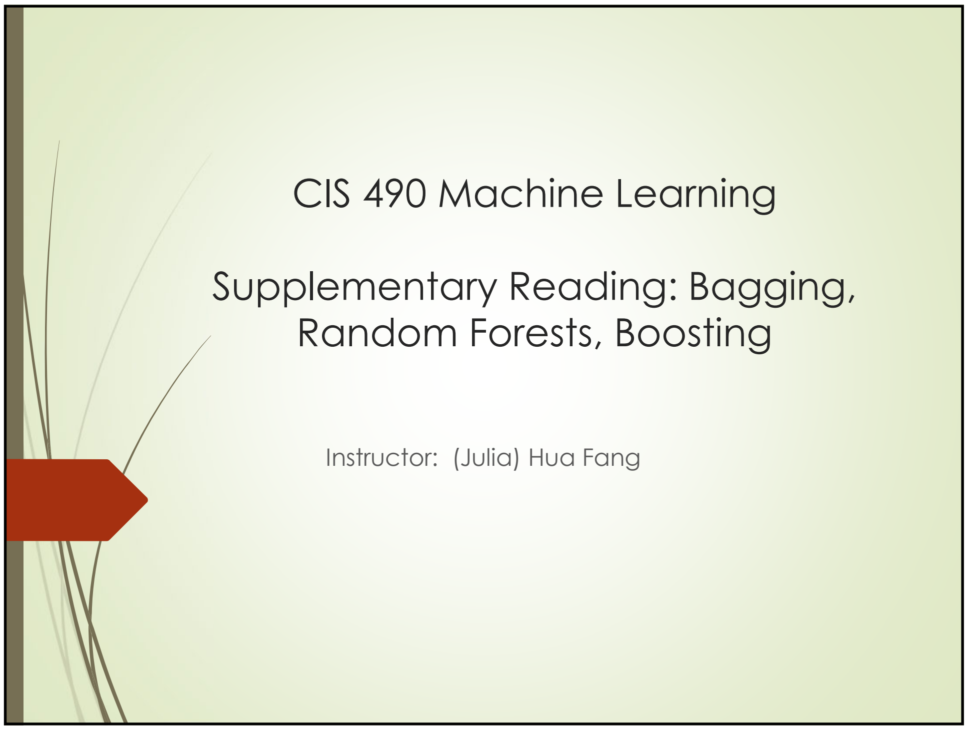


CIS 490 Machine Learning

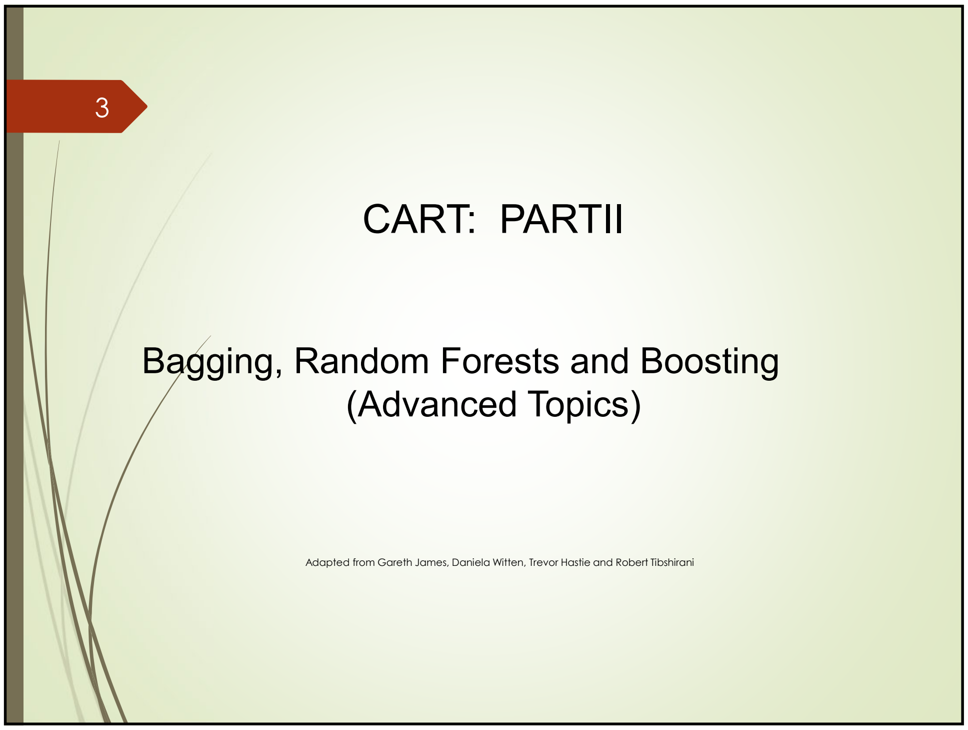
Supplementary Reading: Bagging, Random Forests, Boosting

Instructor: (Julia) Hua Fang

Remedies

However, by aggregating many decision trees, the predictive performance of trees can be substantially improved. We introduce these concepts next.

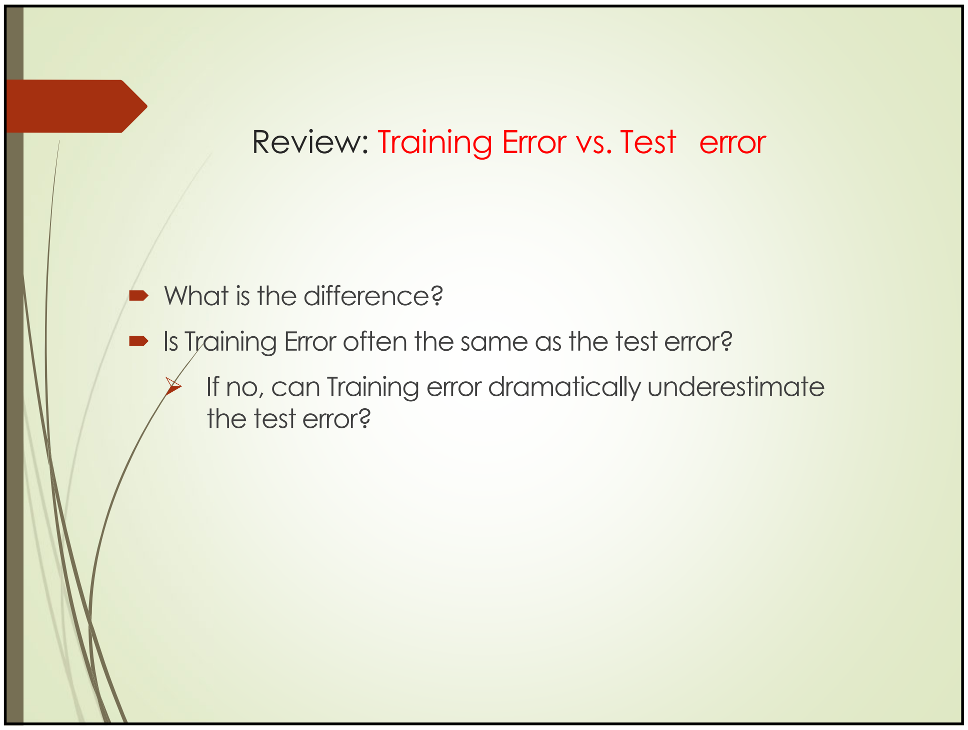
-- Bagging
-- Random Forest
-- Boosting



CART: PARTII

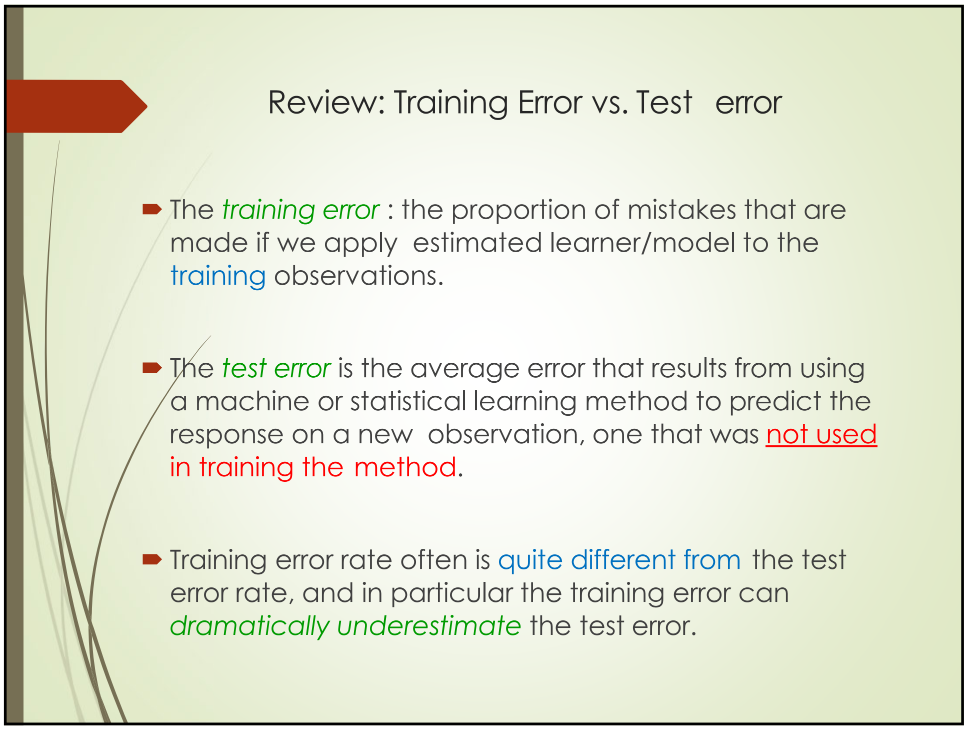
Bagging, Random Forests and Boosting
(Advanced Topics)

Adapted from Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani



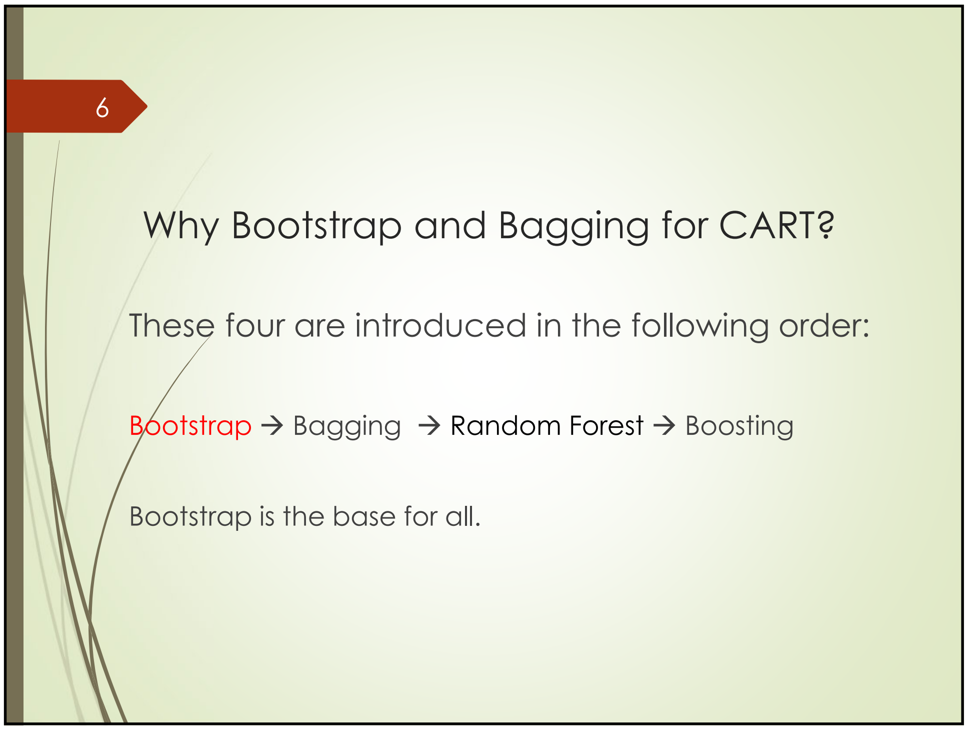
Review: Training Error vs. Test error

- What is the difference?
- Is Training Error often the same as the test error?
  - If no, can Training error dramatically underestimate the test error?



Review: Training Error vs. Test error

- The *training error* : the proportion of mistakes that are made if we apply estimated learner/model to the training observations.
- The *test error* is the average error that results from using a machine or statistical learning method to predict the response on a new observation, one that was not used in training the method.
- Training error rate often is quite different from the test error rate, and in particular the training error can *dramatically underestimate* the test error.



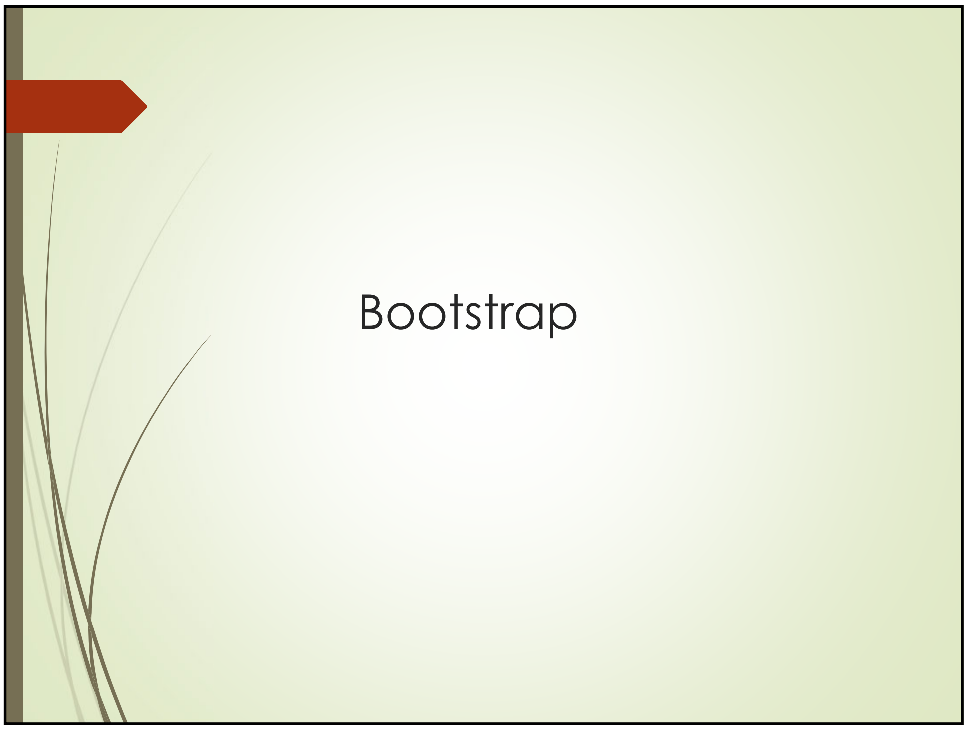
6

Why Bootstrap and Bagging for CART?

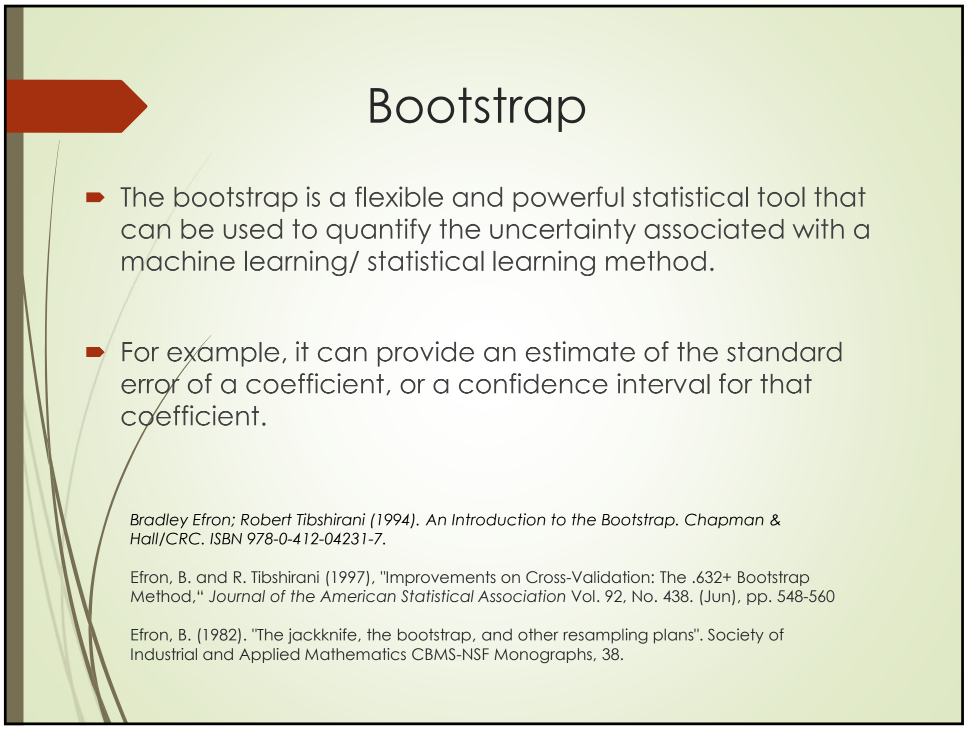
These four are introduced in the following order:

Bootstrap → Bagging → Random Forest → Boosting

Bootstrap is the base for all.



Bootstrap



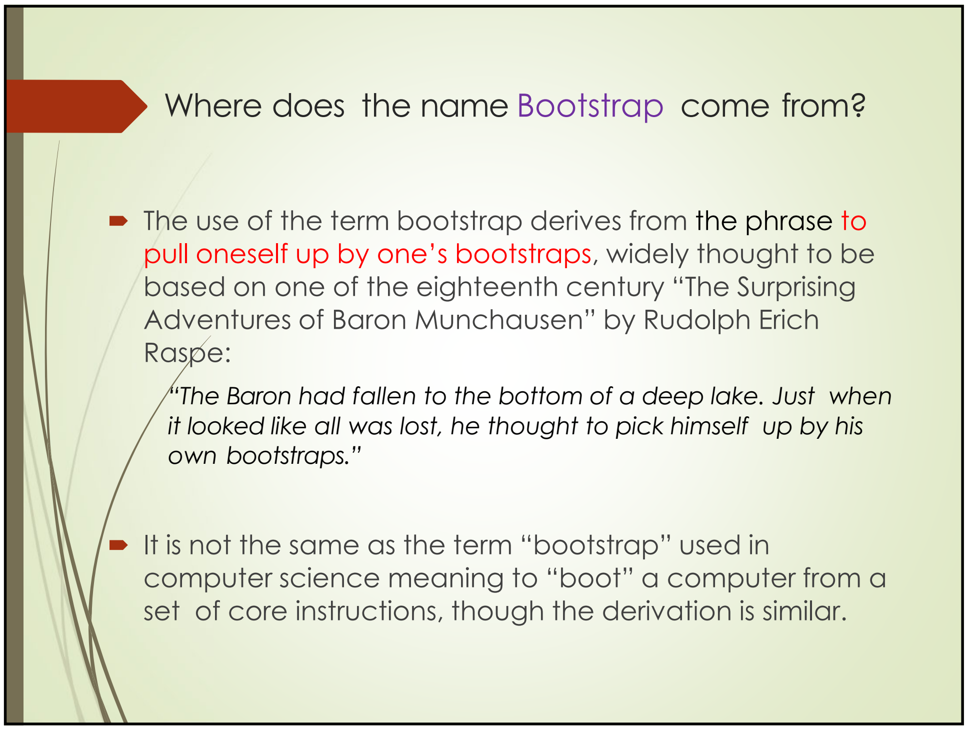
Bootstrap

- The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a machine learning/ statistical learning method.
- For example, it can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.

Bradley Efron; Robert Tibshirani (1994). An Introduction to the Bootstrap. Chapman & Hall/CRC. ISBN 978-0-412-04231-7.

Efron, B. and R. Tibshirani (1997), "Improvements on Cross-Validation: The .632+ Bootstrap Method,“ *Journal of the American Statistical Association* Vol. 92, No. 438. (Jun), pp. 548-560

Efron, B. (1982). "The jackknife, the bootstrap, and other resampling plans". Society of Industrial and Applied Mathematics CBMS-NSF Monographs, 38.

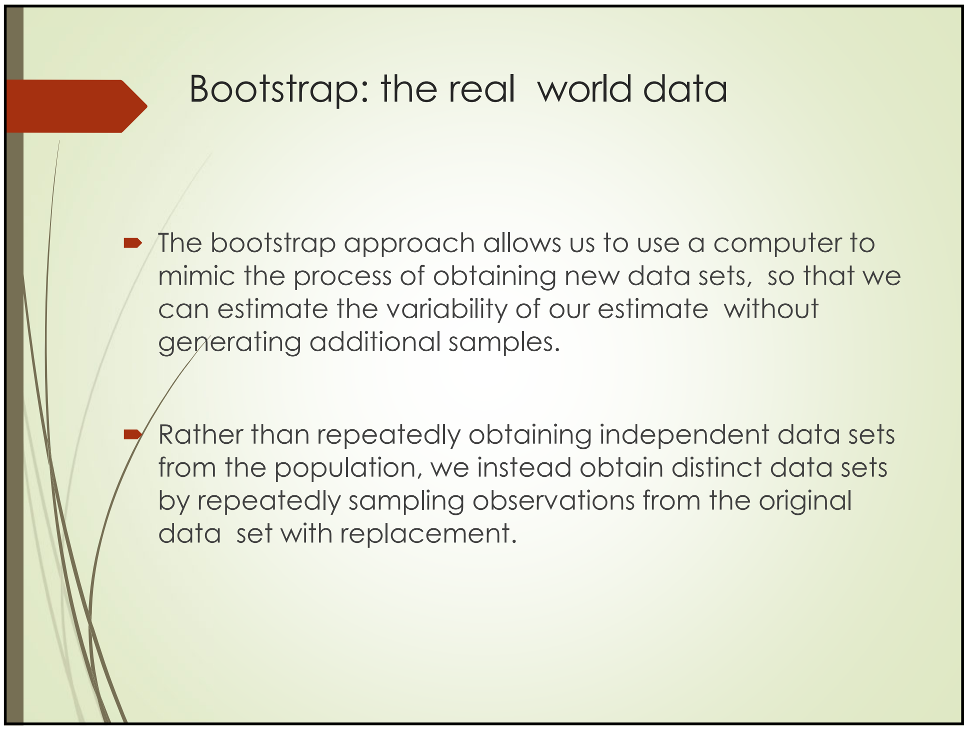


Where does the name Bootstrap come from?

- The use of the term bootstrap derives from the phrase to pull oneself up by one's bootstraps, widely thought to be based on one of the eighteenth century "The Surprising Adventures of Baron Munchausen" by Rudolph Erich Raspe:

  "The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps."

- It is not the same as the term "bootstrap" used in computer science meaning to "boot" a computer from a set of core instructions, though the derivation is similar.



Bootstrap: the real world data

- The bootstrap approach allows us to use a computer to mimic the process of obtaining new data sets, so that we can estimate the variability of our estimate without generating additional samples.

- Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set with replacement.

Bootstrap: the real world data

- Each of these "bootstrap data sets" is created by sampling with replacement, and is the same size as our original dataset.

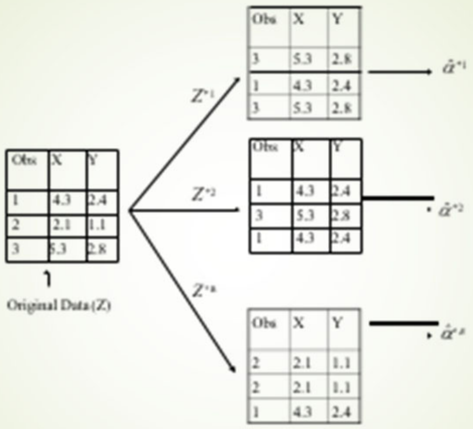
  - As a result some observations may appear more than once in a given bootstrap data set and some not at all.

  Suppl.: *One can show that on average, each bootrapped sample makes use of around two-thirds of the observations.*

  (essentially, to show that $\lim_{n\to\infty}(1-1/n)^n = e^{-1} \approx 1/3$, when n gets larger)

Efron, B. and R. Tibshirani (1997), "Improvements on Cross-Validation: The .632+ Bootstrap Method," *Journal of the American Statistical Association* Vol. 92, No. 438. (Jun), pp. 548-560

Bootstrap: Illustrative Example with just 3 observations



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

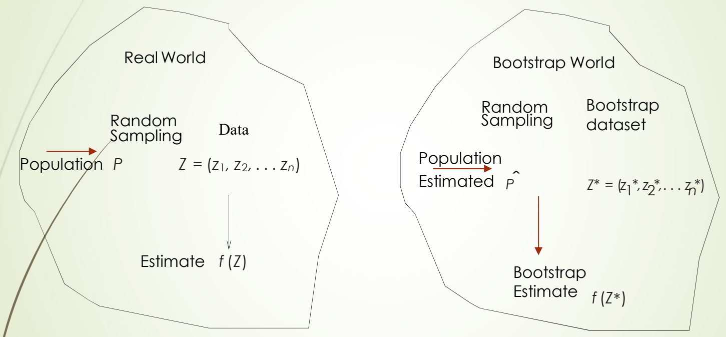
Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
1	4.3	2.4
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

A graphical illustration of the bootstrap approach on a small sample containing n = 3 observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α .

A general picture for the bootstrap

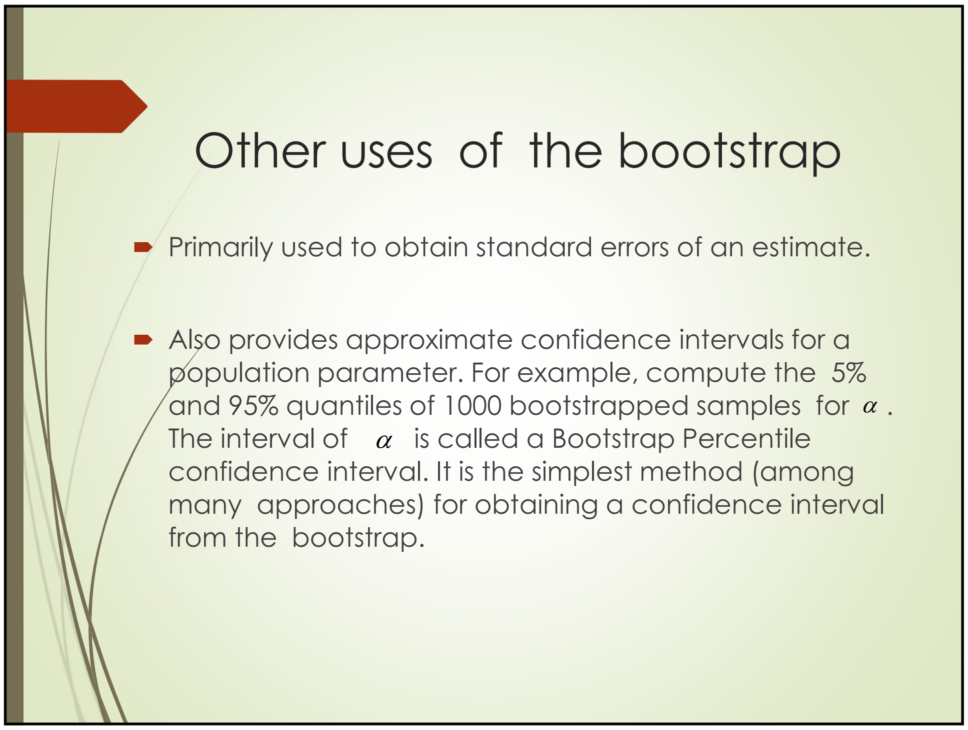


Bootstrap: Specific Steps

- Denoting the first bootstrap data set by **Z*1**, we use **Z*1** to produce a new bootstrap estimate for a, which we call $\hat{\alpha}^{*1}$
- This procedure is repeated **B** times for some large value of **B** (say 100 or 1000), in order to produce B different bootstrap data sets, **Z*1**, **Z*2**, . . . , **Z*B** , and **B** corresponding α estimates, $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}$.
- We estimate the standard error of these bootstrap estimates using the formula

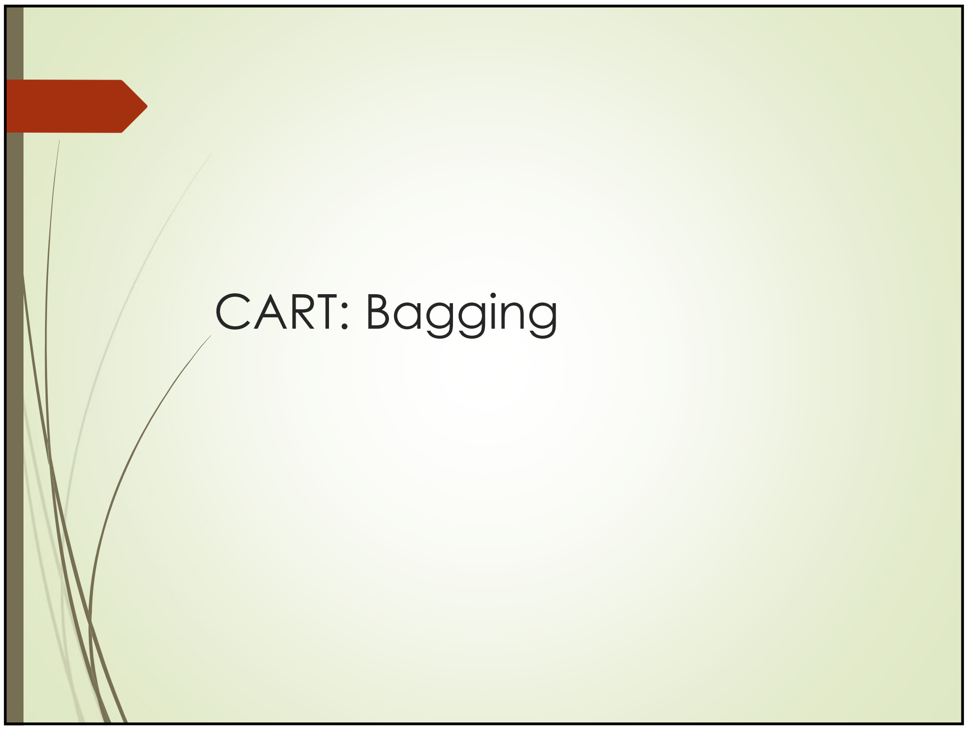
$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \bar{\hat{\alpha}}^{*}\right)^2}.$$

- This serves as an estimate of the standard error of a^ estimated from the original data set.



Other uses of the bootstrap

- Primarily used to obtain standard errors of an estimate.
- Also provides approximate confidence intervals for a population parameter. For example, compute the 5% and 95% quantiles of 1000 bootstrapped samples for α . The interval of α is called a Bootstrap Percentile confidence interval. It is the simplest method (among many approaches) for obtaining a confidence interval from the bootstrap.



CART: Bagging

CART: Bagging

Bagging: Bootstrap **agg**regat**ing** is a method that result in low variance.

Bootstrap aggregation/aggregating, or *bagging*, is a general-purpose procedure for reducing the variance of a machine learning/statistical learning method; we introduce it here because it is particularly useful and frequently used in the context of decision trees.

- Given a set of n independent observations $\mathbf{Z_1, \ldots, Z_n}$, each with variance $\boldsymbol{\sigma^2}$, the variance of the mean $\bar{Z}$ of the observations is given by $\boldsymbol{\sigma^2/n}$.

  ie., averaging a set of observations reduces variance.

  (Note: this is not practical because we generally do not have access to multiple training sets)

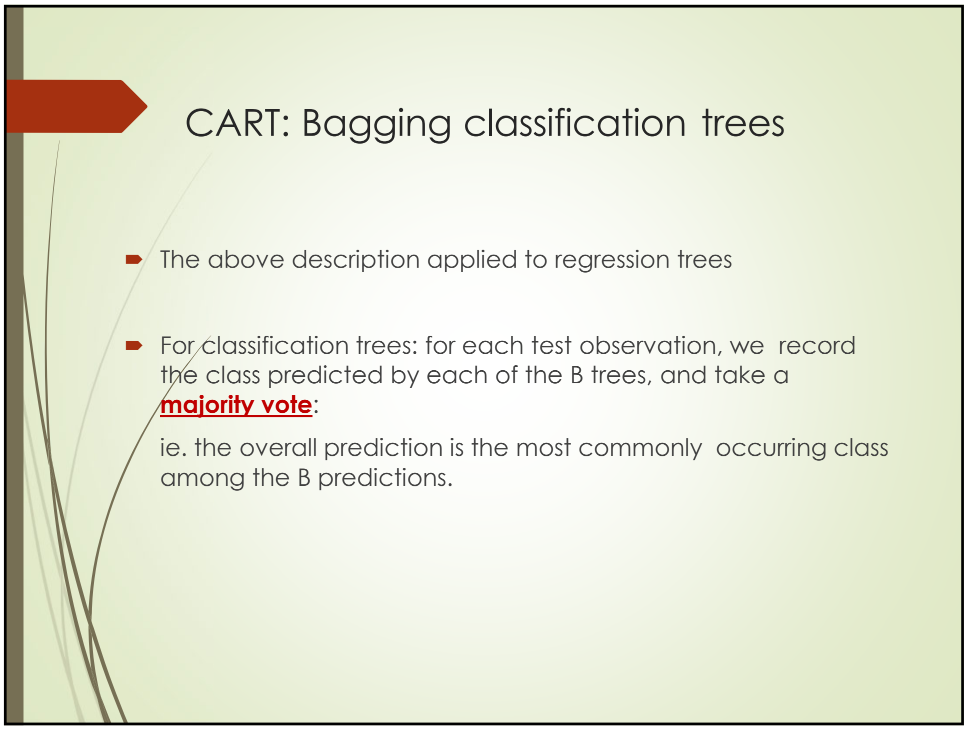
CART: Bagging— continued

- Instead, we can bootstrap, by taking repeated samples from the (single) training data set.
- In this approach we generate B different bootstrapped training data sets. We then train our method on the bth bootstrapped training set in order to get $\hat{f}^{*b}(x)$, the prediction at a point x. We then average all the predictions to obtain

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

This is called *bagging*.

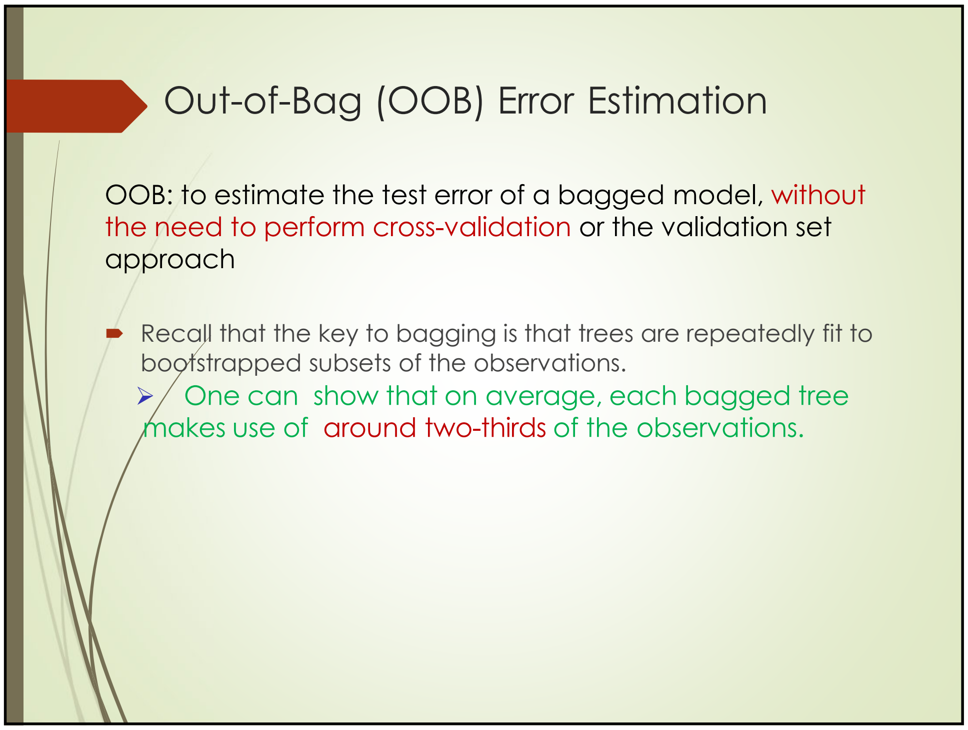
Bootstrap
Construct B (hundreds) of trees (no pruning) .
Learn a classifier for each bootstrap sample and average them



CART: Bagging classification trees

- The above description applied to regression trees
- For classification trees: for each test observation, we record the class predicted by each of the B trees, and take a **majority vote**:

  ie. the overall prediction is the most commonly occurring class among the B predictions.



Out-of-Bag (OOB) Error Estimation

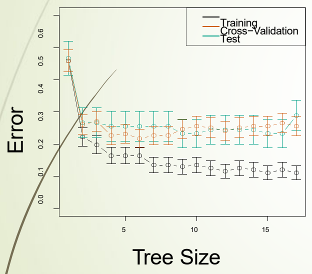
OOB: to estimate the test error of a bagged model, without the need to perform cross-validation or the validation set approach

- 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.

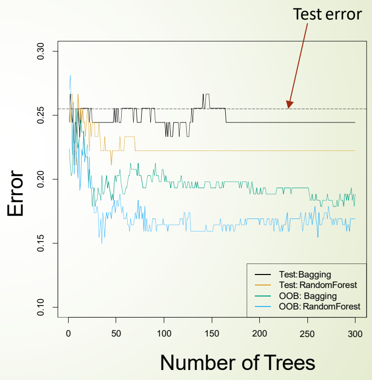
Out-of-Bag (OOB) Error Estimation: cont.

- 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

Bagging: the Heart Disease data

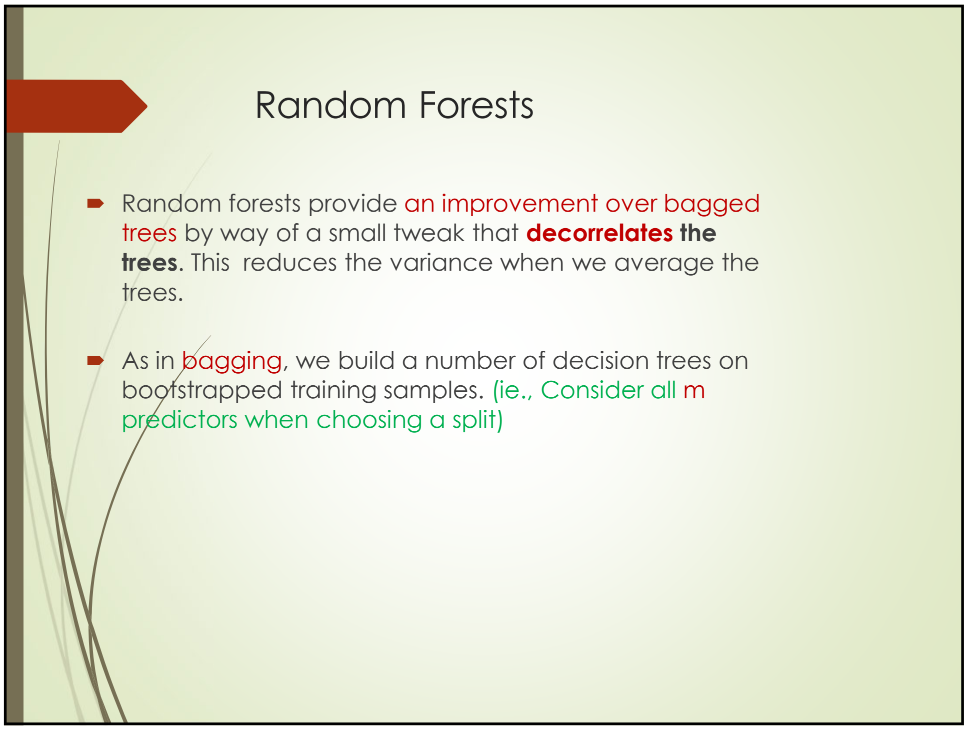


Recall: **Single tree** with pruning



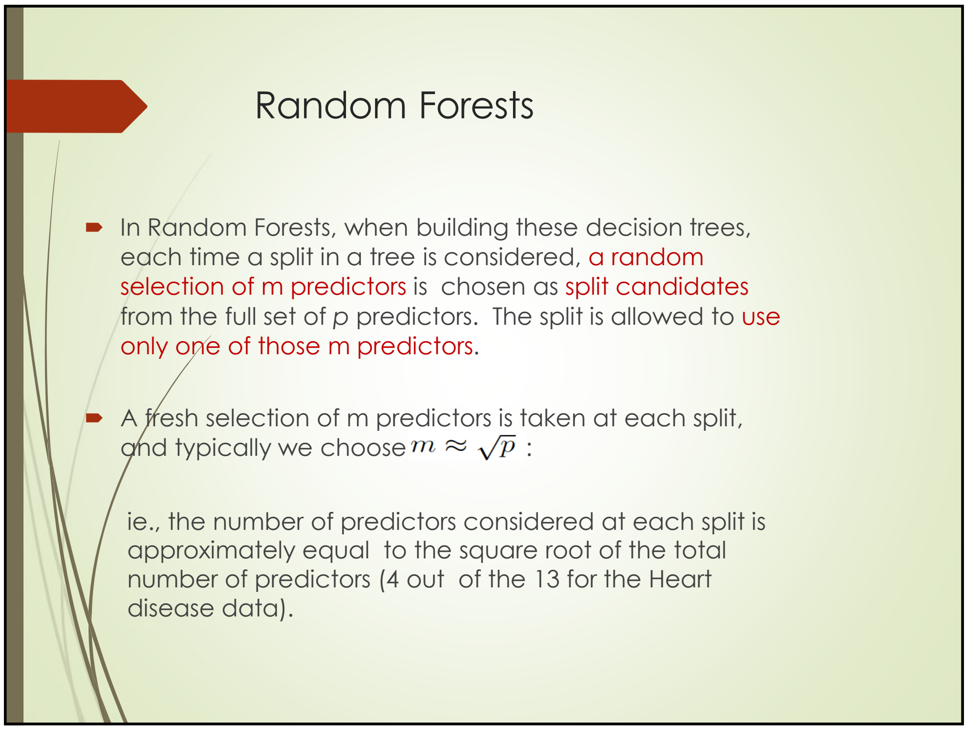
Bagging: up to 300 trees

CART: Random Forests



Random Forests

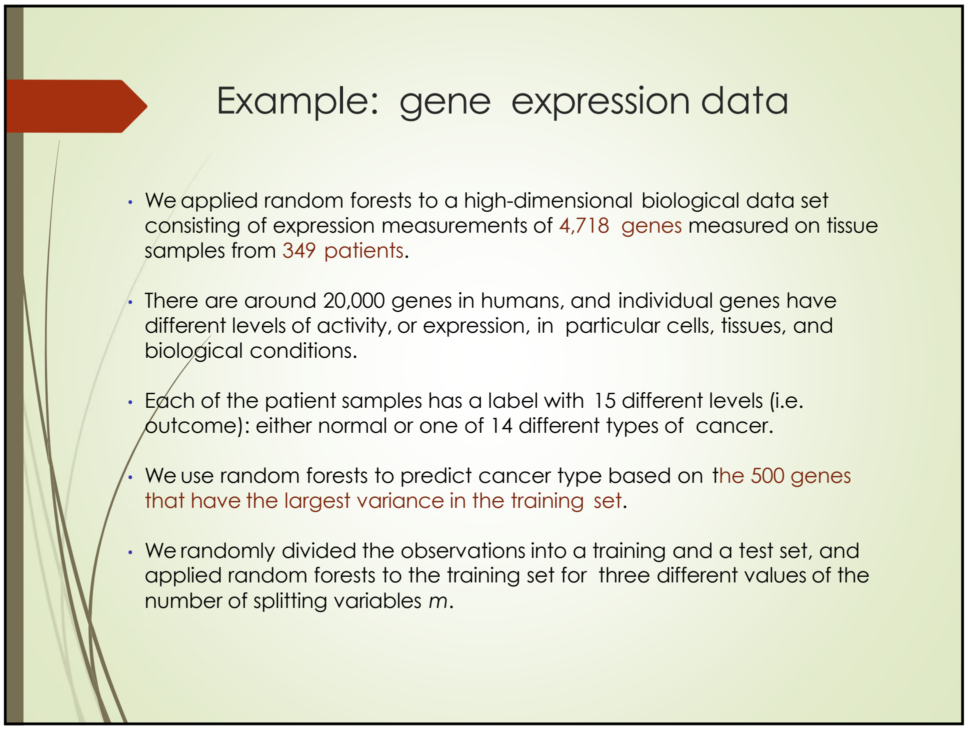
- Random forests provide an improvement over bagged trees by way of a small tweak that **decorrelates the trees**. This reduces the variance when we average the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples. (ie., Consider all m predictors when choosing a split)



Random Forests

- In Random Forests, when building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of *p* predictors. The split is allowed to use only one of those m predictors.
- A fresh selection of m predictors is taken at each split, and typically we choose $m \approx \sqrt{p}$:

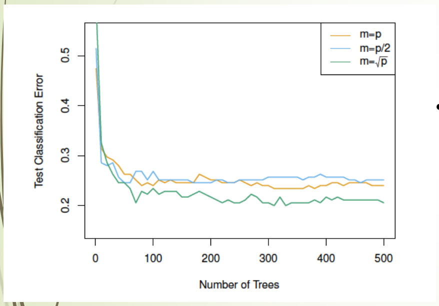
  ie., the number of predictors considered at each split is approximately equal to the square root of the total number of predictors (4 out of the 13 for the Heart disease data).



Example: gene expression data

- We applied random forests to a high-dimensional biological data set consisting of expression measurements of 4,718 genes measured on tissue samples from 349 patients.
- There are around 20,000 genes in humans, and individual genes have different levels of activity, or expression, in particular cells, tissues, and biological conditions.
- Each of the patient samples has a label with 15 different levels (i.e. outcome): either normal or one of 14 different types of cancer.
- We use random forests to predict cancer type based on the 500 genes that have the largest variance in the training set.
- We randomly divided the observations into a training and a test set, and applied random forests to the training set for three different values of the number of splitting variables *m*.

Results: gene expression data

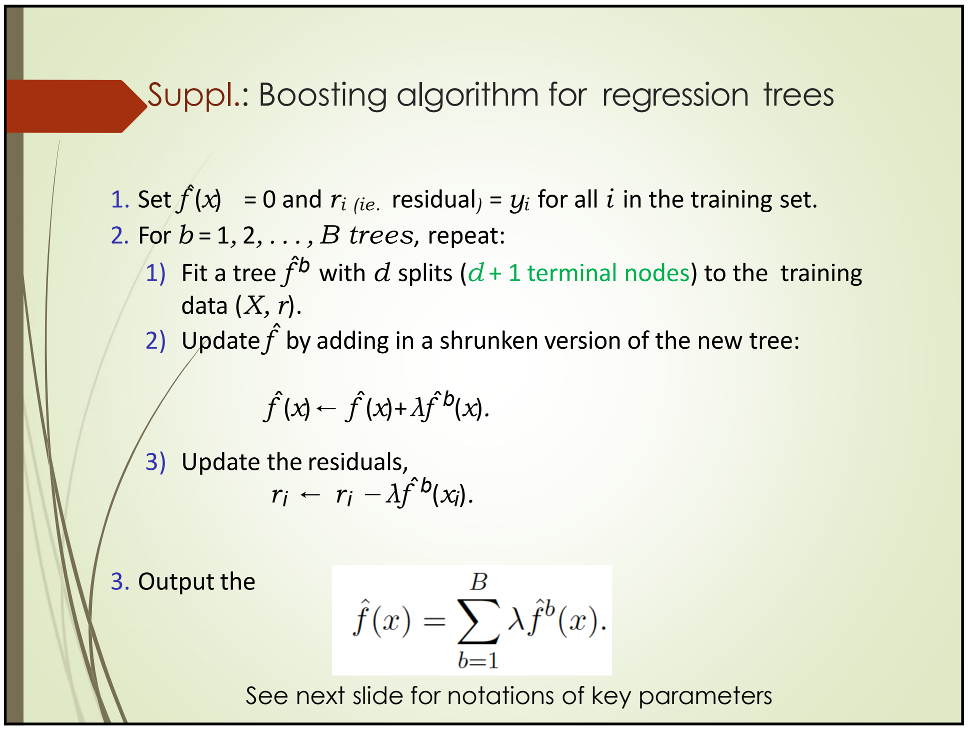


- Results from random forests for the fifteen-class gene expression data set with p= 500 predictors.
- Random forests ($m \approx \sqrt{p}$): a slight improvement over bagging (m = p). A single classification tree has an error rate of 45.7%

Suppl. CART: Boosting

Like bagging, boosting is a general approach that can be applied to many statistical learning methods for regression or classification. We only discuss boosting for CART.

- Recall that bagging involves creating multiple copies of the original training data set using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model.
- Boosting works in a similar way, except that the trees are grown sequentially: each tree is grown using information from previously grown trees.
  - Boosting does not involve bootstrap sampling; instead each tree is fit on a modified version of the original data set



Suppl.: Boosting algorithm for regression trees

1. Set $\hat{f}(x) = 0$ and r_i *(ie.* residual*)* = y_i for all i in the training set.
2. For $b = 1, 2, \ldots, B$ *trees*, repeat:
   1) Fit a tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes) to the training data (X, r).
   2) Update $\hat{f}$ by adding in a shrunken version of the new tree:

      $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$

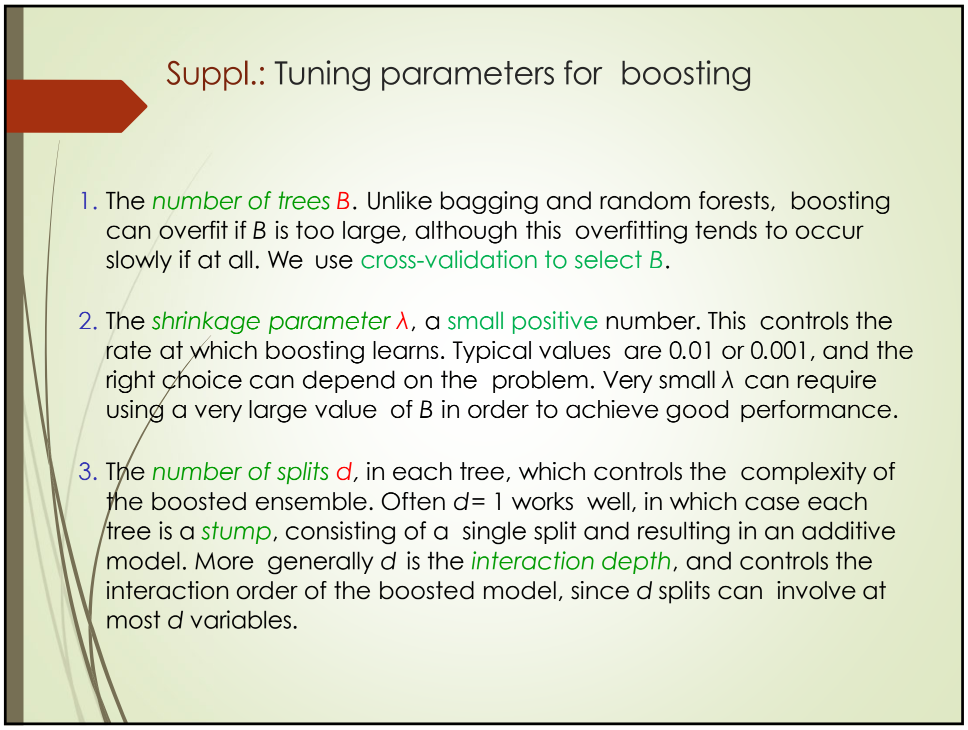
   3) Update the residuals,

      $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$

3. Output the

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

See next slide for notations of key parameters

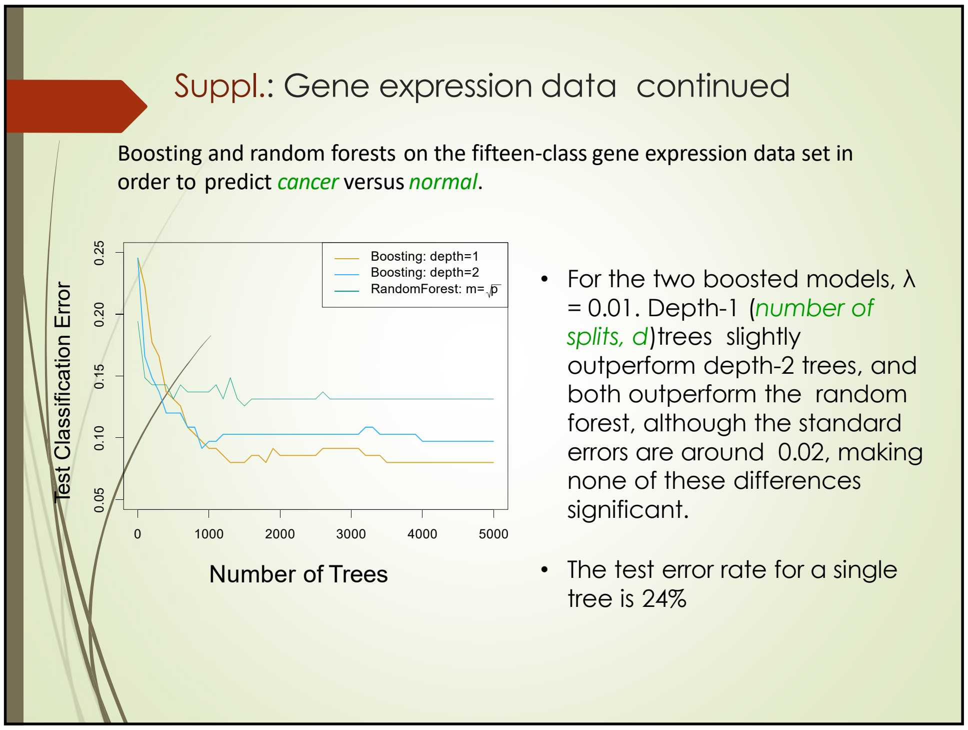


Suppl.: Tuning parameters for boosting

1. The *number of trees B*. Unlike bagging and random forests, boosting can overfit if *B* is too large, although this overfitting tends to occur slowly if at all. We use cross-validation to select *B*.

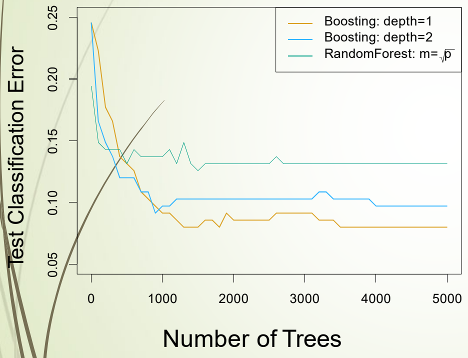
2. The *shrinkage parameter λ*, a small positive number. This controls the rate at which boosting learns. Typical values are 0.01 or 0.001, and the right choice can depend on the problem. Very small *λ* can require using a very large value of *B* in order to achieve good performance.

3. The *number of splits d*, in each tree, which controls the complexity of the boosted ensemble. Often *d*= 1 works well, in which case each tree is a *stump*, consisting of a single split and resulting in an additive model. More generally *d* is the *interaction depth*, and controls the interaction order of the boosted model, since *d* splits can involve at most *d* variables.

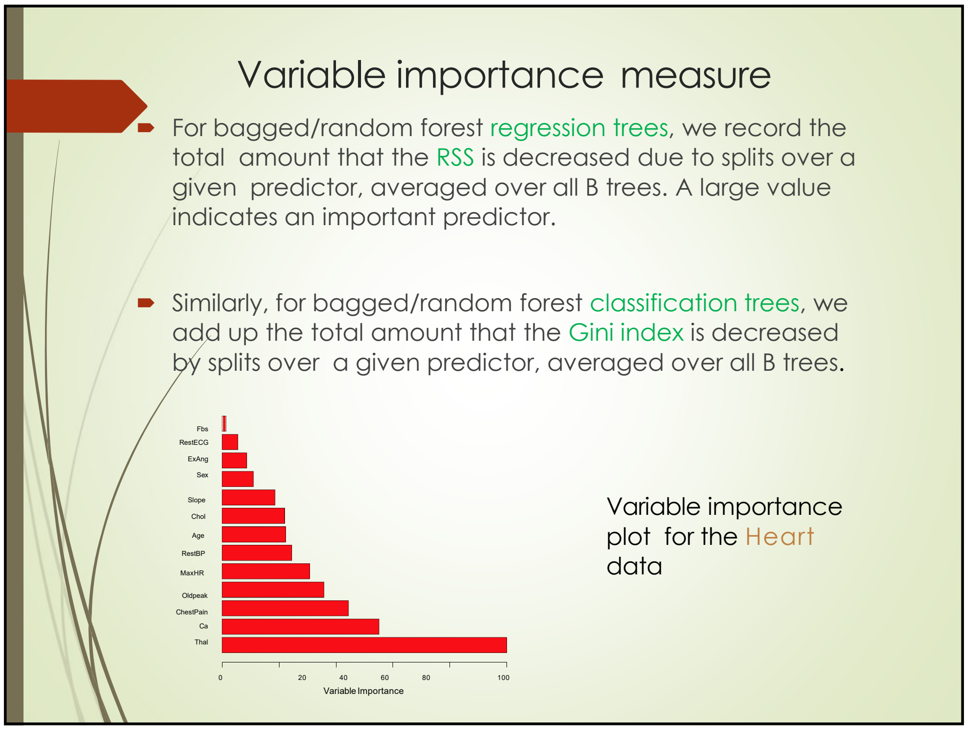


Suppl.: Gene expression data continued

Boosting and random forests on the fifteen-class gene expression data set in order to predict *cancer* versus *normal*.

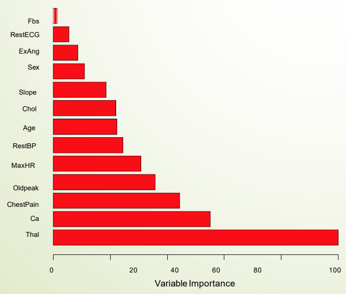


- For the two boosted models, λ = 0.01. Depth-1 (*number of splits, d*)trees slightly outperform depth-2 trees, and both outperform the random forest, although the standard errors are around 0.02, making none of these differences significant.
- The test error rate for a single tree is 24%

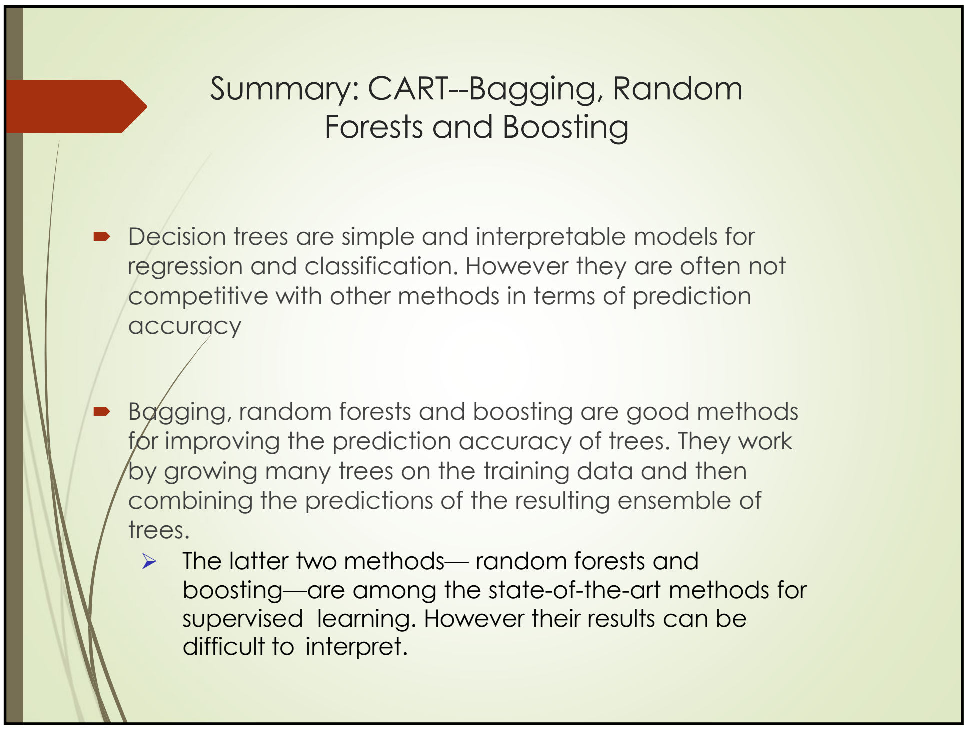


Variable importance measure

- For bagged/random forest regression trees, we record the total amount that the RSS is decreased due to splits over a given predictor, averaged over all B trees. A large value indicates an important predictor.
- Similarly, for bagged/random forest classification trees, we add up the total amount that the Gini index is decreased by splits over a given predictor, averaged over all B trees.

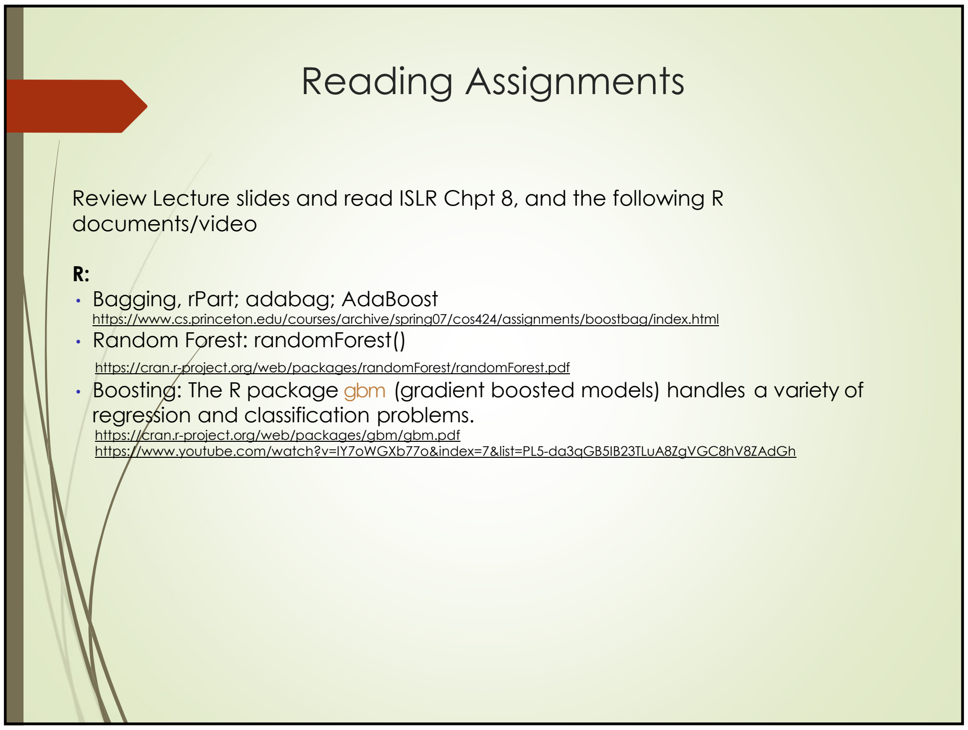


Variable importance plot for the Heart data



Summary: CART--Bagging, Random Forests and Boosting

- Decision trees are simple and interpretable models for regression and classification. However they are often not competitive with other methods in terms of prediction accuracy
- Bagging, random forests and boosting are good methods for improving the prediction accuracy of trees. They work by growing many trees on the training data and then combining the predictions of the resulting ensemble of trees.
  - The latter two methods— random forests and boosting—are among the state-of-the-art methods for supervised learning. However their results can be difficult to interpret.



Reading Assignments

Review Lecture slides and read ISLR Chpt 8, and the following R documents/video

R:

- Bagging, rPart; adabag; AdaBoost
  https://www.cs.princeton.edu/courses/archive/spring07/cos424/assignments/boostbag/index.html
- Random Forest: randomForest()
  https://cran.r-project.org/web/packages/randomForest/randomForest.pdf
- Boosting: The R package gbm (gradient boosted models) handles a variety of regression and classification problems.
  https://cran.r-project.org/web/packages/gbm/gbm.pdf
  https://www.youtube.com/watch?v=IY7oWGXb77o&index=7&list=PL5-da3qGB5IB23TLuA8ZgVGC8hV8ZAdGh