ORA Machine Learning

Michael J Kowalchuck

mkowalch@andrew.cmu.edu

Algorithms

- Decision Tree
- Random Forest
  - A forest of decision trees
- JRip (coming soon)
  - Tree of rules
Trees

- Connected acyclic graph with a root

Decision Tree

- Every node is associated with a variable in the data
- Every branch is a value that the parent node can take
- Every leaf has a dependent variable value associated with it
Decision Tree Overview

- Can be used for classification or regression problems
- Ora’s decision tree can only do Classification for now
  - Classification is where we are predicting a variable with discrete categories
- Still useful with unbalanced data (almost all positives or almost all negatives)
- Useful for finding the most important variables
- Weak learner
- Tends towards overfitting
- Walking up the tree from a leaf gives interesting subgroups

Tennis Data

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>FALSE</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>TRUE</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>FALSE</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>FALSE</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>FALSE</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>TRUE</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>TRUE</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>FALSE</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>FALSE</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>FALSE</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>TRUE</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>TRUE</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>FALSE</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>TRUE</td>
<td>No</td>
</tr>
</tbody>
</table>
DT Algorithm

- Pick the splitting variable
  - Pick variable with highest information gain or gini impurity
  - Break data set into subsets, one for each value splitting variable can take
  - create child nodes for each subset and repeat the process for each child

Choosing the Splitting Variable

- Information gain
  - Difference in entropy between parent and entropy of all child nodes
  - Problem: if there are too many unique values for a variable
- Gini impurity
  - How accurate the current split is
  - Useful in regression
  - Not gini coefficient that is something else
DT as Weak Learner

- Weak learners generally do not perform very well
  - Sometimes barely above random chance
- Decision trees performance can improve by picking the right parameters
- Weak learners can do well in bagging

Decision Tree Parameters

- Minimum samples per node
- Maximum tree depth
- Without these the algorithm will perfectly overfit the data
Random Forest

- Random Forest is based on Bagging
  - Ensemble method
  - Bagging is Bootstrap Aggregation

Bootstrap

- If you have a sample of some population that is independent and identically distributed
  - Resample from your sample with replacement (so the same sample can be taken multiple times) until you get a new sample of the same size as the original
  - For each resample calculate your statistic
Build Random Forest

- For every tree in the forest, create a subsample of the data with replacement
  - Size of the forest is a parameter to the algorithm
- For every subsample, create a decision tree
  - These trees are allowed to overfit
    - When deciding which variable to split on only a subset of available variables is considered
      - The size of this subset is $\sqrt{\text{variable}\_\text{count}}$
    - This is the difference from bagged decision trees and random forest