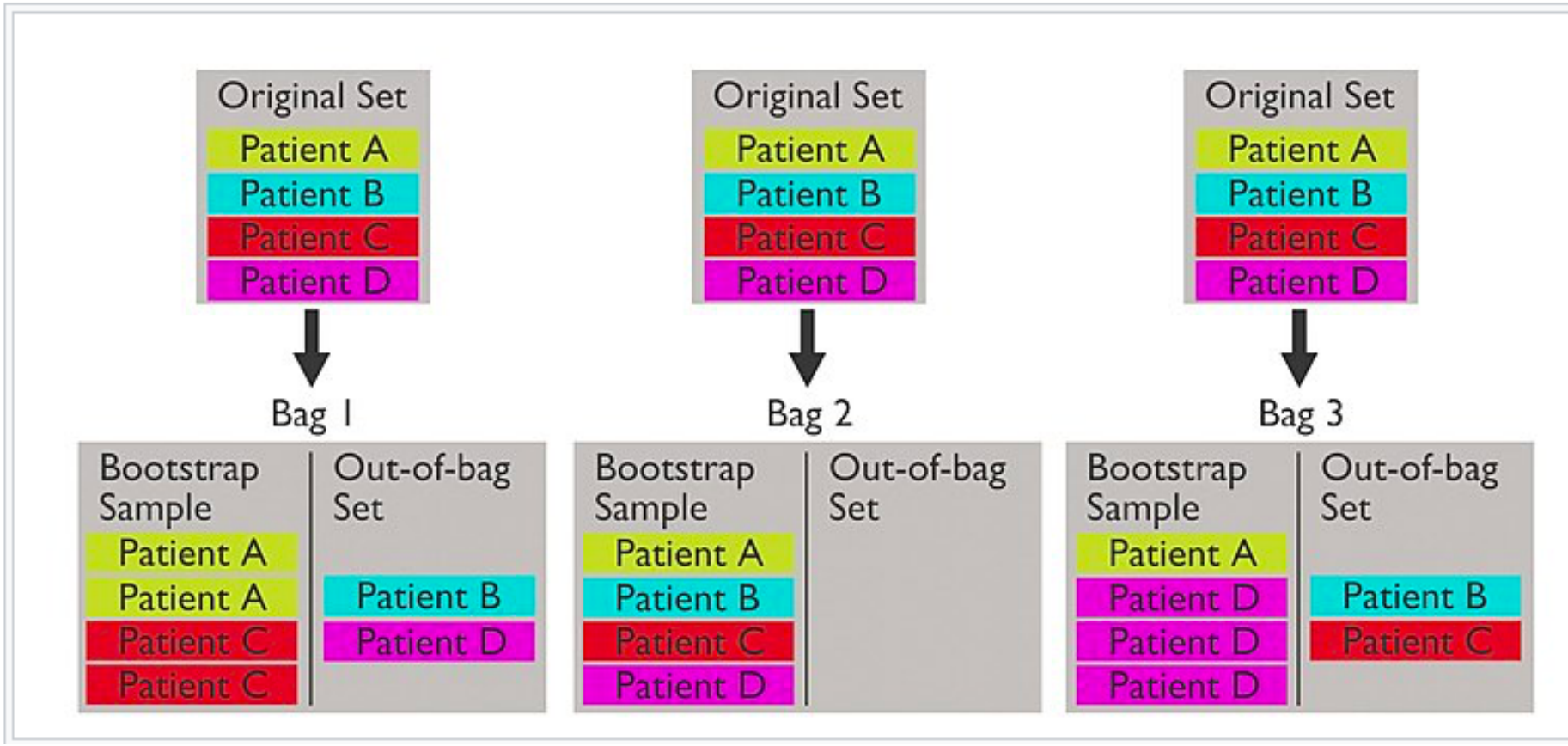
An important feature of random forests is its use of out-of-bag (OOB) samples: for each observation $z_i = (x_i, y_i)$, construct its random forest predictor by averaging only those trees corresponding to bootstrap samples in which z_i did not appear. In details, the calculation for OOB error is

- Find all trees that are not trained by the OOB instance.
- Take the majority vote of these models' result for the OOB instance, compared to the true value of the OOB instance.
- Compile the OOB error for all instances in the OOB dataset.

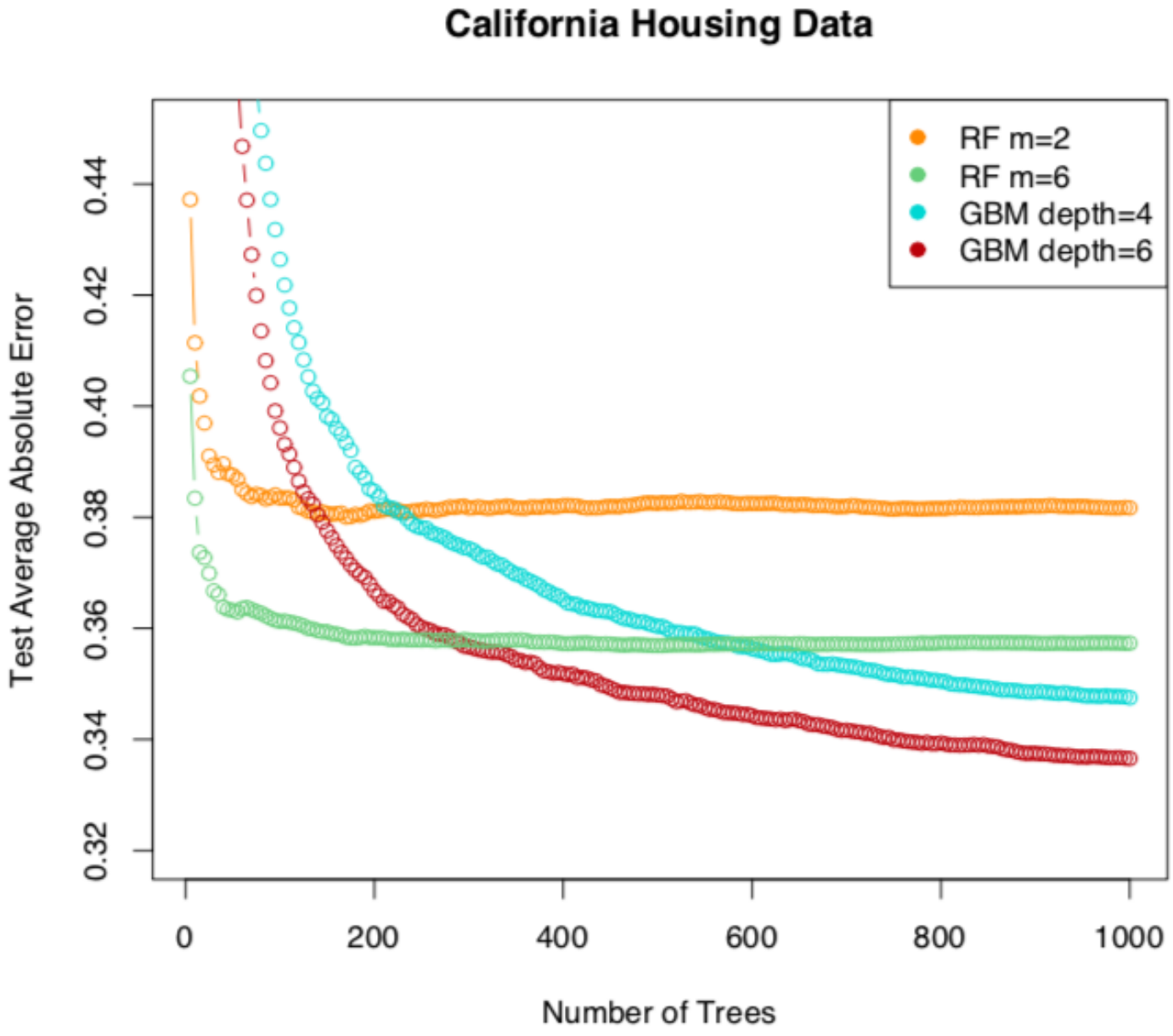
Comments:

- An OOB error estimate is almost identical to that obtained by N-fold cross-validation.
- Unlike many other nonlinear estimators, random forests can be fit in one sequence, with cross-validation being performed along the way. Once the OOB error stabilizes, the training can be terminated.

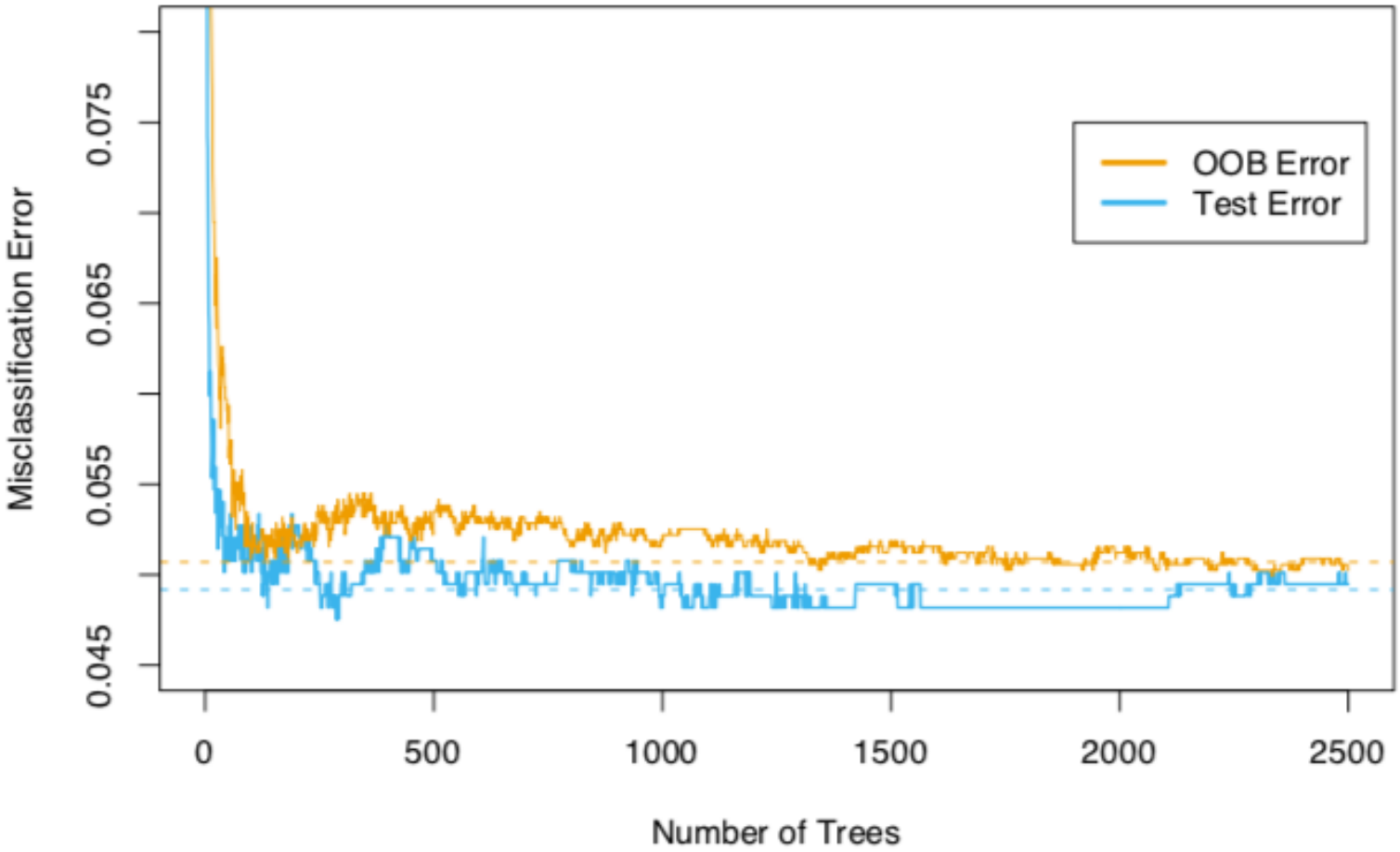
Illustration of OOB



Example: Random Forest vs Boosting



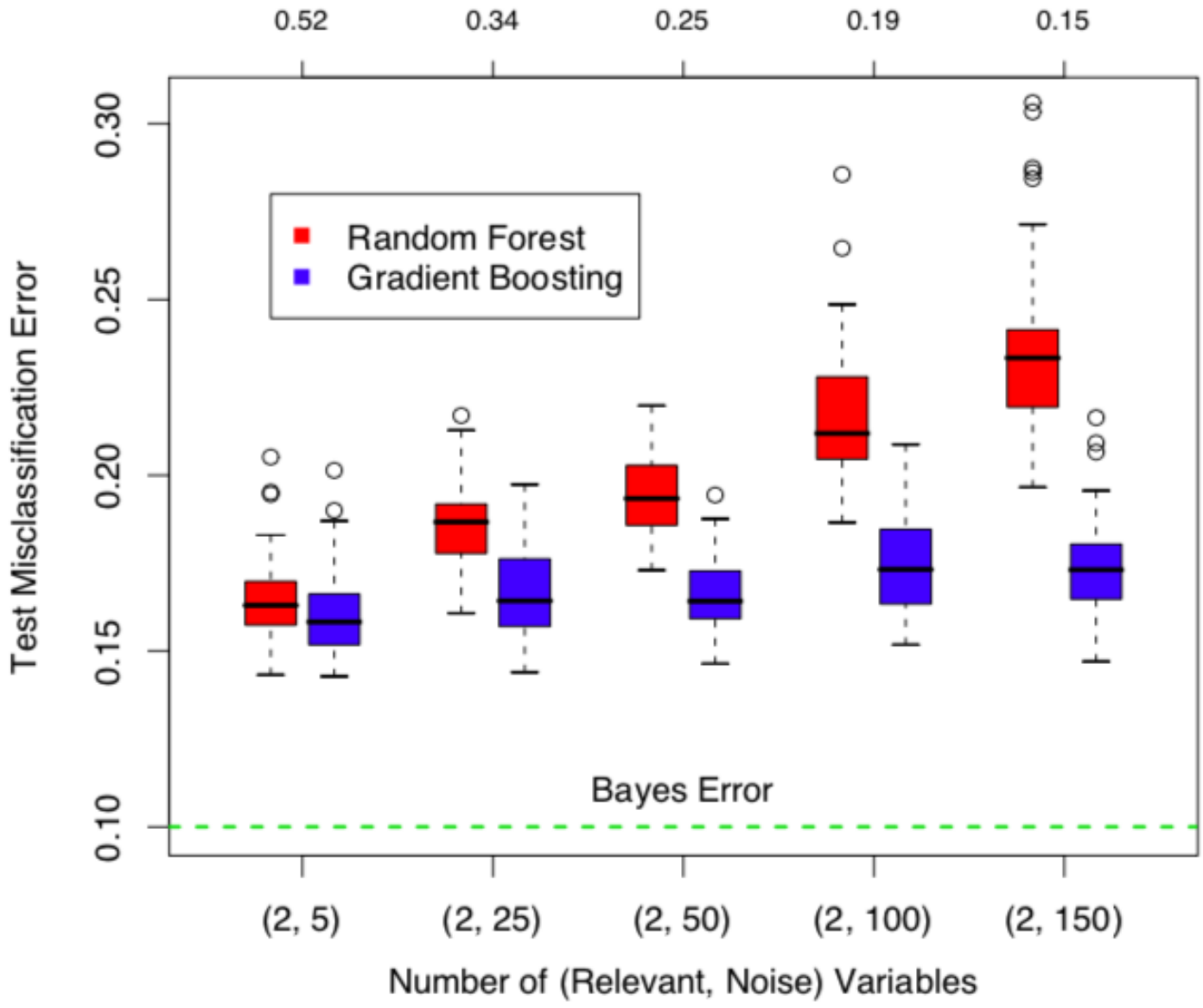
Example: OOB error vs Test error



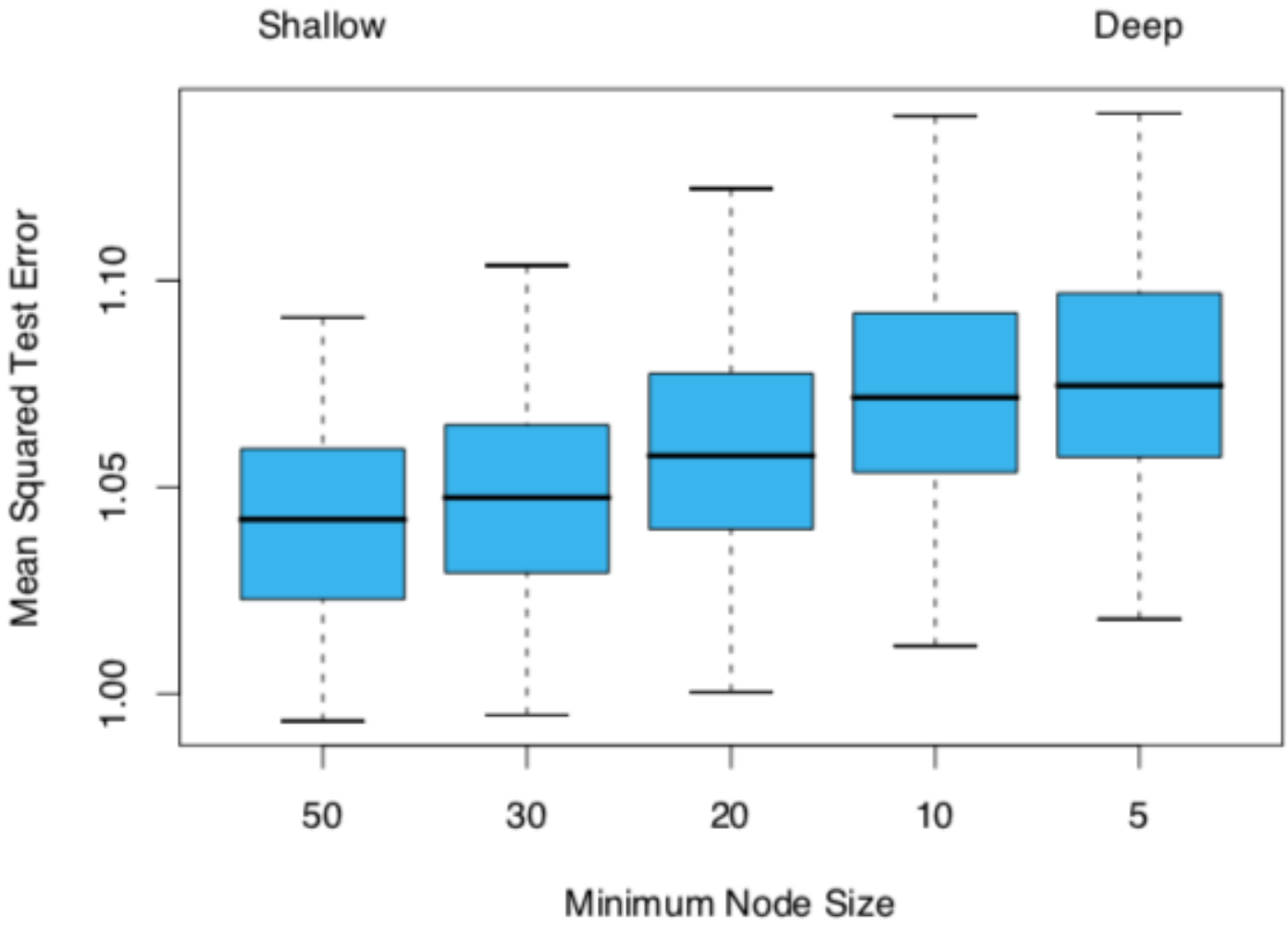
Random Forest and Overfitting

- When the number of variables is large, but the fraction of relevant variables small, random forests are likely to perform poorly with small m.
- Another claim is that random forests "can not overfit" the data.
- Classifiers are less sensitive to variance, and this effect of overfitting is seldom seen with random-forest classification.

Example: Effects of Noise Variables



Example: Effects of Depth on Random Forest



Discussion: Correlation of Trees in Random Forests