Lecture 5 DT cont.

Oct 8 2008
Review of last lecture

• What is decision tree
• What decision boundaries do decision trees produce
  – Syntactically different trees can represent the same decision boundaries
  – In such cases, we prefer smaller trees
  – flexible *hypothesis space*
• How to learn a decision tree?
  – A greedy approach
  – At each step, choose the test that reduce the most uncertainty about class labels
Choosing the test based on training error

- Perform 1-step look-ahead search and choose the attribute that gives the lowest error rate on the training data

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Training examples

\[
\begin{array}{c|c|c|c}
4 & 4 & x_1 \\
0 & 1 \\
1 & 3 \\
3 & 1 \\
J=2
\end{array}
\]

\[
\begin{array}{c|c|c|c}
4 & 4 & x_2 \\
0 & 1 \\
2 & 2 \\
J=4
\end{array}
\]

\[
\begin{array}{c|c|c|c}
4 & 4 & x_3 \\
0 & 1 \\
2 & 2 \\
J=4
\end{array}
\]
Unfortunately, this measure does not always work well, because it does not detect cases where we are making “progress” toward a good tree.
A Better Heuristic from Information Theory

• Let $X$ have the following probability distribution:

<table>
<thead>
<tr>
<th>$P(X = 0)$</th>
<th>$P(X = 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

• The **entropy** of $X$, denoted $H(X)$, is defined as

$$H(X) = -P_0 \log_2 P_0 - P_1 \log_2 P_1$$

$$H(X) = -P_0 \log_2 P_0 - ... - P_k \log_2 P_k$$ if there are $k$ possible values

• $-\log P(X=x)$ measures the **surprise** of value $x$:
  If $P(X=x)$ is small, $x$ is a surprising value to take, $-\log P(x)$ is large

• **Entropy** can be considered as the average surprise of a random variable, which is also referred to as the uncertainty of a random variable
Entropy

- Entropy is a concave function downward

Minimum uncertainty occurs when $p_0=0$ or 1
Mutual Information

- If we use entropy to measure uncertainty, we end up measuring the mutual information between a candidate test variable $X$ and class label $Y$:

$$I(X, Y) = H(Y) - H(Y | X)$$

  - Uncertainty of $Y$
  - Remaining uncertainty of $Y$ after knowing the value of $X$

- $H(Y|X)$ is called the conditional entropy of $Y$ given $X$
  - Measures the uncertainty of $Y$ after knowing the value of $X$

$$H(Y | X) = \sum_x P(X = x)H(Y | X = x)$$

$$= -\sum_x P(X = x)\sum_y P(Y = y | X = x)\log P(Y = y | X = x)$$

  - The probability of $X=x$
  - The uncertainty of $Y$ when $X=x$
$P(X_1=0) = 0.6677 \quad P(X_1=1) = 0.3333$

$H(Y|X_1=0) = -0.6 \log 0.6 - 0.4 \log 0.4 = 0.9710$

$H(Y|X_1=1) = -0.8 \log 0.8 - 0.2 \log 0.2 = 0.7219$

$I(X_1, Y) = 0.0304$

$H(Y) = 0.9183$

$H(Y|X_1) = 0.6667 \times 0.9710 + 0.3333 \times 0.7219 = 0.8873$
Information Gain

- This is called the **information gain** criterion: choose $X$ that maximizes mutual information between $X$ and $y$

$$\arg \max_{j} I(X_j; Y) = \arg \max_{j} H(Y) - H(Y | X_j)$$

$$= \arg \min_{j} H(Y | X_j)$$

- Information gain is just one of the methods for selecting tests in decision tree learning

- There are other methods as well, but they use the same general approach based on different uncertainty measures
Choosing the Best Feature: A General View

Benifit of split = \( U(S) - [P_L \times U(S_L) + P_R \times U(S_R)] \)

Expected Remaining Uncertainty (Impurity)

<table>
<thead>
<tr>
<th>Measures of Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
</tr>
<tr>
<td>min{p, 1 - p}</td>
</tr>
<tr>
<td>Entropy</td>
</tr>
<tr>
<td>(-p \log p - (1 - p) \log 1 - p)</td>
</tr>
<tr>
<td>Gini Index</td>
</tr>
<tr>
<td>(2p(1 - p))</td>
</tr>
</tbody>
</table>
Issues with Multi-nominal Features

• Multi-nominal features: more than 2 possible values
• Comparing two features, one is binary, the other has 100 possible values, which one you expect to have higher mutual information with $Y$?
  – The conditional entropy of $Y$ given this feature will be low
  – But is this meaningful?
  – This bias will inherently prefer such multinomial features to binary features
  – Method 1: To avoid this, we can rescale the conditional entropy:
    \[
    \arg \min_j \frac{H(Y | X_j)}{H(X_j)} = \arg \min_j \frac{\sum_x P(X_j = x)H(Y | X_j = x)}{-\sum_x P(X_j = x) \log P(X_j = x)}
    \]
  – Method 2: Test for one value versus all of the others
  – Method 3: Group the values into two disjoint sets and test one set against the other
Continuous Features

• Test against a threshold

• How to compute the best threshold $\theta_j$ for $X_j$?
  – Sort the examples according to $X_j$.
  – Move the threshold $\theta$ from the smallest to the largest value
  – Select $\theta$ that gives the best information gain
  – Trick: only need to compute information gain when class label changes
Over-fitting

- Decision tree has a very flexible hypothesis space
- As the nodes increase, we can represent arbitrarily complex decision boundaries
- This can lead to over-fitting

Possibly just noise, but the tree is grown larger to capture these examples
Over-fitting
Avoid Overfitting

• Early stop
  – Stop growing the tree when data split does not offer large benefit

• Post pruning
  – Separate training data into training set and validating set
  – Evaluate impact on validation set when pruning each possible node
  – Greedily prune the node that most improves the validation set performance
Effect of Pruning

![Graph showing the effect of pruning on training and test data.](image)
Revisit some of the issues

• Is decision tree robust to outliers?
• Is decision tree sensitive to irrelevant features?
• Is decision tree computational efficient?