Machine Learning (CS60050) – Weekly Report

Kaushal Banthia (19CS10039)

Week 4: 1st – 3rd September, 2021

Topics Covered:

- Gain Function
- Hypothesis Space Search (ID3)
- Overfitting the Data
- Attributes with many values
- Unknown Attribute Values and Attributes with Costs
- Gini Index
- Regression Tree
- Advantages and Disadvantages of Decision Trees
- Sequential Covering Algorithms
- General to Specific Beam Search
- Beyond Decision Trees

Summary (Topic Wise):

- Gain Function

  - Value lies between 0 & 1.
  - Gain of 0 means a 50/50 split of the positive and negative training examples.
  - Gain of 1 from perfect uncertainty to perfect certainty.

- Hypothesis Space Search (ID3)

  - The hypothesis space is complete. (Target function included)
  - Input: Training examples (Attributes and the target attribute)

```
1. Create a Root Node
      • If all the training examples are positive or negative, return a
        single node with the same label.
      • If the attribute is empty, return a single node with the most
        frequent label.
2. Select the best attribute A
3.     For each possible value v_i of A
4.         Form the subset S of training examples, whose value of A is v_i
5.         If S is empty
6.             Return by adding a leaf node below with most frequent label
7.         Else
8.             Recursively call ID3 with S to form the subtree.
```

  - ID3 has a preference bias (incomplete search strategy).
  - It also has a preference for short trees, and for those with high information gain attributes near the root.

- Overfitting the Data
  - May not have the best generalization in performance, due to noise in data, or very little data.
  - h overfits the training data, if there exists another hypothesis h', such that h' has greater error than h on the training data, but smaller error on the test data than h.
  - Avoid splitting when a very deep tree is required to fit just one odd training example.
  - To avoid overfitting:
    - Prepruning: Stop growing the tree at some point during construction when it is determined that there is not enough data to make reliable choices.
    - Postpruning: Grow the full tree and then remove nodes that seem not to have sufficient evidence. (More popular)
  - Methods of evaluating subtrees to prune
    - Cross Validation: Reserve hold-out set to evaluate utility (more popular).
    - Statistical testing: Test if the observed regularity can be dismissed as likely to occur by chance.
    - Minimum Description Length: Is the additional complexity of the hypothesis smaller than remembering the exceptions? (Regularization in other contexts)
  - Reduced Error Pruning: A post-pruning, cross validation approach

```
1. Partition training data into "grow" set and "validation" set.
2. Build a complete tree for the "grow" data
3. Until accuracy on validation set decreases, do:
4.     For each non-leaf node in the tree
5.         Temporarily prune the tree below and replace it by majority vote.
6.         Test the accuracy on the validation set
7.     Permanently prune the node with the greatest increase in accuracy on
   the validation test.
```

  - The problem is the it uses less data to construct the tree.

- Attributes with many values
  - If an attribute has many values, gain will select it
  - Solution: Use GainRatio instead
  - $GainRatio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)}$
  - $SplitInformation(S,A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|}$, where S_i is the subset of S for which A has value v_i

- Unknown Attribute Values and Attributes with Costs
  - To handle missing values of attribute A, use the training examples anyway and sort through the tree.
  - If node n tests A
    - assign most common value of A among other examples sorted to node n.
    - Assign most common value of A among other examples with same target value
    - Assign probability p_i to each possible value v_i of A and assign fraction p_i of example to each descendant in the tree
  - To learn a consistent tree with low expected cost, replace gain by $\frac{Gain^2(S,A)}{Cost(A)}$

- Gini Index

  - $Gini = 1 - \sum_i p(i)^2$
  - After applying attribute A, $Gini(A) = \sum_{v \in A} p(v) \sum_i p(i|v)\big(1 - p(i|v)\big)$
  - Gini can be interpreted as the expected error rate.

- Regression Tree

  - Uses a set of attributes to predict the value instead of a class label.
  - Computes the sum of squared errors.
  - Partition the attribute space into a set of rectangular subspaces, each with its own predictor, with the simplest predictor being a constant value.
  - To minimize the square error on the learning sample, the prediction at a leaf is the average output of the learning cases reaching that leaf.
  - Impurity of a sample defined by the variance of the output in that learning sample (LS): $I(LS) = var_{y|LS}\{y\} = E_{y|LS}\{\big(y - E_{y|LS}\{y\}\big)^2\}$
  - The best split is the one that reduces the most variance
  - $\Delta I(LS, A) = var_{y|LS}\{y\} - \sum_a \frac{|LS|_a}{|LS|} var_{y|LS_a}\{y\}$
  - Pruning is application of prepruning or postpruning. In postpruning, the tree that minimizes the squared error on VS is selected.

- Advantages and Disadvantages of Decision Trees

  - Advantages: Very fast, Flexible, and are easy to interpret.
  - Disadvantages: Instability of trees (due to high variance), Not always have the best accuracy, as compared to the other algorithms.

- Sequential Covering Algorithms

  - Learn rules one by one, from training examples and then remove those that have are covered by the rule. Iterate this process till all examples have been covered.
  - Learn one rule. (General to specific beam search). Involves a greedy DFS.

- General to Specific Beam Search

  - Initialize h with the most general rule and maintain a candidate list of the k top performing rules.

```
1. For each rule in the candidate list
2.     Specialize with a constraint (A=v) by adding in CNF
3.     Consider all possible attribute value pairs in the Training Examples
   (excluding any attribute sets that are already present in the antecedent)
4.     Update h with best performance measure(Entropy of class distribution)
5. Update the candidate list with k top performing rules
6. Iterate above till it covers all attributes in the list
7. Choose the best and form the consequent by assigning the most frequent
   target value in the covered set.
```

- Beyond Decision Trees

  - Oblique Decision Trees
    - A combination of multiple attributes checked in a node.
    - A hyperplane dividing into two halves of a space.
    - Usually small (Highly dependent on the choice of hyperplanes, central axis projection and perceptron training).
  - Random Decision Forest
    - Multiple trees from randomly constructed subspaces.
    - For m attributes, 2^m subspaces.
    - An aggregation policy on the class labels obtained from each tree, like averaging, voting, posterior probability of a class etc.

Concepts Challenging to Comprehend: None yet.

Interesting and Exciting Concepts: ID3 and Gini Index

Concepts not understood: None yet.

A novel idea: The aggregation policy for a random decision forest can also be done based upon the frequency of the outputs. It could also combine various parameters and have their combined output as the answer, like having voting and averaging done together.