Today:
• What is machine learning?
• Decision tree learning
• Course logistics

Readings:
• “The Discipline of ML”
• Mitchell, Chapter 3
• Bishop, Chapter 14.4

Machine Learning:

Study of algorithms that
• improve their performance $P$
• at some task $T$
• with experience $E$

well-defined learning task: $<P,T,E>$
Learning to Predict Emergency C-Sections

Data:

<table>
<thead>
<tr>
<th>Patient001 (true)</th>
<th>Patient002 (true)</th>
<th>Patient003 (true)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age: 29</td>
<td>Fetal-pocket: no</td>
<td>Diagnosis: no</td>
</tr>
<tr>
<td>First-trimester: yes</td>
<td>Delivery: yes</td>
<td>Pre-eclampsia: yes</td>
</tr>
<tr>
<td>Emergency C-Section: ?</td>
<td>Ultrasound: normal</td>
<td>Emergency C-Section: ?</td>
</tr>
</tbody>
</table>

9714 patient records, each with 215 features

One of 18 learned rules:

If   No previous vaginal delivery, and
     Abnormal 2nd Trimester Ultrasound, and
     Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

Over training data: 26/41 = .63,
Over test data: 12/20 = .60

Learning to detect objects in images

(Prof. H. Schneiderman)

Example training images for each orientation
Learning to classify text documents

- Company home page vs Personal home page vs University home page vs ...

Reading a noun (vs verb) [Rustandi et al., 2005]
Machine Learning - Practice

- Speech recognition
- Object recognition
- Text analysis
- Mining Databases

Text analysis

- PAC Learning Theory (supervised concept learning)
  - # examples ($m$)
  - representational complexity ($H$)
  - error rate ($\epsilon$)
  - failure probability ($\delta$)
  - $m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$

Other theories for

- Reinforcement skill learning
- Semi-supervised learning
- Active student querying

... also relating:

- # of mistakes during learning
- learner’s query strategy
- convergence rate
- asymptotic performance
- bias, variance
Machine Learning in Computer Science

• Machine learning already the preferred approach to
  – Speech recognition, Natural language processing
  – Computer vision
  – Medical outcomes analysis
  – Robot control
  – …

• This ML niche is growing (why?)
Machine Learning in Computer Science

• Machine learning already the preferred approach to
  – Speech recognition, Natural language processing
  – Computer vision
  – Medical outcomes analysis
  – Robot control
  – …

• This ML niche is growing
  – Improved machine learning algorithms
  – Increased data capture, networking, new sensors
  – Software too complex to write by hand
  – Demand for self-customization to user, environment

Function Approximation and Decision tree learning
Function approximation

Problem Setting:
• Set of possible instances $X$
• Unknown target function $f : X \rightarrow Y$
• Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$

Input:
• Training examples $\{<x^{(i)}, y^{(i)}>\}$ of unknown target function $f$

Output:
• Hypothesis $h \in H$ that best approximates target function $f$

A Decision tree for
$F: <\text{Outlook, Humidity, Wind, Temp}> \rightarrow \text{PlayTennis}$

Each internal node: test one attribute $X_i$
Each branch from a node: selects one value for $X_i$
Each leaf node: predict $Y$ (or $P(Y|X \in \text{leaf})$)
Decision Tree Learning

Problem Setting:
• Set of possible instances $X$
  – each instance $x$ in $X$ is a feature vector
  – e.g., $<$Humidity=low, Wind=weak, Outlook=rain, Temp=hot$>$
• Unknown target function $f : X \rightarrow Y$
  – $Y$ is discrete valued
• Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$
  – each hypothesis $h$ is a decision tree
  – trees sorts $x$ to leaf, which assigns $y$

Input:
• Training examples $\{ <x^{(i)}, y^{(i)} > \}$ of unknown target function $f$

Output:
• Hypothesis $h \in H$ that best approximates target function $f$
**Decision Trees**

Suppose \( X = \langle X_1, \ldots, X_n \rangle \)
where \( X_i \) are boolean variables

How would you represent \( Y = X_2 X_5 \)? \( Y = X_2 \lor X_5 \)

How would you represent \( X_2 X_5 \lor X_3 X_4 (\neg X_7) \)

---

**A Tree to Predict C-Section Risk**

Learned from medical records of 1000 women

Negative examples are C-sections

\[
[833+, 167-] .83+ .17-
| Fetal_Presentation = 1: [822+, 116-] .88+ .12-
  | Previous_Csection = 0: [767+, 81-] .90+ .10-
  | Primiparous = 0: [399+, 13-] .97+ .03-
  | Primiparous = 1: [368+, 68-] .84+ .16-
  | Fetal_Distress = 0: [334+, 47-] .88+ .12-
  | Fetal_Distress = 1: [34+, 21-] .62+ .38-
  | Previous_Csection = 1: [55+, 35-] .61+ .39-
Fetal_Presentation = 2: [3+, 29-] .11+ .89-
Fetal_Presentation = 3: [8+, 22-] .27+ .73-\]
Top-Down Induction of Decision Trees

node = Root

Main loop:
1. A ← 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

Which attribute is best?

Entropy

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

$H(X) = - \sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$

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

Why? Information theory:
- Most efficient code assigns $-\log_2 P(X=i)$ bits to encode the message $X=i$
- So, expected number of bits to code one random $X$ is:

$$\sum_{i=1}^{n} P(X = i)(- \log_2 P(X = i))$$
\textbf{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 $S$
  \[ H(S) = -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus \]

\textbf{Entropy}

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

\[ H(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i) \]

Specific conditional entropy $H(X|Y=v)$ of $X$ given $Y=v$:

\[ H(X|Y = v) = -\sum_{i=1}^{n} P(X = i|Y = v) \log_2 P(X = i|Y = v) \]

Conditional entropy $H(X|Y)$ of $X$ given $Y$:

\[ H(X|Y) = \sum_{v \in \text{values}(Y)} P(Y = v) H(X|Y = v) \]

Mutual information (aka Information Gain) of $X$ and $Y$:

\[ I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \]
Information Gain is the mutual information between input attribute $A$ and target variable $Y$.

Information Gain is the expected reduction in entropy of target variable $Y$ for data sample $S$, due to sorting on variable $A$.

$$Gain(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y|A)$$

Training Examples

<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>
Selecting the Next Attribute

Which attribute is the best classifier?

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

Humidity
- High \[ [3+4-] \] \[ E = 0.985 \]
- Normal \[ [6+1-] \] \[ E = 0.392 \]

Wind
- Weak \[ [6+2-] \] \[ E = 0.811 \]
- Strong \[ [3+3-] \] \[ E = 1.000 \]

Gain (\( S, \) Humidity)
\[ = 0.940 - (7/14) \times 0.985 - (7/14) \times 0.392 \]
\[ = 0.151 \]

Gain (\( S, \) Wind)
\[ = 0.940 - (8/14) \times 0.811 - (6/14) \times 1.0 \]
\[ = 0.048 \]

\[ [D1, D2, ..., D14] \]
\[ [9+5-] \]

Outlook
- Sunny \[ [D1, D2, D8, D9, D11] \] \[ [2+3-] \]
- Overcast \[ [D3, D7, D12, D13] \] \[ [4+6-] \]
- Rain \[ [D4, D5, D6, D10, D14] \] \[ [3+2-] \]

Which attribute should be tested here?

\( S_{\text{Sunny}} = \{D1, D2, D8, D9, D11\} \)

Gain (\( S_{\text{Sunny}}, \) Humidity)
\[ = 0.970 - (3/5) \times 0.0 - (2/5) \times 0.0 = 0.970 \]

Gain (\( S_{\text{Sunny}}, \) Temperature)
\[ = 0.970 - (2/5) \times 0.0 - (2/5) \times 1.0 - (1/5) \times 0.0 = 0.570 \]

Gain (\( S_{\text{Sunny}}, \) Wind)
\[ = 0.970 - (2/5) \times 1.0 - (3/5) \times 0.918 = 0.19 \]
Decision Tree Learning Applet


Which Tree Should We Output?

- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?

*Occam’s razor: prefer the simplest hypothesis that fits the data*
Why Prefer Short Hypotheses? (Occam’s Razor)

Arguments in favor:

Arguments opposed:

Why Prefer Short Hypotheses? (Occam’s Razor)

Argument in favor:
• Fewer short hypotheses than long ones
  ➔ a short hypothesis that fits the data is less likely to be a statistical coincidence
  ➔ highly probable that a sufficiently complex hypothesis will fit the data

Argument opposed:
• Also fewer hypotheses with prime number of nodes and attributes beginning with “Z”
• What’s so special about “short” hypotheses?
Overfitting in Decision Trees

Consider adding noisy training example #15:

Sunny, Hot, Normal, Strong, PlayTennis = No

What effect on earlier tree?

Overfitting

Consider error of hypothesis $h$ over

- training data: $error_{\text{train}}(h)$
- entire distribution $\mathcal{D}$ of data: $error_{\mathcal{D}}(h)$

Hypothesis $h \in H$ overfits training data if there is an alternative hypothesis $h' \in H$ such that

\[ error_{\text{train}}(h) < error_{\text{train}}(h') \]

and

\[ error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h') \]
Overfitting in Decision Tree Learning

![Graph showing accuracy vs size of tree](image)

Avoiding Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune
Reduced-Error Pruning

Split data into training and validation set

Create tree that classifies training set correctly

Do until further pruning is harmful:

1. Evaluate impact on validation set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves validation set accuracy

- produces smallest version of most accurate subtree
- What if data is limited?

Effect of Reduced-Error Pruning

![Graph showing the effect of Reduced-Error Pruning on accuracy over the size of the tree. The graph compares accuracy on training data, test data, and test data during pruning.]
Continuous Valued Attributes

Create a discrete attribute to test continuous

- \( Temperature = 82.5 \)
- \( (Temperature > 72.3) = t, f \)

<table>
<thead>
<tr>
<th>Temperature:</th>
<th>40</th>
<th>48</th>
<th>60</th>
<th>72</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlayTennis:</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Attributes with Many Values

Problem:

- If attribute has many values, \( Gain \) will select it
- Imagine using \( Date = Jun. 3.1996 \) as attribute

One approach: use \( GainRatio \) instead

\[
GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}
\]

\[
SplitInformation(S, A) \equiv - \sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}
\]

where \( S_i \) is subset of \( S \) for which \( A \) has value \( v_i \)
What you should know:

• Well posed function approximation problems:
  – Instance space, X
  – Sample of labeled training data \{ <x^{(i)}, y^{(i)}> \}
  – Hypothesis space, \( H = \{ f: X \rightarrow Y \} \)

• Learning is a search/optimization problem over \( H \)
  – Various objective functions
    • minimize training error (0-1 loss)
    • among hypotheses that minimize training error, select smallest (?)

• Decision tree learning
  – Greedy top-down learning of decision trees (ID3, C4.5, …)
  – Overfitting and tree/rule post-pruning
  – Extensions…