**Artificial Intelligence**

Name and ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment 5**

**Due date: November 30 at 11:59pm**

**Problem 1** (4 points)

Read Chapter 18 and answer the following questions:

1. What are the main advantages and drawbacks of using Ockham’s razor in learning
2. What are the main limitation of the decision-tree learning?

**Problem 2** (6 points)

This file contains eight training data and three test data, and the attributes refer to whether a book will be expensive at the local bookstore.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classification | Bind type | Style of book | Color pictures? | Is the book well known? | Length of book |
| Positive | Hardcover | Novel | Nocolor | Popular | Long |
| Positive | Softcover | Textbook | Nocolor | Popular | Long |
| Negative | Softcover | Novel | Nocolor | Popular | Short |
| Positive | Hardcover | Textbook | Color | Popular | Short |
| Positive | Hardcover | Photojournal | Color | Unknown | Short |
| Negative | Softcover | Textbook | Nocolor | Unknown | Short |
| Positive | Hardcover | Photojournal | Color | Popular | Long |
| Negative | Softcover | Novel | Color | Unknown | Short |

Use these data to construct a decision tree; you should compute the information gains to decide which attributes are more important. For each node of the tree, indicate the corresponding information gain.

**Problem 3** (10 points)

Implement a program for building decision trees. It should read a file with training and test examples, use the training examples to build a tree, and then classify the text examples. The only required output is the classification of the test examples; it does not have to include the tree itself (if you output also the tree, you will have 5 points bonus). The input format is as follows:

<classification> <attribute> <attribute> … <attribute>

…

<classification> <attribute> <attribute> … <attribute>

…

<attribute> <attribute> … <attribute>

…

<attribute> <attribute> … <attribute>

The training examples are above the blank line, and the test examples are below. <classification> is either “positive” or “negative, and each <attribute> is a string of lower-case letters. The length of an attribute is at most twenty characters; successive attributes are separated by one or more spaces. For instance, the following file includes three training examples and two test examples

positive fish low good noplate

negative beef high bad plate

negative fish medium good plate

fish high bad noplate

beef medium good plate

**Sample file:**

The file contains six training values and two test values. These values correspond to whether a coin will be valuable to a collector, and the fields correspond to the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Classification | How many made? | How old is coin? | How much wear is on the coin? |
| Positive | Rare | New | Low |
| Positive | Rare | Old | Low |
| Positive | Common | Old | Low |
| Negative | Rare | Old | High |
| Negative | Common | New | Low |
| Negative | Common | New | High |

From the above table, it should be apparent that the wear on the coin contains the most information for the first classification, as the value for wear is "low" in all of the positive cases. The actual information calculation looks like this:

First, I(p/(p+n), n/(p+n)) can be calculated as I(3/6, 3/6) = -0.5\*lg(0.5) - 0.5\*lg(0.5) = 1. For the case of I(1, 0) (where 0 is the logarithmic singularity), I() goes to zero.

Next, the gain from each attribute can be computed as:

|  |  |  |
| --- | --- | --- |
| Gain(Rarity) = | 1 - [(3/6)\*I(2/3, 1/3) + (3/6)\*I(1/3, 2/3)] | = 0.0817 |
| Gain(Age) = | 1 - [(3/6)\*I(2/3, 1/3) + (3/6)\*I(1/3, 2/3)] | = 0.0817 |
| Gain(Wear) = | 1 - [(4/6)\*I(3/4,1/4) + (2/6)\*I(0,1)] | = 0.4591 |

As expected, we choose Wear to be the first branch of our decision tree. Since the test data both have high wear, they will be classified as Negative, meaning that they do not have much collector's value. The tree should branch again to handle any test cases that have a low wear attribute (using one of the remaining attributes).

**Submitting your assignment**

* Submission via Canvas Assignment.
  + It is your responsibility to submit these assignments in a timely fashion.
* All files should be zipped together.
* Name of your file must include your last name and ID number
* There should be a readme file explaining in detail the exact steps to be taken to compile and execute the code files and the title page
* In case of any code errors, partial credit may be offered based on the code and documentation.

**Late Submission Policy**

* Late work will be not accepted.

**Grading Criteria:**

* Minus 10 points if code does not compile.
* Minus 7 points if it compiles but does not run.
* Further deduction will be depended on your code