Decision Tree Algorithm

Andrew Kusiak
Intelligent Systems Laboratory
2139 Seamans Center
The University of Iowa
Iowa City, IA 52242 - 1527
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak
Tel. 319-335 5934

Concept Learning System (CLS)

• CLS proposed by Hunt et al. (1966)
• Precursor of ID3 (Interactive Dichotomizer 3) algorithm developed by Quinlan (1986)
• Precursor of C4.5 (successor of ID3) algorithm developed by Quinlan

CLS Algorithm

Divide and conquer method!

Notation:
• $S$ training set
• $n$ number of examples in $S$
• $c_1, c_2, \ldots, c_C$ classes of examples

Goal:
Divide the training data set into disjoint sets $S_1, S_2, \ldots, S_N$ so that they create a partition based on a single feature

CLS Based Algorithms

• ID3 Algorithm uses Shannon's entropy as a criterion for determining the most discriminating feature
• C4.5 Algorithm works with continuous features by using a discretization scheme. It also handles cases with missing data. The windowing scheme reduces memory requirements by dividing the training data set into subsets (called windows).

ID3 Algorithm (Quinlan, 1986)

Quinlan's data set

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit rating</th>
<th>buys computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>30-40</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31-40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&gt;30</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;30</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31-40</td>
<td>medium</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

Output: A Decision Tree for “buys_computer”

Information Gain in Decision Tree Induction
- Assume that using attribute A a set S will be partitioned into sets \{S_1, S_2, \ldots, S_v\}, i.e., v attribute A values
  - If \( S_i \) contains \( p_i \) examples of \( P \) and \( n_i \) examples of \( N \), the entropy, or the expected information needed to classify objects in all subtrees \( S_i \) is
    \[
    E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)
    \]
- The encoding information that would be gained by branching on \( A \)
  \[
  Gain(A) = I(p, n) - E(A)
  \]

Example

Attribute Selection by Information Gain Computation
- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”
- Information \( I(p, n) = I(9, 5) = 0.940 \)
- Compute the entropy for:
  - age
    | age | p  | n  | \( I(p, n) \) |
    |-----|----|----|---------------|
    | <=30| 2  | 1  | 0.971         |
    | 30…40| 4  | 0  |               |
    | >40 | 3  | 2  | 0.971         |

Decision Tree Revisited
**Classification accuracy** (CA) of a rule set is the ratio of the number of correctly classified objects from the test set and all objects in the test set.

\[
\text{Accuracy} = \frac{A + D}{A + B + C + D}
\]

- **Sensitivity** (true positive rate) = \(\frac{A}{A+C}\)
- **Specificity** (true negative rate) = \(\frac{D}{B+D}\)
- **Positive predicted value** = \(\frac{A}{A+B}\)
- **Negative predicted value** = \(\frac{D}{C+D}\)

**From Decision Trees to Rules**

**Avoiding Overfitting the Data**
References
