ECS629/759 Artificial Intelligence: Assignment 2

Due date: Fri 24 March 2017

The aim of this assignment is to implement the ID3 algorithm in Java to perform decision tree learning and classification for objects with discrete (String-valued) attributes. You will be given two input files, one containing labelled training examples, and the other containing examples which have no label, which your programme has to classify. In each file the examples will be described by a list of attributes, one example per line, with the attribute values separated by commas (comma-separated value or CSV format). You may assume that none of the attribute names or values contains a comma or other punctuation. The first line of each input file contains the names of the attributes. The last attribute on each line in the training file is the class that the example belongs to. Apart from the class, which only appears in the training data, the other attributes will be in the same order in both files.

You are provided with a skeleton programme to get you started. The skeleton programme takes care of the text processing: reading and parsing the CSV files, and finding the set of values that each attribute can have. You may assume that no attribute has a value in the testing data that does not also occur in the training data. The skeleton programme also contains most of the data structures that you need to write your methods. You need to complete the methods `classify()` and `train()`. Do NOT change their specification or any of the predefined data structures, or else you might fail automatic marking. You may add new methods and variables to the ID3 class as they are needed.

You are also provided with two sets of test files (each containing the two input files and the correct output file). Test with (replacing the relevant file names):

```
java ID3 TrainFile.csv TestFile.csv >MyOutput.csv
diff MyOutput.csv OutputFile.csv
```

The `diff` command should give no output if your code is correct. Note that the given test files are very simple tests (the data is taken from the questions in Tutorial 5, so you can check the code step by step). You will need to design your own tests to make sure your code functions correctly under all legal input conditions (e.g. 1 class; 3 or more classes; and cases where the training set can not be perfectly classified).

Marks will be allocated as follows (subject to the usual late penalties):

- 50%: correct functioning of `train()` method
- 20%: correct functioning of `classify()` method
- 30%: code and report describing how your program works. The report should be MAXIMUM 1 page, PDF format ONLY. Originality, design decisions, code quality and ability to follow these instructions will be assessed.

Using the link on your landing page, submit your assignment as a single zip file containing the following 2 files only:

- source code (Java source code, filename `ID3.java`)
- report (PDF format, filename `report.pdf`)

Do NOT:

- put any directory structure in the zip file
- submit more than one pdf file
- submit a Word document
- change the specification of the `classify()` and `train()` methods, nor the `Tree` data structure
- add any statements that print anything to standard output or standard error (if you use print statements for debugging, then either remove them, comment them out, or disable them before submitting your code)