CS60050 Machine Learning - Weekly Report

Nisarg Upadhyaya (19CS30031)

Week 4: 1-3 September

1 Topics covered

- ID3 Algorithm
- Inductive Bias in Decision Tree Learning
- Issues with Decision Trees
- Extensions of basic algorithm
- Conclusion

2 Summary

2.1 ID3 Algorithm

It is a recursive algorithm to create a decision tree which takes as input $Examples$, $Attributes$ and $Target-Attribute$. Create a root node for the tree. If all $Examples$ are positive(negative) then return this single node with label positive(negative). If $Attributes$ is empty return node with label set to most common value of $Target-Attribute$ in $Examples$. Otherwise assign the root with decision attribute A with highest information gain and for every possible value v_i of A:

1. Add a new branch from tree root corresponding to test $A = v_i$
2. Let $Examples_{v_i}$ be the subset of $Examples$ with value v_i for A
3. If $Examples_{v_i}$ is empty then to this branch add leaf node with label set to most common value of $Target-Attribute$ in $Examples$. Else add subtree by calling ID3 with parameters $Examples_{v_i}$, $Attributes - \{A\}$ and $Target - Attribute$

The hypothesis space is a space of all decision trees. ID3's hypothesis space of all decision trees is a complete space of finite discrete-valued functions on the available attributes.

2.2 Inductive Bias in Decision Tree Learning

There are two important points:

1. Shorter trees are preferred over longer trees.
2. Preference for trees with high information gain attributes near the root.

The bias comes from its search strategy and is termed as *preference/search* bias. For Candidate-Elimination it came from the restriction of the set of hypothesis and that is termed as *restriction/language* bias.

2.3 Issues with Decision Trees

Decision trees are prone to overfitting the data. The recursive procedure grows each branch deeply enough to perfectly classify the training examples. This can lead to overfitting when there is noise or error in the data with the tree becoming long at some points just to accomodate one odd training example which then affects the generalization over unseen examples. There are two broad approaches, pre-pruning (stop growing the tree at some point) and post-pruning (grow full tree and the remove nodes) of which the second is more popular.

2.3.1 Reduced Error Pruning

Split data into training and validation set. Build tree using training set. Measure the accuracy on validation set and prune leaf nodes one by one until the accuracy starts decreasing on the validation set. Each time we prune the node which causes the largest increase in the accuracy on validation set.

2.3.2 Rule Post Pruning

In this we allow the tree to grow completely. Convert this tree into equivalent set of rules (one for each path from root to leaf). Prune rules by removing any preconditions which result in improving its estimated accuracy.

2.4 Extensions of basic algorithm

1. Continuous Valued Attributes: These can be incorporated by introducing threshold(s) to the value of the attribute. Threshold(s) partition the continuous space into discrete spaces which can then be added to the decision tree.

2. Attributes with Many Values: These can create problem by inducing many small partitions which makes the performance poor on unseen examples. This is resolved by adding *split information* to the gain calculated before. *Split Information* term discourages the selection of attributes with many uniformly distributed values.

3. Unknown Attribute Values: We can assign the missing attribute value the one that is most common among training examples at that node. Other complex methods include taking into consideration the probability distribution of the value of the missing attribute at that particular node.

4. Attributes with Associated Cost: Introduce an additional cost term with the gain calculated.

5. *Gini Gain* and *Gain Ratio* (point 2) are some other impurity measures which can be used.

6. Regression Trees: They predict a value from a set of attributes instead of a class label. Work by calculating sum of squared errors instead of information gain. Similar steps of pre-pruning or post-pruning can be used in regression trees as well.

2.5 Conclusion

Decision trees are very fast and flexible. They can handle large datasets easily and also incorporate several attribute types, missing values and there are several strategies and heuristics to choose from to improve their performance. However, they are unstable due to the high variance and do not compete very well with other algorithms in terms of accuracy.

3 Challenging concepts

None

4 Interesting concepts

It was interesting to see how pruning a tree can improve accuracy on unseen examples.

5 Concepts not understood

None

6 Ideas

The ID3 algorithm doesn't backtrack, it selects the locally optimum attribute. It would be interesting to investigate the possibility of allowing backtracking, possibly through some efficient method such as dynamic programming to see if we can improve the overall tree structure and attribute selection at each node, instead of greedily selecting it based on current gain values only.