CS 678 Machine Learning
Programming Assignment #3

Automated Classification using Decision Trees

Specification

The basic idea is to write a program that, given a collection of training data for a classification problem, generates a Decision Tree via the ID3 algorithm.

Background

Decision trees are hierarchical data structures functioning as classifier systems. They are constructed based on a set of training data for which the value of the target function is known (i.e. a form of Supervised Learning). ID3 is a greedy algorithm that generates shortest-path decision trees.

Resources

- Your text contains a psuedocode presentation of the ID3 algorithm (Figure 9.3).
- A tutorial describing the operation of the ID3 algorithm has been posted on the course web page (see Decision Tree Generation).
- The course web page also includes a link to the UCI Machine Learning Repository, a good source of databases culled from many different domains.

Implementation

Implement the basic ID3 algorithm to create a decision tree classifier.

```
ID3 (S)
    if all examples in S are of the same class
        return a leaf with that class label
    else if there are no more attributes to test
        return a leaf with the most common class label
    else
        choose the attribute a that maximizes the information gain of S
        let attribute a be the decision for the current node
        add a branch from the current node for each possible value v of the attribute a
        for each branch
            "sort" examples down the branches based on their value v of attribute a
            recursively call ID3(S_v) on the set of examples in each branch
```

To implement the algorithm:

- Use a measure of purity (e.g. Entropy):

$$\text{Entropy}(S) \equiv -\sum_{i=1}^{k} p_i \log_2 p_i$$

where S is the collection of examples, k is the number of categories, and p_i is the ratio of the cardinality of category i to the cardinality of S, as in $p_i = N_i / N$

- Then use the formula for Information Gain:

$$\text{Gain}(S, a) = \text{Entropy}(S) - \sum_{v=\text{values}(a)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where *values(a)* is the set of all possible values for attribute a, and S_v is the subset of set S for which attribute a has value v.

Data Sets

Sample datasets have been posted on the course Web page. Datafile format is:

NumTargets
    // number of targets (classifications)
T: *targetNames*
    // names of targets on one line
NumAttributes
    // number of attributes
A: *attributeName* numAttributeValues *attributeValues*
    // one attribute per line; each attribute takes multiple values
NumExamples
    // number of training data examples
D: *attributeValues targetValue*
    // one example per line

You may assume discrete (nominal) attribute values for all training data. You may also modify the input files in any way you choose. You may of course use any language/platform.

Requirements

Submit a written report and be prepared to present your solution to the class:

- □ Include complete documentation of your code.
- □ Describe your approach, any interesting problems encountered or experiments performed, packages used, etc.
- □ Demonstrate/test the effectiveness of your classifier.
- □ Include a discussion/analysis of your results.
- □ Extract the *rule-base* (IF-THEN) or visualize your decision tree.

Further Investigation (extra credit)

- □ Find/create a different problem domain and dataset
- □ Incorporate numeric-valued training data
- □ Add "*Prediction* mode" operation to your program (i.e. input an unseen example and use the decision tree to output a prediction/classification)
- □ Experiment with using weighted training data
- □ Experiment with alternative splitting functions
- □ Implement pruning