Classification:
Decision Trees & Overfitting

These slides were assembled by Eric Eaton, with grateful acknowledgement of the many others who made their course materials freely available online. Feel free to reuse or adapt these slides for your own academic purposes, provided that you include proper attribution. Please send comments and corrections to Eric.
Summary of Decision Trees (so far)

- Decision tree induction $\rightarrow$ choose the best attribute
  - Choose split via information gain
  - Build tree greedily, recursing on children of split
  - Stop when we achieve homogeneity
    - i.e., when all instance in a child have the same class
Noisy Data

• Many kinds of “noise” can occur in the examples:
  – Two examples have same attribute/value pairs, but different classifications
  – Some values of attributes are incorrect because of errors in the data acquisition process or the preprocessing phase
  – The instance was labeled incorrectly (+ instead of -)

• Also, some attributes are irrelevant to the decision-making process
  – e.g., color of a die is irrelevant to its outcome
Overfitting in Decision Trees

• Irrelevant attributes can result in overfitting the training example data
  – If hypothesis space has many dimensions (large number of attributes), we may find meaningless regularity in the data that is irrelevant to the true, important, distinguishing features

• If we have too little training data, even a reasonable hypothesis space will ‘overfit’
Overfitting in Decision Trees

Consider adding a noisy training example to the following tree:

What would be the effect of adding:

<outlook=sunny, temperature=hot, humidity=normal, wind=strong, playTennis=No>?
Overfitting in Decision Trees

Consider error of hypothesis $h$ over

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

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

$$\text{error}_{\text{train}}(h) < \text{error}_{\text{train}}(h')$$

and

$$\text{error}_{\mathcal{D}}(h) > \text{error}_{\mathcal{D}}(h')$$
Overfitting in Decision Trees

![Graph showing the accuracy of a decision tree model as a function of the size of the tree (number of nodes). The graph compares the performance on training data (solid line) and test data (dashed line).]
Avoiding Overfitting in Decision Trees

How can we avoid overfitting?

• Stop growing when data split is not statistically significant
• Acquire more training data
• Remove irrelevant attributes (manual process – not always possible)
• Grow full tree, then post-prune

How to select “best” tree:

• Measure performance over training data
• Measure performance over separate validation data set
• Add complexity penalty to performance measure
Reduced-Error Pruning

Split data into *training* and *validation* sets

Grow tree based on *training set*

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 node that most improves *validation set* accuracy
Pruning Decision Trees

• Pruning of the decision tree is done by replacing a whole subtree by a leaf node.
• The replacement takes place if a decision rule establishes that the expected error rate in the subtree is greater than in the single leaf.
• For example,

<table>
<thead>
<tr>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Color</strong></td>
<td><strong>Color</strong></td>
</tr>
<tr>
<td>red</td>
<td>red</td>
</tr>
<tr>
<td>blue</td>
<td>blue</td>
</tr>
<tr>
<td>1 positive</td>
<td>1 positive</td>
</tr>
<tr>
<td>0 negative</td>
<td>3 negative</td>
</tr>
<tr>
<td>2 negative</td>
<td>1 negative</td>
</tr>
</tbody>
</table>

Based on Example from M. desJardins & T. Finin
Effect of Reduced-Error Pruning

![Graph showing the effect of reduced-error pruning on accuracy over the size of the tree (number of nodes). The graph compares accuracy on training data, test data, and test data during pruning.](image)

Based on Slide by Pedro Domingos
Effect of Reduced-Error Pruning

On training data it looks great

But that’s not the case for the test data

The tree is pruned back to the red line where it gives more accurate results on the test data
Summary: Decision Tree Learning

• Widely used in practice

• Strengths include
  – Fast and simple to implement
  – Can convert to rules
  – Handles noisy data

• Weaknesses include
  – Univariate splits/partitioning using only one attribute at a time --- limits types of possible trees
  – Large decision trees may be hard to understand
  – Requires fixed-length feature vectors
  – Non-incremental (i.e., batch method)
Summary: Decision Tree Learning

- Representation: decision trees
- Bias: prefer small decision trees
- Search algorithm: greedy
- Heuristic function: information gain or information content or others
- Overfitting / pruning

Slide by Pedro Domingos
### Comparison of Learning Methods

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Neural Nets</th>
<th>SVM</th>
<th>Trees</th>
<th>MARS</th>
<th>k-NN, Kernels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural handling of data of “mixed” type</td>
<td>▼▼</td>
<td>▼▼</td>
<td>▲▲</td>
<td>▲</td>
<td>▼▼</td>
</tr>
<tr>
<td>Handling of missing values</td>
<td>▼▼</td>
<td>▼▼</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>Robustness to outliers in input space</td>
<td>▼▼</td>
<td>▼▼</td>
<td>▲</td>
<td>▼</td>
<td>▲</td>
</tr>
<tr>
<td>Insensitive to monotone transformations of inputs</td>
<td>▼▼</td>
<td>▼▼</td>
<td>▲</td>
<td>▼</td>
<td>▼▼ ▼▼</td>
</tr>
<tr>
<td>Computational scalability (large N)</td>
<td>▼▼</td>
<td>▼▼</td>
<td>▲</td>
<td>▼</td>
<td>▼▼</td>
</tr>
<tr>
<td>Ability to deal with irrelevant inputs</td>
<td>▼▼</td>
<td>▼▼</td>
<td>▲</td>
<td>▼</td>
<td>▼▼</td>
</tr>
<tr>
<td>Ability to extract linear combinations of features</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▼</td>
<td>▼▼</td>
</tr>
<tr>
<td>Interpretability</td>
<td>▼▼</td>
<td>▼▼</td>
<td>▼</td>
<td>▼</td>
<td>▼▼</td>
</tr>
<tr>
<td>Predictive power</td>
<td>▲</td>
<td>▲</td>
<td>▼</td>
<td>▼</td>
<td>▼▼</td>
</tr>
</tbody>
</table>

[Table 10.3 from Hastie, et al. Elements of Statistical Learning, 2nd Edition]