EE 511
Decision Trees

Instructor: Hanna Hajishirzi
hannaneh@washington.edu

Slides adapted from Ali Farhadi, Mari Ostendorf, Pedro Domingos, Carlos Guestrin, and Luke Zettelmoyer
A learning problem: predict fuel efficiency

From the UCI repository (thanks to Ross Quinlan)

<table>
<thead>
<tr>
<th>mpg</th>
<th>cylinders</th>
<th>displacement</th>
<th>horsepower</th>
<th>weight</th>
<th>acceleration</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>75to78</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>75to78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>70to74</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>70to74</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>75to78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>8</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>79to83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>75to78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>75to78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>79to83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>75to78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>5</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>75to78</td>
<td>europe</td>
</tr>
</tbody>
</table>

- 40 Records
- Discrete data (for now)
- Predict MPG
- Need to find: $f : X \rightarrow Y$

From the UCI repository (thanks to Ross Quinlan)
How to Represent our Function?

\[ f(\text{cylinders, displacement, horsepower, weight, acceleration, modelyear, maker}) \rightarrow \text{mpg} \]

Conjunctions in Propositional Logic?

\[ \text{maker}=\text{asia} \land \text{weight}=\text{low} \]

Need to find “Hypothesis”: \( f : X \rightarrow Y \)
Restricted Hypothesis Space

• Many possible representations
• Natural choice: *conjunction* of attribute constraints
• For each attribute:
  – Constrain to a specific value: eg *maker*=asia
  – Don’t care: ?
• For example
  
  \[
  \text{maker} \quad \text{cyl} \quad \text{displace} \quad \text{weight} \quad \text{accel} \quad \ldots.
  \]
  
  asia \quad ? \quad ? \quad ? \quad \text{low} \quad ?

  Represents *maker*=asia $\land$ *weight*=low
Consistency

• Say an “example is consistent with a hypothesis” when the example *logically satisfies* the hypothesis

• Hypothesis:  \( \text{maker}=\text{asia} \land \text{weight}=\text{low} \)

  \[ \text{maker} \quad \text{cyl} \quad \text{displace} \quad \text{weight} \quad \text{accel} \quad \ldots \]
  
  asia \quad ? \quad ? \quad low \quad ?

• Examples:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>asia</td>
<td>5</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>...</td>
</tr>
<tr>
<td>usa</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>...</td>
</tr>
</tbody>
</table>
Hypotheses: decision trees $f : X \rightarrow Y$

- Each internal node tests an attribute $x_i$
- Each branch assigns an attribute value $x_i = v$
- Each leaf assigns a class $y$
- To classify input $x$: traverse the tree from root to leaf, output the labeled $y$
Hypothesis space

- How many possible hypotheses?
- What functions can be represented?

<table>
<thead>
<tr>
<th>mpg</th>
<th>cylinders</th>
<th>displacement</th>
<th>horsepower</th>
<th>weight</th>
<th>acceleration</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>75to78</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>75to78</td>
<td>europe</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>70to74</td>
<td>asia</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>70to74</td>
<td>asia</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75to78</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>5</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>75to78</td>
<td>europe</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>75to78</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75to78</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>8</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75to78</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>75to78</td>
<td>europe</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>5</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>75to78</td>
<td>europe</td>
<td></td>
</tr>
</tbody>
</table>
What functions can be represented?

- Decision trees can represent any boolean function!
- But, could require exponentially many nodes...

cyl=3 ∨ (cyl=4 ∧ (maker=asia ∨ maker=europa)) ∨ ...
Hypothesis space

• How many possible hypotheses?

• What functions can be represented?

• How many will be consistent with a given dataset?

• How will we choose the best one?
  • Lets first look at how to split nodes, then consider how to find the best tree
What is the Simplest Tree?

Is this a good tree?

predict mpg=bad

Means: correct on 22 examples, incorrect on 18 examples

[22+, 18-]
A Decision Stump

mpg values: bad good

root
22 18
pchance = 0.001

cylinders = 3
0 0
Predict bad

cylinders = 4
4 17
Predict good

cylinders = 5
1 0
Predict bad

cylinders = 6
8 0
Predict bad

cylinders = 8
9 1
Predict bad
Recursive Step

Take the original dataset...

And partition it according to the value of the attribute we split on.

mpg values: bad good

root
22 18
pchance = 0.001

cylinders = 3  cylinders = 4  cylinders = 5  cylinders = 6  cylinders = 8
0 0
4 17
1 0
8 0
9 1

Predict bad Predict good Predict bad Predict bad Predict bad

Records in which cylinders = 4
Records in which cylinders = 5
Records in which cylinders = 6
Records in which cylinders = 8
Records in which cylinders = 4

Records in which cylinders = 5

Records in which cylinders = 6

Records in which cylinders = 8
Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia.
A full tree

mpg values: bad  good

root
22 18
pchange = 0.001

---
cyliners = 3
  0 0
  Predict bad
  pchange = 0.001

cyliners = 4
  4 17
  Predict bad
  pchange = 0.135

---
cyliners = 5
  1 0
  Predict bad

cyliners = 6
  8 0
  Predict bad
  pchange = 0.085

cyliners = 8
  9 1
  Predict bad

---
maker = america
  0 10
  Predict good
  pchange = 0.317

maker = asia
  2 5
  pchange = 0.717

maker = europe
  2 2
  Predict bad

---
horsepower = low
  0 0
  Predict bad

horsepower = medium
  2 1
  pchange = 0.894

horsepower = high
  0 1
  Predict good
  pchange = 0.717

---
acceleration = low
  1 0
  Predict bad

acceleration = medium
  1 1
  (unexpandable)

acceleration = high
  0 0
  Predict bad
  pchange = 0.001

---
modelyear = 70to74
  0 0
  Predict bad

modelyear = 75to78
  0 1
  Predict good

modelyear = 79to83
  1 0
  Predict bad

---
modelyear = 84to89
  0 0
  Predict bad

---
modelyear = 90to96
  0 0
  Predict bad

---
modelyear = 97to02
  0 0
  Predict bad
Are all decision trees equal?

• Many trees can represent the same concept
• But, not all trees will have the same size!
  – e.g., $\phi = (A \land B) \lor (\neg A \land C) -- ((A \text{ and } B) \text{ or } (\text{not } A \text{ and } C))$

Which tree do we prefer?
• Smaller tree has more examples at each leaf!
Learning decision trees is hard!!!

• Learning the simplest (smallest) decision tree is an NP-complete problem [Hyafil & Rivest ’76]

• Resort to a greedy heuristic:
  – Start from empty decision tree
  – Split on **next best attribute (feature)**
  – Recurse
So far ...

- Decision trees
- They will overfit
- How to split?
- When to stop?
What defines a good attribute?

Ideal split

Which one do you prefer?
Splitting: choosing a good attribute

Would we prefer to split on $X_1$ or $X_2$?

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Idea: use counts at leaves to define probability distributions, so we can measure uncertainty!
Measuring uncertainty

• Good split if we are more certain about classification after split
  – Deterministic good (all true or all false)
  – Uniform distribution bad
  – What about distributions in between?

\[
\begin{array}{|c|c|c|c|}
\hline
P(Y=A) & P(Y=B) & P(Y=C) & P(Y=D) \\
1/2 & 1/4 & 1/8 & 1/8 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|}
\hline
P(Y=A) & P(Y=B) & P(Y=C) & P(Y=D) \\
1/4 & 1/4 & 1/4 & 1/4 \\
\hline
\end{array}
\]
Entropy

Entropy $H(Y)$ of a random variable $Y$

$$H(Y) = - \sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

More uncertainty, more entropy!

Information Theory interpretation:

$H(Y)$ is the expected number of bits needed to encode a randomly drawn value of $Y$ (under most efficient code)
Entropy Example

\[ H(Y) = - \sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i) \]

\[ P(Y=t) = \frac{5}{6} \]
\[ P(Y=f) = \frac{1}{6} \]

\[ H(Y) = - \frac{5}{6} \log_2 \frac{5}{6} - \frac{1}{6} \log_2 \frac{1}{6} \]
\[ = 0.65 \]
Conditional Entropy

Conditional Entropy $H(Y \mid X)$ of a random variable $Y$ conditioned on a random variable $X$

$$H(Y \mid X) = - \sum_{i=1}^{k} P(Y = y_i \mid X = x_j) \sum_{j=1}^{v} P(X = x_j) \log_2 P(Y = y_i \mid X = x_j)$$

Example:

$P(X_1=t) = \frac{4}{6}$

$P(X_1=f) = \frac{2}{6}$

$H(Y \mid X_1) = - \frac{4}{6} (1 \log_2 1 + 0 \log_2 0)$

$- \frac{2}{6} (\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2})$

$= \frac{2}{6}$
Information gain

Decrease in entropy (uncertainty) after splitting

\[ IG(X) = H(Y) - H(Y \mid X) \]

- IG(X) is non-negative (\( \geq 0 \))
- Prove by showing \( H(Y \mid X) \leq H(X) \), with Jensen’s inequality

In our running example:

\[ IG(X_1) = H(Y) - H(Y \mid X_1) \]
\[ = 0.65 - 0.33 \]

\( IG(X_1) > 0 \rightarrow \) we prefer the split!
Learning decision trees

• Start from empty decision tree
• Split on **next best attribute (feature)**
  – Use, for example, information gain to select attribute:
    \[
    \arg \max_i IG(X_i) = \arg \max_i H(Y) - H(Y | X_i)
    \]
• Recurse
Suppose we want to predict MPG

Look at all the information gains...
First split looks good! But, when do we stop?
Don’t split a node if all matching records have the same output value.
Don’t split a node if none of the attributes can create multiple non-empty children.
Base Case Two: No attributes can distinguish
Base Cases: An idea

- **Base Case One**: If all records in current data subset have the same output then don’t recurse.
- **Base Case Two**: If all records have exactly the same set of input attributes then don’t recurse.

**Proposed Base Case 3:**
If all attributes have zero information gain then don’t recurse.

• *Is this a good idea?*
The problem with Base Case 3

\[ y = a \text{ XOR } b \]

The information gains:

The resulting decision tree:
If we omit Base Case 3:

\[ y = a \text{ XOR } b \]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Is it OK to omit Base Case 3?
The test set error is much worse than the training set error...

...why?
Decision trees will overfit!!!

• Standard decision trees have no learning bias
  – Training set error is always zero!
    • (If there is no label noise)
  – Lots of variance
  – Must introduce some bias towards simpler trees

• Many strategies for picking simpler trees
  – Fixed depth
  – Fixed number of leaves
  – Or something smarter...
Decision trees will overfit!!!
mpg values: bad good

Consider this split
How to Build Small Trees

Two reasonable approaches:

• **Optimize on the held-out (development) set**
  – If growing the tree larger hurts performance, then stop growing!!!
  – Requires a larger amount of data...

• **Use statistical significance testing**
  – Test if the improvement for any split is likely due to noise
  – If so, don’t do the split!
Using Significance Test to avoid overfitting

• Build the full decision tree as before
• But when you can grow it no more, start to prune:
  – Beginning at the bottom of the tree, delete splits in which $p_{\text{chance}} > \text{MaxPchance}$
  – Continue working you way up until there are no more prunable nodes

$\text{MaxPchance}$ is a magic parameter you must specify to the decision tree, indicating your willingness to risk fitting noise
Pruning example

• With MaxPchance = 0.05, you will see the following MPG decision tree:

When compared to the unpruned tree
• improved test set accuracy
• worse training accuracy

<table>
<thead>
<tr>
<th></th>
<th>Num Errors</th>
<th>Set Size</th>
<th>Percent Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
<td>5</td>
<td>40</td>
<td>12.50</td>
</tr>
<tr>
<td>Test Set</td>
<td>56</td>
<td>352</td>
<td>15.91</td>
</tr>
</tbody>
</table>
MaxPchance

- Technical note: MaxPchance is a regularization parameter that helps us bias towards simpler models.

How to choose the value of this parameter?
Real-Valued inputs

What should we do if some of the inputs are real-valued?

Infinite number of possible split values!!!

Finite dataset, only finite number of relevant splits!

<table>
<thead>
<tr>
<th>mpg</th>
<th>cylinders</th>
<th>displacement</th>
<th>horsepower</th>
<th>weight</th>
<th>acceleration</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>4</td>
<td>97</td>
<td>75</td>
<td>2265</td>
<td>18.2</td>
<td>77</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>199</td>
<td>90</td>
<td>2648</td>
<td>15</td>
<td>70</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>121</td>
<td>110</td>
<td>2600</td>
<td>12.8</td>
<td>77</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>350</td>
<td>175</td>
<td>4100</td>
<td>13</td>
<td>73</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>198</td>
<td>95</td>
<td>3102</td>
<td>16.5</td>
<td>74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>108</td>
<td>94</td>
<td>2379</td>
<td>16.5</td>
<td>73</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>113</td>
<td>95</td>
<td>2228</td>
<td>14</td>
<td>71</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>302</td>
<td>139</td>
<td>3570</td>
<td>12.8</td>
<td>78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Finite dataset, only finite number of relevant splits!
"One branch for each numeric value" idea:

Hopeless: with such high branching factor will shatter the dataset and overfit
Threshold splits

- **Binary tree:** split on attribute $X$ at value $t$
  - One branch: $X < t$
  - Other branch: $X \geq t$

- **Requires small change**
  - Allow repeated splits on same variable
  - How does this compare to "branch on each value" approach?
The set of possible thresholds

- Binary tree, split on attribute $X$
  - One branch: $X < t$
  - Other branch: $X \geq t$
- Search through possible values of $t$
  - Seems hard!!!
- But only finite number of $t$’s are important
  - Sort data according to $X$ into $\{x_1, \ldots, x_m\}$
  - Consider split points of the form $x_i + (x_{i+1} - x_i)/2$
Picking the best threshold

• Suppose $X$ is real valued with threshold $t$

• Want $IG(Y|X:t)$: the information gain for $Y$ when testing if $X$ is greater than or less than $t$

• Define:
  - $H(Y|X:t) =$
    \[ H(Y|X < t) \ P(X < t) + H(Y|X \geq t) \ P(X \geq t) \]
  - $IG(Y|X:t) = H(Y) - H(Y|X:t)$
  - $IG^*(Y|X) = \max_t IG(Y|X:t)$

• Use: $IG^*(Y|X)$ for continuous variables
### Example with MPG

<table>
<thead>
<tr>
<th>Input</th>
<th>Value</th>
<th>Distribution</th>
<th>Info Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>cylinders</td>
<td>&lt; 5</td>
<td></td>
<td>0.48268</td>
</tr>
<tr>
<td></td>
<td>&gt;= 5</td>
<td></td>
<td>0.428205</td>
</tr>
<tr>
<td>displacement</td>
<td>&lt; 198</td>
<td></td>
<td>0.48268</td>
</tr>
<tr>
<td></td>
<td>&gt;= 198</td>
<td></td>
<td>0.379471</td>
</tr>
<tr>
<td>horsepower</td>
<td>&lt; 94</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;= 94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>weight</td>
<td>&lt; 2789</td>
<td></td>
<td>0.159982</td>
</tr>
<tr>
<td></td>
<td>&gt;= 2789</td>
<td></td>
<td></td>
</tr>
<tr>
<td>acceleration</td>
<td>&lt; 18.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;= 18.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>modelyear</td>
<td>&lt; 81</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;= 81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>maker</td>
<td>america</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>europe</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example tree for our continuous dataset

mpg values: bad good

root

22 18
pchance = 0.000

cylinders < 5
cylinders >= 5

4 17
18 1
pchance = 0.001
pchance = 0.003

horsepower < 94
carpower >= 94

1 17
3 0
pchance = 0.274
Predict bad
Predict bad

acceleration < 19
acceleration >= 19

18 0
0 1
pchance = 0.237
Predict good
Predict good

maker = america
maker = asia
maker = europe

0 10
0 5
1 2
pchance = 0.270
Predict good
Predict good

Displacement < 116
Displacement >= 116

0 2
1 0
Predict good
Predict bad
What you need to know about decision trees

- Decision trees are one of the most popular ML tools
  - Easy to understand, implement, and use
  - Computationally cheap (to solve heuristically)
- Information gain to select attributes (ID3, C4.5, ...)
- Presented for classification, can be used for regression and density estimation too
- Decision trees will overfit!!!
  - Must use tricks to find “simple trees”, e.g.,
    - Fixed depth/Early stopping
    - Pruning
Acknowledgements

• Some of the material in the decision trees presentation is courtesy of Andrew Moore, from his excellent collection of ML tutorials:
  – http://www.cs.cmu.edu/~awm/tutorials