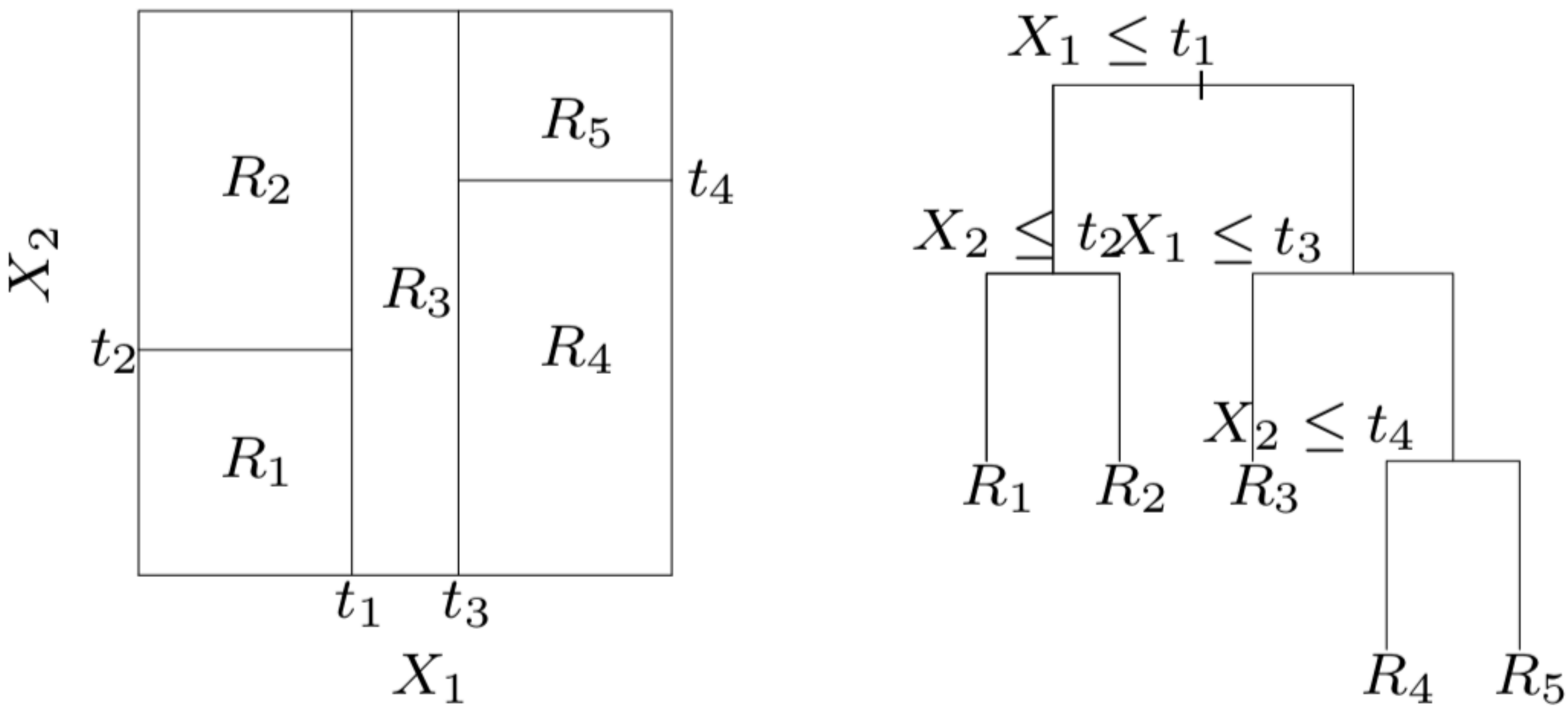
Outline

- Review: DT
- Bagging
- Random Forests

Decision trees, review



Decision Trees, review

Pros

- Popular - highly interpretable.
- Model-free (don’t assume an underlying distribution).
- Fast (well, super fast!)
- Suitable for both regression and classification problems.

Cons

Prediction “accuracy” isn’t that great - inherently high variance

Decision Trees, review

Pros

- Popular - highly interpretable.
- Model-free (don’t assume an underlying distribution).
- Fast (well, super fast!)
- Suitable for both regression and classification problems.

Low Bias

Cons

Prediction “accuracy” isn’t that great - inherently high variance

High Variance

Ensemble methods

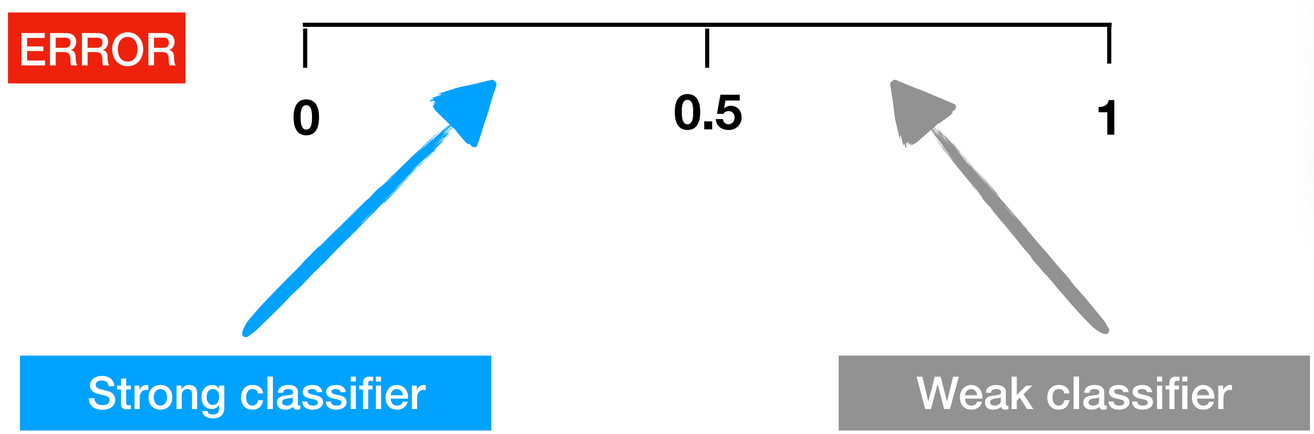
Dietterich (1999) and (2000)

- Bagging — *Breiman, 1996*
- Random Forests — *Breiman, 1996*, ***2001***

Ensemble methods

Dietterich (1999) and (2000)

- Bagging — *Breiman, 1994*
- Random Forests — *Breiman, 1996, **2001***



We can understand the *bagging* effect in terms of a *consensus* of **independent** weak learners!

see also “Wisdom of Crowds” (Surowiecki, 2004)

Outline

- Review: DT
- **Bagging**
- Random Forests

Bagging

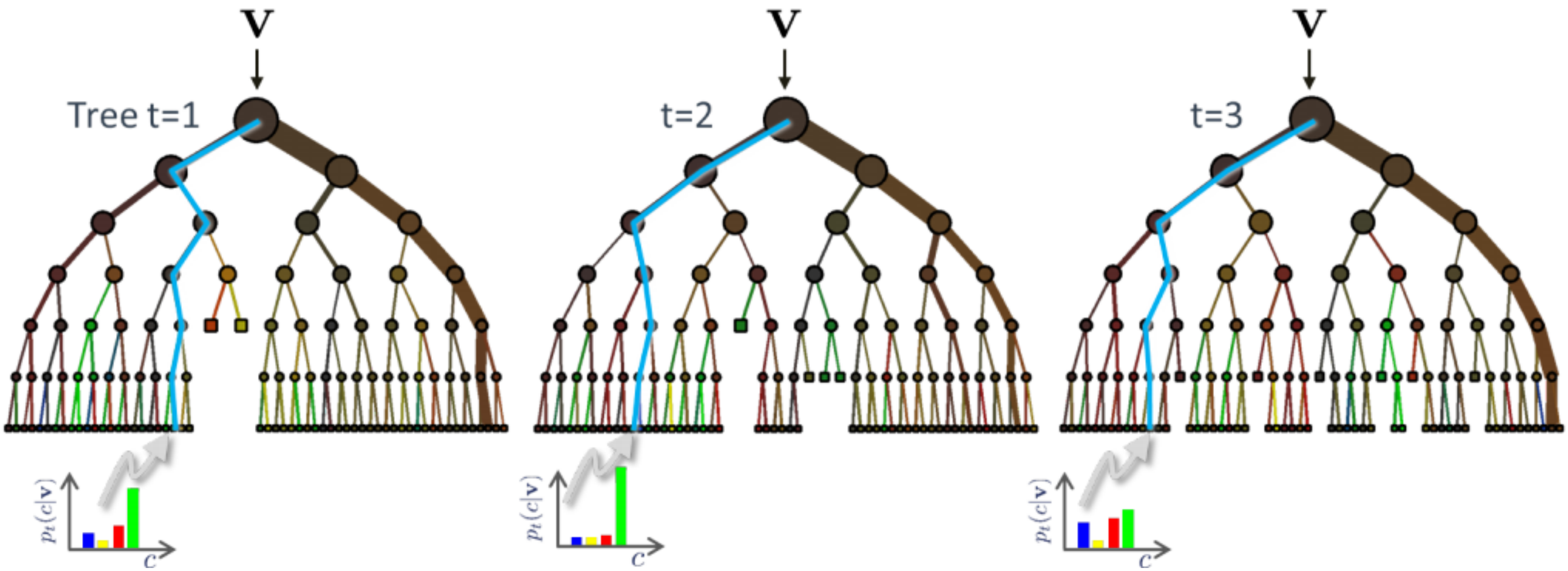
Breiman, 1994,1996

➡ Bootstrap Aggregating; averages predictions over collection of bootstrap samples.

- creates B bootstrap replicates
- fits model to each replicate
- companies predictions via averaging or voting

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

Bagging, schematic view



Classification: Majority vote

Regression: Average

Example: Bagging

Simulated data with n=30, two classes, and 5 features

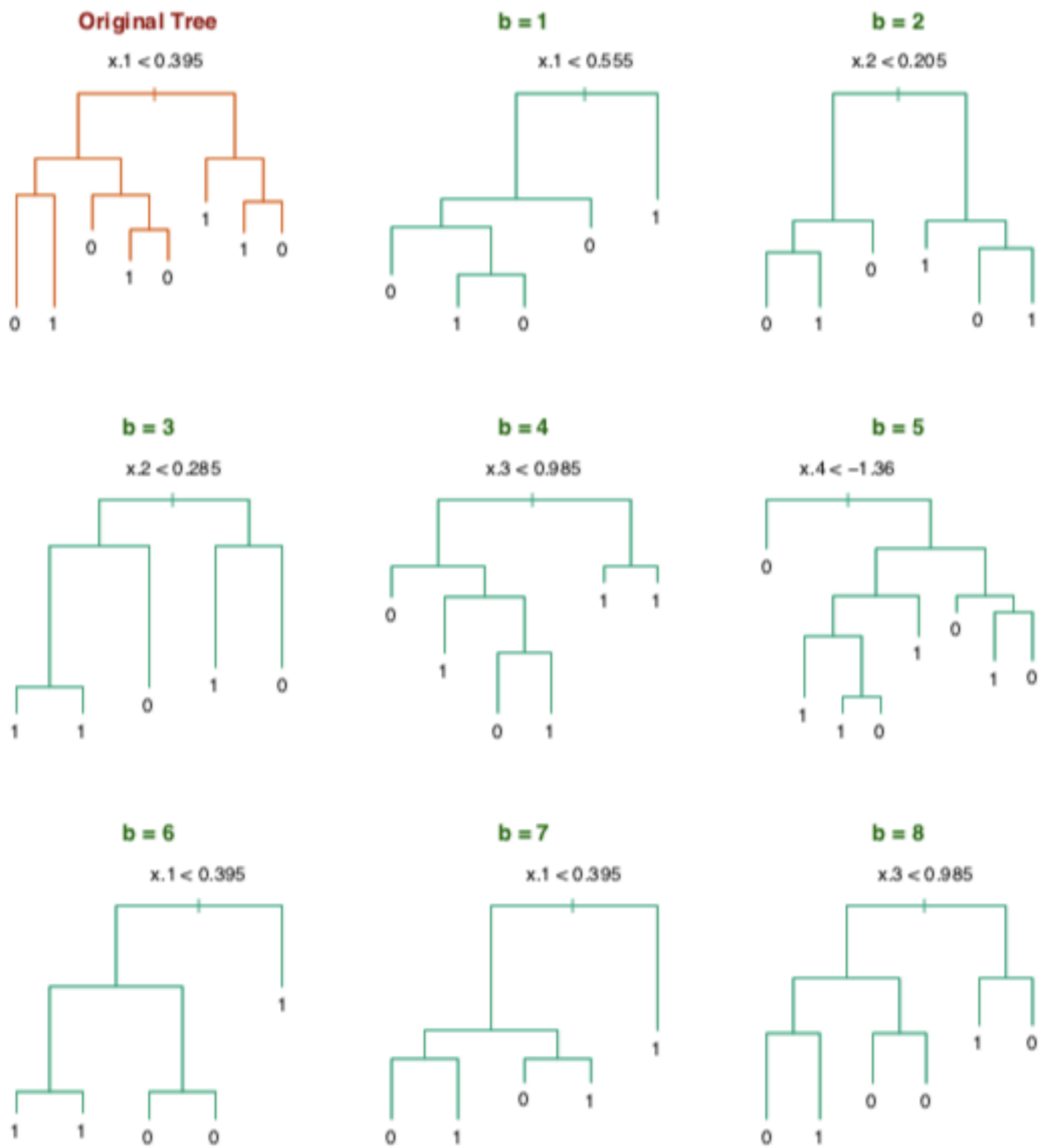
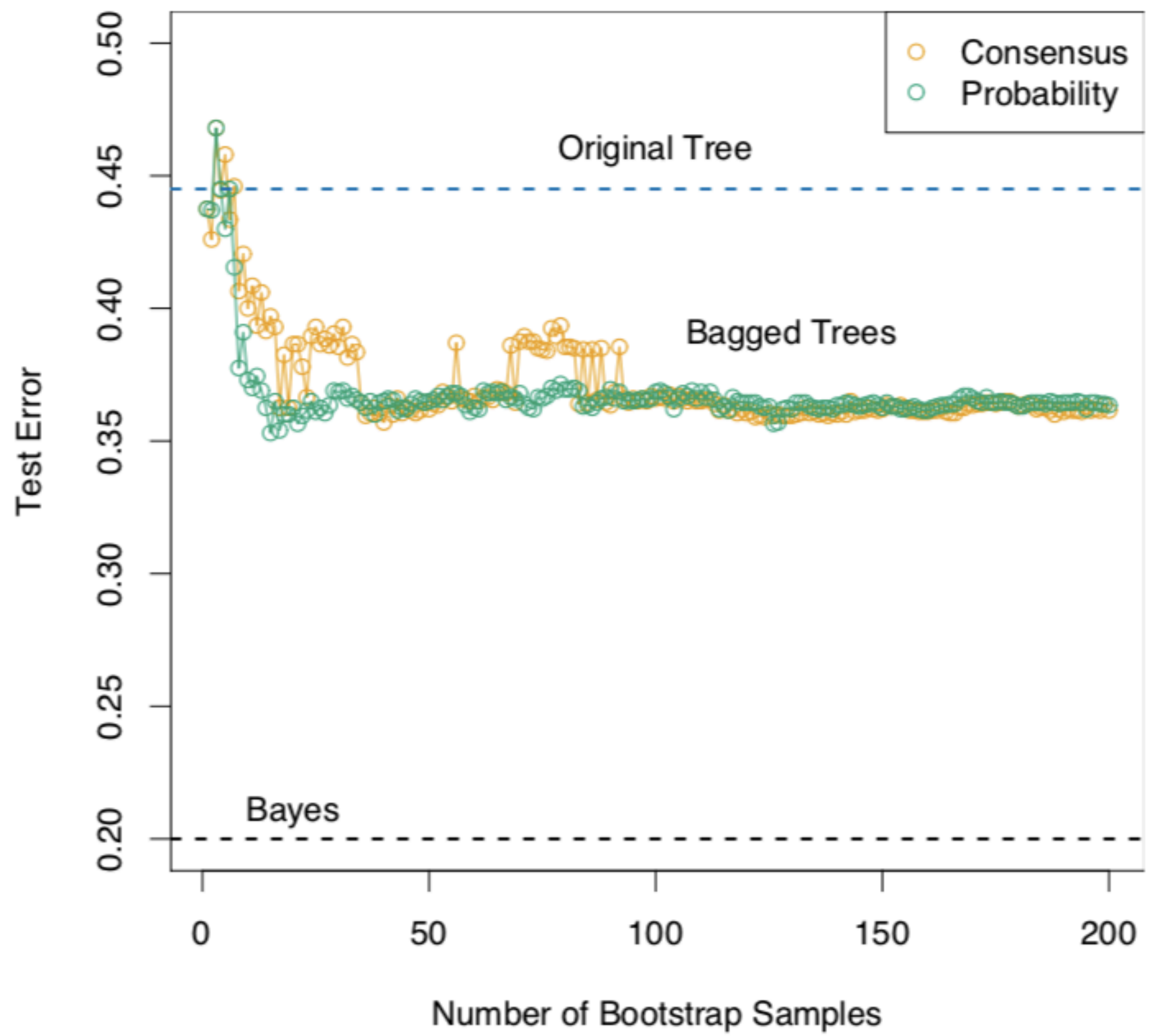


Figure 8.9 (Hastie et al.)

Bagging performance



Bagging helps decrease the misclassification rate of the classifier (evaluated on large independent test set)

Figure 8.10 (Hastie et al.)

Bagging properties

Pros

- Stabilises unstable procedures (models)
- Easily parallizable
- Fast (well, super fast!)
- Each tree grown in bagging is **i.i.d** — expectation of average is same as expectation of one of them

Cons

- Loss of interpretability
- Computational complexity

Bagging properties

Pros

- Stabilises unstable procedures (models)
- Easily parallizable
- Fast (well, super fast!)
- Each tree grown in bagging is **i.i.d** — expectation of average is same as expectation of one of them

Cons

- Loss of interpretability
- Computational complexity

Bagging issue(s)!

- An average of B **i.i.d**. random variables, each with variance $\boldsymbol{\sigma^2}$, has variance: $\boldsymbol{\sigma^2/B}$

- If **i.d.** (identical but not independent) and pair correlation ρ is present, then the variance is:

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

As ***B*** increases the second term disappears but the first term remains

Does bagging generate correlated trees?

Size of the correlation of bagged trees *limits benefits of averaging* —> reduce correlation between trees without increasing variance too much!

Outline

- Review: DT
- Bagging
- **Random Forests**

Random Forests (Brieman 2001)

- A substantial modification of bagging that builds a large collection of *de-correlated trees*, and then averages them.

  ➡ a bagged classifier using decision trees,

  ➡ each split only considers a random group of features,

  > *Before each split, select* $\boldsymbol{m \leq p}$ *of the input variables at random as candidates for splitting.*

  ➡ tree is grown to maximum size without pruning,

  ➡ final predictions obtained by aggregating over the *B* trees,

$$\hat{f}_{\mathrm{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b).$$

- Θ_b characterizes the ***b***th random forest tree in terms of split variables, cut-points at each node, and terminal-node values.

RF: Algorithm

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

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

RF Performance

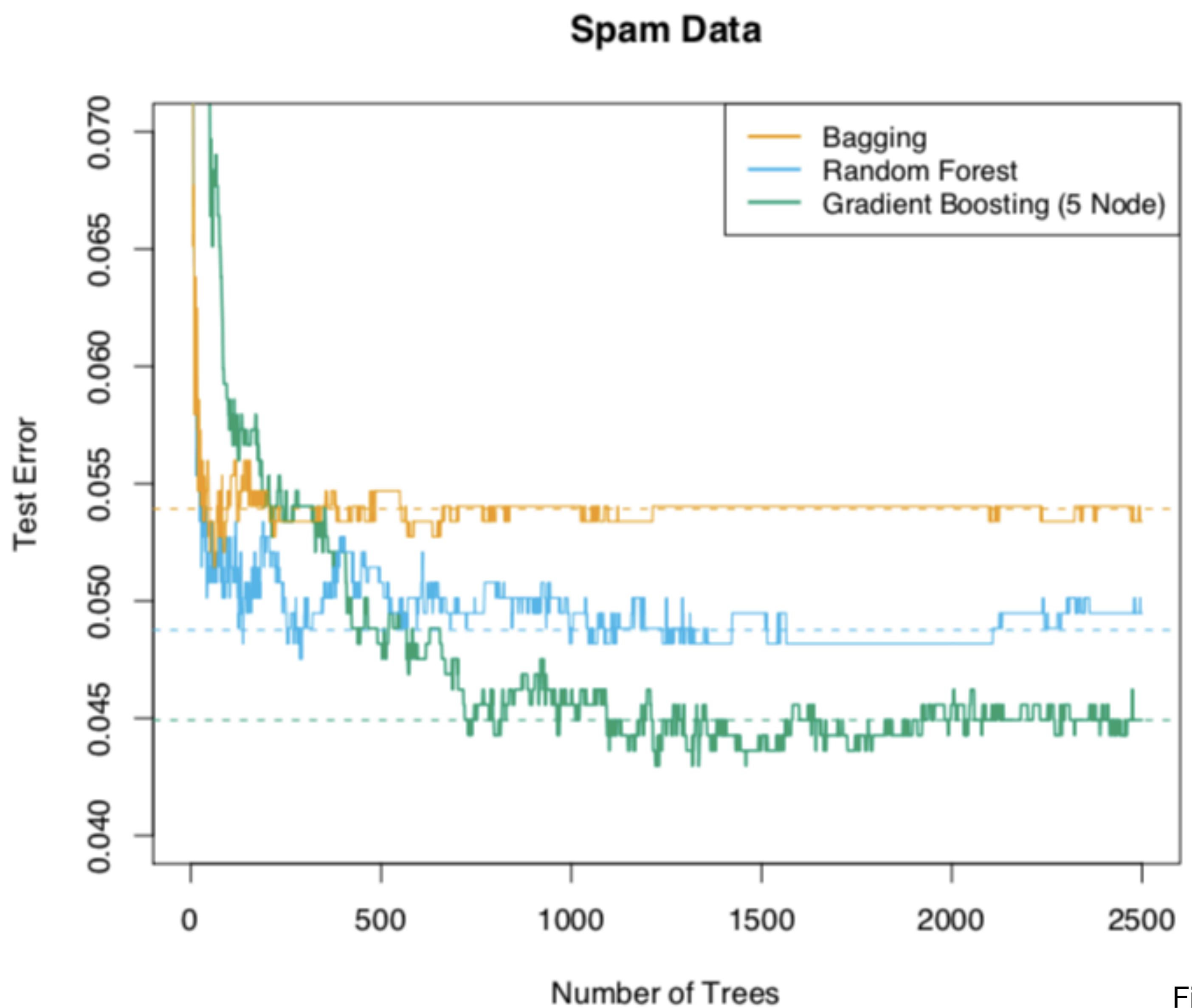


Figure 15.1 (Hastie et al.)

RF: Parameters and details

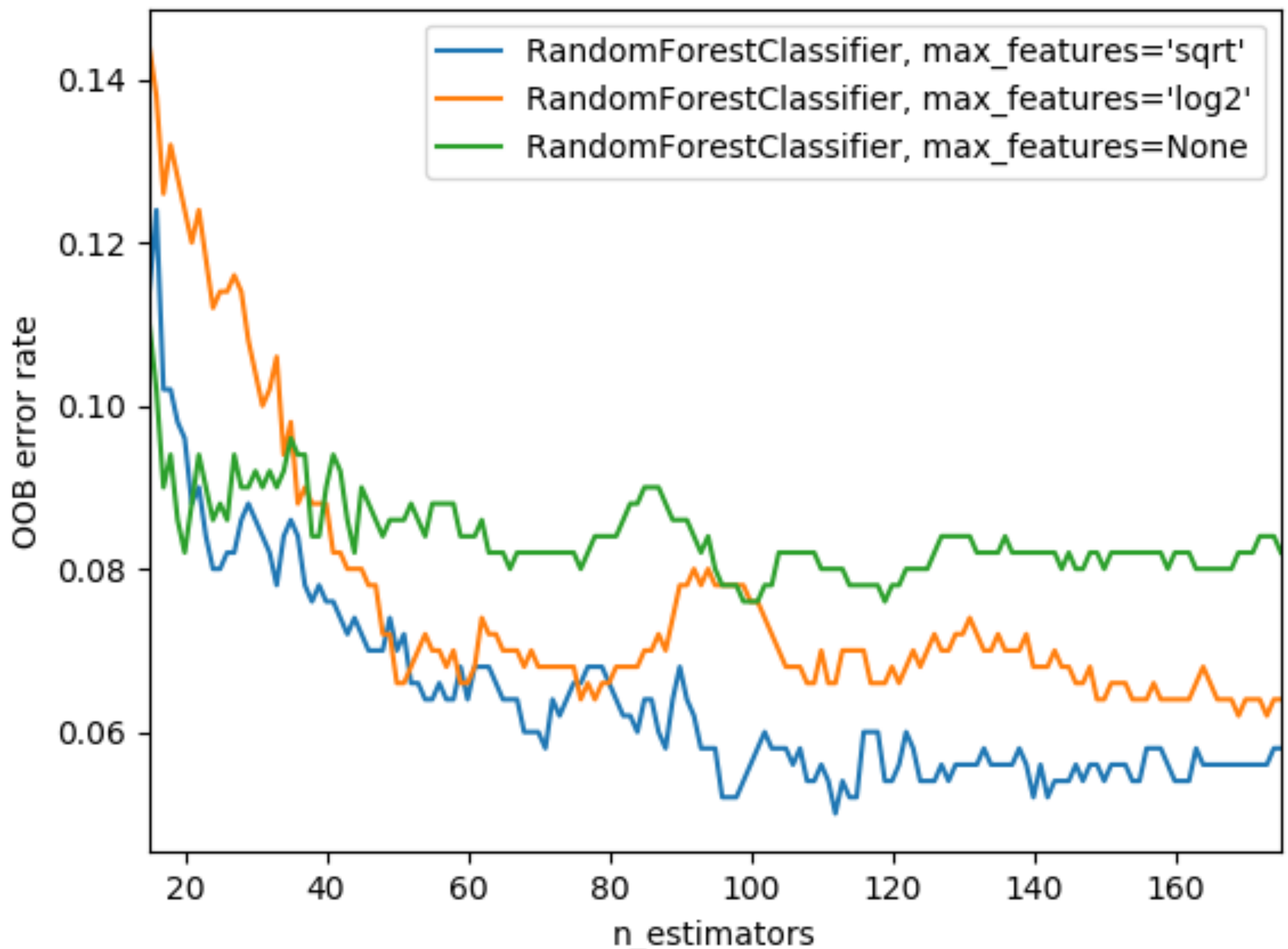
- n_estimators
- node size
- *m <=p (number of features)*
    - For classification, the default value for ***m* is** $\sqrt{p}$ and the minimum node size is one.
    - For regression, the default value for m is ***p/3*** and the minimum node size is five.

OOB: Out of Bag Samples

No cross validation?

- Out-of-bag samples (**OOB**)?
- For each observation, construct its random forest predictor by averaging only those trees corresponding to bootstrap samples in which observation does not appear.
- OOB estimates almost identical to N-Fold cross-validation.
- Once OOB stabilises, training can be stopped.

OOB Error



T. Hastie, R. Tibshirani and J. Friedman, “Elements of Statistical Learning Ed. 2”, p592-593, Springer, 2009.

Variable importance

- For b-th tree, OOB samples are passed down tree and accuracy recorded

- Values for j-th variable are randomly permuted in OOB samples and accuracy again computed

- Decrease in accuracy is used as measure of importance

RF: summary

- State of the art method, generally one of the most accurate general-purpose learners available
- Handles a large number of input variables without overfitting
- Easy to train and tune
- Reduces correlation amongst bagged trees by considering only a subset of variables at each split

RF methods software

Random Forests
Leo Breiman and Adele Cutler

Random Forests(tm) is a trademark of Leo Breiman and Adele Cutler and is licensed exclusively to Salford Systems for the commercial release of the software. Our trademarks also include RF(tm), RandomForests(tm), RandomForest(tm) and Random Forest(tm).

classification/clustering regression survival analysis

NEW graphics

Statistical Methods for Prediction and Understanding.

Phil Cutler

Leo Breiman's and **Adele Cutler** maintain a random forest website where the software is freely available, it is included in every ML/STAT package

http://www.stat.berkeley.edu/~breiman/RandomForests/

sklearn

```
>>> from sklearn.ensemble import BaggingClassifier, RandomForestClassifier
```

References and reading

➡T Hastie, R Tibshirani, J Friedman, “The Elements of Statistical Learning” Sec. 8.7 & Chp. 15
https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf

➡L Breiman “Random Forests”, Machine Learning, 45(1), 5-32,2001 Learning

➡ A Geron, Hands on ML, Ch. 6 and 7 (pp.167–190)

Exercise: Classify handwritten digits using DT, Bagging and RF

- MNIST dataset: 70,000 small images of handwritten digits

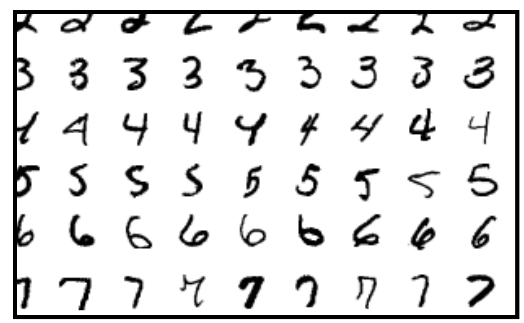
  Modified National Institute of Standards and Technology Database (handwritten by high school students and employees of the US Census Bureau)

- Each digit is 28 x 28 pixels ie, 784 features

```
>>> from sklearn.datasets import fetch_mldata
>>> mnist = fetch_mldata('MNIST original', data_home=custom_data_home)
```

```
X, y = mnist["data"], mnist["target"]
X.shape
```

```
(70000, 784)
```



http://yann.lecun.com/exdb/mnist/

Exercise:
Classify handwritten digits using DT, Bagging and RF

- Compare misclassification rates between the three classifiers.
- Tune both Bagging and RF clf on: number of estimators and minimum node size.
- Tune RF classifier's number of features (m<=p), including that m=p and compare with Bagging results.
- Produce and explain OOB error estimate for both.

documentation

http://scikit-learn.org/stable/modules/ensemble.html#bagging-meta-estimator

http://scikit-learn.org/stable/modules/ensemble.html#random-forests