Today at a glance
- Decision trees for classification and prediction
  - example of football predictions
  - choice of attributes
  - attribute sensitivity
  - decision trees – what’s good, what’s bad
- The reality of decision trees
  - the size matters? …. Again?!
  - splitting rules – problems and solutions
- Attributes
  - continuous values
  - missing values
- Pruning

What defines a data mining task?
- Task-relevant data
- Kinds of knowledge to be mined
  - characterization,
  - discrimination,
  - association,
  - Classification/prediction,
  - clustering
- Domain (background) knowledge
  - Concept hierarchies – support multiple levels of abstraction
  - User beliefs – support identification of expected/unexpected patterns
- Experimental design (includes interestingness measures)
  - Presentation & visualization of discovered patterns
- Analysis of results

Data classification
- Two step process
  - Build model
  - Use model to classify new inputs
- Some terms
  - Training samples
  - Attributes
  - Supervised vs unsupervised learning
- Comparison criteria for algorithms
  - Predictive accuracy
  - Speed
  - Robustness
  - Scalability
  - Interpretability

Classification vs. Prediction
- Classification – predict class labels
  - Example: Classification of manufactured objects as defective or not defective.
- Prediction – predict continuous values
  - Example: Given velocity and current location, predict location after a given amount of time t.

Decision trees – prediction example
Football game prediction system
- Predict the outcome of a football game (will our team win or lose).
- Decision factors - location, weather, team record, opponent record.
- Decision factor values -

<table>
<thead>
<tr>
<th>Location</th>
<th>Weather</th>
<th>Own Record</th>
<th>Opponent Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Rain</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>Away</td>
<td>Cold</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td></td>
<td>Moderate</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td></td>
<td>Hot</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Solutions - win or lose
**Prediction example**

<table>
<thead>
<tr>
<th>Week</th>
<th>Locat.</th>
<th>Weath</th>
<th>Own r</th>
<th>Opp. r</th>
<th>Own</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Home</td>
<td>Hot</td>
<td>Good</td>
<td>Good</td>
<td>Win</td>
</tr>
<tr>
<td>2</td>
<td>Home</td>
<td>Rain</td>
<td>Good</td>
<td>Averg.</td>
<td>Win</td>
</tr>
<tr>
<td>3</td>
<td>Away</td>
<td>Moder.</td>
<td>Good</td>
<td>Averg.</td>
<td>Loss</td>
</tr>
<tr>
<td>4</td>
<td>Home</td>
<td>Hot</td>
<td>Good</td>
<td>Poor</td>
<td>Win</td>
</tr>
<tr>
<td>5</td>
<td>Away</td>
<td>Cold</td>
<td>Good</td>
<td>Good</td>
<td>Loss</td>
</tr>
<tr>
<td>6</td>
<td>Away</td>
<td>Hot</td>
<td>Averg.</td>
<td>Averg.</td>
<td>Loss</td>
</tr>
<tr>
<td>7</td>
<td>Home</td>
<td>Moder.</td>
<td>Averg.</td>
<td>Good</td>
<td>Loss</td>
</tr>
<tr>
<td>8</td>
<td>Away</td>
<td>Cold</td>
<td>Poor</td>
<td>Averg.</td>
<td>Win</td>
</tr>
</tbody>
</table>

**Prediction test**

<table>
<thead>
<tr>
<th>Week</th>
<th>Location</th>
<th>Weath</th>
<th>Own Rec.</th>
<th>Opp. Rec.</th>
<th>Pred.</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Home</td>
<td>Hot</td>
<td>Poor</td>
<td>Poor</td>
<td>No-Data</td>
<td>Win</td>
</tr>
<tr>
<td>10</td>
<td>Home</td>
<td>Modera</td>
<td>Good</td>
<td>Average</td>
<td>Loss</td>
<td>Win</td>
</tr>
<tr>
<td>11</td>
<td>Away</td>
<td>Cold</td>
<td>Good</td>
<td>Good</td>
<td>Win</td>
<td>Win</td>
</tr>
<tr>
<td>12</td>
<td>Home</td>
<td>Hot</td>
<td>Good</td>
<td>Average</td>
<td>Win</td>
<td>Loss</td>
</tr>
<tr>
<td>13</td>
<td>Home</td>
<td>Modera</td>
<td>Good</td>
<td>Average</td>
<td>Loss</td>
<td>Win</td>
</tr>
<tr>
<td>14</td>
<td>Away</td>
<td>Cold</td>
<td>Good</td>
<td>Average</td>
<td>Win</td>
<td>Loss</td>
</tr>
<tr>
<td>15</td>
<td>Home</td>
<td>Cold</td>
<td>Average</td>
<td>Good</td>
<td>Loss</td>
<td>Loss</td>
</tr>
<tr>
<td>16</td>
<td>Away</td>
<td>Modera</td>
<td>Poor</td>
<td>Poor</td>
<td>Loss</td>
<td>Loss</td>
</tr>
</tbody>
</table>

**Sensitivity study - location**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Good</td>
<td>Hot</td>
<td>Good</td>
<td>Good</td>
<td>Win</td>
<td>Win</td>
</tr>
<tr>
<td>2</td>
<td>Average</td>
<td>Rain</td>
<td>Good</td>
<td>Average</td>
<td>Win</td>
<td>Win</td>
</tr>
<tr>
<td>3</td>
<td>Average</td>
<td>Moderate</td>
<td>Good</td>
<td>Average</td>
<td>Loss</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>Average</td>
<td>Hot</td>
<td>Good</td>
<td>Poor</td>
<td>Win</td>
<td>Win</td>
</tr>
<tr>
<td>5</td>
<td>Good</td>
<td>Cold</td>
<td>Good</td>
<td>Good</td>
<td>Loss</td>
<td>?</td>
</tr>
<tr>
<td>6</td>
<td>Good</td>
<td>Hot</td>
<td>Average</td>
<td>Average</td>
<td>Less</td>
<td>Less</td>
</tr>
<tr>
<td>7</td>
<td>Poor</td>
<td>Moderate</td>
<td>Good</td>
<td>Average</td>
<td>Loss</td>
<td>Less</td>
</tr>
<tr>
<td>8</td>
<td>Poor</td>
<td>Cold</td>
<td>Poor</td>
<td>Average</td>
<td>Win</td>
<td>?</td>
</tr>
<tr>
<td>9</td>
<td>Average</td>
<td>Hot</td>
<td>Average</td>
<td>Poor</td>
<td>Loss</td>
<td>Loss</td>
</tr>
<tr>
<td>10</td>
<td>Average</td>
<td>Moderate</td>
<td>Good</td>
<td>Average</td>
<td>Win</td>
<td>Win</td>
</tr>
<tr>
<td>11</td>
<td>Good</td>
<td>Cold</td>
<td>Good</td>
<td>Average</td>
<td>Loss</td>
<td>Less</td>
</tr>
<tr>
<td>12</td>
<td>Good</td>
<td>Hot</td>
<td>Good</td>
<td>Average</td>
<td>Loss</td>
<td>?</td>
</tr>
<tr>
<td>13</td>
<td>Average</td>
<td>Moderate</td>
<td>Good</td>
<td>Average</td>
<td>Win</td>
<td>?</td>
</tr>
<tr>
<td>14</td>
<td>Average</td>
<td>Cold</td>
<td>Good</td>
<td>Average</td>
<td>Loss</td>
<td>Less</td>
</tr>
<tr>
<td>15</td>
<td>Poor</td>
<td>Cold</td>
<td>Average</td>
<td>Good</td>
<td>Loss</td>
<td>Less</td>
</tr>
<tr>
<td>16</td>
<td>Average</td>
<td>Moderate</td>
<td>Average</td>
<td>Poor</td>
<td>Loss</td>
<td>Less</td>
</tr>
</tbody>
</table>

**Decision trees – good and bad**

- Discovers rules from examples - potential unknown rules could be induced.
- Avoids knowledge elicitation problems - system knowledge can be acquired through past examples.
- Can produce new knowledge.
- Can uncover critical decision factors.
- Can eliminate irrelevant decision factors.
- Can uncover contradictions.

- Difficult to choose good decision factors.
- Difficult to understand rules.
- Applicable only for classification problems.
Today at a glance

- Decision trees for classification and prediction
- Example of football predictions
- Choice of attributes
- Attribute sensitivity
- Decision trees – what’s good, what’s bad
- The reality of decision trees
  - The size matters? … Again?!?
  - Splitting rules – problems and solutions
- Attributes
  - Continuous values
  - Missing values
- Pruning

A simple decision tree

A Real decision tree

The top-down procedure

- A.K.A. Recursive Partitioning
  - Find “best” attribute test to install at root
  - Split data on root test
  - Find “best” attribute tests to install at each new node
  - Split data on new tests
  - Repeat until:
    - All nodes are pure
    - All nodes contain fewer than k cases
    - Distributions at nodes indistinguishable from chance
    - Tree reaches predetermined max depth
    - No more attributes to test

Splitting rules

- Information Gain = reduction in entropy due to splitting on an attribute
- Entropy = expected number of bits needed to encode the class of a randomly drawn + or – example using the optimal info-theory coding

\[
\text{Entropy} = -p_+ \log_2 p_+ - p_- \log_2 p_-
\]

\[
\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)
\]
Potential problems?

- Problem with Node Purity and Information Gain:
  - prefer attributes with many values
  - extreme cases:
    - Social Security Numbers
    - patient ID’s

Potential solution

\[
\text{GainRatio} \left( S, A \right) = \text{Entropy} \left( S \right) - \sum_{v \in \text{Values}(A)} \frac{S_v}{S} \log_2 \frac{S_v}{S}
\]

Information gain

Gain ratio

Gain ratio for equal sized n-Way Splits

Info gain vs. Gain ratio

GINI index

\[
\text{GINI}_{node}(Node) = 1 - \sum_{c \in \text{classes}} \left[ p(c) \right]^2
\]

\[
\text{GINI}_{split}(A) = \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{GINI}(N_v)
\]
Today at a glance

- Decision trees for classification and prediction
  - example of football predictions
  - choice of attributes
  - attribute sensitivity
  - decision trees – what’s good, what’s bad
- The reality of decision trees
  - the size matters? …. Again?!
  - splitting rules – problems and solutions
- Attributes
  - continuous values
  - missing values
- Pruning

Continuous values

- Overfitting

Pre-pruning

- Evaluate splits before installing them:
  - don’t install splits that don’t look worthwhile
  - when no worthwhile splits to install, done
- Seems right, but:
  - hard to properly evaluate split without seeing what splits would follow it (use lookahead?)
  - some attributes useful only in combination with other attributes
  - suppose no single split looks good at root node?

Post-pruning

- Grow decision tree to full depth (no pre-pruning)
- Prune-back full tree by eliminating splits that do not appear to be warranted statistically
- Use train set, or an independent prune/test set, to evaluate splits
- Stop pruning when remaining splits all appear to be warranted
- Alternate approach: convert to rules, then prune rules

Goal of pruning

- Optimal
  - Maximum expected accuracy (test set)
  - Minimum size tree
  - Minimum depth tree
  - Fewest attributes tested
  - Easiest to understand
- Test order not always important for accuracy
  - Sometimes random splits perform well

Advantages of decision trees

- TDIDT is relatively fast, even with large data sets (10^6) and many attributes (10^3)
  - advantage of recursive partitioning: only process all cases at root
- Small-medium size trees usually intelligible
- Can be converted to rules
- TDIDT does feature selection
- TDIDT often yields compact models
- Decision tree representation is understandable

Decision trees are intelligible

Well, a correction ....

Not all Decision trees are intelligible.
Predicting probabilities with trees

- Small Tree
  - few leaves
  - few discrete probabilities

- Large Tree
  - many leaves
  - few cases per leaf
  - few discrete probabilities
  - probability estimates based on small/noisy samples

Probability estimation trees

- Smooth large trees
  - correct estimates from small samples at leaves

- Average many trees
  - average of many things each with a few discrete values is more continuous
  - averages improve quality of estimates

- Both

Weaknesses of Decision trees

- Large or complex trees can be just as unintelligible as other models
- Trees don’t easily represent some basic concepts such as M-of-N, parity …
- Don’t handle real-valued parameters as well as Booleans
- If model depends on summing contribution of many different attributes, DTs probably won’t do well
- DTs that look very different can be same/similar
- Usually poor for predicting continuous values
- Propositional (as opposed to 1st order)
- Recursive partitioning: run out of data fast as descend tree

Popular Decision tree packages

- ID3 (ID4, ID5, …) [Quinlan]
  - research code with many variations introduced to test new ideas
- CART: Classification and Regression Trees [Breiman]
  - best known package to people outside machine learning
  - 1st chapter of CART book is a good introduction to basic issues
- C4.5 (C5.0) [Quinlan]
  - most popular package in machine learning community
  - both decision trees and rules
- IND (INDuce) [Buntine]
  - decision trees for Bayesians (good at generating probabilities)
  - available from NASA Ames for use in U.S.

And Last But Not Least When to Use Decision Trees

- Model intelligibility is important
- Problem does not depend on many features
  - modest subset of features contains relevant info
  - not vision
- Speed of learning is important
- Linear combinations of features not critical
- Medium to large training sets

Thank you !!!