Bagging and random forests

MATH 6312
Department of mathematics, UTA

ESL 8.7, 8.7.1, 15.1, 15.2, 15.3, 15.3.1, 15.3.2, 15.4.2

Optional reading: ESL15.3.2

Revisit: CART

- Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
- To fit the tree model, we first grow a large tree, and then prune it using the cross validation.
- Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches.
- However, by aggregating many decision trees, the predictive performance of trees can be substantially improved.

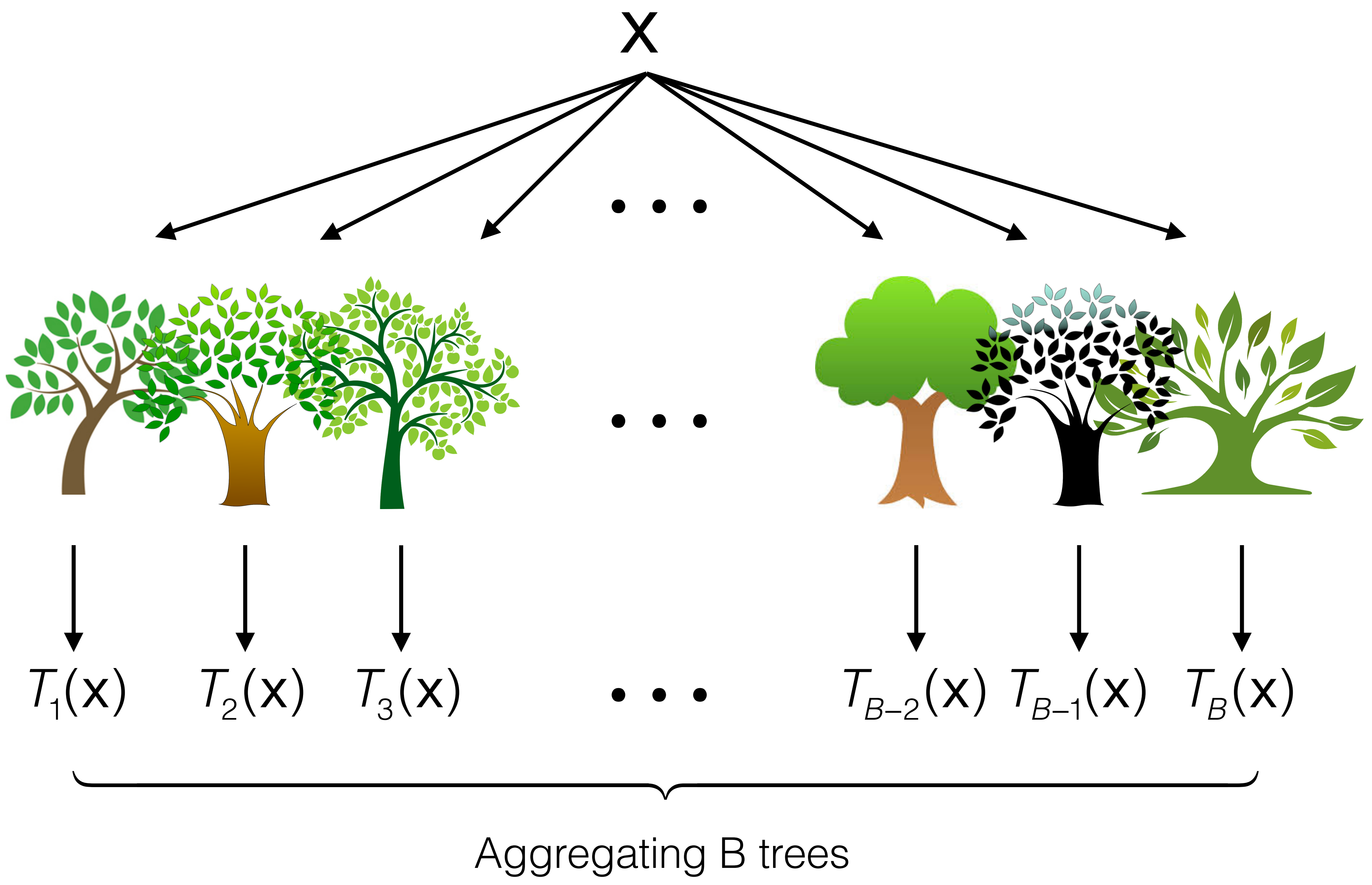
Single tree

- Trees generally suffer from high variance because they are quite unstable.

- A smaller change in the observed data can lead to a dramatically different sequence of splits, and hence a different prediction.

- High variance can lead to high expected prediction error

Aggregating many trees



$\mathbf{x}$

$\cdots$

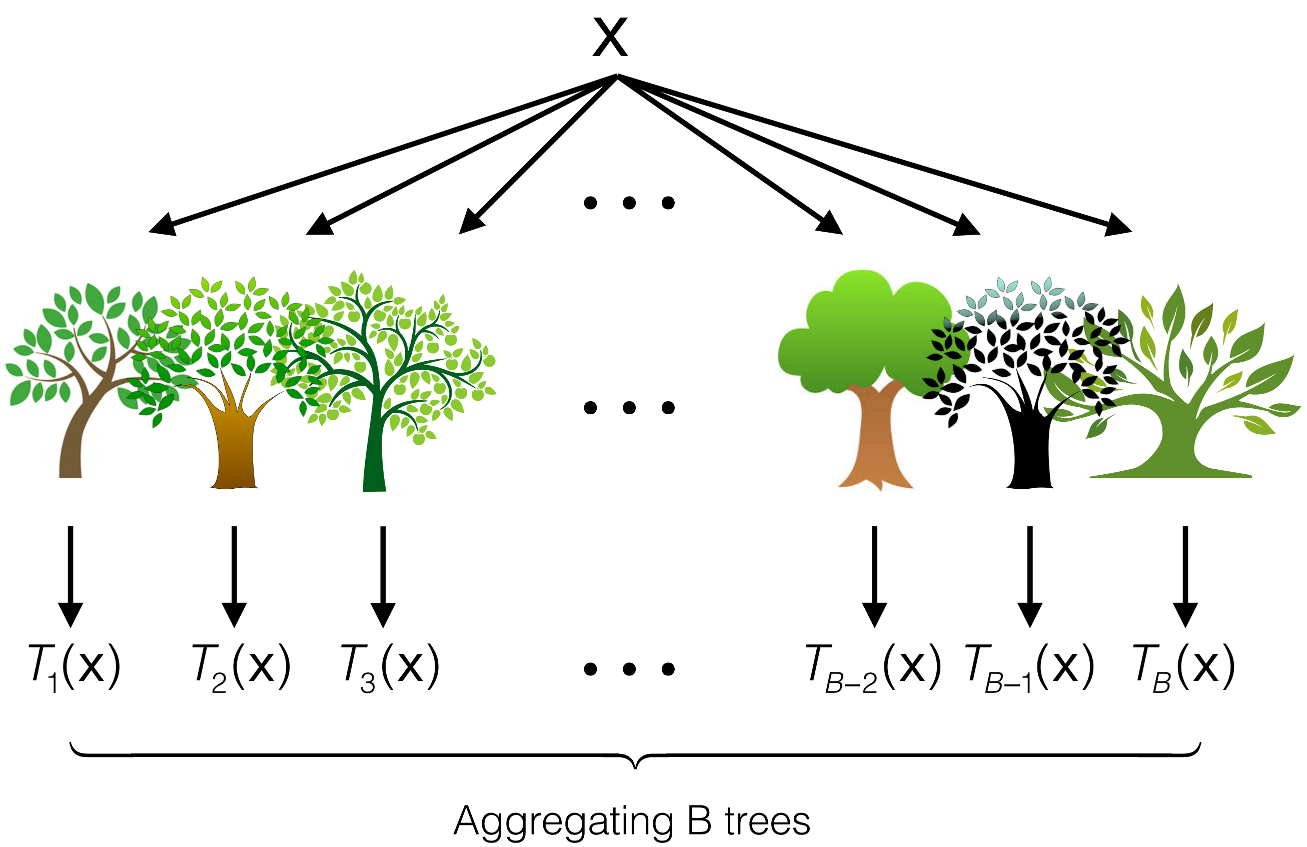
$\cdots$

$T_1(\mathbf{x})$ $T_2(\mathbf{x})$ $T_3(\mathbf{x})$ $\cdots$ $T_{B-2}(\mathbf{x})$ $T_{B-1}(\mathbf{x})$ $T_B(\mathbf{x})$

Aggregating B trees

- We aggregate many trees (forming forests) for the variance reduction in prediction.

How to aggregate results from B trees?



- A simple average of prediction at a point x.

$$\frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

- A class which receives majority vote

$$\mathit{majority\ \ vote}\ \{T_b(x)\}_1^B$$

  - eg. classification at x from B=5 trees are {email, spam, email, email, email}. Then, we use majority voting (email) to aggregate results

How to train many trees?

- If we can afford huge samples, each tree can be trained using independent samples from the population.

- In practice, we cannot afford huge samples, so bootstrap samples are used to train trees.

- Then, we aggregate trees trained based on each bootstrapped data -> bootstrap aggregation.

Bootstrap

- The basic idea underlying the bootstrap is that we can estimate the true F (unknown) using the empirical distribution.

- The empirical distribution of F is

$$z = (y, x) \qquad P_{\hat{F}}(Z = z) = \begin{cases} \frac{1}{N} & \text{if } z = z_i \text{ for some } i \\ 0 & \text{otherwise} \end{cases}$$

  - The empirical distribution assigns equal probability (1/N) at each observation z_i.

- Sample drawn from the empirical distribution is called the bootstrap sample. Unlike sampling from F, we can draw bootstrap samples as many as we want.

Bootstrap samples

- A bootstrap sample of size m: $\{z_1^*, \ldots, z_m^*\}$
- Recall the empirical distribution:

$$P_{\hat{F}}(Z = z) = \begin{cases} \frac{1}{N} & \text{if } z = z_i \text{ for some } i \\ 0 & \text{otherwise} \end{cases}$$

  - Each bootstrap sample is drawn independently as follows: $z_j^* = z_i$ with probabilty $\frac{1}{N}$ for all i, j
  - This is equivalent to randomly draw m samples with replacement from $\{z_1, \ldots, z_N\}$

Bagging (=Bootstrap aggregation)

- Bagging is a general-purpose procedure for reducing the variance of a statistical learning method (not just for trees).

- First, we draw B bootstrap training samples (m=N). Second, we train a base learner $T_b(x)$ on the b*th* bootstrapped training set. We then use average or majority voting to aggregate base learners.

- By aggregating models, we can reduce the variance of the prediction.

  - Discuss how the bagging reduces the variance. $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$

Continued: bagging

- For the classification, majority voting (consensus strategy) does not provide a good estimate of class probability. Can you think why?

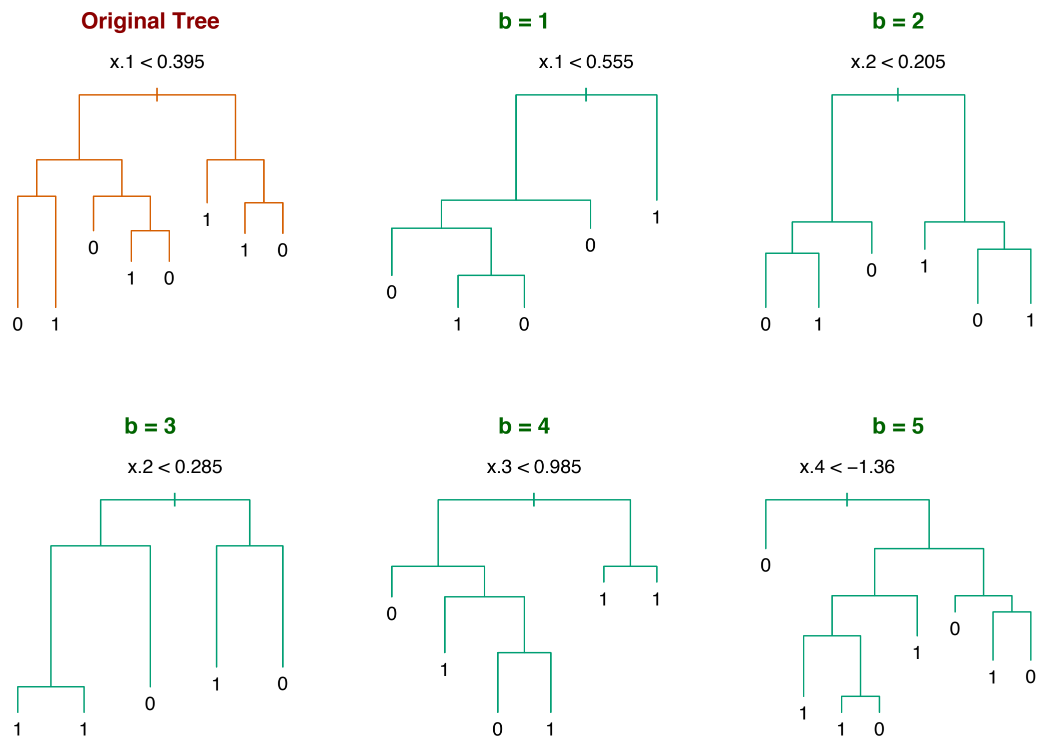
- One can use the bagging estimate of class probability by averaging class probabilities from each model trained using bootstrap samples.

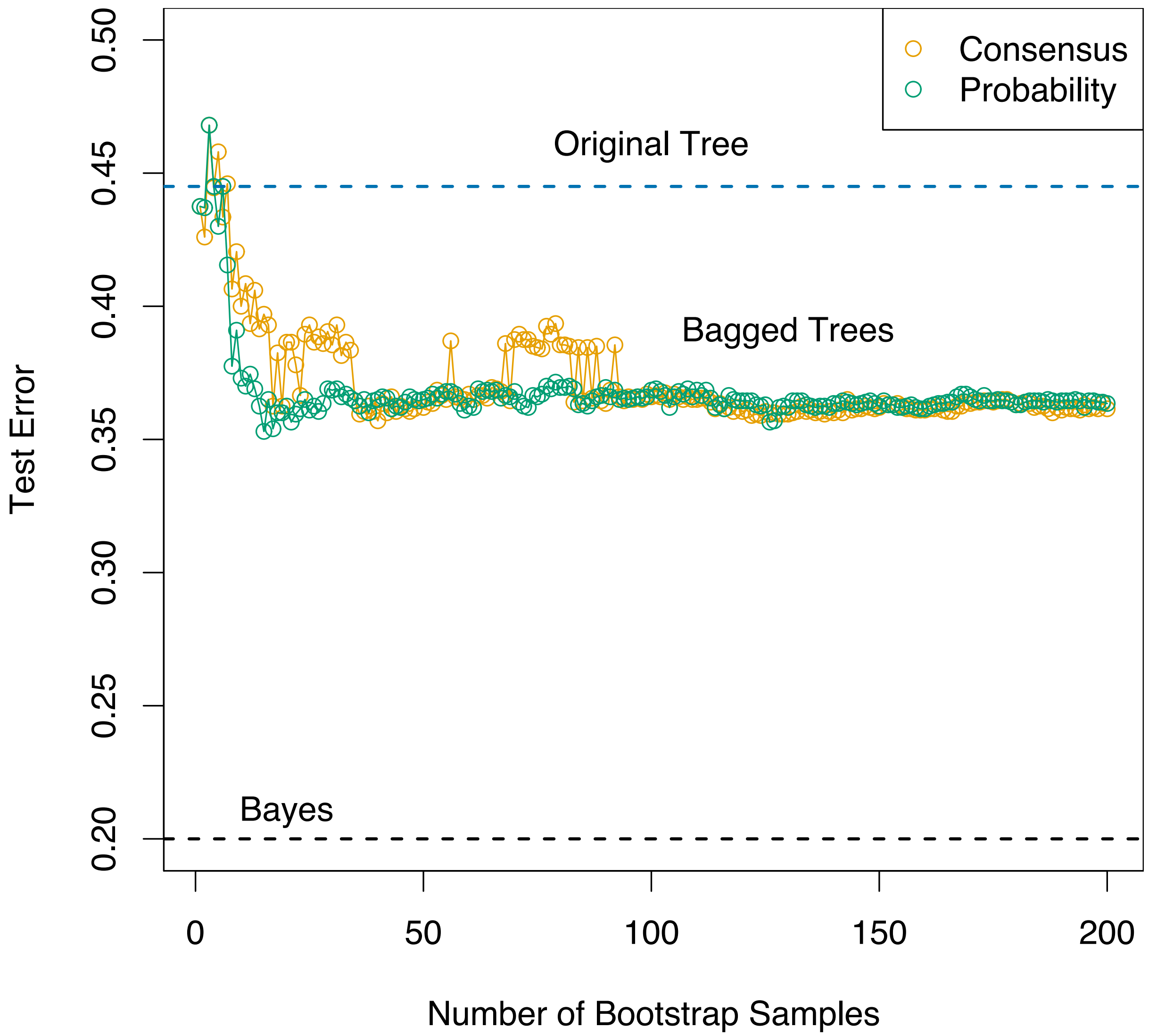
$$\hat{p}_k^{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{p}_k^{b}(x)$$

- Then, the bagged classifier is $\hat{f}^{bag}(x) = \underset{k}{\operatorname{argmax}}\, \hat{p}_k^{bag}(x)$

- This form of bagging (probability strategy) is preferred to get the estimate of class probability. Sometimes, it performs better than the consensus strategy.

Probability vs consensus





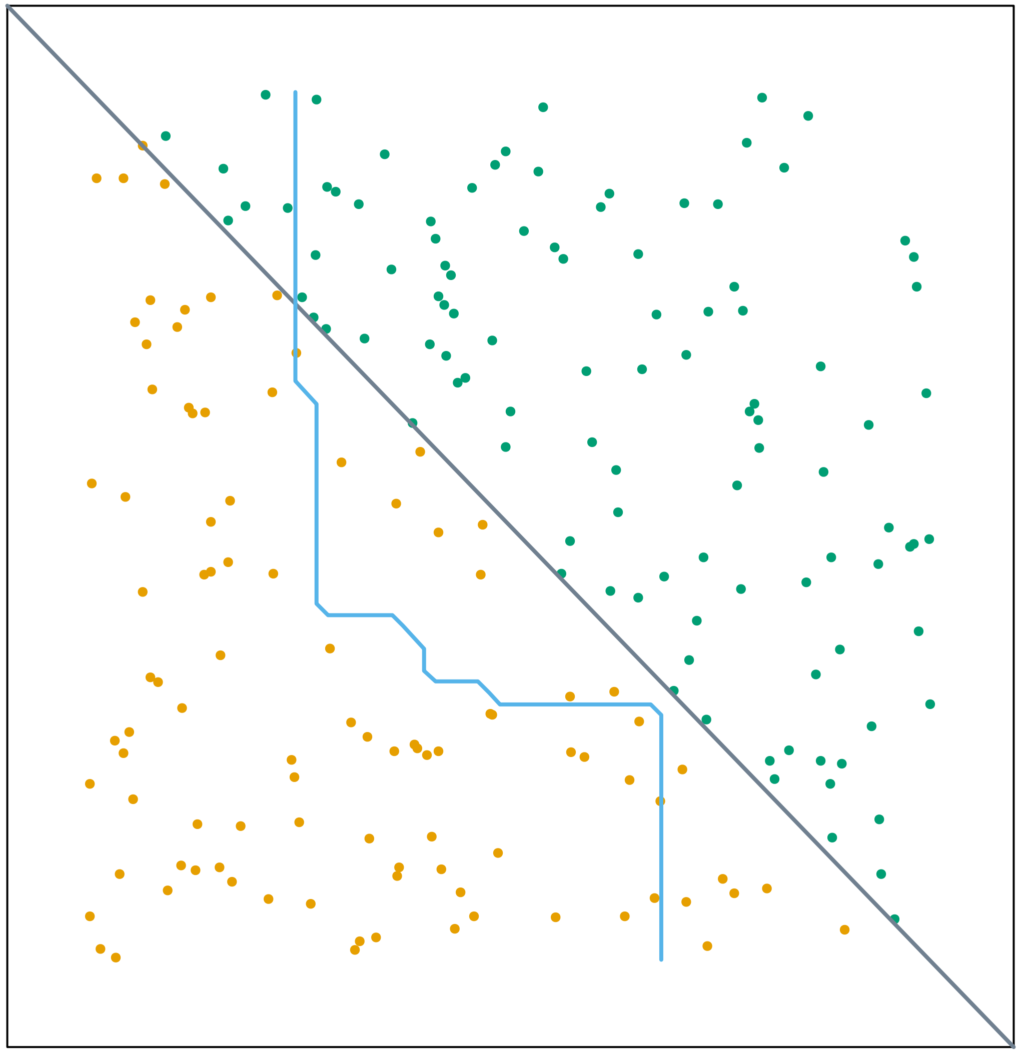
- ❖ No pruning is considered.
- ❖ Trees have high variance due to the correlation in the predictors.
- ❖ Bagging succeeds in smoothing out this variance and hence reducing the test error.

Orange points correspond to the majority vote, while the green points average the probabilities .

- Sometimes, bagging fails to improve the prediction accuracy, and can show worse performance than an original model.
    - Usually, it happens when we perform the bagging with a bad base learner.
    - Also, when bagged trees are highly correlated, it leads to very inaccurate prediction.
- There are some disadvantage of bagging.
    - Loss of interpretability: trees are highly interpretable. however, bagged trees are an average of many trees, so we lose the interpretative ability.
    - Computational complexity: the model training needs to done for B bootstrapped training sets.

Bagging increases model space

Bagged Decision Rule



- The diagonal line: true decision boundary.
- We use the single split tree (decision stump) as a base classifier.
- Bagged trees with B = 50 bootstrap samples is shown by the blue curve
- Bagging increases somewhat the space of models of the base classifier, but fails to represent the very basic decision boundary.

Out-of-Bag Error Estimation

- It turns out that there is a very straightforward way to estimate the test error of a bagged model.

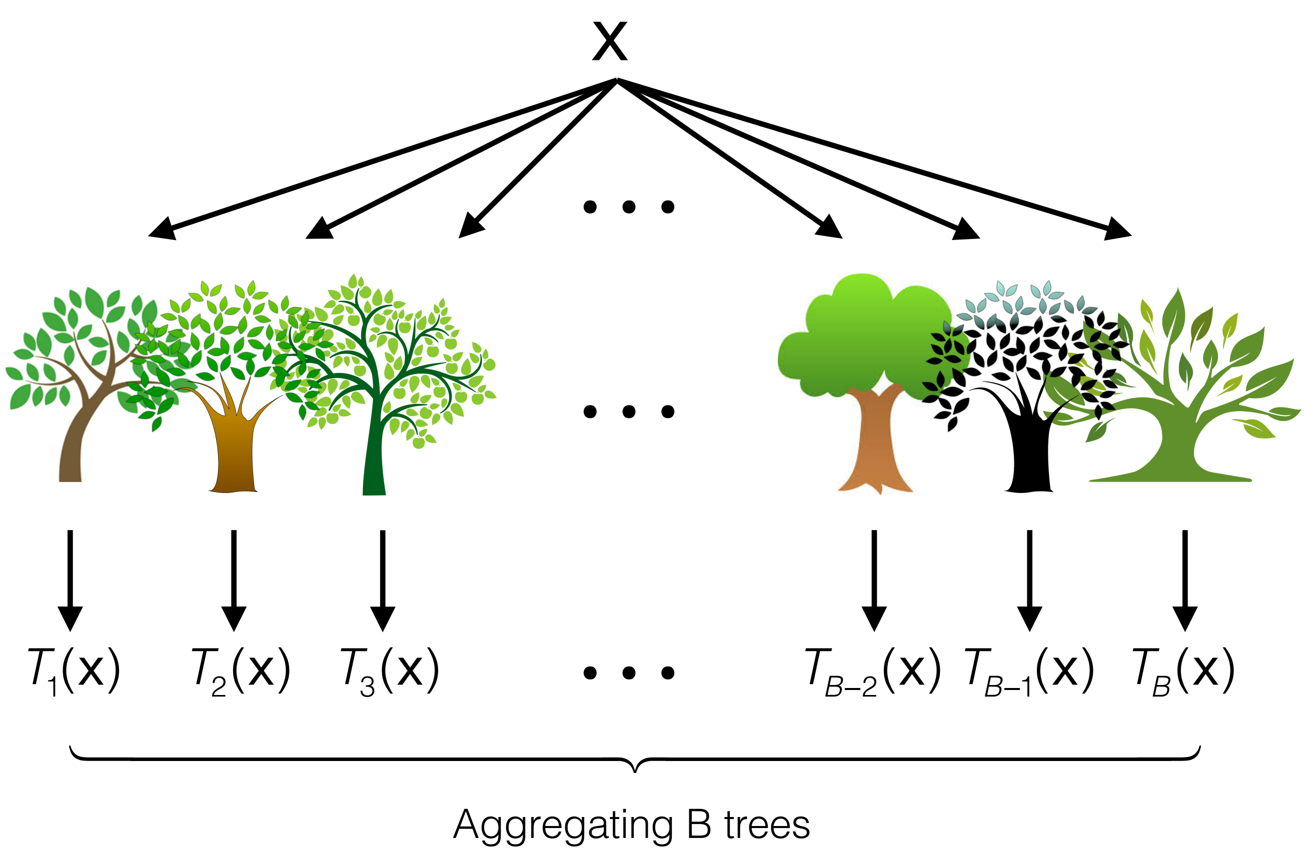
- Recall that the key to bagging is that trees are repeatedly fit to bootstrapped subsets of the observations. One can show that on average, each bagged tree makes use of around two-thirds of the observations.

- The remaining one-third of the observations not used to fit a given bagged tree are referred to as the out-of-bag (OOB) observations.

- We can predict the response for the i*th* observation using each of the trees in which that observation was OOB. This will yield around B/3 predictions for the i*th* observation, which we average.

- This estimate is essentially the LOO cross-validation error for bagging, if B is large. -> No extra computation for CV is needed.

Random forests



- Bagging is one way to aggregate many trees to improve the prediction accuracy.
- However, bagged trees are not independent, so they could degrade performance of the original tree.
- How to make bagged trees to have small correlations among themselves?

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 $\{T_b\}_1^B$.
```

additional randomization has de-correlation effects!

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

- Random forests = bagged trees + additional randomization (before each split, a splitting variable is chosen from m ≤ p predictors selected at random).

- Variance of bagged trees: $$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$$

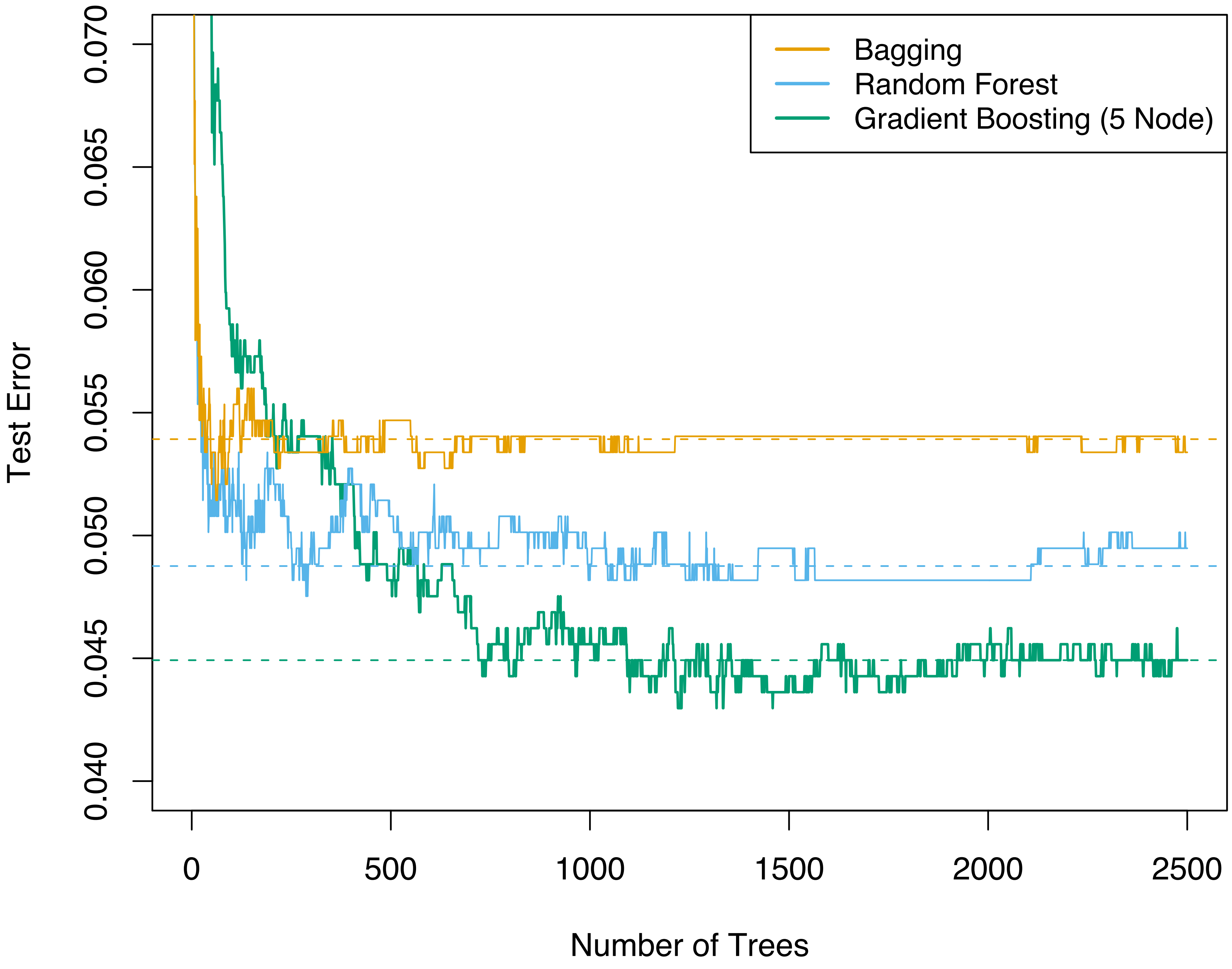
  ρ : correlation among bagged trees

  σ^2 : variance of a base tree

- As m decreases, the correlation decreases, and the 2nd term goes to zero as B goes to ∞.

  - Question: what happen to the prediction error if m is very small ?

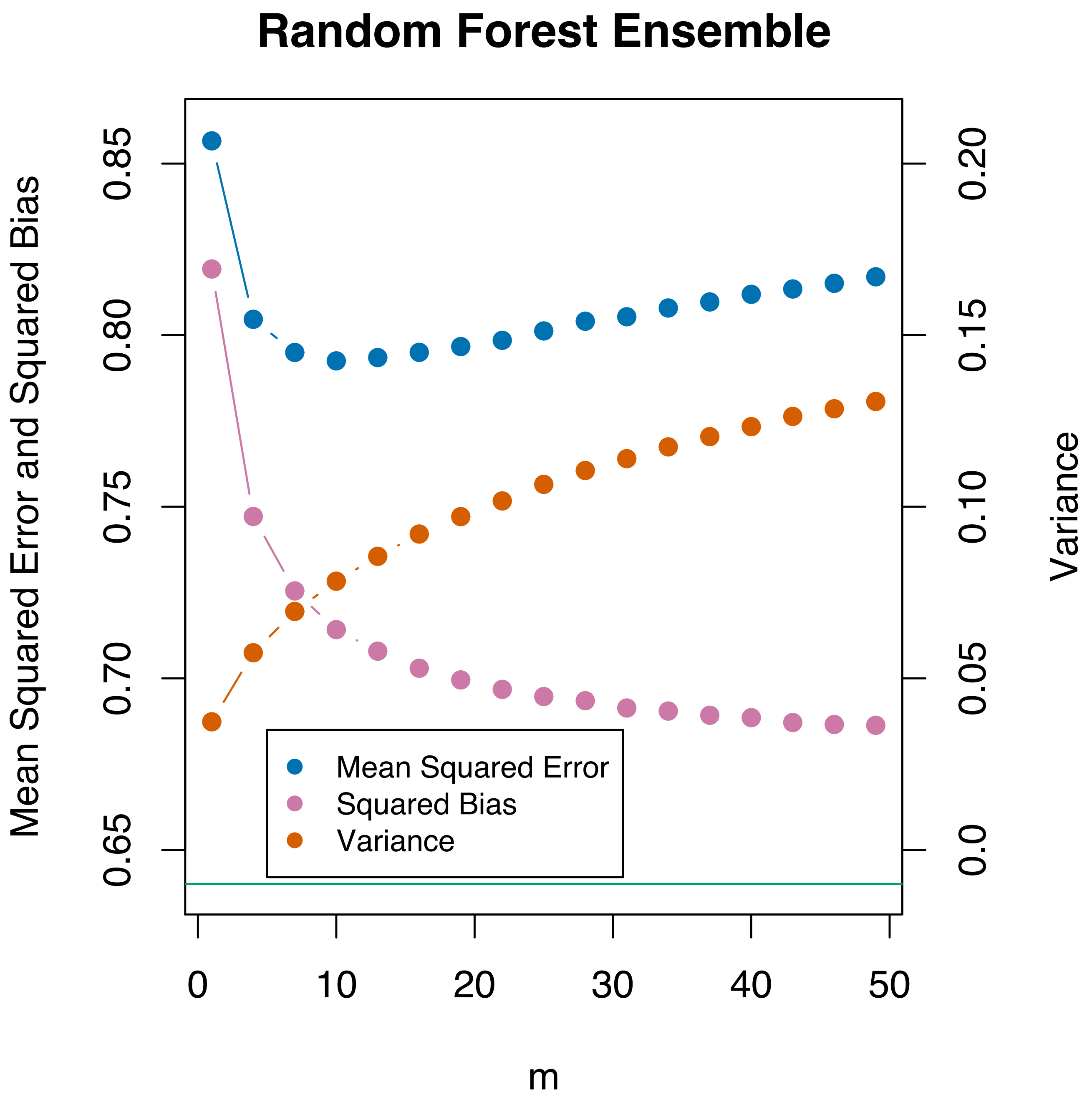
Bagging vs random forests: spam data



- Bagged trees are stabilized at B=250, while at B=750 random forests continues to improve.

More details on random forests (RF)

- How to choose the number of candidate predictors m and stopping rule for recursive binary splitting?
  - Classification: m is normally chosen to be the square root of p and the minimum node size is one.
  - Regression: m is normally chosen to be p/3 and the minimum node size is five.
- The best value for m depend on the underlying problem, and it is treated as a tuning parameter.



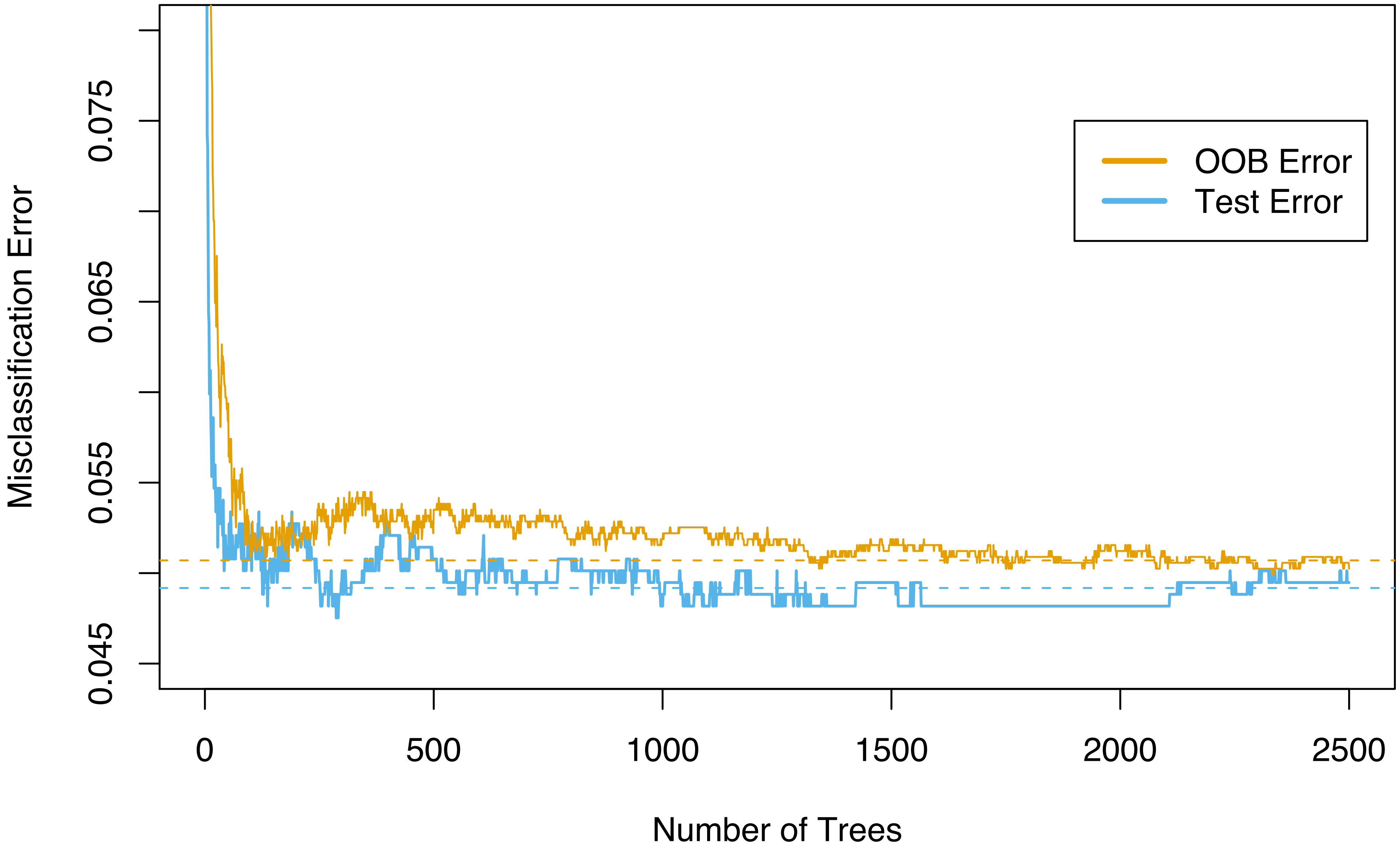
- Bias-variance tradeoff in the choice of m.

- As m decreases, the bias increases, and the variance decreases (de-correlation effect!).

- Green horizontal line is the square bias of a single tree.

More details on RF

- How long should you train the RF? (ie. how to choose B?)
  - An OOB error estimate is almost identical to that obtained by leave-one-out cross validation.
  - 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 (the same stopping rule can be applied for bagged trees).



- OOB error computed on the spam training data,

- OOB error is stabilized at 1500 trees, so that averaging 1500 trees is sufficient, and we can stop training the RF at B=1500.

Pros and cons of RF

- Random forests have smaller prediction variance and therefore usually a better general performance
  - Easy to tune parameters
  - Can model nonlinear class boundaries
  - OOB error “for free” (no computation for CV)
  - Rather slow / black box (hard to get insights)