CS 4824/ECE 4424: Function Approximation

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.
Supervised function approximation

- Problem setting
  - Set of possible instances $X$
  - Unknown target function $f$
  - Set of function hypotheses: $H = \{ h | h: X \to Y \}$

- Input
  - Training examples \{<X^{(i)}, Y^{(i)}>\} of unknown function $f$

- Output
  - Hypothesis $h \in H$ that best approximates $f$
### Example data

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Example function approximator

- Each internal node
  - Tests one attribute $X_i$

- Each branch from a node
  - Selects on value for $X_i$

- Each leaf node:
  - Predicts $Y$ or $P(Y|X \in \text{leaf})$

A decision tree for $F: \langle \text{Outlook, Humidity, Wind, Temp} \rightarrow \text{PlayTennis?} \rangle$
Dynamics of the function approximator

- Set of possible instances $X$
  - Each instance $x$ is a feature vector

- Unknown target function $f$
  - $Y$ is discrete valued

- Set of function hypotheses: $H = \{h \mid h: X \rightarrow Y\}$
  - Each hypothesis $h$ is a decision tree
  - Tree sorts $x$ to leaf, which assigns $y$
Dynamics of the function approximator

- Set of possible instances $X$
  - Each instance $x$ is a feature vector

- Unknown target function $f$
  - $Y$ is discrete valued

- Set of function hypotheses: $H = \{h \mid h: X \rightarrow Y\}$
  - Each hypothesis $h$ is a decision tree
  - Tree sorts $x$ to leaf, which assigns $y$

Q. How many decision trees are possible?
Function approximation using decision trees

- Suppose $X = <X_1, ..., X_n>$, $X_i \in \{0, 1\}$

- How would you represent $Y = X_2 X_5$? $Y = X_2 \lor X_5$?

- Or a more complicated one $X_2 X_5 \lor X_3 X_4 (\neg X_1)$?
Function approximation using decision trees

Q. Can we represent arbitrary boolean (or discrete-valued) functions using decision trees?
Decision tree as function approximator

- Decision trees are expressive
  - Can represent any Boolean (or discrete-valued) functions
  - This makes decision trees universal function approximator
Top-down induction of decision trees

\[ \text{node} = \text{Root} \]

Main loop:
1. \( A \leftarrow \text{the “best” decision attribute for next node} \)
2. Assign \( A \) as decision attribute for node
3. For each value of \( A \), create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

**Intuition:** top-down greedy growth of decision tree using “best” attribute until all examples are perfectly classified.

**Q. How to pick “best” attribute?**
Sample Entropy

- $S$ is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in $S$
- $p_\ominus$ is the proportion of negative examples in $S$
- Entropy measures the impurity of $S$

\[ H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus \]
Entropy

- Entropy $H(X)$ of a random variable $X$ is defined as:
  - $H(X) = -\sum_i P(X=i) \log_2 P(X=i)$

- Specific conditional entropy $H(X | Y=v)$ is
  - $H(X | Y=v) = -\sum_i P(X=i | Y=v) \log_2 P(X=i | Y=v)$

- Conditional entropy $H(X | Y)$ is
  - $H(X | Y) = \sum_{v \in \text{values}(Y)} P(Y=v) H(X | Y=v)$

- Mutual information (a.k.a. information gain) of $X$ and $Y$
  - $I(X, Y) = H(X) - H(X | Y) = H(Y) - H(Y | X)$
Information gain

- Mutual information (a.k.a. information gain) of $X$ and $Y$
  - $I(X, Y) = H(X) - H(X | Y) = H(Y) - H(Y | X)$

- Information Gain is the expected reduction in entropy of target variable $Y$ for data sample $S$, due to sorting on variable $A$
  - $Gain (S, A) = I_s(A, Y) = H_s(Y) - H_s(Y | A)$

Q. How to pick “best” attribute?
A. One that reduces entropy the most. i.e. highest information gain
## Example data

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play/Tennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Selecting the “best” attribute

\[
S: [9+, 5-] \\
E = 0.940
\]

**Humidity**

- **High**
  - [3+, 4-]
  - \( E = 0.985 \)

- **Normal**
  - [6+, 1-]
  - \( E = 0.592 \)

**Gain** \((S, \text{Humidity})\)
\[
= 0.940 - \left( \frac{7}{14} \right) 0.985 - \left( \frac{7}{14} \right) 0.592 \\
= 0.151
\]

\[
S: [9+, 5-] \\
E = 0.940
\]

**Wind**

- **Weak**
  - [6+, 2-]
  - \( E = 0.811 \)

- **Strong**
  - [3+, 3-]
  - \( E = 1.00 \)

**Gain** \((S, \text{Wind})\)
\[
= 0.940 - \left( \frac{8}{14} \right) 0.811 - \left( \frac{6}{14} \right) 1.0 \\
= 0.048
\]
Questions to think about…

Is there more than one decision tree that will perfectly sort the data?

If so, which one do you choose and why?