Decision Trees
Decision Trees - play golf?

Outlook?

- sunny
- cloudy
- rain

Temp.

- <95
- >95

Yes

No

Leaves partition the instance space
Decision trees

- Popular C4.5 excellent “off the shelf” alg.
- Efficient hypothesis space
  - Variable sized: bigger trees for more complicated hypotheses
  - Can mix discrete and numeric attributes
- Small trees are understandable
- Decision tree successes: (many)
  - flying (Sammuti et.al. 92), better than simulator pilots
  - BP GasOil (Mitchie ‘86) for separation at off-shore platforms (10 man-years by hand, done in 100 man days), saved many millions
Finding a good tree

- Want accuracy on training data but don’t want to memorize it
- Small tree good for generalization
- Finding smallest tree consistent with data is NP hard (how many trees?)
- Use greedy search to build trees top down -- can miss XOR
Decision tree algorithm

• If data pure (all one class) then return a leaf predicting the class
• Otherwise:
  – Pick a test to split on (discrete attribute or attribute+threshold)
  – Insert node with test into tree
  – split training data based on test, and recursively construct each branch from proper portion of data
How to chose test?

• Use a *splitting criteria* like number of mistakes or *information gain*

• Often viewed as “*impurity*” measure, goal: minimize the impurity

• Try all possible splits and use the one that optimizes the criteria

• Often use criteria related to information theory
Impurity measures

• If have $n$ total examples with $n_+$ positive examples, let $p = n_+ / n$

• Examples of 2-class impurity functions (as function of $p$)
  – Gini index: $2p(1-p)$
  – Entropy: $-p \log p - (1-p) \log (1-p)$
  – Error rate: $1 - \max[p, 1-p]$
  – Generalized entropy for multiple classes
Impurity of split

• If have set S of $n$ examples split into $S_1$ and $S_2$. Let $n_1$ be number examples in $S_1$ and $p_1$ be fraction of $S_1$ labeled positive ($n_2$, $p_2$ similar for $S_2$)

• Badness of split is average impurity:
  \[
  \frac{n_1 \text{ impurity}(p_1) + n_2 \text{ impurity}(p_2)}{n}
  \]

• Pick split with least badness

• Generalizes to multi-way splits (need to penalize them too)
Decision Trees can overfit

• Overfitting - modeling the particulars of the data set rather than the underlying pattern

• **Def:** Hypothesis \( h \in H \) overfits the data if there is an \( h' \in H \) such that \( h \) better on training data but \( h' \) generalizes better.

• Complexity of decision trees lets them fit the noise
ID3 overfits (Mitchell)
Avoiding overfitting:

- Prune (replace subtrees with leaves) to reduce variance
- PrePruning – stop early (faster)
- PostPruning (more popular/accurate)
  - Based on validation set
  - Rule post-pruning:
    - convert to rules, generalize if accuracy improves, and apply rules in order of accuracy
Rule Extraction from Trees

C4.5 Rules
(Quinlan, 1993)

\begin{itemize}
\item \textbf{R1:} IF (age $> 38.5$) AND (years-in-job $> 2.5$) THEN $y = 0.8$
\item \textbf{R2:} IF (age $> 38.5$) AND (years-in-job $\leq 2.5$) THEN $y = 0.6$
\item \textbf{R3:} IF (age $\leq 38.5$) AND (job-type='A') THEN $y = 0.4$
\item \textbf{R4:} IF (age $\leq 38.5$) AND (job-type='B') THEN $y = 0.3$
\item \textbf{R5:} IF (age $\leq 38.5$) AND (job-type='C') THEN $y = 0.2$
\end{itemize}
Decision tree comments:

• Several packages e.g. ID3, C4.5
  Weka has J48 and BFTree
• Often heavily engineered to handle
  missing data, overfitting, numeric vs.
  nominal attributes, etc.

• Multivariate trees
• Regression trees
Model Selection in Trees:
Random Forests

• Pick small random subset of features to try at each node rather than exhaustive search
• Build many trees and predict with most frequent prediction
• Subset saves time, robust against missing data
• Ensemble reduces variance - don’t need pruning
• Ho ‘95, Brieman ‘01
Exercise

1. Make up an XOR.arff data file with 12 examples. Each example $\mathbf{x}$ labeled with $x_1 \text{ XOR } x_2$, and add 10 irrelevant and random binary features.

2. Try Weka’s ID3 on the XOR data.

3. Try Weka’s J48 on the soybean data - is the tree understandable?
## Tree and kNN comparison

<table>
<thead>
<tr>
<th></th>
<th>Decision Trees</th>
<th>Nearest Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>Trees - flexible</td>
<td>Instance based, flexible</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>mixed</td>
<td>Usually Numeric</td>
</tr>
<tr>
<td>interpretable</td>
<td>If small tree</td>
<td>Only in 1 or 2 dimensions</td>
</tr>
<tr>
<td>Missing values</td>
<td>Tricks</td>
<td>Training set no, but ok for test points</td>
</tr>
<tr>
<td>Noise/outliers</td>
<td>Good with pruning</td>
<td>Good with kNN</td>
</tr>
</tbody>
</table>
# Tree and kNN Robustness

<table>
<thead>
<tr>
<th></th>
<th>Decision tree</th>
<th>Nearest neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotone transformation</td>
<td>Great</td>
<td>Very bad</td>
</tr>
<tr>
<td>Irrelevant features</td>
<td>Fair</td>
<td>Very bad</td>
</tr>
<tr>
<td>Computation time</td>
<td>OK</td>
<td>Lazy Learning Prediction hard</td>
</tr>
</tbody>
</table>
Decision Tree Summary:

• Model: trees (flexible size/complexity)
• Data: numeric and nominal
• Interpretable: if small tree,
• Noise/outliers: OK with pruning
• Irrelevant features: fair
• Missing attributes: some tricks
• Computation time: OK training, fast prediction