CS 678 Machine Learning
Programming Assignment #3

Decision Tree Construction via ID3

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. they are a form of Supervised Learning). ID3 is a greedy algorithm that generates shortest-path decision trees.

Resources

- A Tutorial describing the operation of the ID3 algorithm has been posted to BlackBoard (see Decision Tree)

Data Sets

Five sample datasets have been posted to BlackBoard. Datafile format is:

- Header: metadata describing the dataset
  attributeName: attributeValues # list of attributes with their possible values
  targetName targetValues # values the target can take
- Data: Training set
  attributeValues, targetValue # one example/line

You may modify/use this information in any way you choose.

The datasets all involve a classification problem:

- Treatment (HW #7): predict how a patient will respond to a new treatment
- CAD: predict if a patient is likely to incur coronary artery disease
- Fishing: is it a good day to fish?
- Contact-lens: suggest the best (out of three) contact-lens option
- Caesarian: predict whether a patient will require a Caesarian section

Implementation

To implement the ID3 algorithm, you will need:

- A measure of purity:

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

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 $p_i = N_i / N$

- 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.

Using these metrics, the basic ID3 algorithm for creating 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 majority 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 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 now at each branch
```

CS 678 Machine Learning
Programming Assignment #3

Requirements

Non-recursive program:

- Execute your program on the Treatment (HW #7) dataset.
- Execute your program through the *1st level* of the Fishing dataset (i.e. bottom of page 2 of the Tutorial).

- Write re-usable functions for the fundamental metrics. Try to generalize your program such that it can work with either dataset.
- Include some type of visualization of the output of your final decision tree (even if it is just an interpretable text representation). You may use any package, utility, or program to create the visualization.
- Validate the correctness of your program as executed on the Treatment dataset by testing it against the DecisionTreeClassifier() found in **`scikit.learn`**.

Submit a written report (PDF):

- Include complete documentation, source code, and program output.
- Describe your approach, data structures used, any interesting problems encountered or experiments performed, any special packages used, etc.
- Be prepared to present your solution in class.

Further Investigation

Ungraded; this is for your own benefit.

- Completeness
    - Implement full recursion. Execute your program to completion of the Fishing, CAD, and Contact-lens datasets.
- Extensions
    - Implement the capability to process numeric (continuous-valued) attributes. Execute your program on the Caesarian dataset.
    - Add "*Classification* mode" to your program (i.e. input an unseen instance and use your decision tree to output a prediction/classification).
    - Extract the *rule-base* (IF-THEN) from your decision tree.
- Structural Enhancements
    - Implement pre/post-pruning. Evaluate performance.
    - Employ bagging (bootstrap aggregating) to implement Random Forests. Evaluate performance.
- Visualization
    - Create a dynamic visualization of your growing tree.