

# DECISION TREES

-INTRODUCTION TO DATA SCIENCE-

ISL 8.1

Lecturer: Darren Homrighausen, PhD

# WHAT IS A (DECISION) TREE?

- Trees involve **stratifying** or **segmenting** the feature 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.

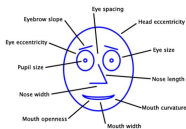
## 3

A phylogenetic tree of the animal kingdom. The tree originates from a 'colonial choanoflagellate ancestor' at the bottom. It branches upwards. The first major split is between 'no nerves' (leading to Porifera) and 'nerves' (leading to all other groups). The 'nerves' branch further splits into 'radial symmetry' (leading to Cnidaria and Pnyhelminthes) and 'bilateral symmetry' (leading to all other groups). The 'bilateral symmetry' branch splits into 'protostomes' (leading to Rotifera, Mollusca, Annelida, and Nemertoda) and 'deuterostomes' (leading to Arthropoda, Echinodermata, and Chordata). Each terminal group is represented by a small icon: Porifera (purple blob), Cnidaria (brown jellyfish), Pnyhelminthes (green worm), Rotifera (blue rotifer), Mollusca (brown shell), Annelida (pink worm), Nemertoda (pink worm), Arthropoda (green insect), Echinodermata (orange starfish), and Chordata (blue fish).

```

graph BT
    Root[colonial choanoflagellate ancestor] --> NoNerves[no nerves]
    Root --> Nerves[nerves]
    NoNerves --> Porifera[Porifera]
    Nerves --> RadialSymmetry[radial symmetry]
    Nerves --> BilateralSymmetry[bilateral symmetry]
    RadialSymmetry --> Cnidaria[Cnidaria]
    RadialSymmetry --> Pnyhelminthes[Pnyhelminthes]
    BilateralSymmetry --> Protostomes[protostomes]
    BilateralSymmetry --> Deuterostomes[deuterostomes]
    Protostomes --> Rotifera[Rotifera]
    Protostomes --> Mollusca[Mollusca]
    Protostomes --> Annelida[Annelida]
    Protostomes --> Nemertoda[Nemertoda]
    Deuterostomes --> Arthropoda[Arthropoda]
    Deuterostomes --> Echinodermata[Echinodermata]
    Deuterostomes --> Chordata[Chordata]
  
```

chernoff face



In the nominating contests so far, Senator Barack Obama has won the vast majority of counties with large black or highly educated populations. Senator Hillary Rodham Clinton lead in less-educated whites. Follow the arrows for a more detailed split.

**Is a county more than 20 percent black?**

**NO** There are not many African-Americans in this county. **YES** This county has a large African-American population.

**Obama wins these counties 383 to 70.**

**And is the high school graduation rate higher than 78 percent?**

**NO** This is a county with less-educated voters. **YES** This is a county with more educated voters.

**Clinton wins these counties 704 to 89.**

**And is the high school graduation rate higher than 87 percent?**

**NO** 78 to 87 percent have a diploma. **YES** This is a highly educated county.

**Obama wins these counties 185 to 36.**

**And where is the county?**

**Northeast or South** **West or Midwest**

**Clinton wins these counties 182 to 79.**

**In 2000, were many households poor?**

**YES** At least 47% earned less than \$30,000. **NO** At least 53% earned more than \$30,000.

**Clinton wins these counties 52 to 25.**

**What's the population density?**

**Very rural** **>61.5 people per sq. mile**

**Obama wins these counties 201 to 83.**

**In 2004, did Bush beat Kerry badly?** (by more than 16.5 percentage points)

**YES** **NO**

**Very Republican**

**Clinton wins these counties 48 to 13.** **Obama wins these counties 56 to 35.**

Note: Chart excludes Florida and Michigan. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine. Text counties are included twice; once for primary voters and once for caucus participants.

Note: Chart excludes Florida and Michigan. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine. Texas counties are included twice; once for primary voters and once for caucus participants.

Sources: Election results via The Associated Press; Census Bureau; Dave Leip's Atlas of U.S. Presidential Elections

ATLANTA COB/  
NEW YORK TIMES

# A MOTIVATING EXAMPLE: REMINDER

Suppose we are interested in predicting whether or not the economy will be in a **recession**

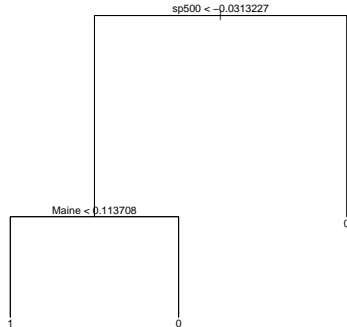
We have quarterly measurements of

- State level economic growth  
(Larger number is better)
- Federal level variables such as GDP, interest rates, employment, S&P 500, ...

Here, we will code the supervisor as

$$Y = \begin{cases} 1 & \text{if recession} \\ 0 & \text{if growth} \end{cases}$$

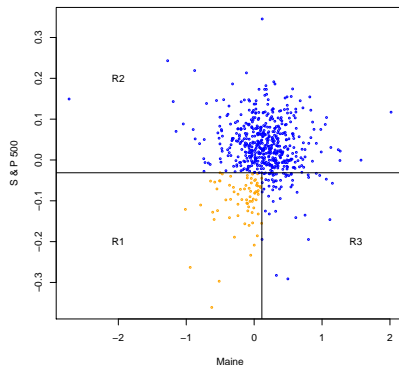
# DENDROGRAM VIEW



## TERMINOLOGY

- We call each split or end point a **node**. Each terminal node is referred to as a **leaf**
  - ▶ This tree has 2 interior nodes and 3 terminal nodes.
- The interior nodes lead to **branches**.
  - ▶ This graph has two main branches (the S&P 500 split).
- Interpret the plot as “Left if true” and “right if false”

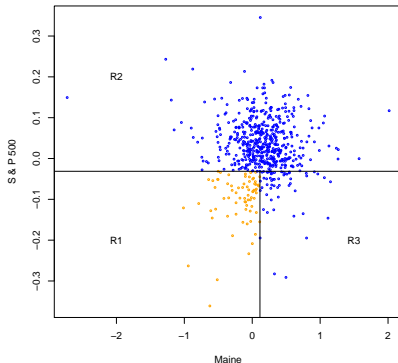
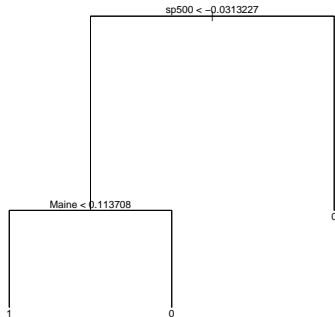
# PARTITIONING VIEW



## 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 economic growth of Maine.

# HOW DO WE BUILD A TREE?

1. Divide the feature space into  $M$  non-overlapping regions  $R_1, \dots, R_M$   
(this is done via greedy, binary splitting)
2. Every observation that falls into a given region  $R_m$  is given the same prediction
  - ▶ **REGRESSION:** The average of the supervisors for a region
  - ▶ **CLASSIFICATION:** 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.
- The tree stops splitting when there are too few observations in a terminal node



# Regression trees

# IMPLICIT MODEL

For a partition  $R_1, \dots, R_M$ , the model for the supervisor is

$$f(X) = \sum_{m=1}^M c_m \mathbf{1}(X \in R_m)$$

We need to find good values for  $M$ ,  $(R_m)_{m=1}^M$ , and  $(c_m)_{m=1}^M$

Generally, searching over all possible regions is infeasible

(This would involve sifting through all  $M \leq n$  and all configurations for  $R_m$ )

So we use a **greedy** approach instead

# REGRESSION TREES

Define the two half-planes

$$r_1(j, s) = \{X | x_j \leq s\} \quad \text{and} \quad r_2(j, s) = \{X | x_j > s\}$$

For squared error loss, we solve

$$\min_{j,s} \left[ \min_{c_1} \sum_{X_i \in r_1(j,s)} (Y_i - c_1)^2 + \min_{c_2} \sum_{X_i \in r_2(j,s)} (Y_i - c_2)^2 \right]$$

This generates, for  $n_k = \sum_{i=1}^n \mathbf{1}(X_i \in r_k)$ ,

$$\hat{c}_k = n_k^{-1} \sum_{i: X_i \in r_k} Y_i$$

The next splits will be conditional on the minimizing  $\hat{s}$

# Classification trees

# CLASSIFICATION TREES

For a given region  $R_m$  and class  $g$ , define training proportions

$$\begin{aligned}\hat{p}_{mg}(X) &= \mathbf{1}(X \in R_m) n_m^{-1} \sum_{i: X_i \in R_m} \mathbf{1}(Y_i = g) \\ &= \begin{cases} \frac{1}{n_m} \sum_{i: X_i \in R_m} \mathbf{1}(Y_i = g) & \text{if } X \text{ is in } R_m \\ 0 & \text{if } X \text{ is not in } R_m \end{cases}\end{aligned}$$

Our classification is

$$\hat{g}(X) = \arg \max_g \hat{p}_{mg}(X)$$

This presumes a given partition ( $R_m$ )

To estimate the partition we need a **loss function**

# HOW DO WE MEASURE QUALITY OF FIT?

There are many possibilities:

CLASSIFICATION ERROR RATE:  $ER = 1 - \max_g(\hat{p}_{mg})$

GINI INDEX:  $GI = \sum_g \hat{p}_{mg}(1 - \hat{p}_{mg})$

CROSS-ENTROPY:  $CE = -\sum_g \hat{p}_{mg} \log(\hat{p}_{mg})$

(Cross-entropy is also known as deviance)

We build a classifier by **growing** a tree that **greedily** minimizes one of these criteria

We would like the tree to have **pure** nodes, in the sense that

$$\max_g \hat{p}_{mg} \approx 1$$

# HOW DO WE MEASURE QUALITY OF FIT?

The  $m^{th}$  node is split by minimizing  $ER$ ,  $GI$ , or  $CE$  over all

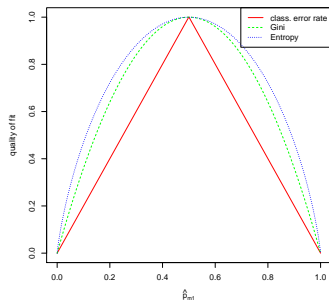
- features
- split points of those feature

such that the loss function is minimized over the 2 new regions

What do these loss functions look like?

# HOW DO WE MEASURE QUALITY OF FIT?

**EXAMPLE:** Suppose  $G = 2$ . Then  $\hat{p} = \hat{p}_{m1} = 1 - \hat{p}_{m2}$



Generally, **GINI INDEX** or **CROSS-ENTROPY** are preferred

(Note: they penalize values of  $\hat{p}$  far from 0 or 1 more severely)



# HOW DO WE MEASURE QUALITY OF FIT?

Additionally, **GINI INDEX** or **CROSS-ENTROPY** are preferred as..

- They are differentiable everywhere
- They are amenable to treating qualitative features with a large number of levels in a computationally efficient way  
(There are  $2^{L-1} - 1$  possibilities for  $L$  levels)  
(See ESL 9.2.4)
- The Gini index can be interpreted in suggestive ways  
(See ESL 9.2.3)

# HOW DO WE MEASURE QUALITY OF FIT?

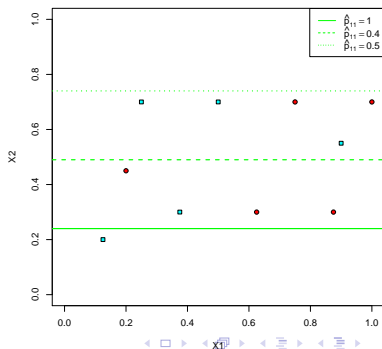
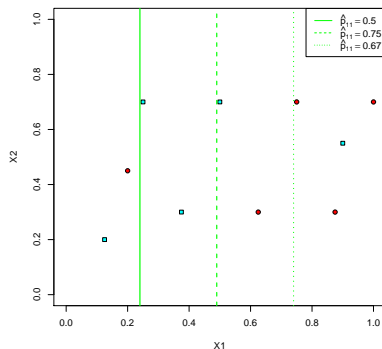
**EXAMPLE:**  $G = p = 2$  and we want to make the first split

(For simplicity, we only look at the loss function at one side of the split)

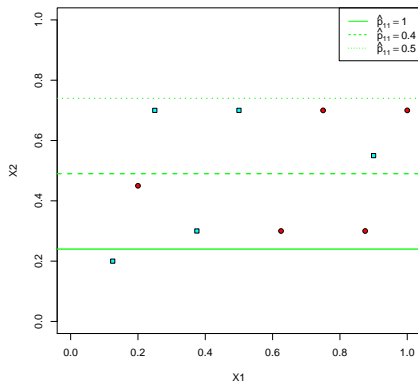
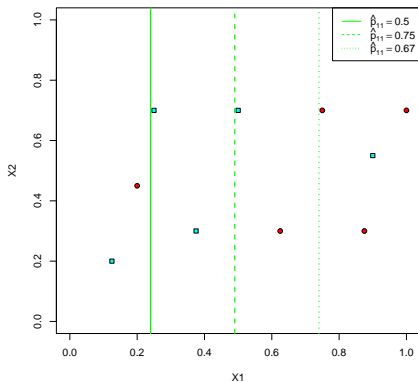
Then  $\hat{p}_{11} = 1 - \hat{p}_{12}$

(Define the 'left' or 'bottom' region as  $R_1$ )

Let's look at some possible splits:

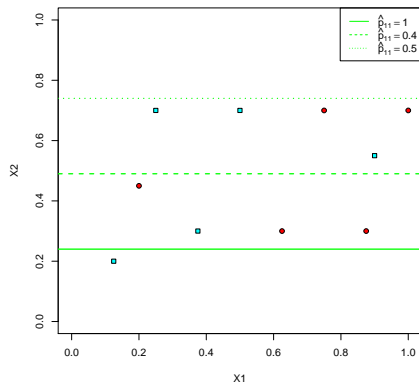
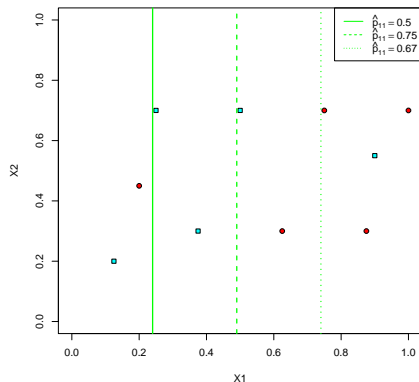


# HOW DO WE MEASURE QUALITY OF FIT?



Where would we split?

# HOW DO WE MEASURE QUALITY OF FIT?



Where would we split if we required  $\geq 2$  observations in a node for  $G_I$  loss?

# THERE'S A PROBLEM

Following this procedure **overfits!**

- The process described so far will fit overly complex trees, leading to poor predictive performance.  
(In fact,  $G_I$  can be interpreted as the variance of Bernoullis)
- Overfit trees mean they have too many leaves.
- To stretch the analogy further, trees with too many leaves must be **pruned**.

# PRUNING THE TREE

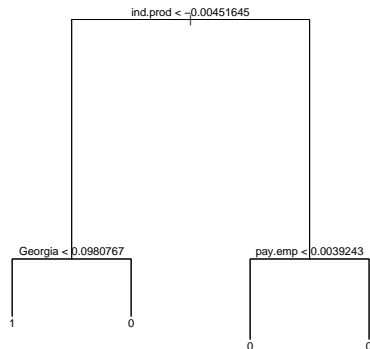
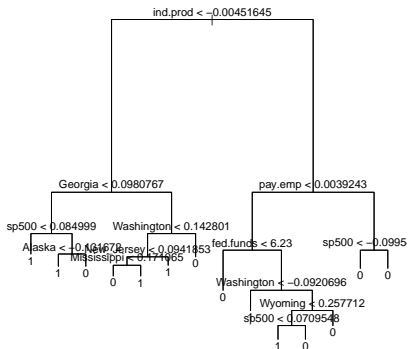
- Using **weakest link pruning** we can obtain a sequence of tree solutions ranging from a tree with no splits to the maximally complex tree  $T_0$ .
- This sequence is a function of a tuning parameter  $\lambda \geq 0$   
(The book uses  $\alpha$ , but I use  $\lambda$  to connect it to previous tuning parameters)
- **Weakest link pruning**: For some loss function  $\ell$

$$\sum_{i=1}^n \ell(Y_i, \hat{g}(X_i)) + \lambda |T|$$

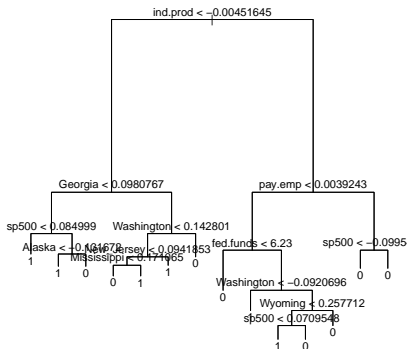
( $|T|$  is the number of terminal nodes or complexity of the tree  $T$ )

- Essentially, we are trading **training fit** (first term) with **model complexity** (second term)
- Now, cross-validation can be used to pick  $\lambda$ .

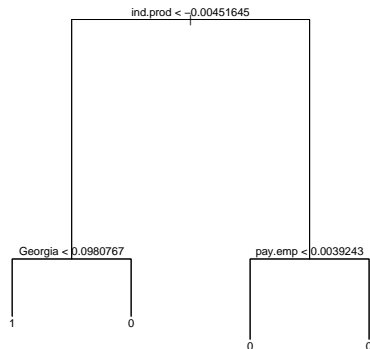
# RESULTS OF TREES ON RECESSION DATA



# RESULTS OF TREES ON RECESSION DATA



Unpruned tree



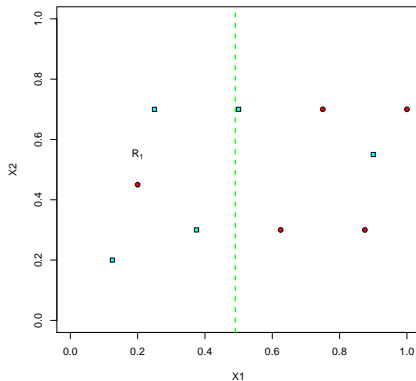
Pruned Tree

The pruned tree is a **subset** of the unpruned tree (**nested**)

There are splits that result the same prediction. **WHY?**



# SPLITS WITH SAME PREDICTION



Suppose we split at vertical, dashed line. Then  $\hat{p}_{11} = 0.75$ .

What happens if we were to now split  $R_1$  at  $X_2 = 0.5$ ?

# TREES IN R

Create a basic, unpruned tree:

```
require(tree)
out.tree = tree(Y~.,data=X,split='gini')
plot(out.tree)
text(out.tree)
```

(There is also the **rpart** package as well)

# TREES IN R

Prune the tree via **cross-validation**

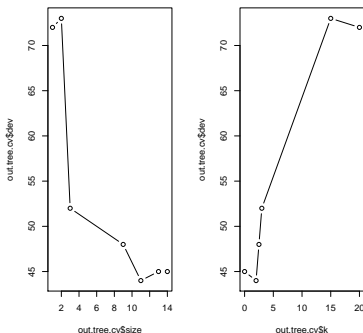
```
out.tree.orig = tree(Y~.,data=X)
out.tree.cv   = cv.tree(out.tree.orig,FUN=prune.misclass)
> names(out.tree.cv)
[1] "size"    "dev"     "k"       "method"
```

# TREES IN R

Prune the tree via **cross-validation**

```
plot(out.tree.cv$size,out.tree.cv$dev,type="b")
```

```
plot(out.tree.cv$k,out.tree.cv$dev,type="b")
```



NOTE:

**k** corresponds to  $\lambda$  in weakest-link pruning.

**dev** means missclassifications in **cv.tree**

# TREES IN R

Prune the tree via cross-validation

```
best.size = out.tree.cv$size[which.min(out.tree.cv$dev)]  
> best.size  
[1] 11  
out.tree = prune.misclass(out.tree.orig,best=best.size)  
class.tree = predict(out.tree,X_0,type='class')
```

# AN INTRODUCTORY EXAMPLE

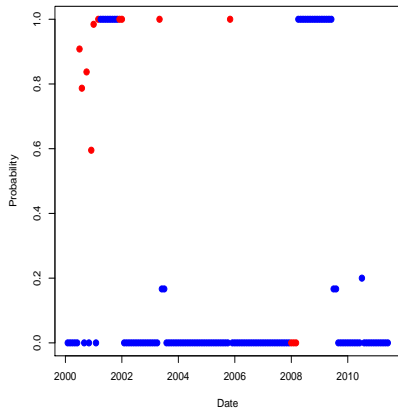
**EXAMPLE:** Use macroeconomic data to predict recessions

We will use data from 1960 through 1999 as **training data**

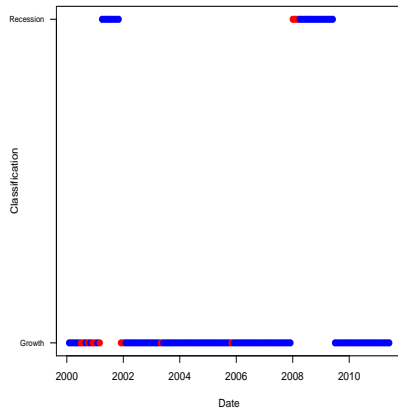
We will use data from 2000 through 2011 as **testing data**

(In the following plots: **correct** vs. **incorrect** classifications)

# 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 the human decision-making process.
- + Trees can easily be displayed graphically no matter the dimension of the data.
- + Trees can easily handle qualitative features 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.