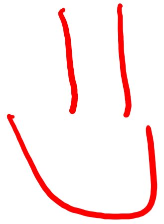
Announcements



Midterm

- Grading over the next few days
- Scores will be included in mid-semester grades

Assignments:

- HW6
  - Out late tonight
  - Due date Tue, 3/24, 11:59 pm

Plan

Last time

- Nearest Neighbor Classification
  - kNN
  - Non-parametric vs parametric

Today

- Decision Trees!

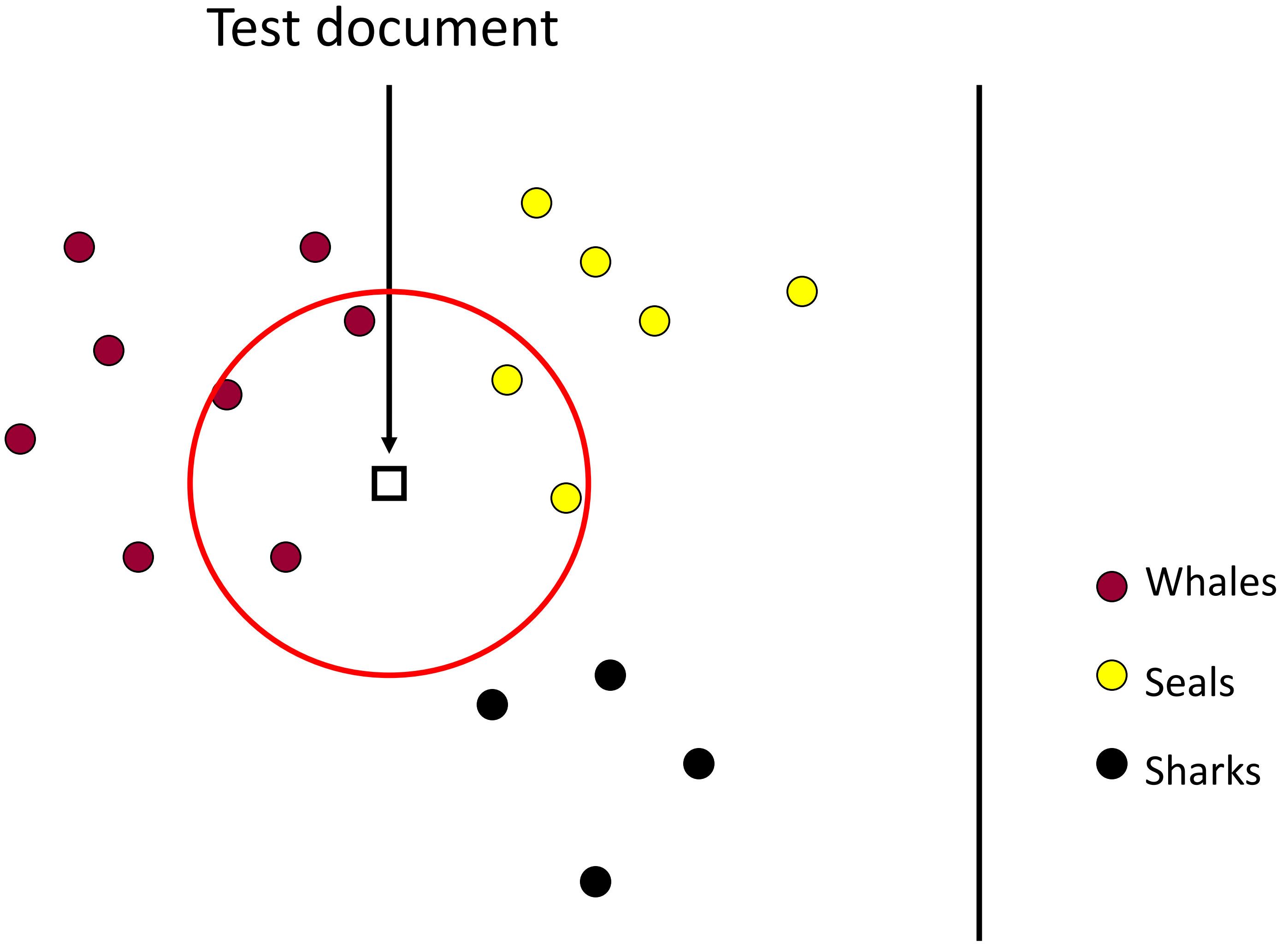
Introduction to Machine Learning

Decision Trees

Instructor: Pat Virtue

k-NN classifier (k=5)

$\frac{3}{5}$ $\frac{2}{5}$



Slide credit: CMU MLD Aarti Singh

k-Nearest Neighbor Classification

Given a training dataset $\mathcal{D} = \{y^{(n)}, \boldsymbol{x}^{(n)}\}_{n=1}^{N}$, $y \in \{1, \dots, C\}$, $\boldsymbol{x} \in \mathbb{R}^m$

and a test input $\boldsymbol{x}_{test}$, predict the class label, $\hat{y}_{test}$:

1) Find the closest k points in the training data to $\boldsymbol{x}_{test}$

$$\mathcal{N}_k(\boldsymbol{x}_{test}, \mathcal{D})$$

2) Return the class label of that closest point

$$\hat{y}_{test} = \underset{c}{\operatorname{argmax}}\, p(Y = c \mid \boldsymbol{x}_{test}, \mathcal{D}, k)$$

$$= \underset{c}{\operatorname{argmax}} \frac{1}{k} \sum_{i \in \mathcal{N}_k(\boldsymbol{x}_{test}, \mathcal{D})} \mathbb{I}\left(y^{(i)} = c\right)$$

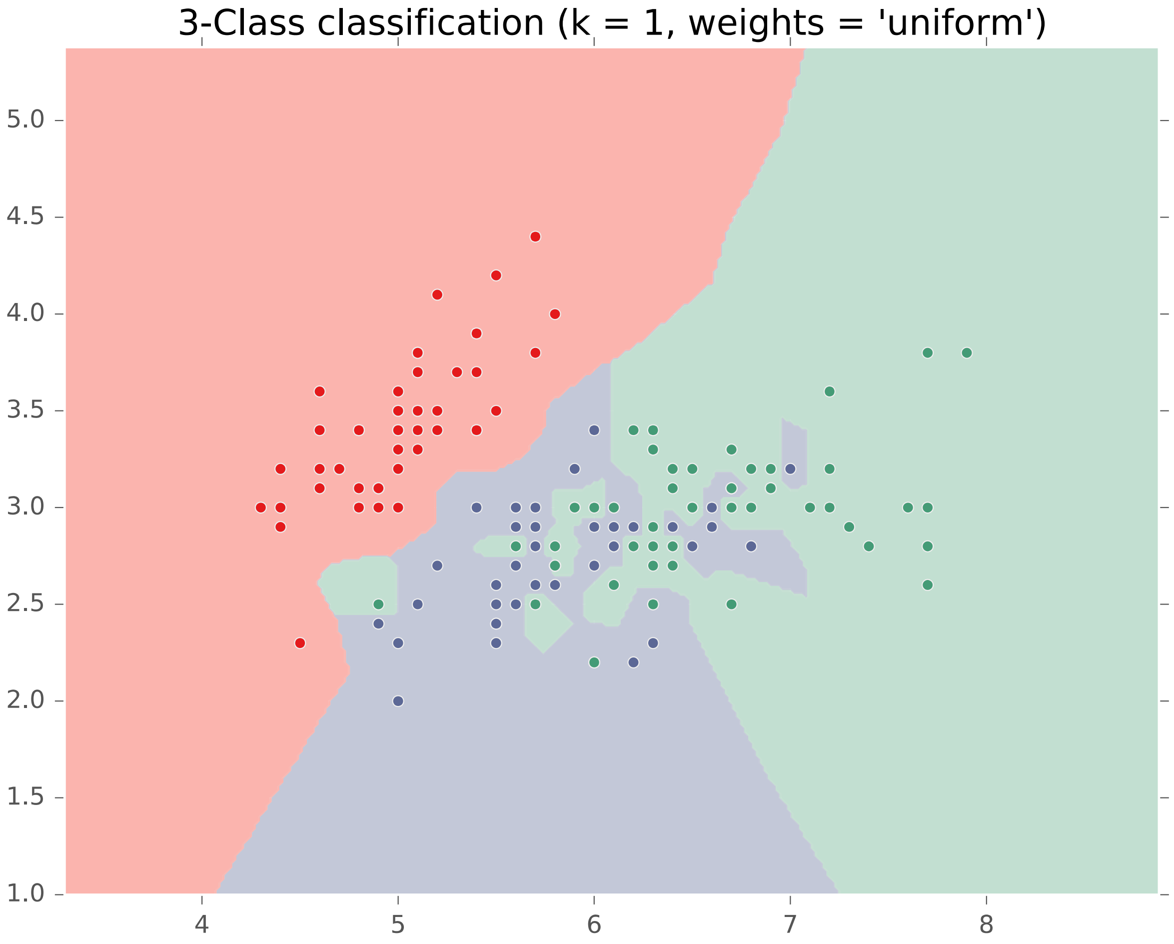
$$\boxed{= \underset{c}{\operatorname{argmax}} \frac{k_c}{k},}$$

where k_c is the number of the k-neighbors with class label c

Slide credit: Murphy, 1.4

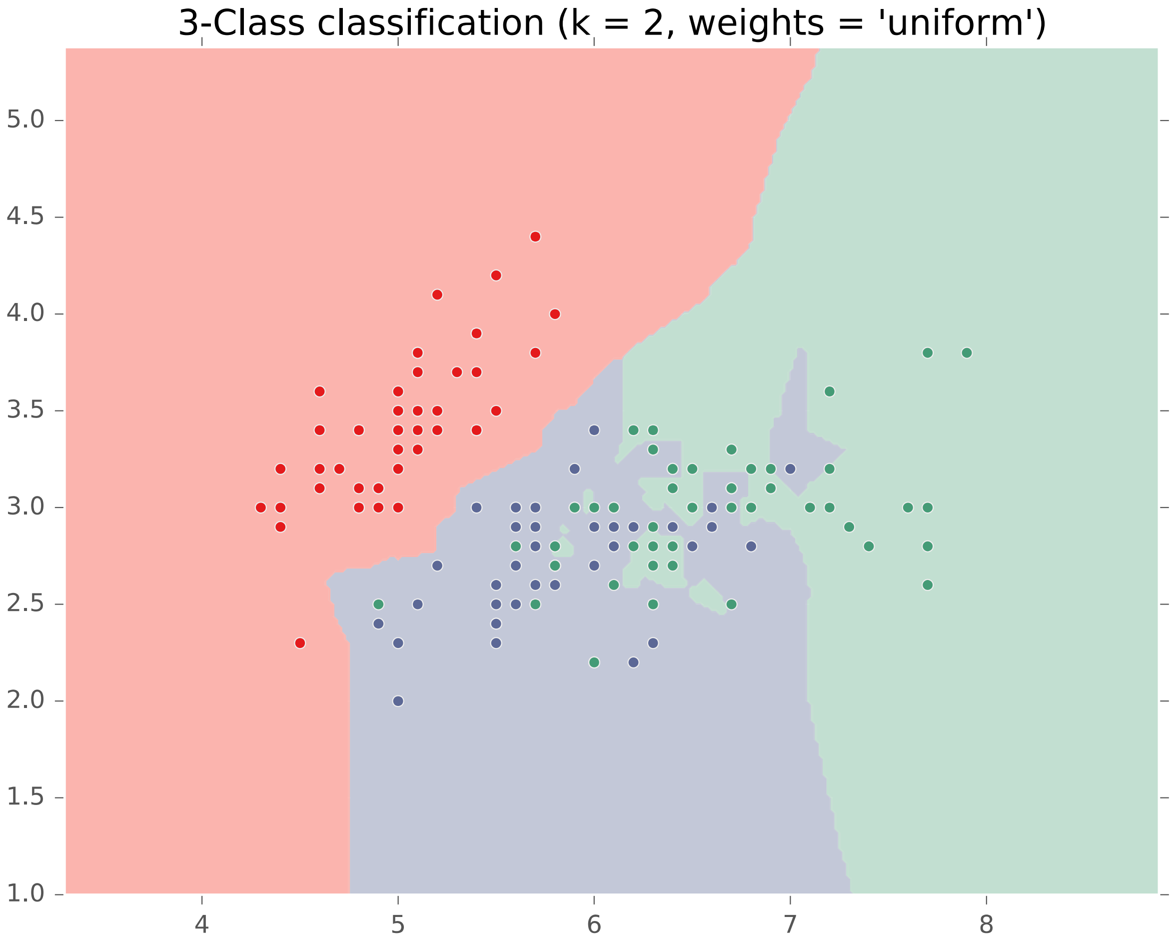
k-NN on Fisher Iris Data

Special Case: Nearest Neighbor



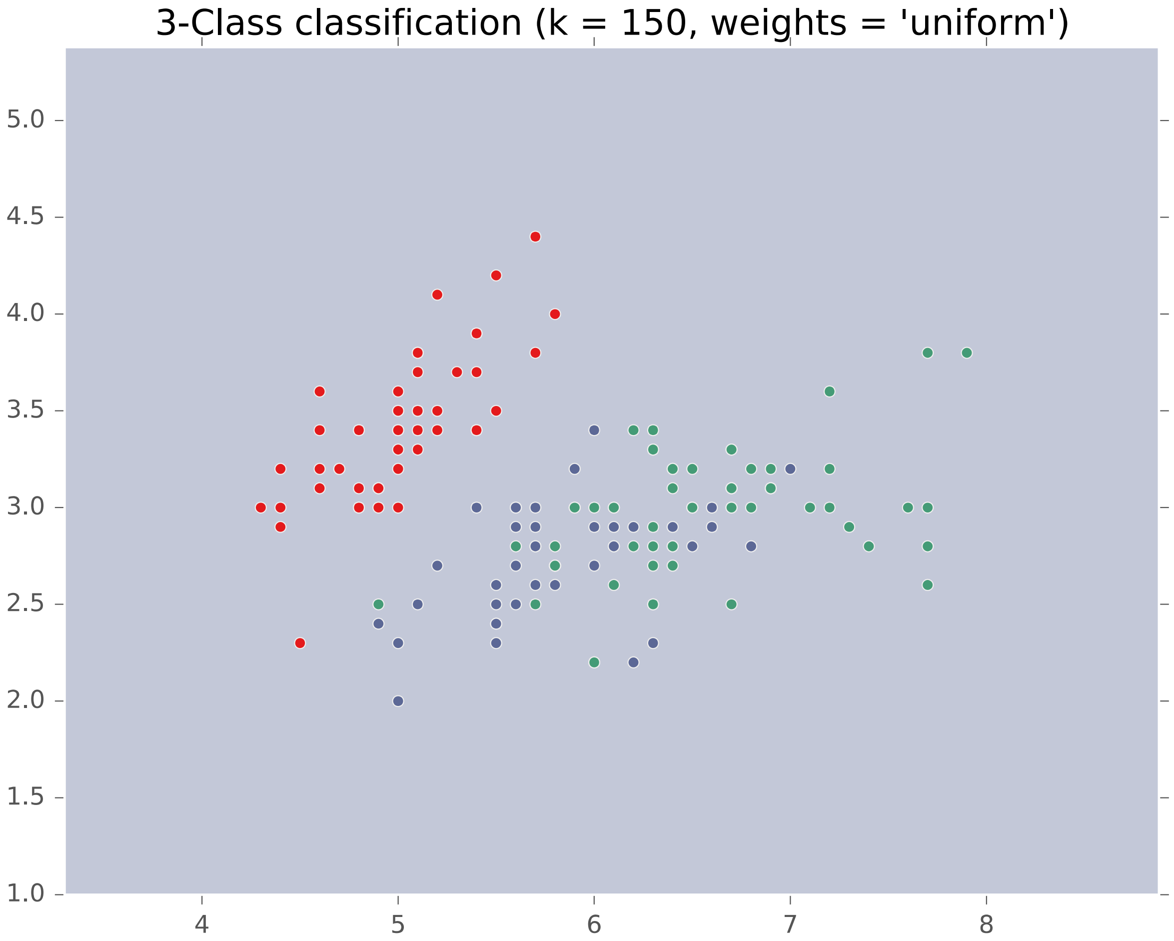
Slide credit: CMU MLD Matt Gormley

k-NN on Fisher Iris Data



k-NN on Fisher Iris Data

Special Case: Majority Vote



Slide credit: CMU MLD Matt Gormley

Decision Trees

First a few tools

Majority vote:

$$\hat{y} = \underset{c}{\operatorname{argmax}} \frac{N_c}{N}$$

Classification error rate:

$$ErrorRate = \frac{1}{N} \sum_n \mathbb{I}(\underline{y_n \neq \hat{y}_n})$$

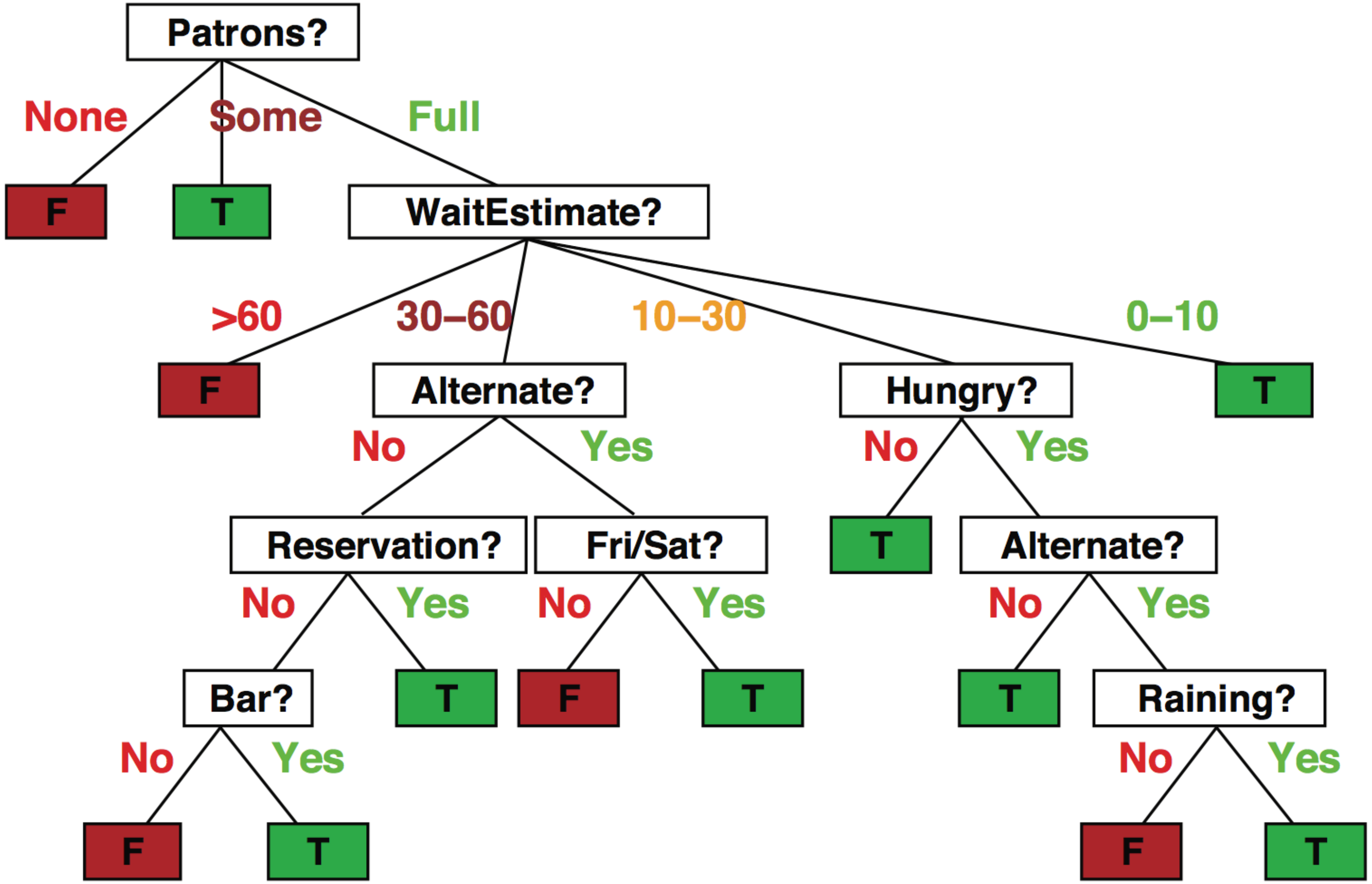
What fraction did we predict incorrectly

Decision trees

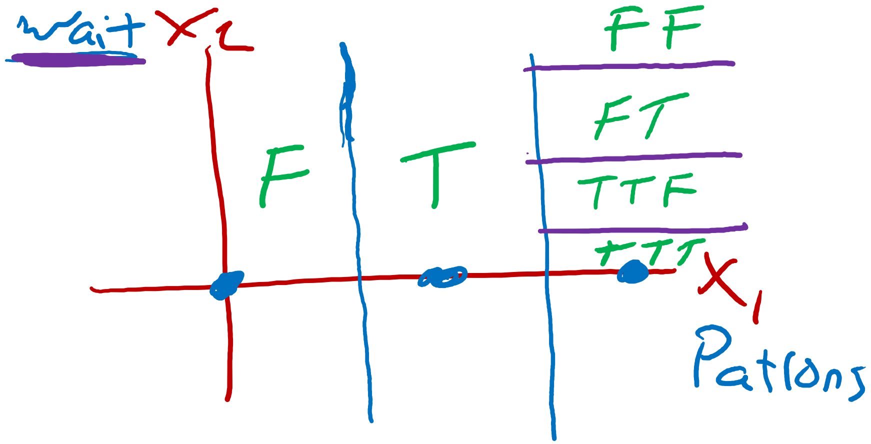
Popular representation for classifiers

- Even among humans!

I've just arrived at a restaurant: should I stay (and wait for a table) or go elsewhere?



Decision trees



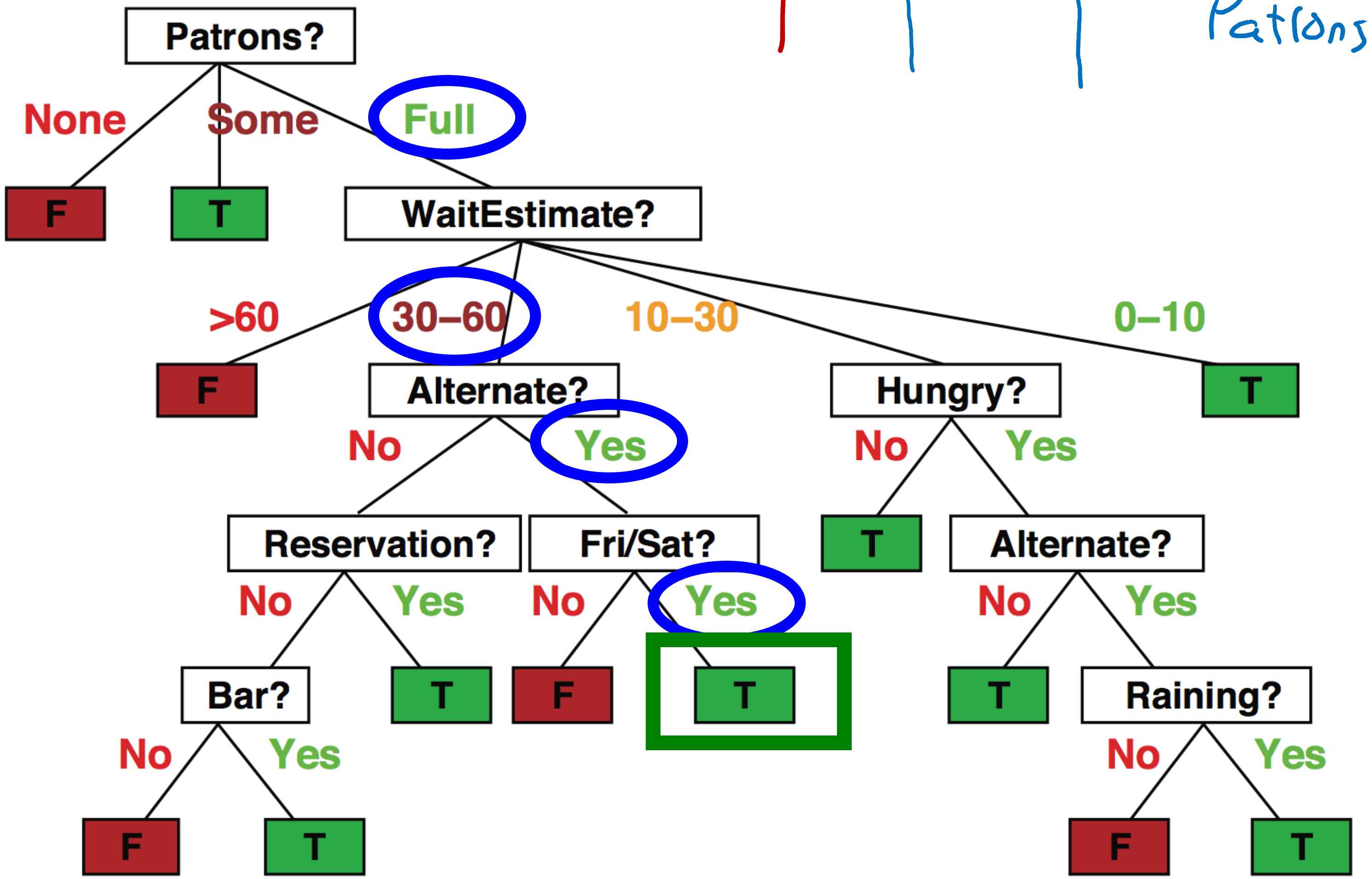
It's Friday night and you're hungry

You arrive at your favorite cheap but really cool happening burger place

It's full up and you have no reservation but there is a bar

The host estimates a 45 minute wait

There are alternatives nearby but it's raining outside



Decision tree ***partitions*** the input space, assigns a label to each partition

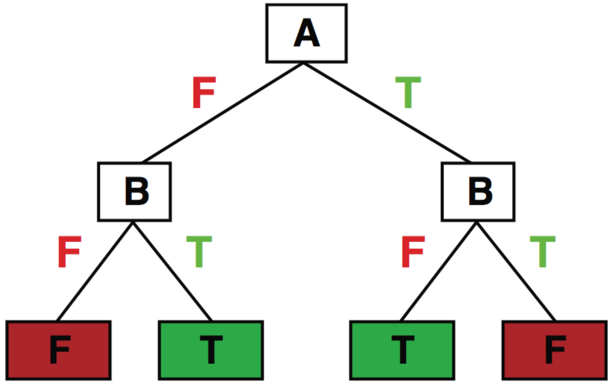
Slide credit: ai.berkeley.edu

Expressiveness

Discrete decision trees can express ***any function*** of the input

E.g., for Boolean functions, build a path from root to leaf for each row of the truth table:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



True/false: there training set exactly

But a tree that simply records the examples is essentially a lookup table

To get generalization to new examples, need a compact tree

Tree to Predict C-Section Risk

Learned from medical records of 1000 women

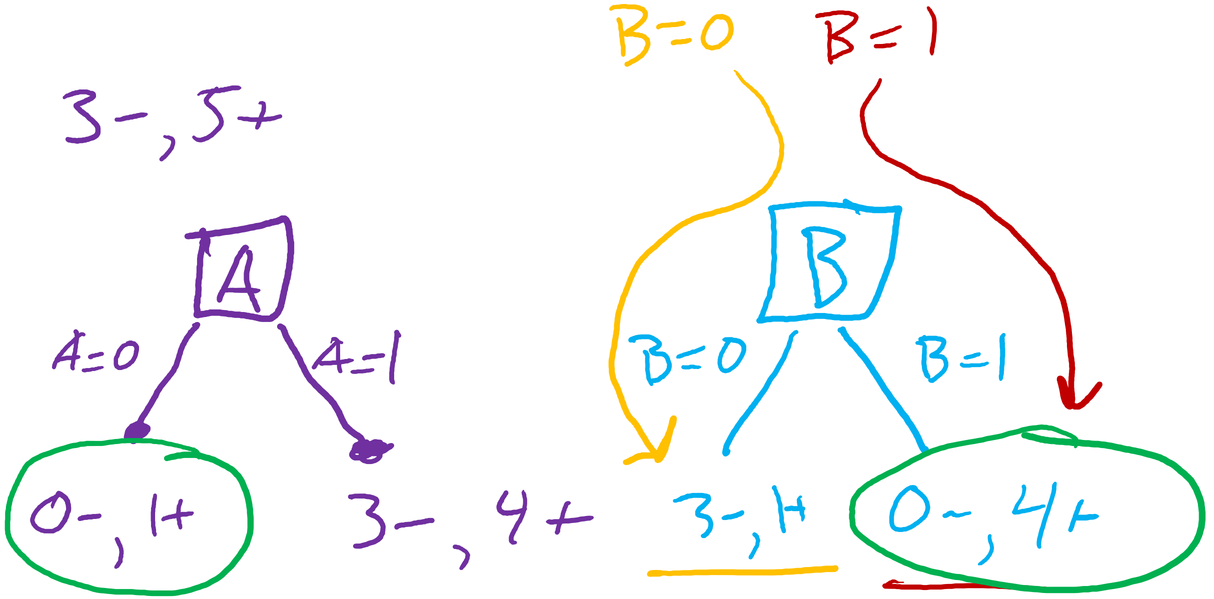
Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

Slide credit: CMU MLD Tom Mitchell

Decision Stumps

Split data based on a single attribute



return - return +

$\frac{1}{8}(3 + 4)$ accuracy

$\frac{1}{8}(1 + 0)$ error rate

Dataset:

Output Y, Attributes A, B, C

Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Piazza Poll 1

Splitting on which attribute {A, B, C} creates a decision stump with the lowest training error?

B

Dataset:

Output Y, Attributes A, B, C

Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Building a decision tree

```
Function BuildTree(n,A)    // n: samples, A: set of attributes
   If empty(A) or all n(L) are the same
       status = leaf
       class = most common class in n(L)
   else
       status = internal
       a ⇐ bestAttribute(n,A)  ?
       LeftNode = BuildTree(n(a=1), A \ {a})
       RightNode = BuildTree(n(a=0), A \ {a})
    end
end
```

Slide credit: CMU MLD Ziv Bar-Joseph

Building a decision tree

```
Function BuildTree(n,A)    // n: samples, A: set of attributes
   If empty(A) or all n(L) are the same
       status = leaf
       class = most common class in n(L)
   else
       status = internal
       a ⇐ bestAttribute(n,A)
       LeftNode = BuildTree(n(a=1), A \ {a})
       RightNode = BuildTree(n(a=0), A \ {a})
    end
end
```

n(L): Labels for samples in this set

Decision: Which attribute?

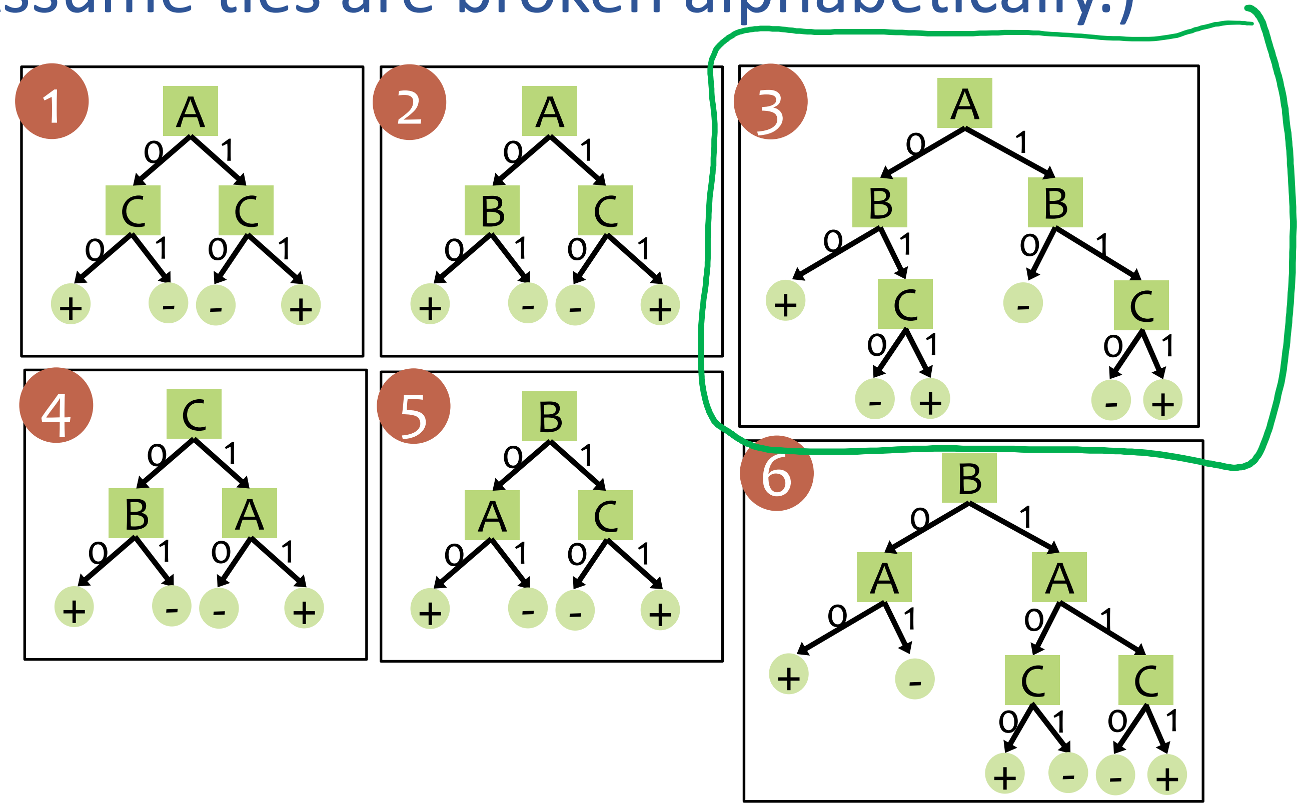
Recursive calls to create left and right subtrees, n(a=1) is the set of samples in n for which the attribute a is 1

Slide credit: CMU MLD Ziv Bar-Joseph

Piazza Poll 2

Which of the following trees would be learned by the the decision tree learning algorithm using "error rate" as the splitting criterion?

(Assume ties are broken alphabetically.)



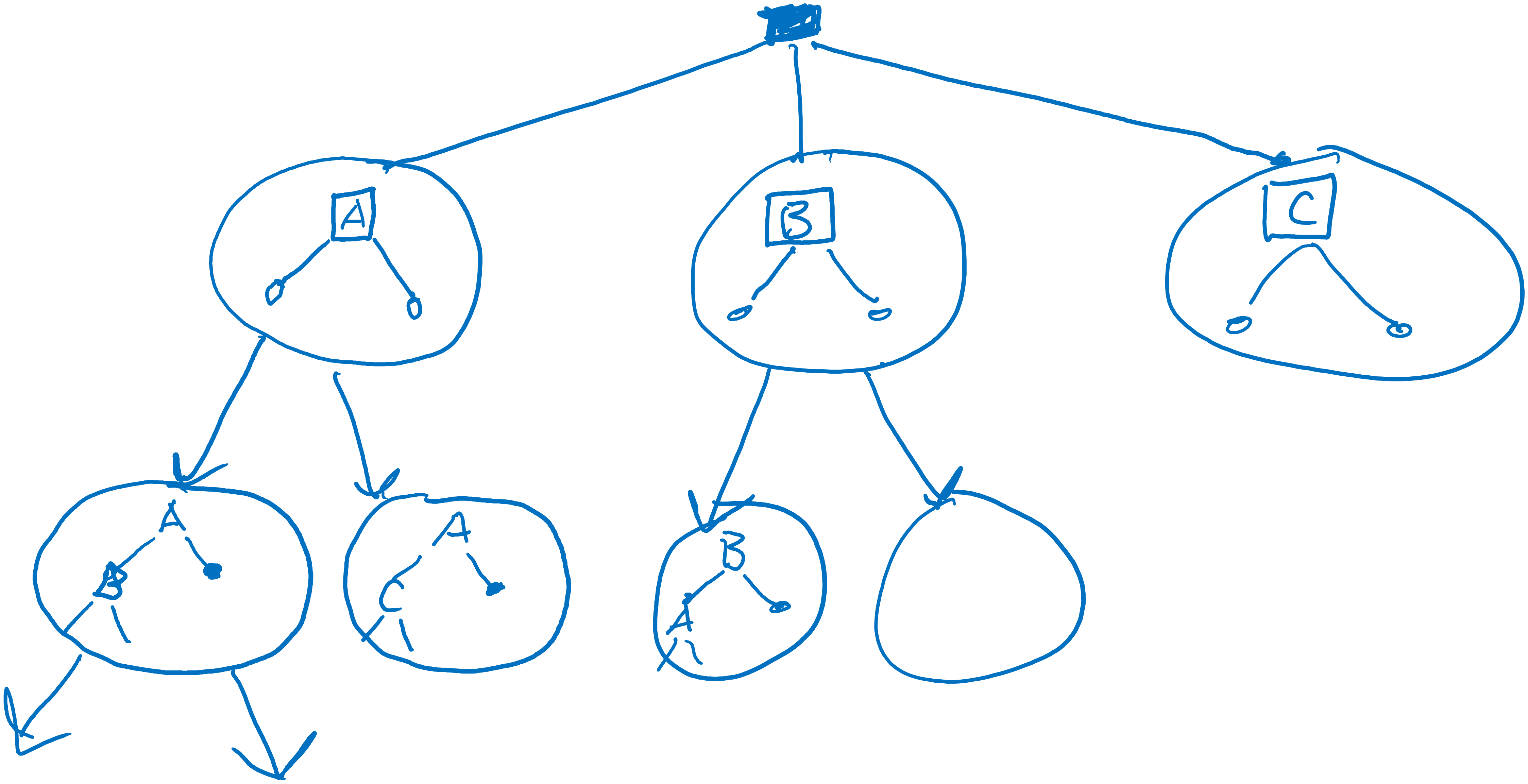
Dataset:

Output Y, Attributes A, B, C

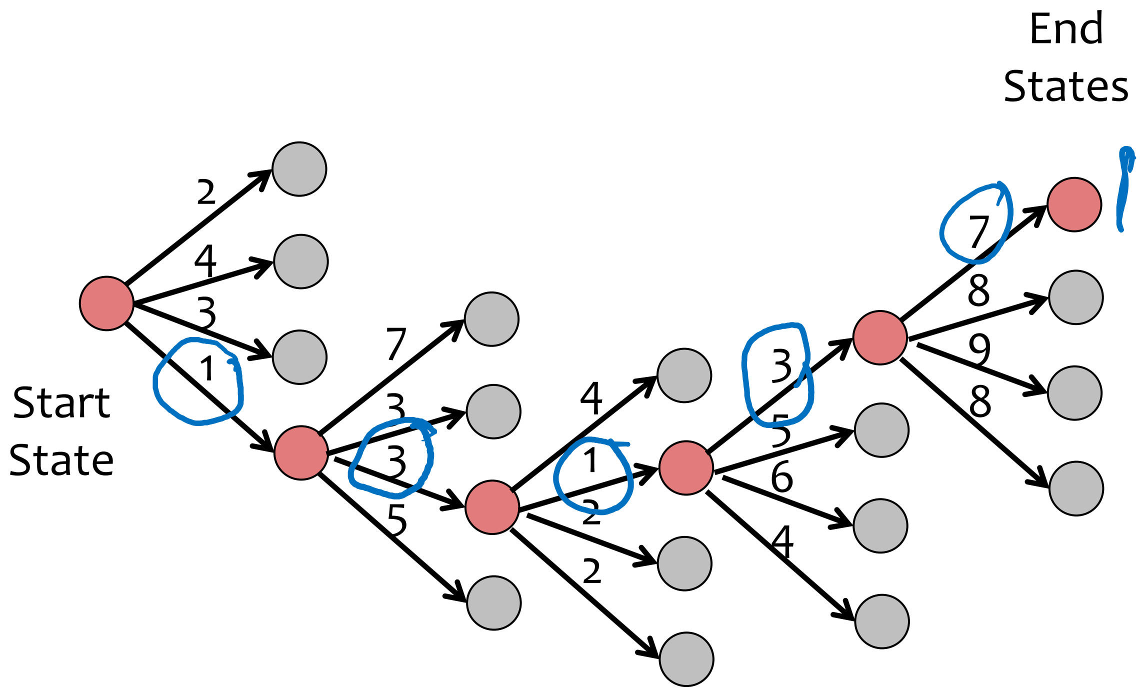
Y	A	B	C
+	0	0	0
+	0	0	1
-	0	1	0
+	0	1	1
-	1	0	0
-	1	0	1
-	1	1	0
+	1	1	1

Slide credit: CMU MLD Matt Gormley

Decision Trees as a Search Problem



Background: Greedy Search



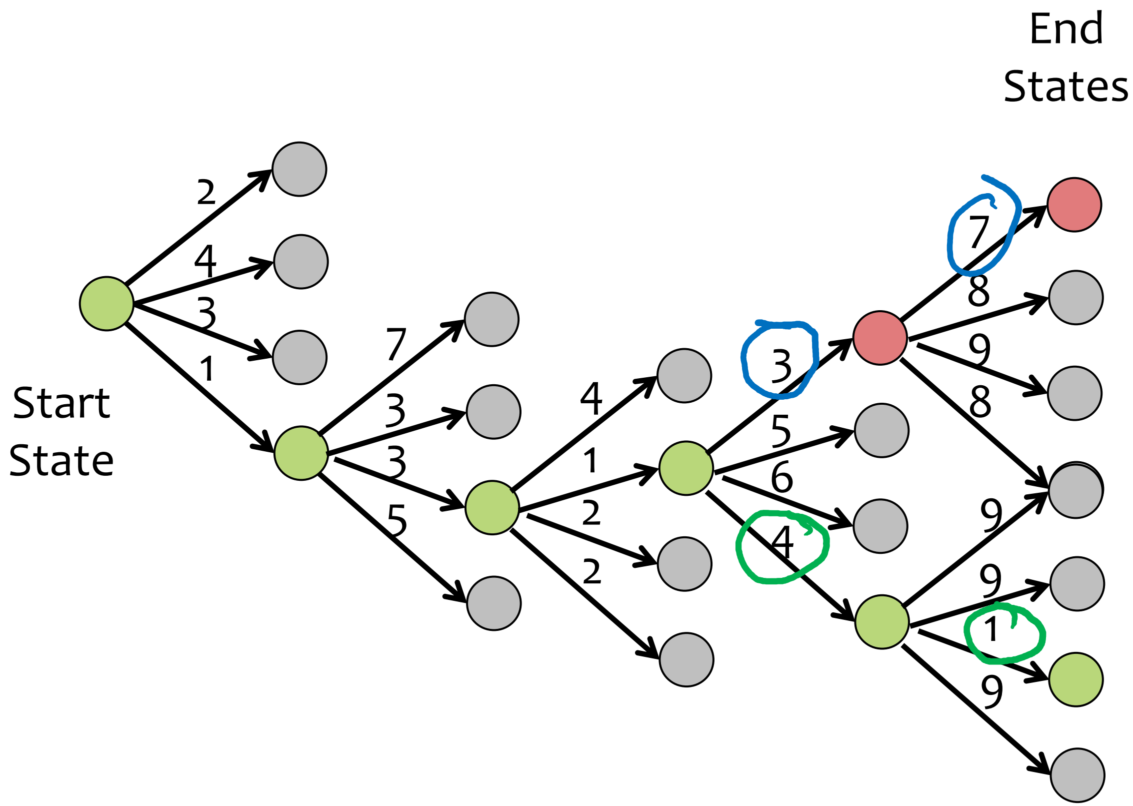
Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- Heuristic method of search (i.e. does *not* necessarily find the best path)

Background: Greedy Search



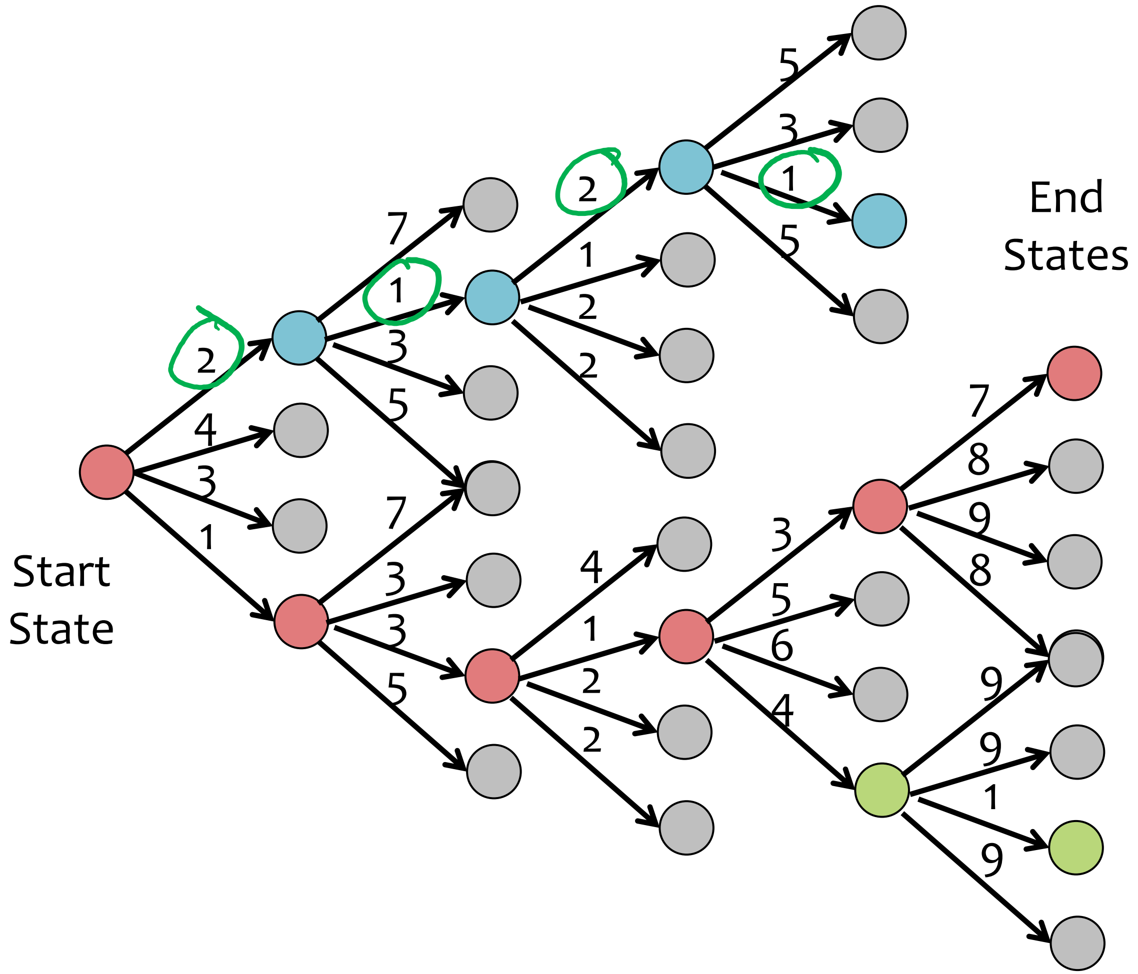
Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- Heuristic method of search (i.e. does *not* necessarily find the best path)

Background: Greedy Search



Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

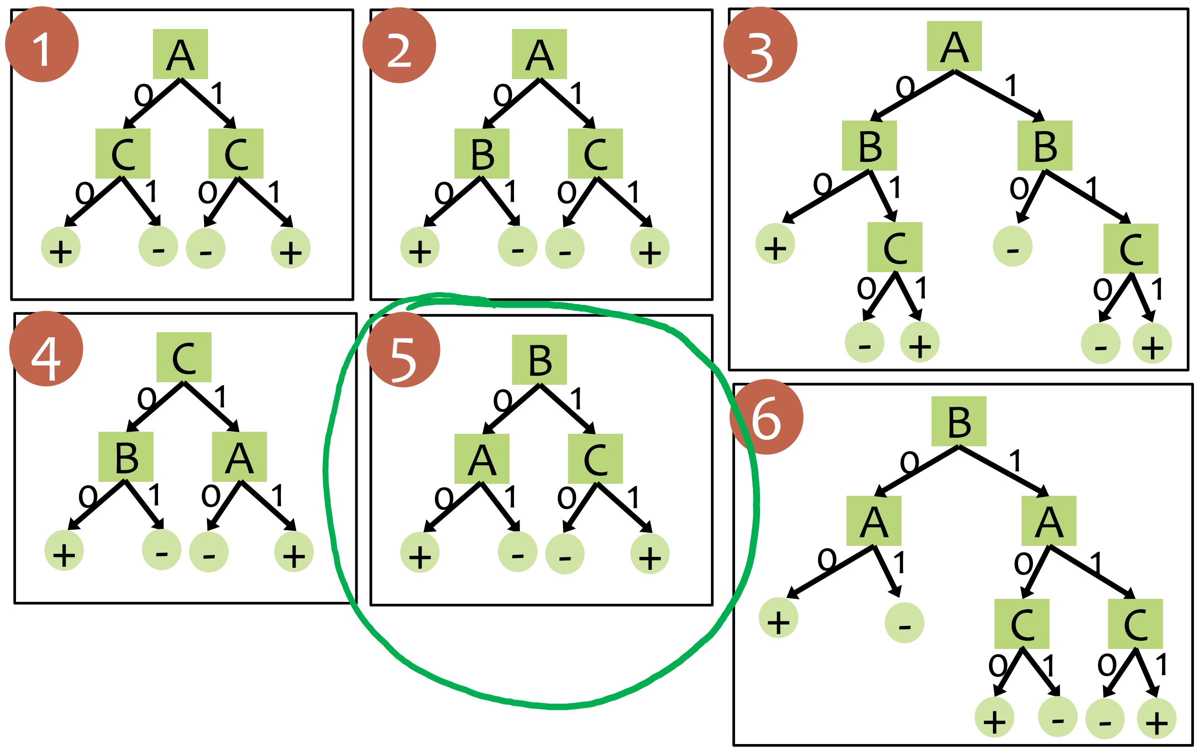
Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- Heuristic method of search (i.e. does *not* necessarily find the best path)

Piazza Poll 3

Suppose you had an algorithm that found **the tree with lowest training error that was as small as possible (i.e. exhaustive global search)**, which tree would it return?

(Assume ties are broken alphabetically.)



Dataset:

Output Y, Attributes A, B, C

Y	A	B	C
+	0	0	0
+	0	0	1
-	0	1	0
+	0	1	1
-	1	0	0
-	1	0	1
-	1	1	0
+	1	1	1

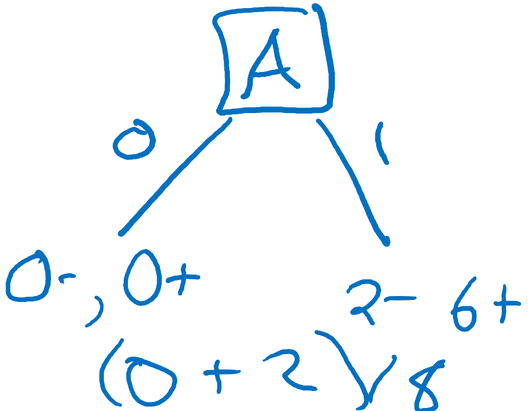
Piazza Poll 4

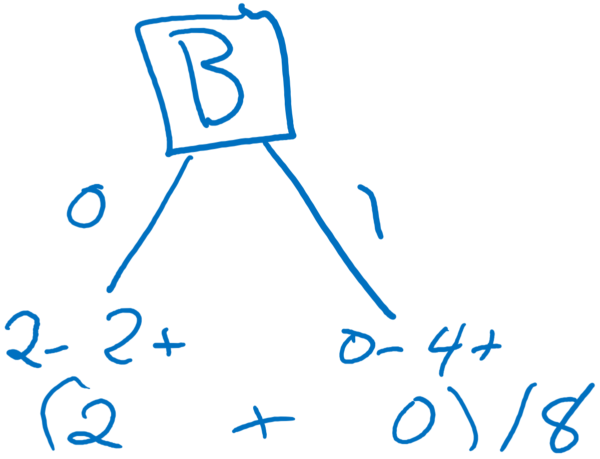
Which attribute {A, B} would error rate select for the next split?

1) A
2) B
3) A or B (tie)
4) I don't know

Dataset:
Output Y, Attributes A and B

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1





Building a decision tree

```
Function BuildTree(n,A)    // n: samples, A: set of attributes
   If empty(A) or all n(L) are the same
       status = leaf
       class = most common class in n(L)
   else
       status = internal
       a ⇐ bestAttribute(n,A)
       LeftNode = BuildTree(n(a=1), A \ {a})
       RightNode = BuildTree(n(a=0), A \ {a})
    end
end
```

n(L): Labels for samples in this set

Decision: Which attribute?

Recursive calls to create left and right subtrees, n(a=1) is the set of samples in n for which the attribute a is 1

Slide credit: CMU MLD Ziv Bar-Joseph

Identifying ‘bestAttribute’

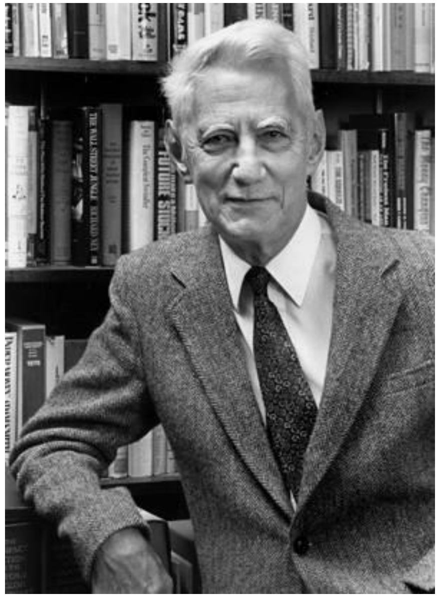
There are many possible ways to select the best attribute for a given set.

We will discuss one possible way which is based on information theory.

Entropy

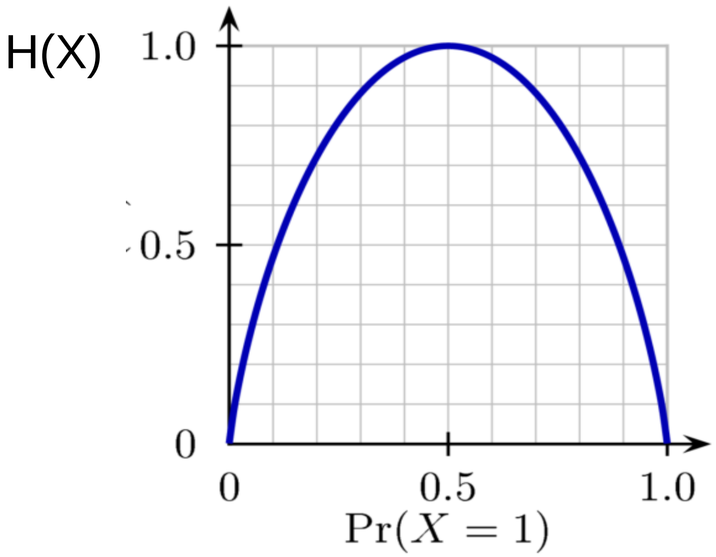
- Quantifies the amount of uncertainty associated with a specific probability distribution
- The higher the entropy, the less confident we are in the outcome
- Definition

$$H(X) = \sum_{c} - p(X = c) \log_2 p(X = c)$$



Claude Shannon (1916 – 2001), most of the work was done in Bell labs

Entropy



- Definition

$$H(X) = \sum_i - p(X = i)\log_2 p(X = i)$$

- So, if P(X=1) = 1 then

$$H(X) = -p(x=1)\log_2 p(X=1) - p(x=0)\log_2 p(X=0)$$
$$= -1\log 1 - 0\log 0 = 0$$

- If P(X=1) = .5 then

$$H(X) = -p(x=1)\log_2 p(X=1) - p(x=0)\log_2 p(X=0)$$
$$= -.5\log_2 .5 - .5\log_2 .5 = -\log_2 .5 = 1$$

Mutual Information

Let X be a random variable with $X \in \mathcal{X}$.
Let Y be a random variable with $Y \in \mathcal{Y}$.

$$\text{Entropy: } H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$$

$$\text{Specific Conditional Entropy: } H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$$

$$\text{Conditional Entropy: } H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$$

$$\text{Mutual Information: } I(Y; X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$$

- For a decision tree, we can use **mutual information** of the output class Y and some attribute X on which to split **as a splitting criterion**
- Given a dataset D of training examples, we can estimate the required probabilities as…

$$P(Y = y) = N_{Y=y}/N$$
$$P(X = x) = N_{X=x}/N$$
$$P(Y = y|X = x) = N_{Y=y,X=x}/N_{X=x}$$

where $N_{Y=y}$ is the number of examples for which $Y = y$ and so on.

Slide credit: CMU MLD Matt Gormley

Mutual Information

Let X be a random variable with $X \in \mathcal{X}$.
Let Y be a random variable with $Y \in \mathcal{Y}$.

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

- **Entropy** measures the **expected # of bits** to code one random draw from X.
- For a decision tree, we want to **reduce the entropy of the random variable we are trying to predict**!

Conditional entropy is the expected value of specific conditional entropy $E_{P(X=x)}[H(Y \mid X = x)]$

Informally, we say that **mutual information** is a measure of the following:
If we know X, how much does this reduce our uncertainty about Y?

Slide credit: CMU MLD Matt Gormley

Decision Tree Learning Example

Dataset:

Output Y, Attributes A and B

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

Which attribute would **mutual information** select for the next split?

1. A
2. B
3. A or B (tie)
4. Neither

Slide credit: CMU MLD Matt Gormley

Decision Tree Learning Example

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y=y) \log_2 P(Y=y)$

Specific Conditional Entropy: $H(Y \mid X=x) = -\sum_{y \in \mathcal{Y}} P(Y=y \mid X=x) \log_2 P(Y=y \mid X=x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X=x) H(Y \mid X=x)$

Mutual Information: $I(Y;X) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

Y	A	B
-	1	0
-	1	0
+	1	0
+	1	0
+	1	1
+	1	1
+	1	1
+	1	1

Slide credit: CMU MLD Matt Gormley