CISC Introduction to Machine Learning
Homework 1

Due: see Canvas

There are two parts: individual problems and group problems. Each student should upload one submission for individual problem. Each group should upload one submission for group problems.

Individual Problem (15pt)

Problem 1. (10pt) Consider the training dataset given below. X_1, X_2, X_3 and X_4 are the attributes/features and Y is the class variable.

Y	X_1	X_2	X_3	X_4
+1	0	1	0	1
+1	1	0	1	0
+1	1	1	1	0
+1	0	0	0	1
+1	1	1	1	0
-1	0	0	1	1
-1	0	0	0	0
-1	0	0	1	0
-1	1	0	0	0
-1	0	0	1	1

(a) (5pt) Learn a decision tree using the ID3 algorithm. Show your computation steps.

(b) (3pt) Draw a decision tree having only 4 leaf nodes, 3 internal nodes and depth bounded by 2, that has 100% accuracy on the given dataset.

(c) (2pt) For the two trees in (a) and (b), which decision tree will you prefer? Please explain your answer.

Problem 2. (5pt) Let x be a vector of n Boolean variables $\{X_1, ..., X_n\}$ and let k be an integer less than n. Let f_k be a target concept which is a disjunction consisting of k literals.

Examples of $f_2 : X_1 \vee X_2, X_1 \vee \neg X_4$, etc.

Examples of f_3: $X_1 \vee X_2 \vee \neg X_{10}, X_1 \vee \neg X_4 \vee \neg X_7$ etc.

State the size of the smallest possible consistent decision tree (namely a decision tree that correctly classifies all possible examples) for f_k in terms of n and k and describe its shape.

What to Turn in

- a pdf file with your answer

Group Problem (35pt)

In this homework, your group will implement and test the decision tree learning algorithm (See Mitchell, Chapter 3). You can use either Java or Python to implement your algorithms.

- Download the two datasets available on Canvas. Each data set is divided into three sets: the *training set*, the *validation set* and the *test set*. Data sets are in CSV format. The first line in the file gives the attribute names. Each line after that is a training (or test) example that contains a list of attribute values separated by a comma. The last attribute is the class-variable. Assume that all attributes take values from the domain {0,1}.
- Implement the decision tree learning algorithm. As discussed in class, the main step in decision tree learning is choosing the next attribute to split on. Implement the following two heuristics for selecting the next attribute.
  1. Information gain heuristic (See Class slides, Mitchell Chapter 3).
  2. Variance impurity heuristic described below.
     Let K denote the number of examples in the training set. Let K_0 denote the number of training examples that have $class = 0$ and K_1 denote the number of training examples that have $class = 1$. The variance impurity of the training set S is defined as:

     $$VI(S) = \frac{K_0}{K}\frac{K_1}{K}$$

     Notice that the impurity is 0 when the data is pure. The gain for this impurity is defined as usual.

     $$Gain(S, X) = VI(S) - \sum_{x \in Values(X)} \Pr(x) VI(S_x)$$

     where X is an attribute, S_x denotes the set of training examples that have $X = x$ and $\Pr(x)$ is the fraction of the training examples that have $X = x$ (i.e., the number of training examples that have $X = x$ divided by the number of training examples in S).
- Implement a function to print the decision tree to standard output. We will use the following format.

```
wesley = 0 :
| honor = 0 :
| | barclay = 0 : 1
| | barclay = 1 : 0
| honor = 1 :
| | tea = 0 : 0
| | tea = 1 : 1
wesley = 1 : 0
```

According to this tree, if wesley = 0 and honor = 0 and barclay = 0, then the class value of the corresponding instance should be 1. In other words, the value appearing before a colon is an attribute value, and the value appearing after a colon is a class value.

Algorithm 1: Post Pruning

Input: An integer L and an integer K
Output: A post-pruned Decision Tree
begin
  Build a decision tree using all the training data. Call it D;
  Let $D_{Best} = D$;
  for $i = 1$ *to* L **do**
    Copy the tree D into a new tree D';
    $M =$ a random number between 1 and K;
    for $j = 1$ *to* M **do**
      Let N denote the number of non-leaf nodes in the decision tree D'. Order the nodes in D' from 1 to N;
      $P =$ a random number between 1 and N;
      Replace the subtree rooted at P in D' by a leaf node. Assign the majority class of the subset of the data at P to the leaf node.;
      /* For instance, if the subset of the data at P contains 10 examples with $class = 0$ and 15 examples with $class = 1$, replace P by $class = 1$ */
    end
    Evaluate the accuracy of D' on the validation set;
    /* accuracy = percentage of correctly classified examples */
    if D' *is more accurate than* D_{Best} **then**
      $D_{Best} = D'$;
    end
  end
  return D_{Best};
end

- Implement the post pruning algorithm given in Algorithm 1 (See also Mitchell, Chapter 3).
- Once we compile your code, we should be able to run it from the command line. Your program should take as input the following six arguments:

```
.\program <L> <K> <training-set> <validation-set> <test-set> <to-print>
L: integer (used in the post-pruning algorithm)
K: integer (used in the post-pruning algorithm)
to-print:{yes,no}
```

  It should output the accuracies on the test set for decision trees constructed using the two heuristics as well as the accuracies for their post-pruned versions for the given values of L and K. If to-print equals yes, it should print the decision tree in the format described above to the standard output.

What to Turn in

- Your **code and a Readme file** for compiling the code.

- A **report file**. For the two datasets on Canvas:
  - Report the accuracy on the test set for decision trees constructed using the two heuristics mentioned above.
  - Choose 10 suitable values for L and K (not 10 values for each, just 10 combinations). For each of them, report the accuracies for the post-pruned decision trees constructed using the two heuristics.