Statistical Machine Learning: Classification

Daniel J. McDonald

Indiana University, Bloomington

mypage.iu.edu/∼dajmcdon

February 24-26, 2015

What is Classification?

Suppose we observe a dataset $\mathcal{D}_n = \{(Y_1, X_1,), \ldots, (Y_n, X_n)\}$.

In regression, we have $Y_i \in \mathbb{R}$ and (generally) use squared error loss

However, sometimes Y_i takes only a few possible values

In this case, we will use zero-one loss instead:

$$R_n(\widehat{f}) = \mathbb{E}[I(Y \neq \widehat{f}(X))]$$

Now, we want to label or classify a new observation (Y, X) such that $\widehat{f}(X) = Y$ as often as possible.

CLASSIFICATION EXAMPLES

- A person arrives at an emergency room with a set of symptoms that could be 1 of 3 possible conditions. Which one is it?
- A online banking service must be able to determine whether each transaction is fraudulent or not, using a customer's location, past transaction history, etc.
- Given a set of individuals' sequenced DNA, can we determine whether various mutations are associated with different phenotypes?
- Take state and national level economic variables from 1960 to the present and predict when the next recession will occur.

CLASSIFICATION EXAMPLES

- A person arrives at an emergency room with a set of symptoms that could be 1 of 3 possible conditions. Which one is it?
- A online banking service must be able to determine whether each transaction is fraudulent or not, using a customer's location, past transaction history, etc.
- Given a set of individuals' sequenced DNA, can we determine whether various mutations are associated with different phenotypes?
- Take state and national level economic variables from 1960 to the present and predict when the next recession will occur.

THE SET-UP

Given $\mathcal{D}_n$, we want to estimate a function $\widehat{f}$ that maps X into Y

The same constraints apply as in the regression scenario.

We want to find an $\widehat{f}$ that makes $R_n(\widehat{f})$ small.

This means:

- We want a classifier that predicts test data, not just the training data.
- Often, this comes with the introduction of some bias to get lower variance and better predictions.

An Introductory Example

Use macroeconomic data to predict recessions

"Nowcasting" and forecasting

Use handful of national-level variables – Federal Funds Rate, Term Spread, Industrial Production, Payroll Employment, S&P500

Also include state-level Payroll Employment

Do state-level variables help?

In this example, we code $Y = 1$ as a recession and $Y = 0$ as growth.

Source: See Owyang, Piger, Wall (2012)

Generalized Linear Models (GLMs)

You have all seen an example of a classifier: Logistic regression.

Logistic regression models probabilities (more specifically, log-odds)

That is, if $\pi(X_i) = Pr(Y_i = 1|X_i)$, then

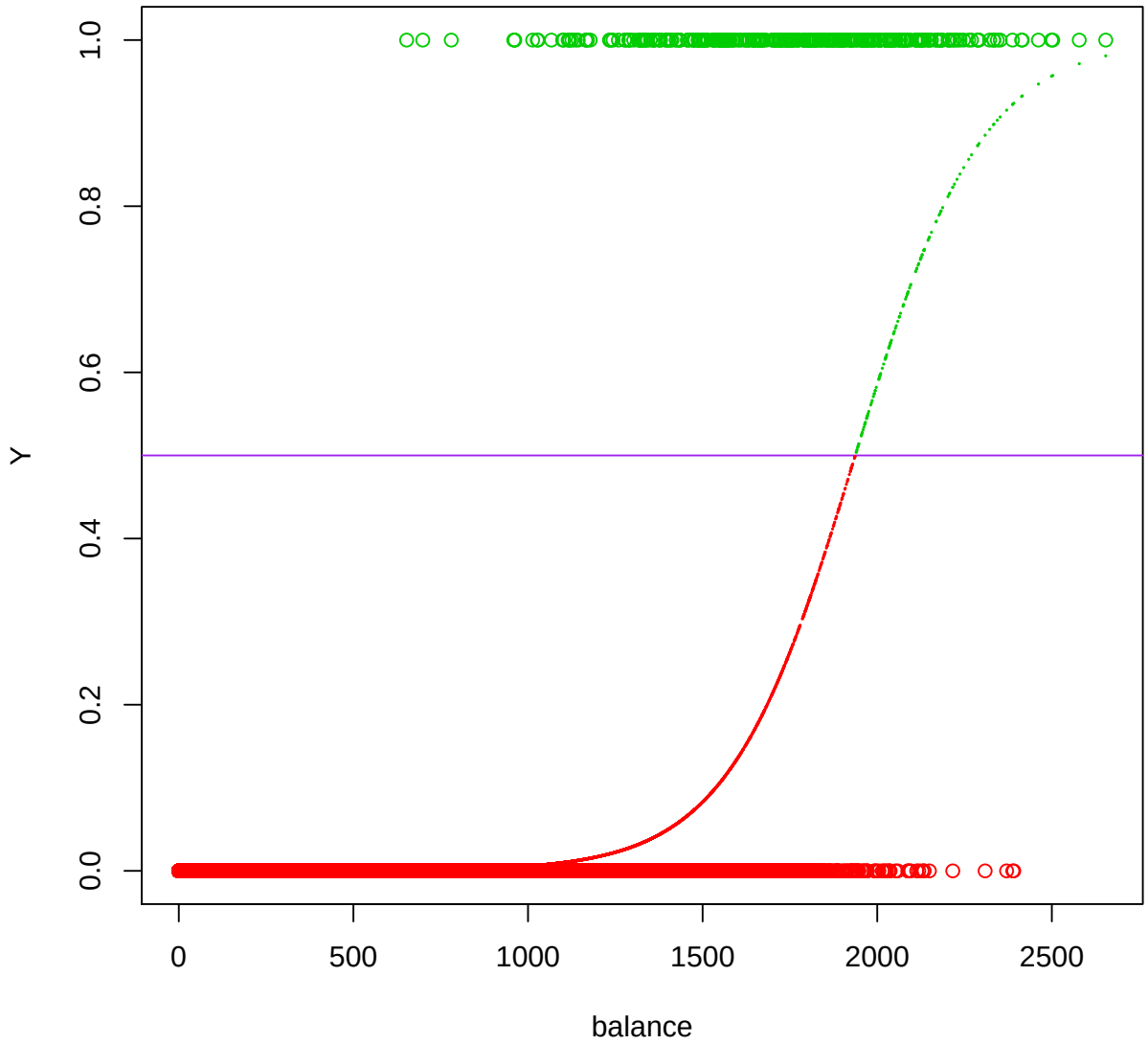
$$\log\left(\frac{\pi(X_i)}{1-\pi(X_i)}\right) = X_i^\top \beta$$

Which has inverse

$$\pi(X_i) = \frac{\exp\{X_i^\top \beta\}}{1+\exp\{X_i^\top \beta\}}$$

Then, we classify $Y = 1$ if and only if $\pi(X) > c$.

GLM EXAMPLE: PREDICTING CREDIT DEFAULT



Generalized Linear Models (GLMs)

Logistic regression works remarkably well as a classifier

However, it has major weaknesses

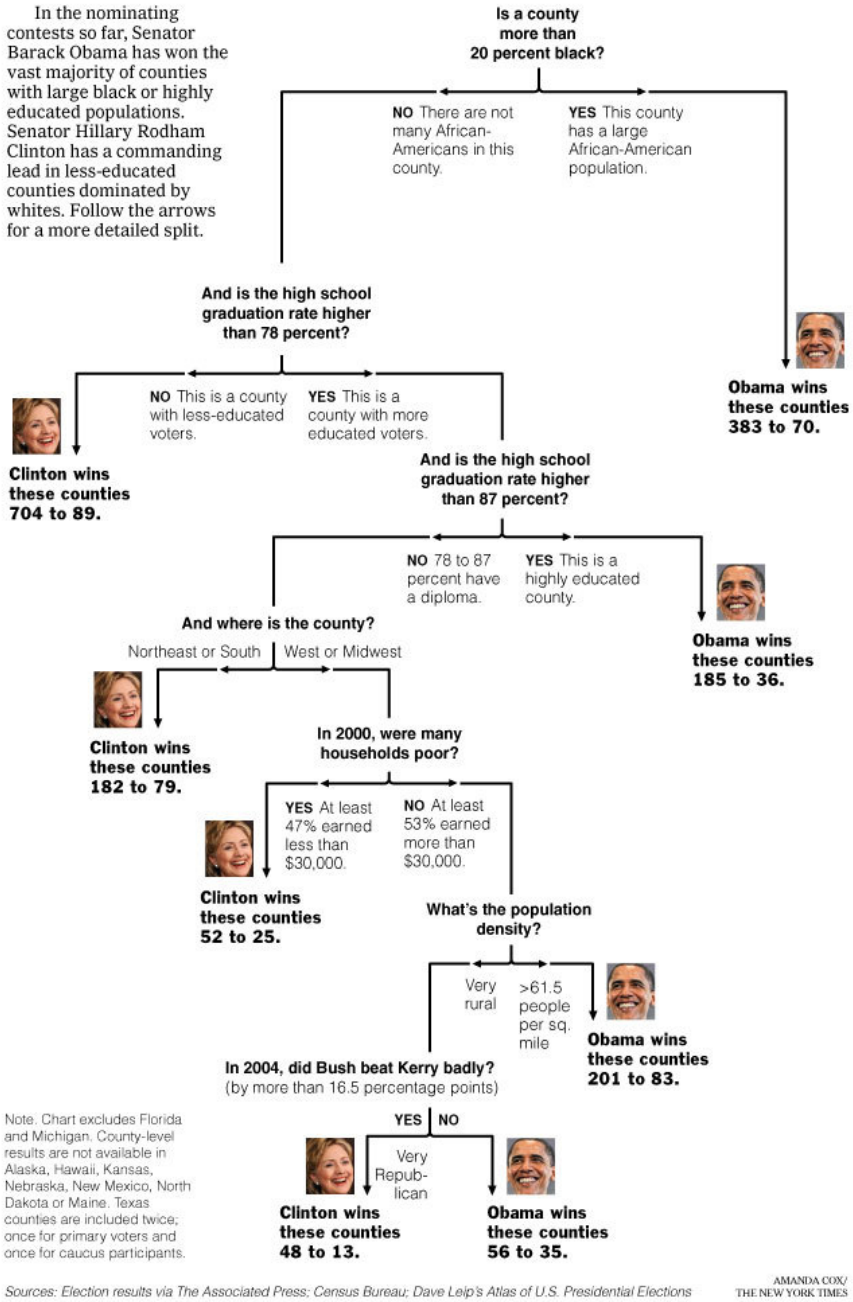
We as a field have developed better methods

Trees

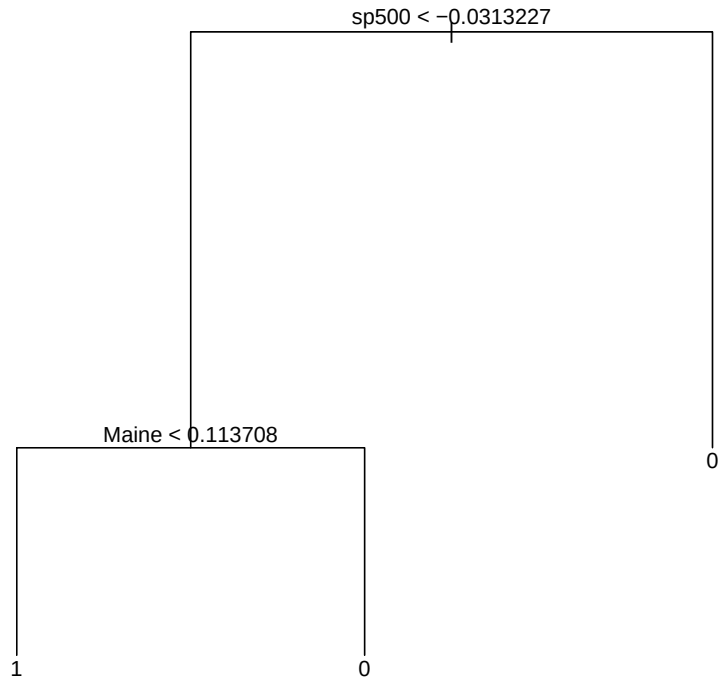
What is a (decision) tree?

- Trees involve stratifying or segmenting the predictor space into a number of simple regions.
- Trees are simple and useful for interpretation.
- Basic trees are not great at prediction.
- More modern methods that use trees are much better.

EXAMPLE TREE



Dendogram view



Terminology

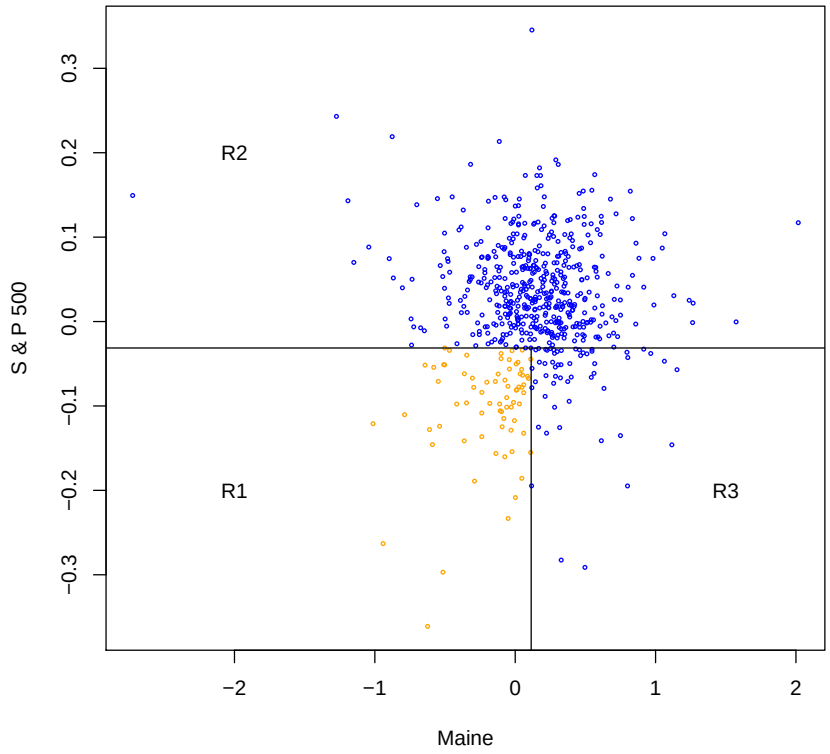
- We call each split or end point a node. Each terminal node is referred to as a leaf.

  This tree has 1 interior node and 2 terminal nodes.

- The interior nodes lead to branches.

  This graph has two main branches (the S&P 500 split).

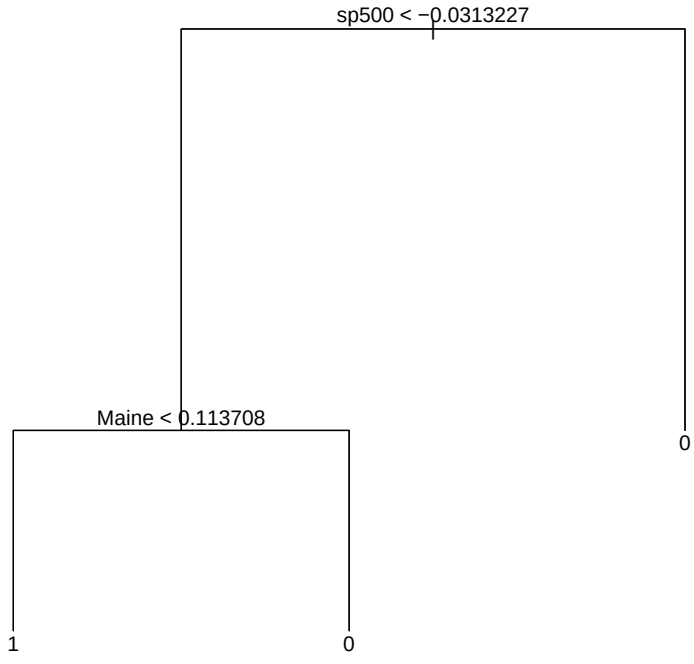
PARTITIONING VIEW

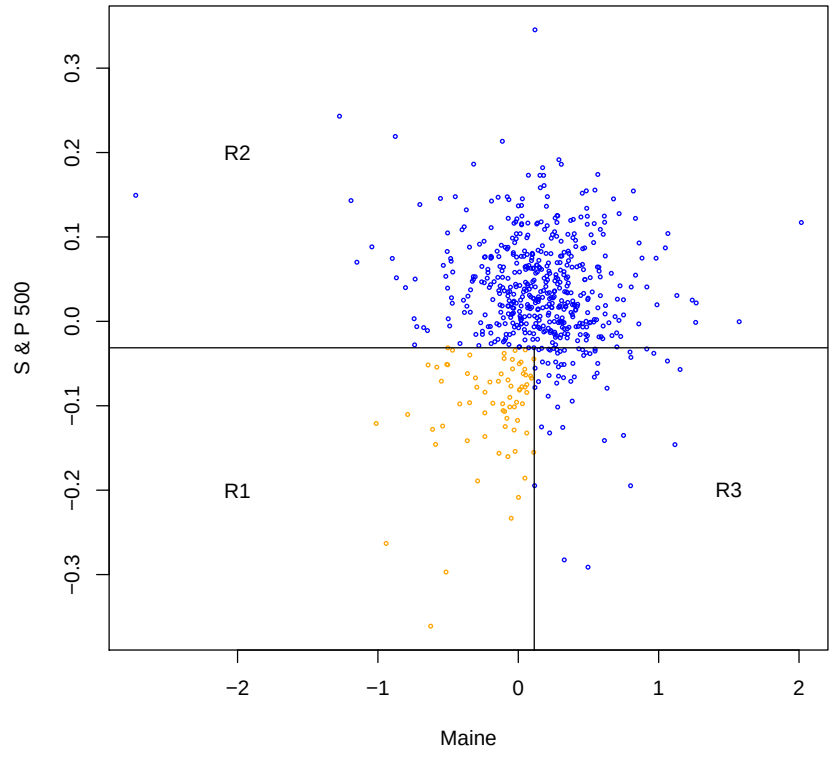


NOTES

- We classify all observations in a region the same.
- The three regions R1, R2, and R3 are the leaves of the tree.

TREE





We can interpret this as

- S&P 500 is the most important variable.
- If S&P 500 is large enough, then we predict no recession.
- If S&P 500 is small enough, then we need to know the change in the Employment Level of Maine.

How do we build a tree?

1. Divide the predictor space into J non-overlapping regions $R_1, \ldots, R_J$ (this is done via greedy, recursive binary splitting).
2. Every observation that falls into a given region R_j is given the same prediction, which is determined by majority (or plurality) vote in that region.

Important:

- Trees can only make rectangular regions that are aligned with the coordinate axis.
- The fit is greedy, which means that after a split is made, all further decisions are conditional on that split.

Does the tree fit?

How do we measure quality of fit?

There are many choices for a metric. Let p_{mk} be the proportion of training observations in the m^{th} region that are from the k^{th} class.

CLASSIFICATION ERROR RATE:	$E = 1 - \max_k(\widehat{p}_{mk})$
GINI INDEX:	$G = \sum_k \widehat{p}_{mk}(1 - \widehat{p}_{mk})$
CROSS-ENTROPY:	$D = -\sum_k \widehat{p}_{mk} \log(\widehat{p}_{mk})$

Both Gini and cross-entropy can be thought of as measuring the purity of the classifier (small if all p_{mk} are near zero or 1). These are preferred over the classification error rate.

We build a classifier by growing a tree that minimizes G or D.

THERE'S A PROBLEM

Following this procedure overfits!

- The process described so far will fit overly complex trees, leading to poor predictive performance.
- Overfit trees mean they have too many leaves.
- To stretch the analogy further, trees with too many leaves must be pruned.

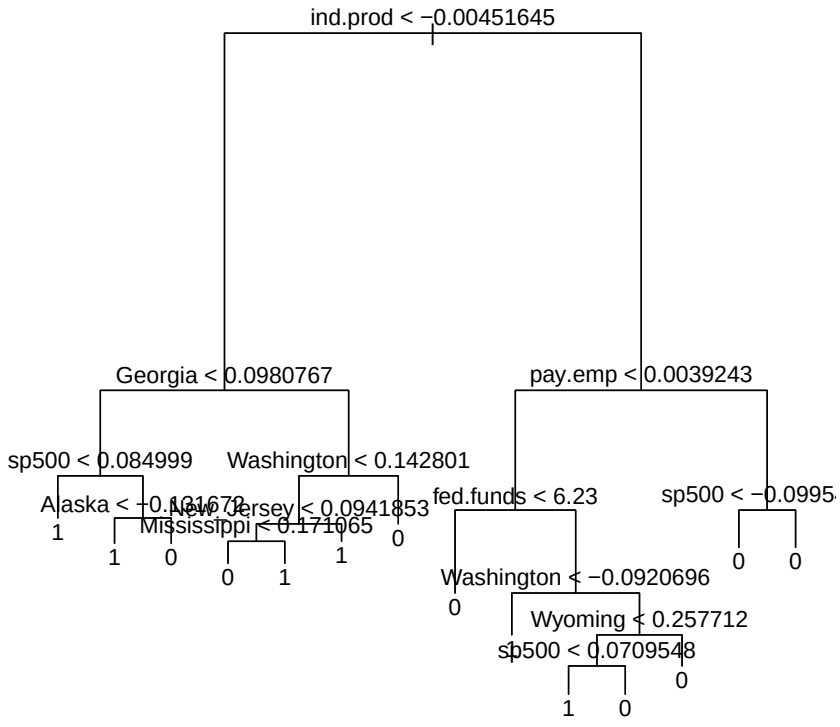
PRUNING THE TREE

- Cross-validation can be used to directly prune the tree, but it is far too expensive (computationally) to use in practice (combinatorial complexity).
- Instead, we use 'weakest link pruning',

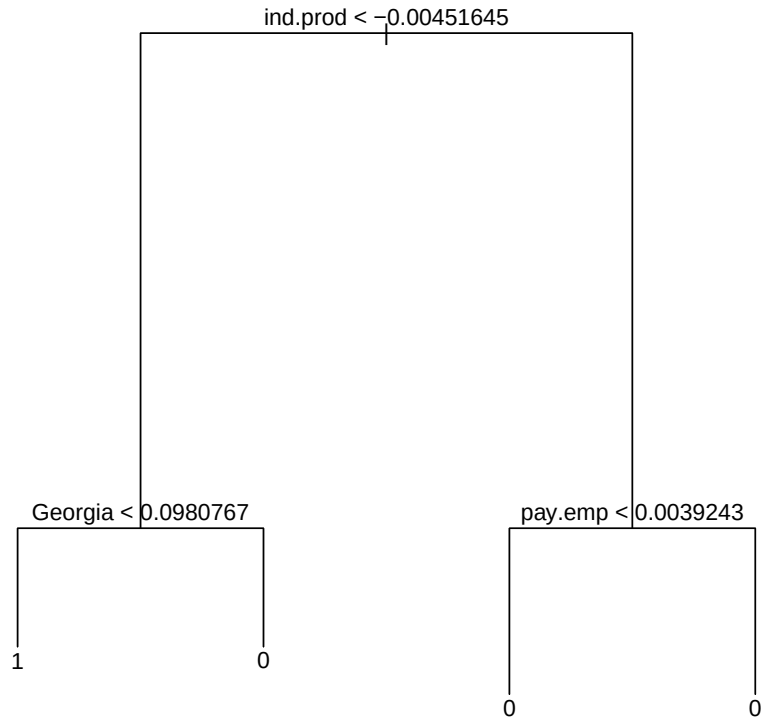
$$\sum_{m=1}^{|T|} \sum_{i \in R_m} I(y_i = \widehat{y}_{R_m}) + \alpha |T|$$

  where $|T|$ is the number of terminal nodes. Essentially, we are trading training fit (first term) with model complexity (second) term (compare to lasso).
- Now, cross-validation can be used to pick α.

RESULTS OF TREES ON RECESSION DATA



Unpruned tree

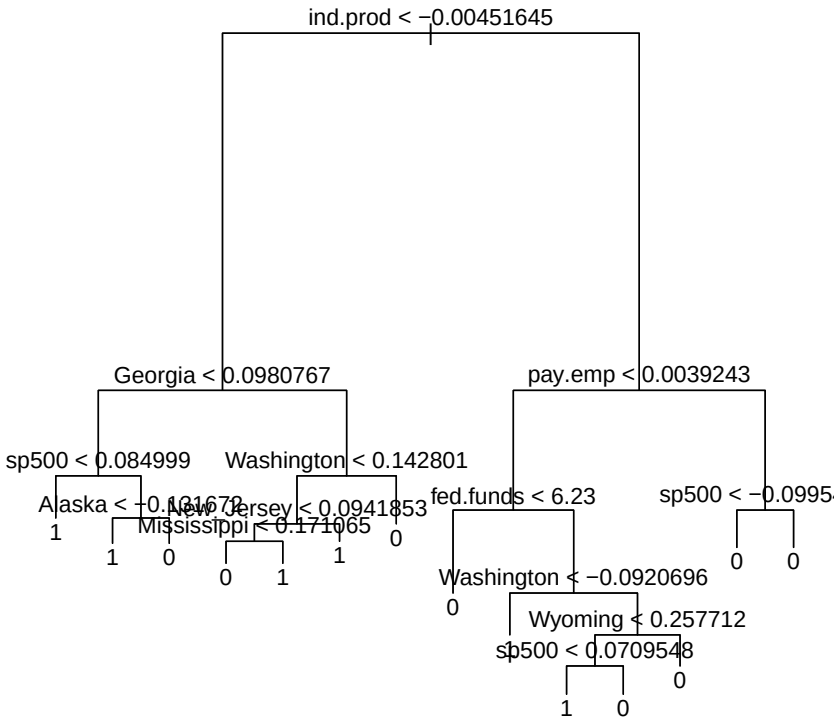


Pruned Tree

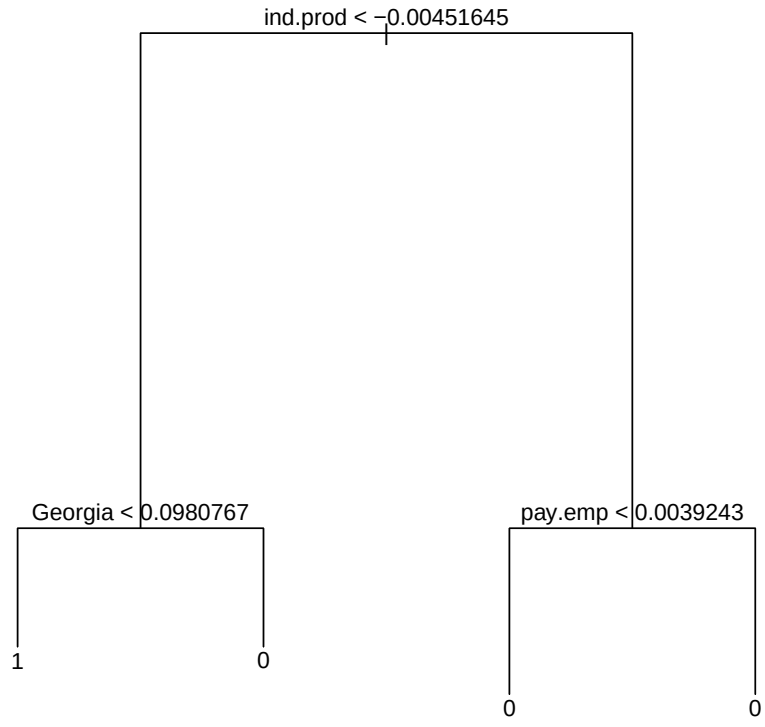
These trees were grown on the training data (from 1960 to 2000)

Now, we use them to predict on the test data (from 2000 to 2011)

RESULTS OF TREES ON RECESSION DATA



Unpruned tree

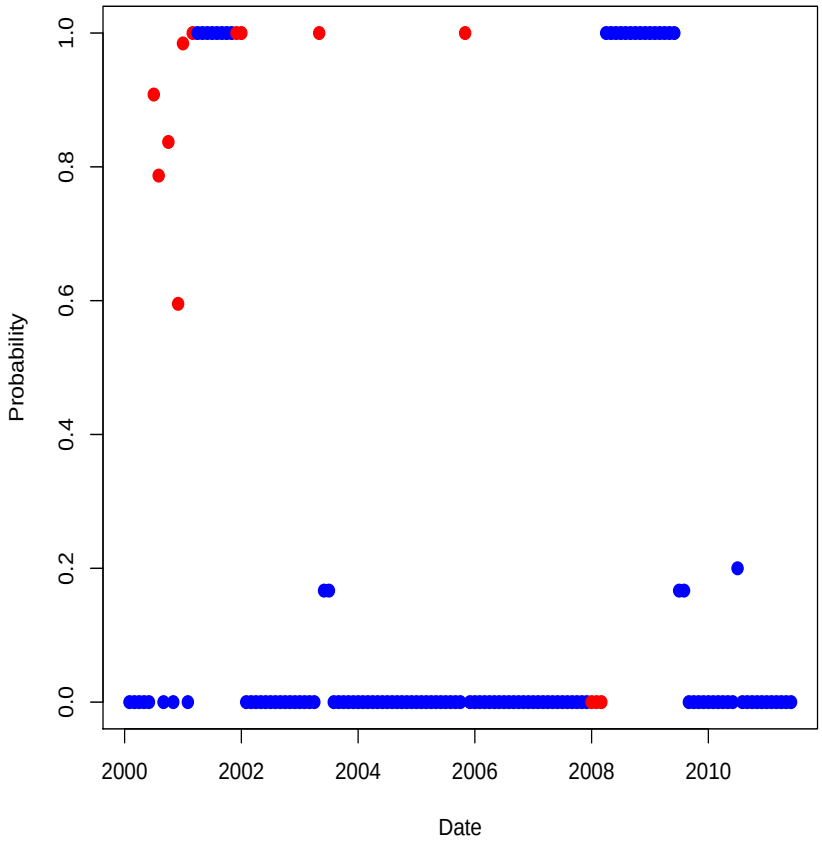


Pruned Tree

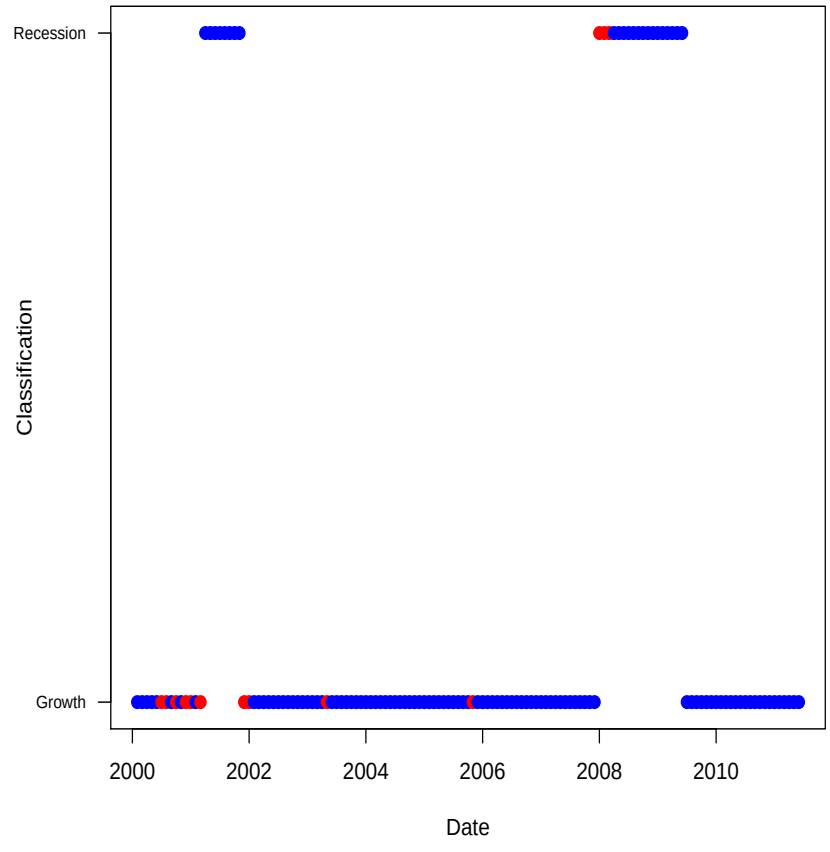
These trees were grown on the training data (from 1960 to 2000)

Now, we use them to predict on the test data (from 2000 to 2011)

RESULTS OF TREES ON RECESSION DATA



Posterior probability of prediction



Predictions

Advantages and disadvantages of trees

- \+ Trees are very easy to explain (much easier than even linear regression).
- \+ Some people believe that decision trees mirror human decision.
- \+ Trees can easily be displayed graphically no matter the dimension of the data.
- \+ Trees can easily handle qualitative predictors without the need to create dummy variables.
- – Trees aren't very good at prediction.

To fix this last one, we can try to grow many trees and average their performance.

Bagging

BAGGING

Many methods (trees included) tend to be designed to have lower bias but high variance. This means that if we split the training data into two parts at random and fit a decision tree to each part, the results could be quite different.

In contrast, a low variance estimator would yield similar results if applied repeatedly to distinct data sets (consider $\widehat{f} = 0$).

Bagging, also known as bootstrap aggregation, is a general purpose procedure for reducing variance. We'll use it specifically in the context of trees, but it can be applied much more broadly.

BAGGING: THE MAIN IDEA

Suppose we have n uncorrelated observations $Z_1, \ldots, Z_n$, each with variance σ^2.

What is the variance of

$$\overline{Z} = \frac{1}{n} \sum_{i=1}^{n} Z_i ?$$

Answer: σ^2/n.

More generally, if we have B separate (uncorrelated) training sets, $1, \ldots, B$, we can form B separate model fits, $\widehat{f}^1(x), \ldots, \widehat{f}^B(x)$, and then average them:

$$\widehat{f}_B(x) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}^b(x)$$

BAGGING: THE MAIN IDEA

Suppose we have n uncorrelated observations $Z_1, \ldots, Z_n$, each with variance σ^2.

What is the variance of

$$\overline{Z} = \frac{1}{n} \sum_{i=1}^{n} Z_i ?$$

Answer: σ^2/n.

More generally, if we have B separate (uncorrelated) training sets, $1, \ldots, B$, we can form B separate model fits, $\widehat{f}^1(x), \ldots, \widehat{f}^B(x)$, and then average them:

$$\widehat{f}_B(x) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}^b(x)$$

BAGGING: THE BOOTSTRAP PART

Of course, this isn't practical as having access to many training sets is unlikely. We therefore turn to the bootstrap to simulate having many training sets.

The bootstrap is a widely applicable statistical tool that can be used to quantify uncertainty without Gaussian approximations.

Let's look at an example.

Bootstrap detour

BOOTSTRAP DETOUR

Suppose we are looking to invest in two financial instruments, X and Y. The return on these investments is random, but we still want to allocate our money in a risk minimizing way. That is, for some $\alpha \in (0, 1)$, we want to minimize

$$\text{Var}(\alpha X + (1 - \alpha)Y)$$

The minimizing α is:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}^2}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}^2}$$

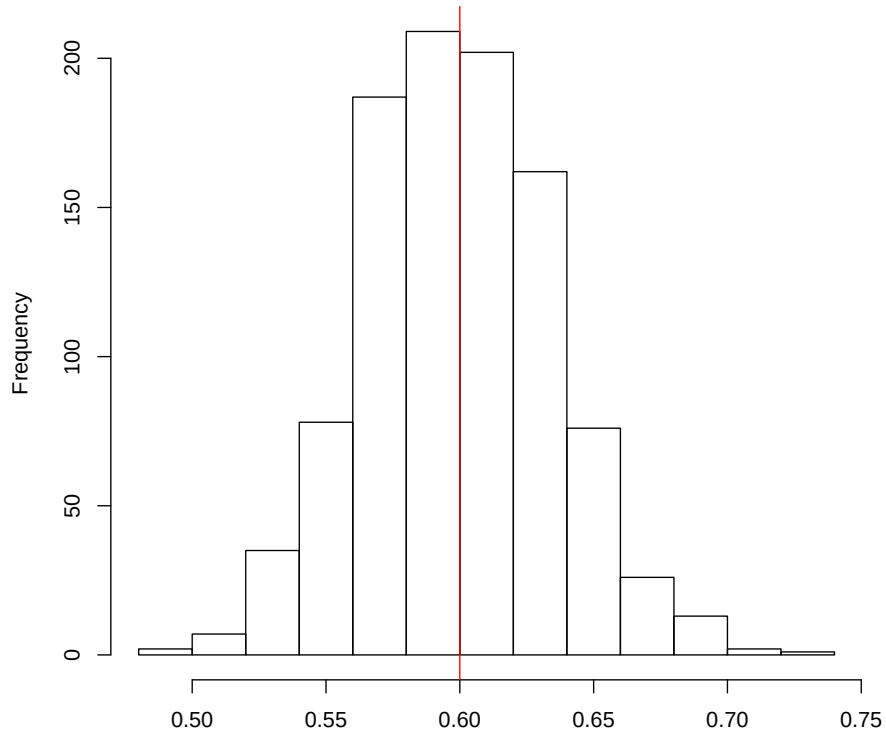
which we can estimate via

$$\widehat{\alpha} = \frac{\widehat{\sigma}_Y^2 - \widehat{\sigma}_{XY}^2}{\widehat{\sigma}_X^2 + \widehat{\sigma}_Y^2 - 2\widehat{\sigma}_{XY}^2}$$

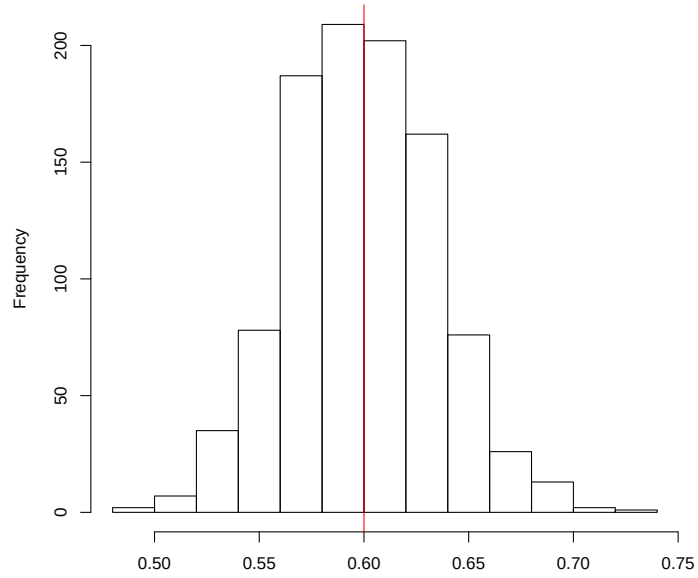
BOOTSTRAP DETOUR

Now that we have an estimator of α, it would be nice to have an estimator of its *variability*. In this case, computing a standard error is very difficult.

Suppose for a moment that we can simulate a large number of draws (say 1000) of the data, which has actual value $\alpha = 0.6$.
Then we could get estimates $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_{1000}$.



BOOTSTRAP DETOUR



This is a histogram of all 1000 estimates. The mean of all of these is:

$$\overline{\alpha} = \frac{1}{1000}\sum_{r=1}^{1000}\widehat{\alpha}_r = 0.599,$$

which is very close to 0.6 (red line), and the standard deviation is

$$\sqrt{\frac{1}{1000-1}\sum_{r=1}^{1000}(\widehat{\alpha}_r - \overline{\alpha})^2} = 0.035.$$

BOOTSTRAP DETOUR

The standard error of 0.035 gives a very good idea of the accuracy of $\widehat{\alpha}$ for a single sample. Roughly speaking, for a new random sample, we expect $\widehat{\alpha} \in (\alpha - 0.035, \alpha + 0.035)$.

In practice, of course, we cannot use this procedure as it relies on being able to draw a large number of (independent) samples from the same distribution as our data. This is where the bootstrap comes in.

We instead draw a large number of samples directly from our observed data. This sampling is done *with replacement*, which means that the same data point can be drawn multiple times.

BOOTSTRAP DETOUR: SMALL EXAMPLE

Suppose we have data $X = (4.3, 3, 7.2, 6.9, 5.5)$.

Then we can draw bootstrap samples, which might look like:

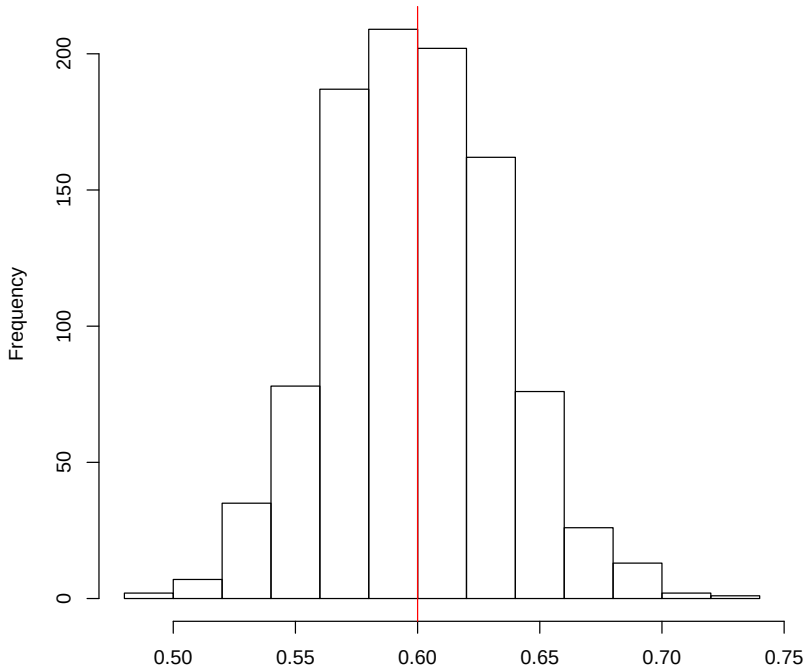
$$
\begin{aligned}
X_1^* &= (7.2, 4.3, 7.2, 5.5, 6.9) \\
X_2^* &= (6.9, 4.3, 3.0, 4.3, 6.9) \\
&\vdots \\
X_B^* &= (4.3, 3.0, 3.0, 5.5, 6.9)
\end{aligned}
$$

It turns out these are all draws from the same (empirical) distribution as our data, which should be close to the actual distribution (Glivenko-Cantelli theorem).

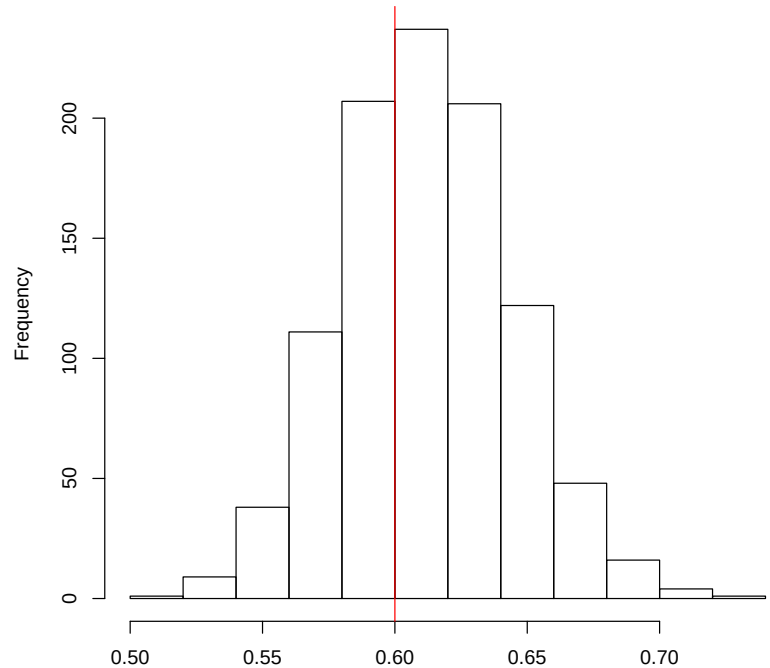
BOOTSTRAP DETOUR

Now, the bootstrap estimator of the standard error is:

$$\mathrm{SE}_B(\widehat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{\alpha}_b* - \frac{1}{B}\sum_{s=1}^{B}\widehat{\alpha}_s^*\right)^2}$$



Sampling distribution of $\widehat{\alpha}$ from repeated draws of same data (impossible)



Sampling distribution of $\widehat{\alpha}$ estimated from bootstrap draws (possible)

BOOTSTRAP: END DETOUR

Summary:

Suppose we have data $Z_1, \ldots, Z_n$ and we want to get an idea of the sampling distribution of some statistic $f(Z_1, \ldots, Z_n)$. Then we do the following

1. Choose some large number of samples, B.
2. For each $b = 1, \ldots, B$, draw a new dataset from $Z_1, \ldots, Z_n$, call it $Z_1^*, \ldots, Z_n^*$.
3. Compute $\widehat{f}_b^* = \widehat{f}(Z_1^*, \ldots, Z_n^*)$.
4. Now, we can estimate the distribution of $\widehat{f}(Z_1, \ldots, Z_n)$ by looking at the B draws, $\widehat{f}_b^*$.

BOOTSTRAP: END DETOUR

Summary:

Suppose we have data $Z_1, \ldots, Z_n$ and we want to get an idea of the sampling distribution of some statistic $f(Z_1, \ldots, Z_n)$. Then we do the following

1. Choose some large number of samples, B.
2. For each $b = 1, \ldots, B$, draw a new dataset from $Z_1, \ldots, Z_n$, call it $Z_1^*, \ldots, Z_n^*$.
3. Compute $\widehat{f}_b^* = \widehat{f}(Z_1^*, \ldots, Z_n^*)$.
4. Now, we can estimate the distribution of $\widehat{f}(Z_1, \ldots, Z_n)$ by looking at the B draws, $\widehat{f}_b^*$.

BOOTSTRAP: END DETOUR

Summary:

Suppose we have data $Z_1, \ldots, Z_n$ and we want to get an idea of the sampling distribution of some statistic $f(Z_1, \ldots, Z_n)$. Then we do the following

1. Choose some large number of samples, B.
2. For each $b = 1, \ldots, B$, draw a new dataset from $Z_1, \ldots, Z_n$, call it $Z_1^*, \ldots, Z_n^*$.
3. Compute $\widehat{f}_b^* = \widehat{f}(Z_1^*, \ldots, Z_n^*)$.
4. Now, we can estimate the distribution of $\widehat{f}(Z_1, \ldots, Z_n)$ by looking at the B draws, $\widehat{f}_b^*$.

BOOTSTRAP: END DETOUR

Summary:

Suppose we have data $Z_1, \ldots, Z_n$ and we want to get an idea of the sampling distribution of some statistic $f(Z_1, \ldots, Z_n)$. Then we do the following

1. Choose some large number of samples, B.
2. For each $b = 1, \ldots, B$, draw a new dataset from $Z_1, \ldots, Z_n$, call it $Z_1^*, \ldots, Z_n^*$.
3. Compute $\widehat{f}_b^* = \widehat{f}(Z_1^*, \ldots, Z_n^*)$.
4. Now, we can estimate the distribution of $\widehat{f}(Z_1, \ldots, Z_n)$ by looking at the B draws, $\widehat{f}_b^*$.

BAGGING: THE BOOTSTRAP PART

Now, instead of having B separate training sets, we do B bootstrap samples. $\widehat{f}_1^*(x), \ldots, \widehat{f}_B^*(x)$, and then average them:

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

This process is known as *Bagging* (bootstrap aggregation).

Bagging trees

BAGGING TREES

The procedure for trees is the following

1. Choose a large number B.
2. For each $b = 1, \ldots, B$, grow an unpruned tree on the b^{th} bootstrap draw from the data.
3. Average all these trees together.

Each tree, since it is unpruned, will have (low/high) variance and (low/high) bias.

BAGGING TREES

The procedure for trees is the following

1. Choose a large number B.
2. For each $b = 1, \ldots, B$, grow an unpruned tree on the b^{th} bootstrap draw from the data.
3. Average all these trees together.

Each tree, since it is unpruned, will have (low/high) variance and (low/high) bias.

BAGGING TREES

The procedure for trees is the following

1. Choose a large number B.
2. For each $b = 1, \ldots, B$, grow an unpruned tree on the b^{th} bootstrap draw from the data.
3. Average all these trees together.

Each tree, since it is unpruned, will have (low/high) variance and (low/high) bias.

Therefore averaging many trees results in an estimator that has lower variance and still low bias.

BAGGING TREES

The procedure for trees is the following

1. Choose a large number B.
2. For each $b = 1, \ldots, B$, grow an unpruned tree on the b^{th} bootstrap draw from the data.
3. Average all these trees together.

Each tree, since it is unpruned, will have (low/high) variance and (low/high) bias.

Therefore averaging many trees results in an estimator that has lower variance and still low bias.

Caveat: Be careful bootstrapping time series data.

Bagging trees: Variable importance measures

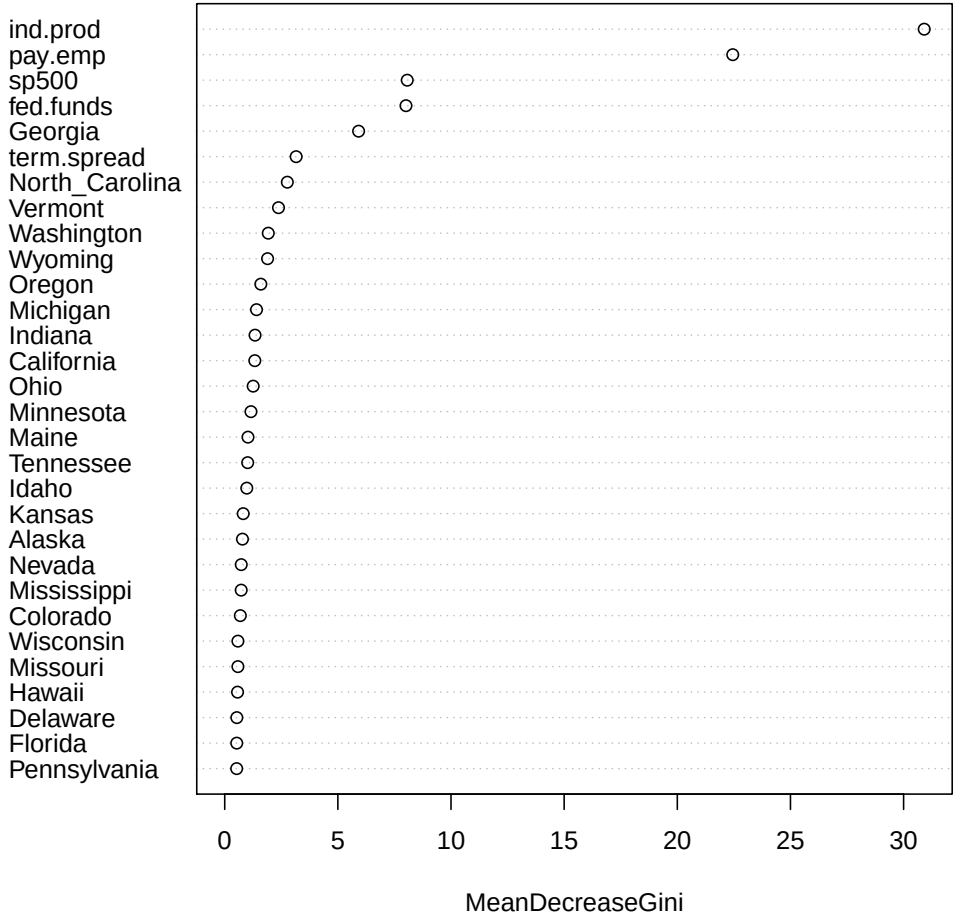
Though bagging can improve predictive performance of trees, the trade-off is we sacrificed some interpretability.

We no longer have that nice diagram that shows the segmentation of the predictor space (or, more accurately, we have B of them).

To recover some information, we can do the following:

1. For each of the b trees and each of the p variables, we record the amount that the Gini index is reduced by the addition of that variable
2. Report the average reduction over all B trees.

BAGGING TREES: VARIABLE IMPORTANCE MEASURES



RANDOM FOREST

Random Forest is a small extension of Bagging, in which the bootstrap trees are decorrelated.

The idea is, we draw a bootstrap sample and start to build a tree.

- At each split, we randomly select m of the possible p predictors as candidates for the split.
- A new sample of size m of the predictors is taken at each split.

Usually, we use about $m = \sqrt{p}$
(this would be 7 out of 56 predictors for GDP data).

In other words, at each split, we aren't even allowed to consider the majority of possible predictors!

RANDOM FOREST

What is going on here?

Suppose there is 1 really strong predictor and many mediocre ones.

- Then each tree will have this one predictor in it,
- Therefore, each tree will look very similar (i.e. highly correlated).
- Averaging highly correlated things leads to much less variance reduction than if they were uncorrelated.

If we don't allow some trees/splits to use this important variable, each of the trees will be much less similar and hence much less correlated.

Bagging is Random Forest when $m = p$, that is, when we can consider all the variables at each split.

Reporting results

There are two main ways classification results are reported:

- Sensitivity/specificity
- Confusion matrix

REPORTING RESULTS: SENSITIVITY/SPECIFICITY

SENSITIVITY:	The proportion of times we label 'recession', given that 'recession' is the correct answer. (True $+$)
SPECIFICITY:	The proportion of times we label 'no recession', given that 'no recession' is the correct answer. (True $-$)

We can think of this in terms of hypothesis testing. If

$$H_0 : \text{ no recession,}$$

then

SENSITIVITY:	$P(\text{reject } H_0 \mid H_0 \text{ is false})$, that is: 1 - P(Type II error)
SPECIFICITY:	$P(\text{accept } H_0 \mid H_0 \text{ is true})$, that is: 1 - P(Type I error)

Reporting results: Confusion matrix

Alternatively, we can report our results in a matrix:

		Truth	
		Growth	Recession
Our Predictions	Growth	(A)	(B)
	Recession	(C)	(D)

For each observation in the test set, we compare our prediction to the truth.

The total number of each combination is recorded in the table.

The overall miss-classification rate is

$$\frac{(\mathrm{B}) + (\mathrm{C})}{(\mathrm{A}) + (\mathrm{B}) + (\mathrm{C}) + (\mathrm{D})} = \frac{(\mathrm{B}) + (\mathrm{C})}{n_{\text{test}}}$$

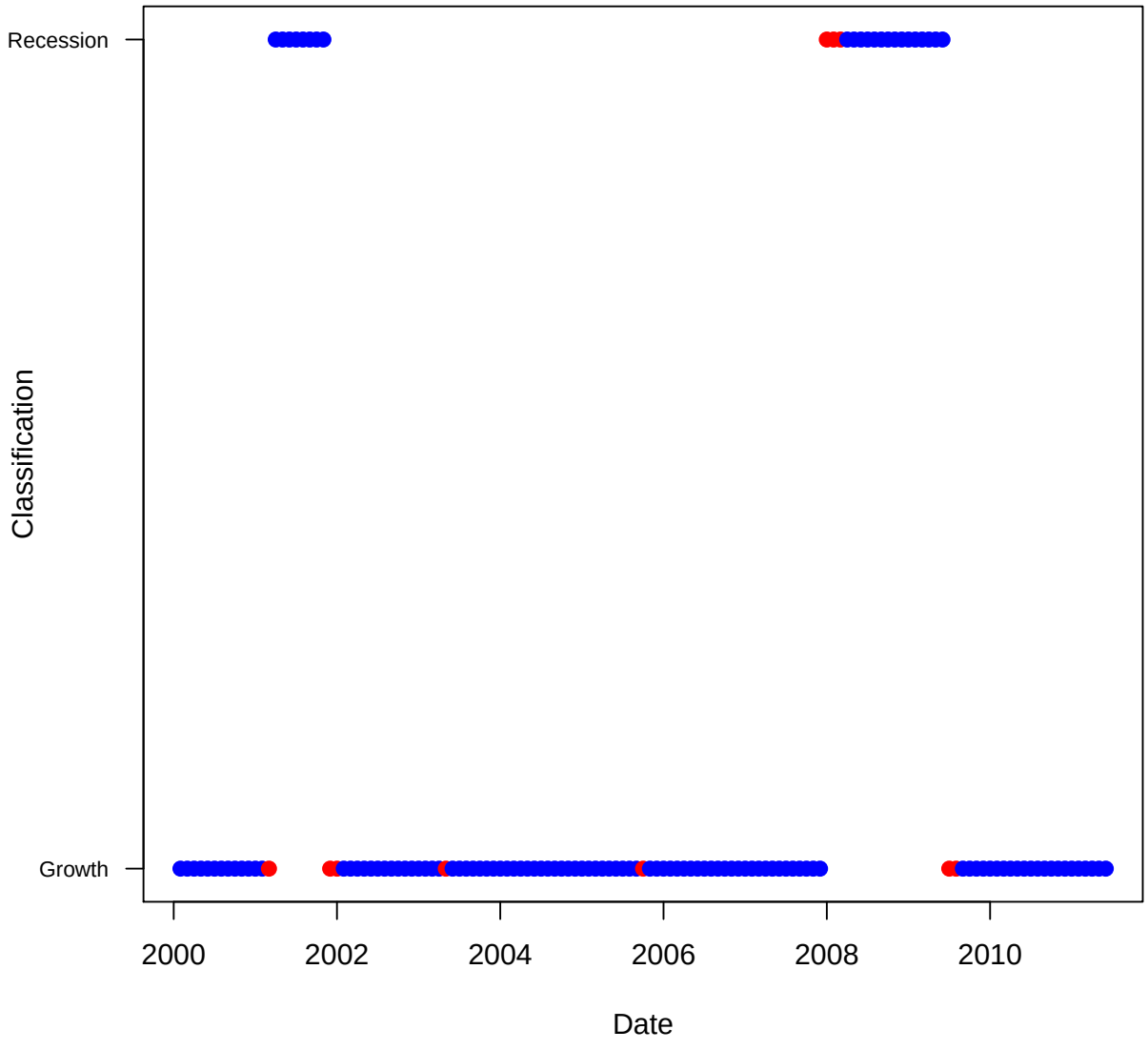
TREE RESULTS: CONFUSION MATRICES

			Truth		
			Growth	Recession	Mis-Class
Our Predictions	NULL	Growth	111	26	
		Recession	0	0	18.9%
	TREE	Growth	99	3	
		Recession	12	23	10.9%
	RANDOM FOREST	Growth	102	5	
		Recession	9	21	10.2%
	BAGGING	Growth	104	3	
		Recession	7	23	7.3%

Tree results: Sensitivity & specificity

	Sensitivity	Specificity
Null	0.000	1.000
Tree	0.884	0.891
Random Forest	0.807	0.918
Bagging	0.884	0.936

Results of Bagging on recession data



Out-of-Bag error estimation (OOB)

One can show that, on average, drawing n samples from n observations with replacement (also known as bootstrap) results in about 2/3 of the observations being selected.

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

We can think of it as a for-free cross-validation.

Each time a tree is grown, we can get its prediction error on the unused observations. We average this over all bootstrap samples.

OUT-OF-BAG ERROR ESTIMATION FOR BAGGING

			Truth		
			Growth	Recession	Miss-Class
Our Predictions	BAGGING	Growth	400	10	
		Recession	21	46	6.5%

CLASSIFICATION RECAP

There are many, many different classification algorithms.

Different methods will work bettter in different situations.

LINEAR:	Logisitic regression,
	linear discriminant analysis (LDA),
	GLMNET, separating hyperplanes
NON-LINEAR:	quadratic discriminant analysis (QDA),
	trees (and associated methods),
	K-nearest neighbors (KNN)
	Support vector machines (SVM)
	Neural networks