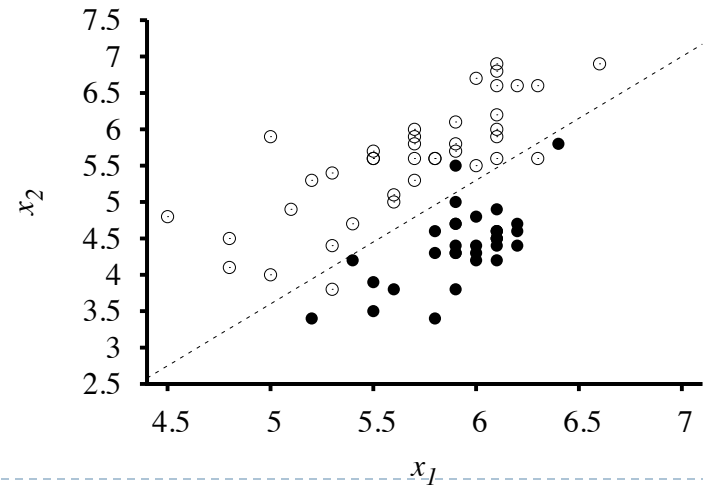
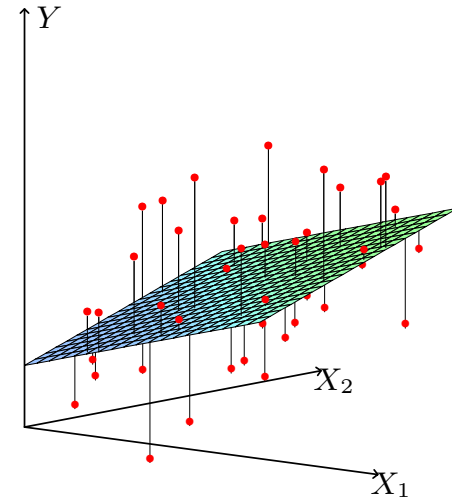


Decision Trees

CS 341 – Lectures 8/9
Dan Sheldon

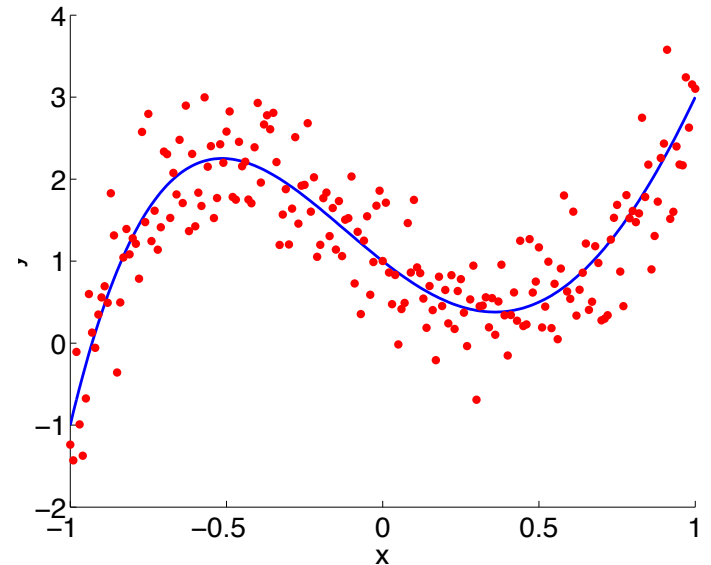
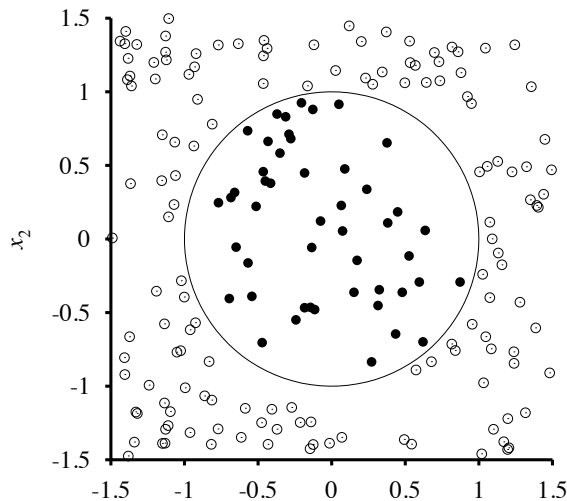
Review: Linear Methods

- ▶ So far, we've looked at *linear* methods
- ▶ Linear regression
 - ▶ Fit a line/plane/hyperplane
- ▶ Logistic regression
 - ▶ Decision boundary is a line/plane/hyperplane



Review: Feature Expansions

- ▶ Non-linearity by trickery
- ▶ **Drawback:** not automatic
→ hard work for us!



$$x \mapsto (1, x, x^2, x^3, \dots)$$

$$(x_1, x_2) \mapsto (1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$$

Next Up: Non-Linear Methods

- ▶ Hypothesis class is intrinsically non-linear
- ▶ Next few topics
 - ▶ Decision trees
 - ▶ Neural networks
 - ▶ Kernels
- ▶ Today: decision trees



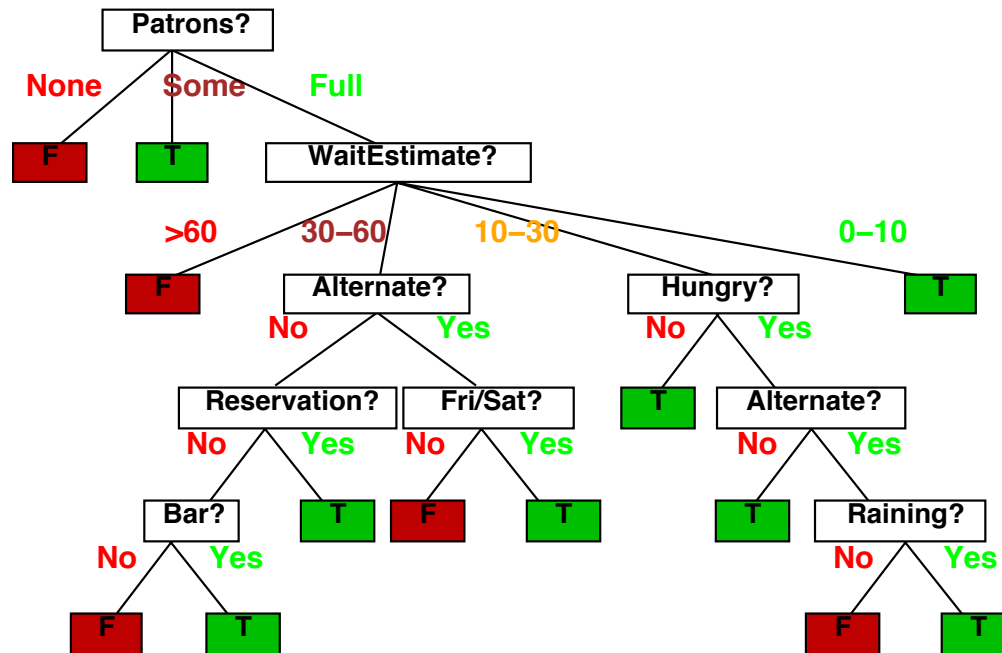
Decision Tree Topics

- ▶ What is a decision tree?
- ▶ Learn a tree from data
 - ▶ Good vs. bad trees
 - ▶ Greedy top-down learning
 - ▶ Entropy
- ▶ Pruning
- ▶ Geometric interpretation



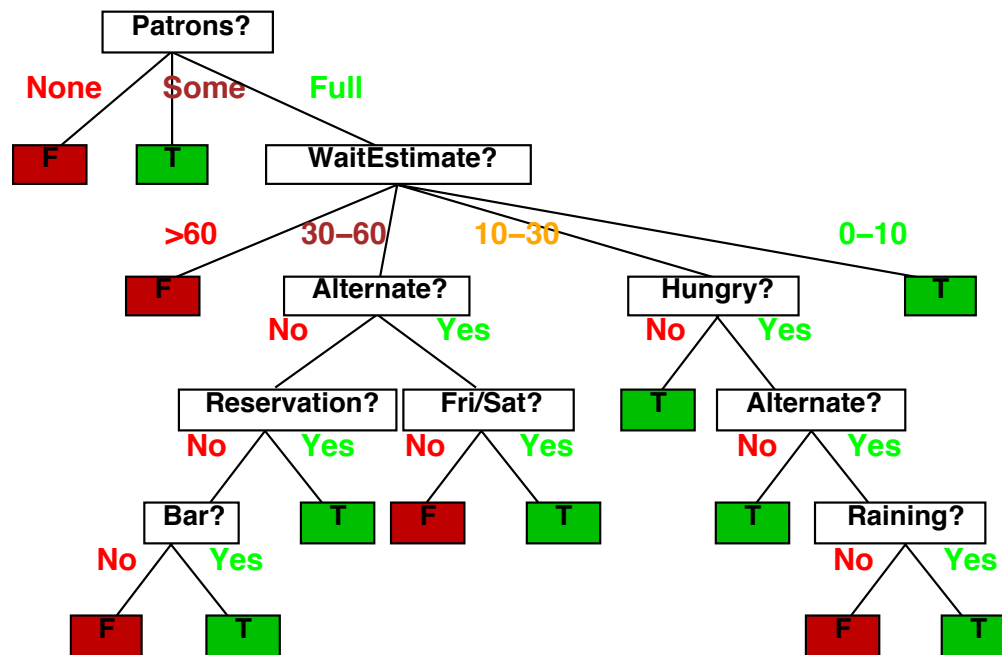
What Is a Decision Tree?

- ▶ Example from R&N: wait for table at restaurant?



What Is a Decision Tree?

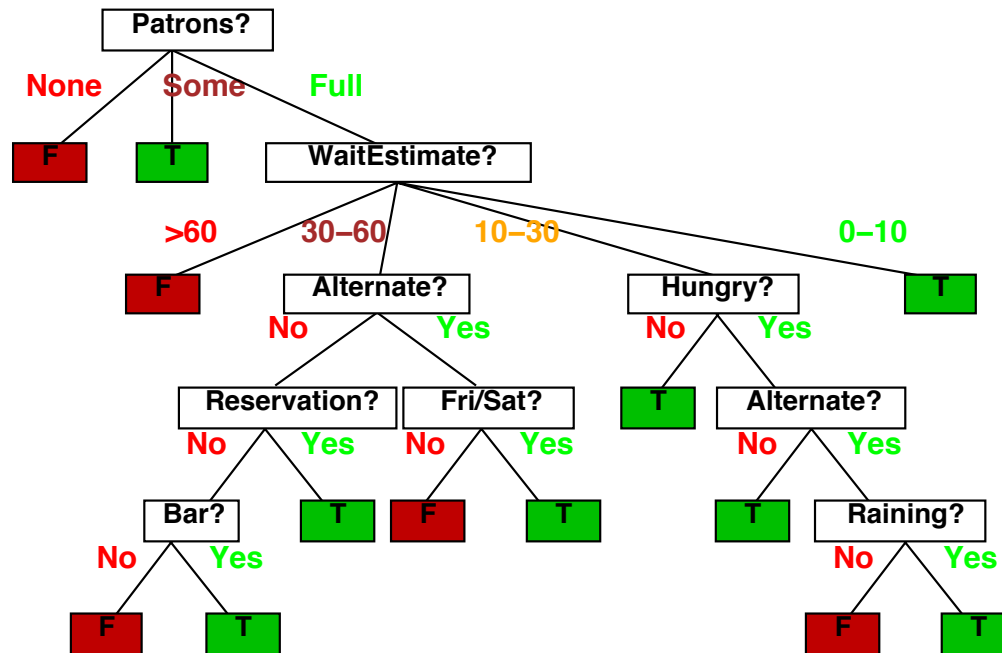
- ▶ Internal nodes: test an **attribute** (feature)
 - ▶ Assume discrete attributes for now
- ▶ Branches: different values of attribute
- ▶ Leaf node: predict a **class label**



What Is a Decision Tree?

- Predict by following path down tree

Example	Attributes										Target Wait?
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
X_4	<i>T</i>	<i>F</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>10–30</i>	<i>T</i>



Decision Trees in the Wild

- ▶ Decision trees are common decision-making tools
 - ▶ You have all seen them!
- ▶ R&N: Car service manuals
- ▶ Viburnum ID
 - ▶ <http://www.hort.cornell.edu/vlb/key/index.html>



[Image: Clemson cooperative extension]

Decision Trees in the Wild

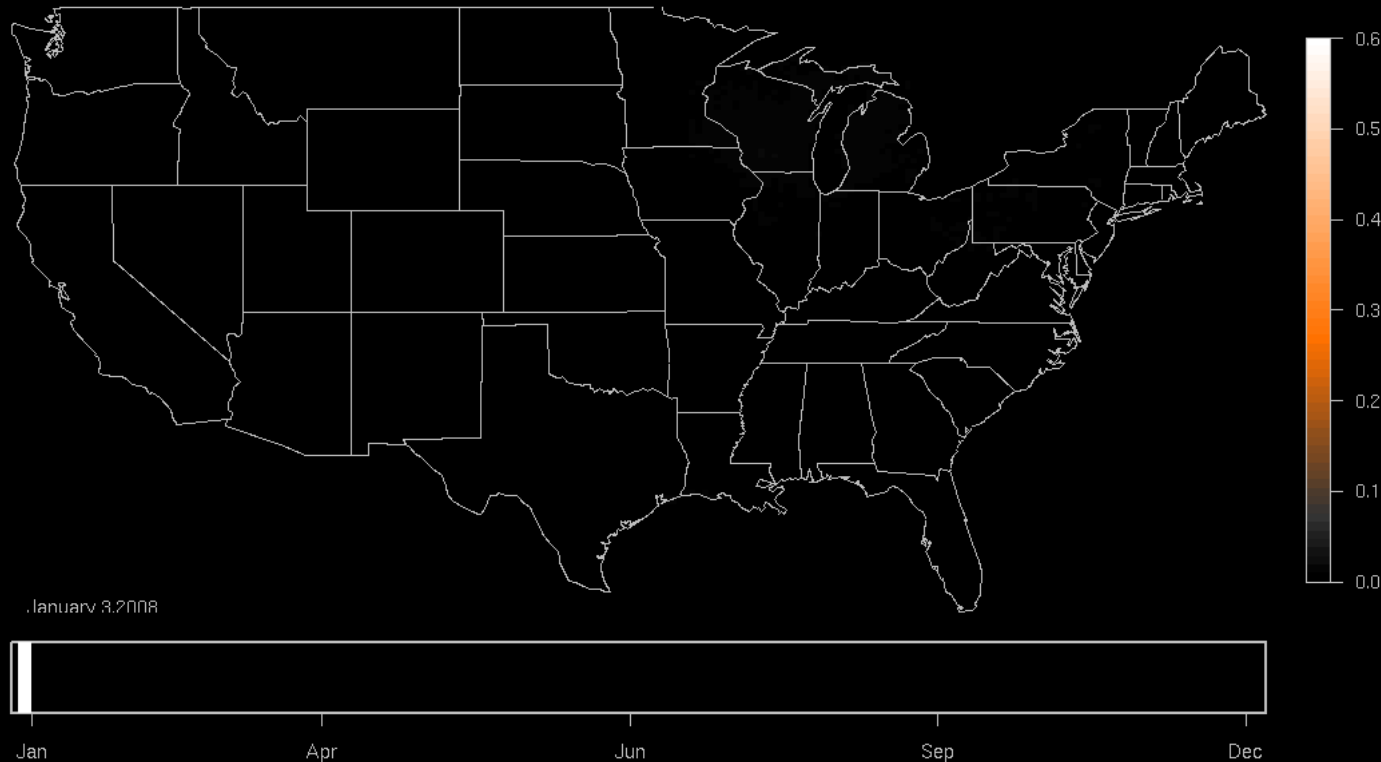
► C-section risk (learned from data)

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+ .05-
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+ .22-
| | | 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-
```

Decision Trees in the Wild



eBird



Why Decision Trees?

- ▶ Easy to understand
 - ▶ Match human decision-making processes
- ▶ Excellent performance
 - ▶ One of the best out-of-the-box methods when used carefully
- ▶ Very flexible hypothesis space
 - ▶ Can learn **complex** decision boundaries
- ▶ Handle messy data well
 - ▶ Missing features
 - ▶ Continuous discrete
 - ▶ Etc.



How to Learn a Tree From Data

► Training data for R&N example

Example	Attributes										Target Wait?
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
X_1	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0–10	T
X_2	Yes	No	No	Yes	Full	\$	No	No	Thai	30–60	F
X_3	No	Yes	No	No	Some	\$	No	No	Burger	0–10	T
X_4	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10–30	T
X_5	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	F
X_6	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0–10	T
X_7	No	Yes	No	No	None	\$	Yes	No	Burger	0–10	F
X_8	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0–10	T
X_9	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	F
X_{10}	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10–30	F
X_{11}	No	No	No	No	None	\$	No	No	Thai	0–10	F
X_{12}	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30–60	T

Exercise

- ▶ Design an algorithm
 - ▶ Input: single training example
 - ▶ Output: decision tree that matches that example

Example	Attributes										Target Wait?
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
X_4	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10–30	T



Exercise (cont')

Example	Attributes										Target Wait?
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
X_4	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Thai</i>	<i>10–30</i>	<i>T</i>



Exercise

► Design a tree to match *all* examples

► Hint: use previous exercise

Example	Attributes										Target Wait?
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
<i>X</i> ₁	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0–10	<i>T</i>
<i>X</i> ₂	Yes	No	No	Yes	Full	\$	No	No	Thai	30–60	<i>F</i>
<i>X</i> ₃	No	Yes	No	No	Some	\$	No	No	Burger	0–10	<i>T</i>
<i>X</i> ₄	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10–30	<i>T</i>
<i>X</i> ₅	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	<i>F</i>
<i>X</i> ₆	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0–10	<i>T</i>
<i>X</i> ₇	No	Yes	No	No	None	\$	Yes	No	Burger	0–10	<i>F</i>
<i>X</i> ₈	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0–10	<i>T</i>
<i>X</i> ₉	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	<i>F</i>
<i>X</i> ₁₀	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10–30	<i>F</i>
<i>X</i> ₁₁	No	No	No	No	None	\$	No	No	Thai	0–10	<i>F</i>
<i>X</i> ₁₂	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30–60	<i>T</i>

A Correct Answer

- ▶ For each training example, create a path from root to leaf

- ▶ What is wrong with this?



What Is a Good Tree?

- ▶ Find *smallest* tree that fits training data
- ▶ A very hard problem
 - ▶ Number of trees on d binary attributes *at least*

$$2^{2^d}$$

- ▶ E.g., with 6 attributes

$$2^{2^6} = 18,446,744,073,709,551,616$$

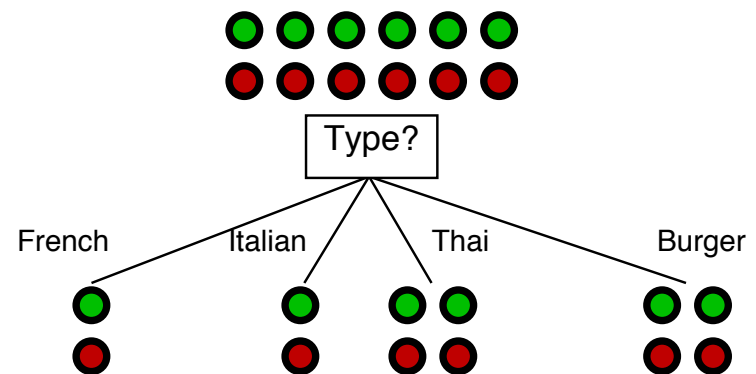
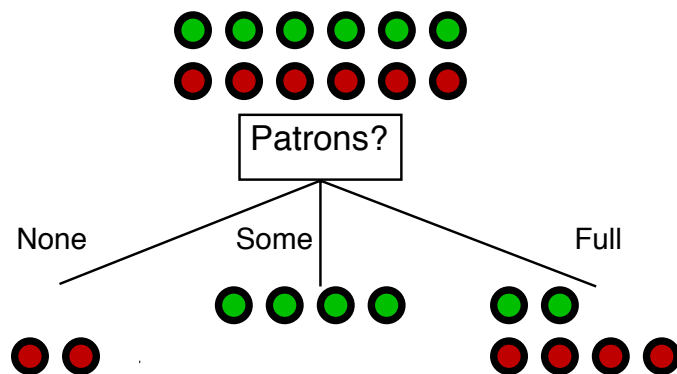
- ▶ Effectively impossible to find *best* tree



Top-Down Greedy Heuristic

► Idea:

- Find “best” attribute to test at root of tree
- Recursively apply algorithm to each branch



► “Best” attribute?

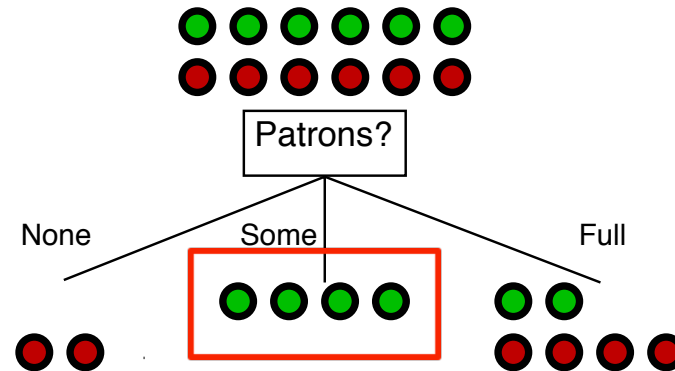
- Ideally, splits the examples into subsets that are "all positive" or "all negative"

Base Cases

- ▶ Keep splitting until
 - ▶ All examples have same label
 - ▶ Run out of examples
 - ▶ Run out of attributes



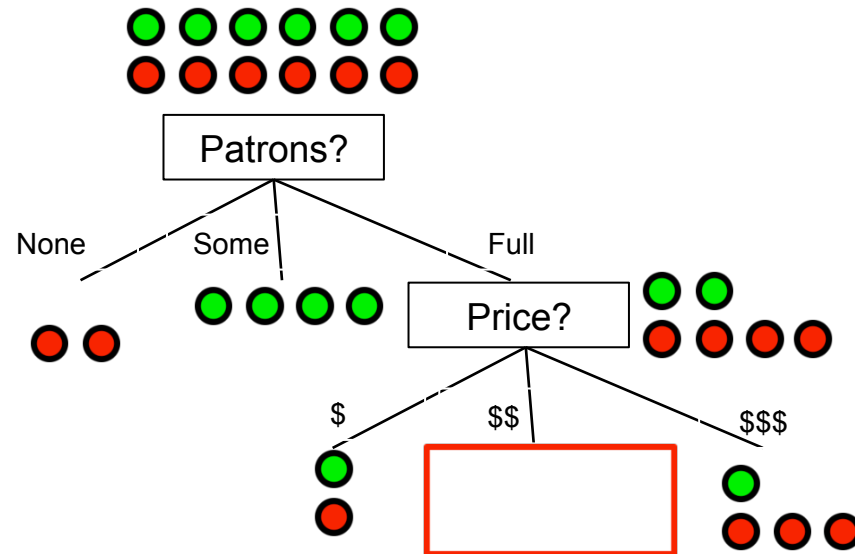
Base Case I



- ▶ All examples have same classification (e.g. true)
 - ▶ Return classification (e.g., true)

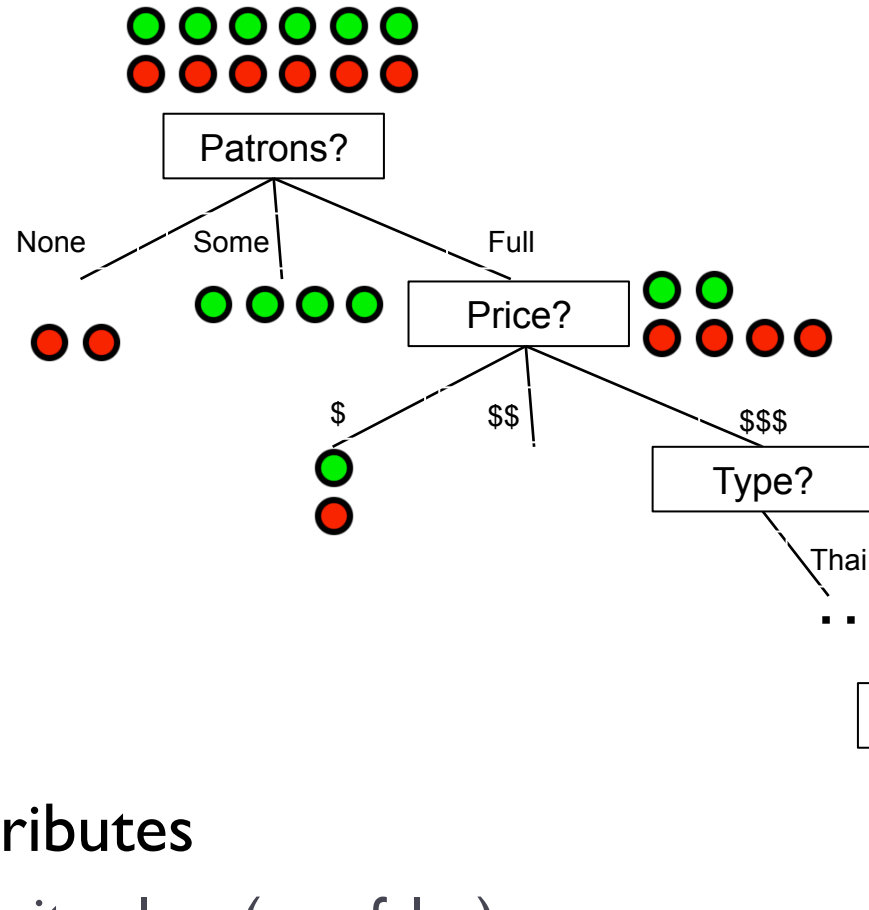


Base Case II



- ▶ No more examples
 - ▶ Return majority class of parent (e.g., false)

Base Case III



- ▶ No more attributes
 - ▶ Return majority class (e.g., false)

The Algorithm

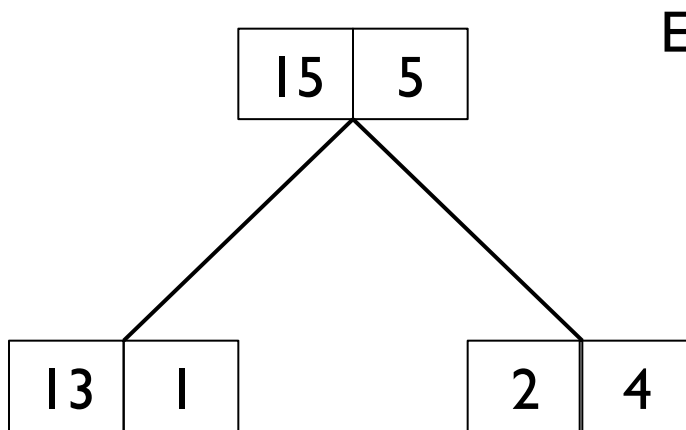
► Put it (almost) all together

```
function DTL(examples, attributes, default) returns a decision tree
  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MODE(examples)
  else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value  $v_i$  of best do
      examplesi ← {elements of examples with best =  $v_i$ }
      subtree ← DTL(examplesi, attributes − best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
  return tree
```

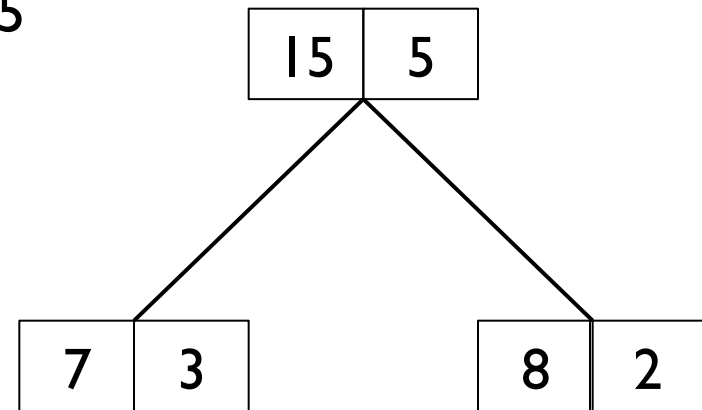


Splitting Criteria

- ▶ An obvious idea is error rate
 - ▶ Total # of errors we would make if we stopped training now



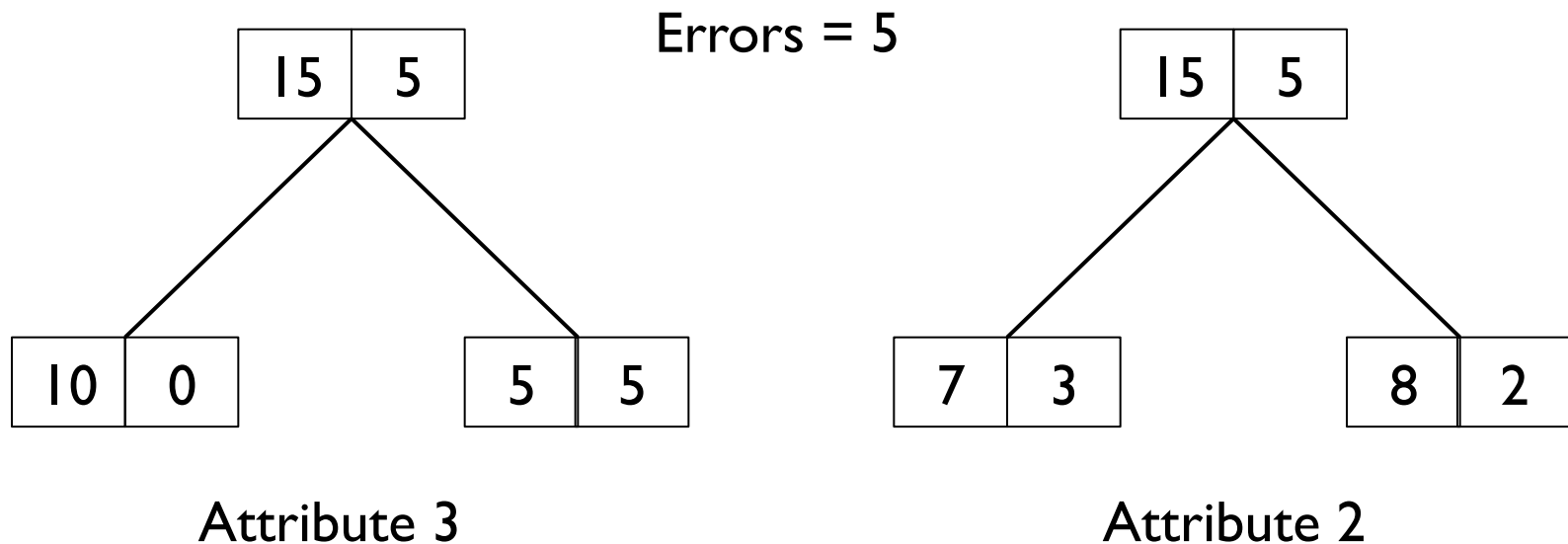
Attribute 1



Attribute 2

Splitting Criteria

- ▶ Problem: error rate fails to differentiate these two splits



- ▶ **HW problem:** show that error rate does not decrease unless children have different majority classes

Entropy

- ▶ Measure of uncertainty from field of Information Theory
 - ▶ Claude Shannon (1948)
- ▶ Consider a biased random coin
 - ▶ $\Pr(X = \text{heads}) = q$
 - ▶ $\Pr(X = \text{tails}) = 1 - q$
- ▶ Define *surprise* of a random variable as
 - ▶ $S(X = x) = -\log_2 \Pr(X = x)$

	$q = 0$	$q = 1/4$	$q = 1/2$
$S(X = \text{heads})$	infinite	2	1
$S(X = \text{tails})$	0	.4150	1



Entropy

	$q = 0$	$q = 1/4$	$q = 1/2$
$S(X = \text{heads})$	infinite	2	1
$S(X = \text{tails})$	0	.4150	1

- ▶ Entropy = average surprise

$$H(X) = - \sum_{\text{values } x} \text{Pr}(X = x) \log \text{Pr}(X = x)$$

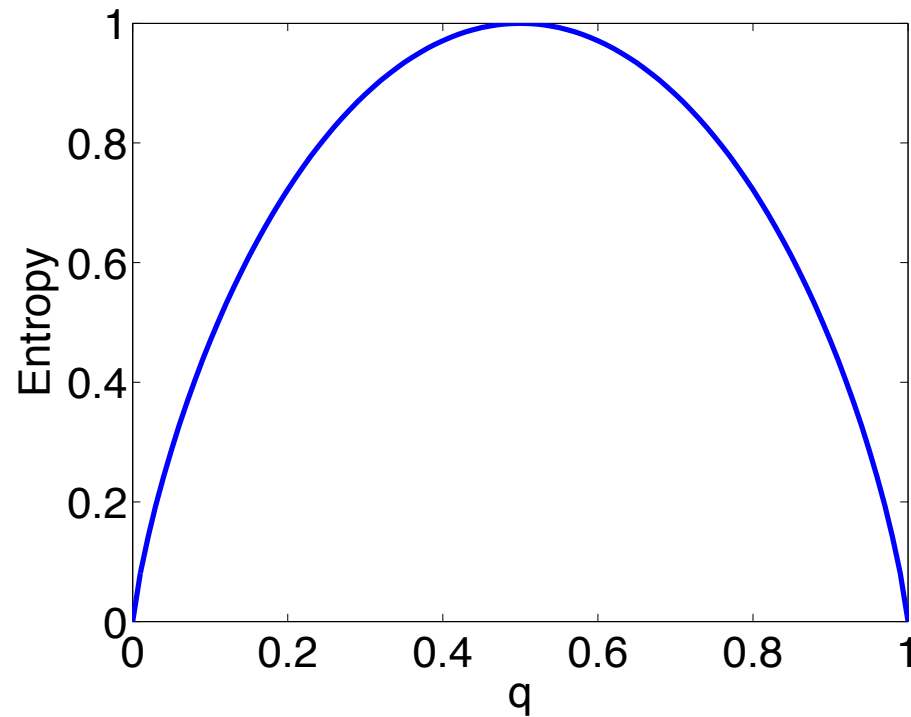
- ▶ For the biased coin

$$B(q) := H(X) = -q \log q - (1 - q) \log(1 - q)$$



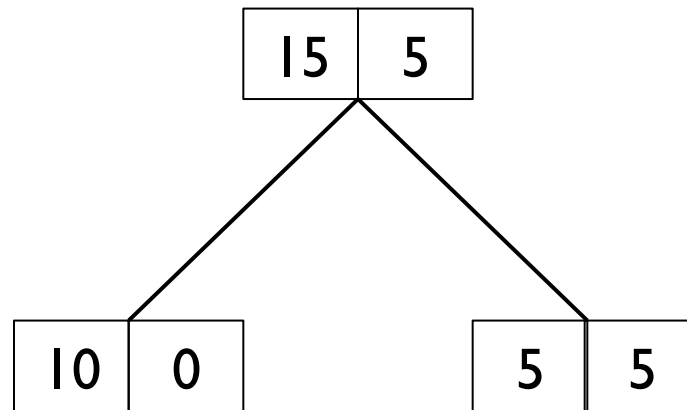
Entropy

► Entropy of a biased coin



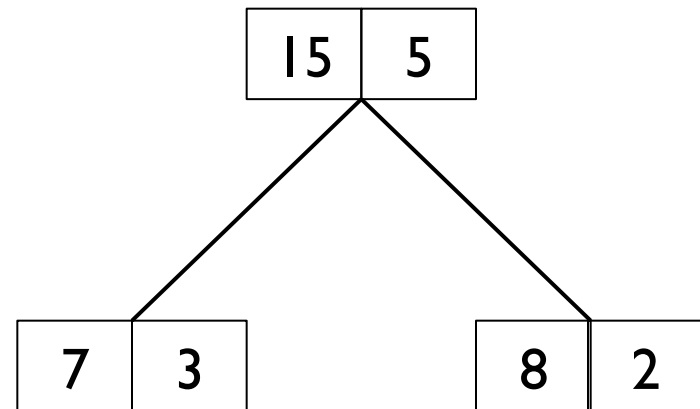
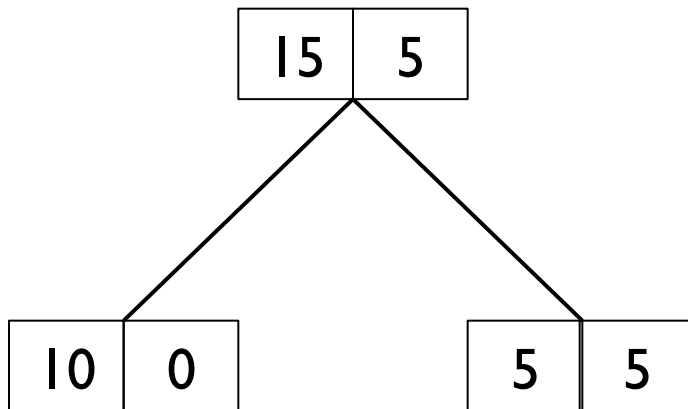
Entropy as Attribute Test

- ▶ Look for split with biggest reduction in entropy
 - ▶ *Information gain* = entropy – (average entropy of children)



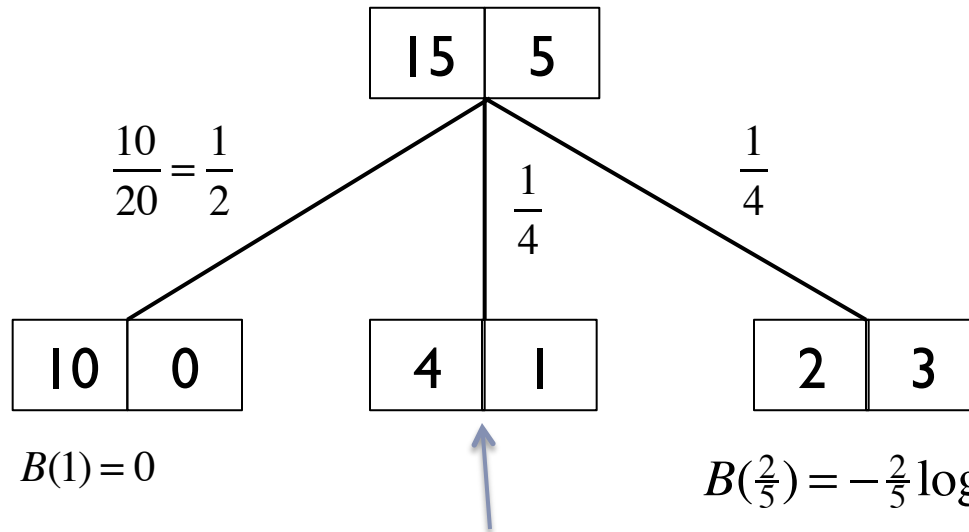
Entropy as Attribute Test

- Back to our previous example



A More General Example

$$B(\frac{3}{4}) = -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} = 0.88$$



Average entropy of children $\frac{1}{2} \cdot 0 + \frac{1}{4} \cdot 0.72 + \frac{1}{4} \cdot 0.97 = 0.42$

Reduction $0.88 - 0.42 = 0.46$

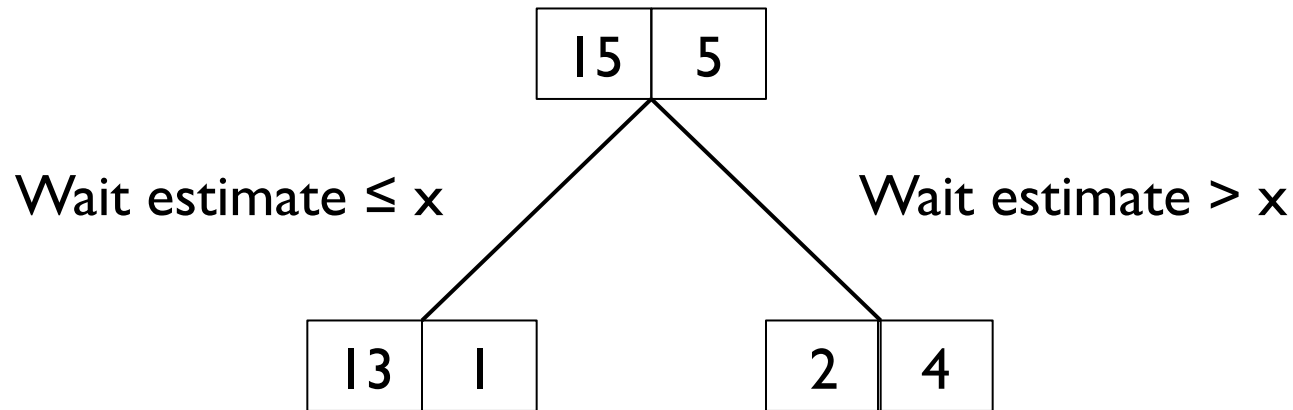


Pruning

- ▶ It is still possible to grow very large trees → overfitting
- ▶ Pruning
 - ▶ First grow a full tree
 - ▶ Then remove some nodes
 - ▶ (Why not just stop growing the tree earlier?)
- ▶ Different pruning criteria
 - ▶ R&N: Chi-squared. How “different” are the children from the parents?
 - ▶ An easier possibility: use a test set to see if removing the node hurts generalization



Continuous Attributes



- ▶ Split on any threshold value x
 - ▶ Select attribute/threshold combination with best reduction in entropy



Miscellaneous

- ▶ Issues to think/read about on your own
- ▶ Entropy criteria tends to favor multi-valued attributes over binary attributes
 - ▶ Why?
 - ▶ Various fixes proposed in literature. Can you think of one?
- ▶ Missing attributes
 - ▶ E.g., medical testing
 - ▶ How to handle these during prediction?
 - ▶ During training?



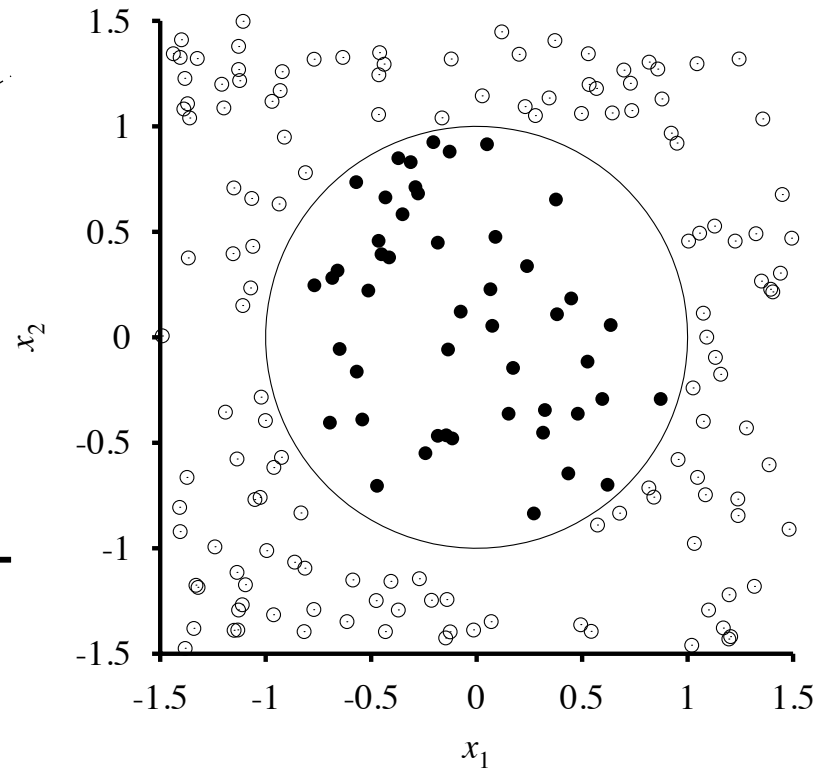
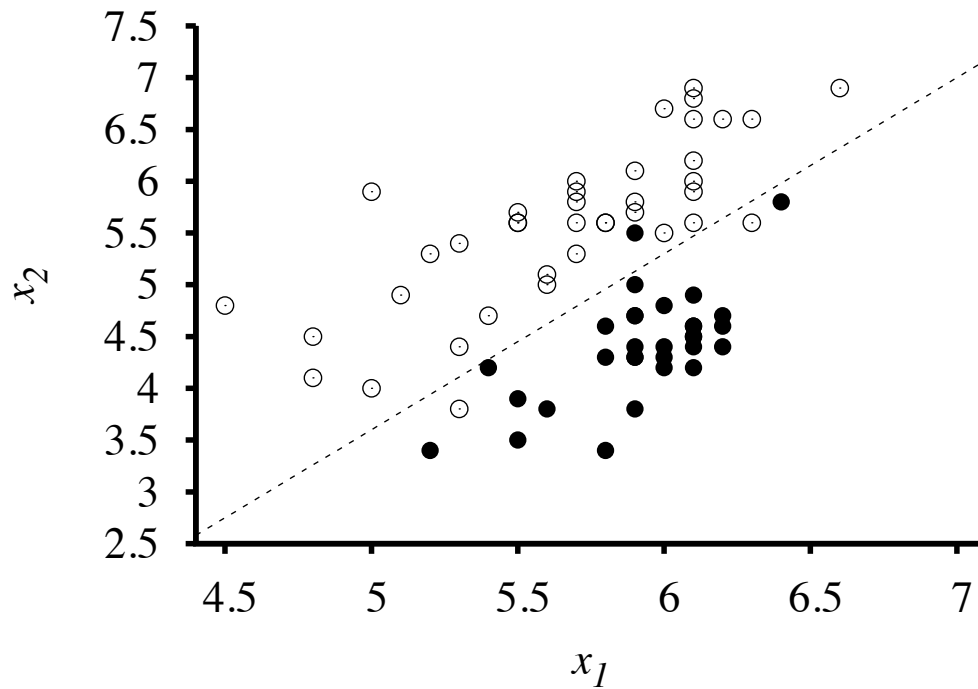
Geometric Interpretation

- ▶ **Axis-parallel splits**



Geometric Interpretation

- Can learn simple and complex boundaries



What You Should Know

- ▶ Top-down greedy learning for decision trees
- ▶ Entropy splitting criteria
- ▶ Continuous attributes

