Artificial Intelligence: Assignment 4 - Decision Trees

Submit by: 28/05/2017

April 25th, 2017

Decision Trees

Using as a reference the material available in Sections 18.1 to 18.3 of our textbook, Artificial Intelligence: a Modern Approach, by Peter Norvig and Stuart Russell (3ed), implement and algorithm for induction of decision trees (similar to the ID3, shown in the book, Figure 18.5). Use as node selection function the information gain defined in page 704.

The input to your algorithm will be a set of examples in CSV (Comma Separated Value) format. This set will have several attributes (columns of your CSV table), being the last one the variable of interest for classification. The output of your program must be in the following format:

```
<attribute>
    value1:
        <attribute>
            value1: class1 (counter1)
            value2: class2 (counter2)
    value2: class3 (counter3)
    value3:
        <attribute>
            value1:
               <attribute>
                   value1: class4 (counter4)
                   value2: class2 (counter5)
            value2: class3 (counter6)
```

Where, `attribute` is the root of each subtree, `value#` is one of the values of the attribute (one of the branches of your tree), `class#` is the class value assigned to that branch in the tree and `counter#` is a counter of the number of examples corresponding to that tree branch.

Examples of datasets for testing can be found in the following files: `http://www.dcc.fc.up.pt/~ines/aulas/1617/IA/t4/restaurant.csv`, `http://www.dcc.fc.up.pt/~ines/aulas/1617/IA/t4/weather.csv` and `http://www.dcc.fc.up.pt/~ines/aulas/1617/IA/t4/iris.csv`, described below.

- `restaurant`: (example from textbook, page 700) contains information about customers and restaurants (type of food, waiting time, price etc), and the class attribute (last column) says if the customer will wait or not to eat in that restaurant. The task is to generate a decision tree (as explained in theoretical class and following the ID3 algorithm available in the textbook). This decision tree must be used later to classify (answer if the customer will wait or not) new cases.
- `weather`: contains information about climate conditions to play tennis. The task is to generate a decision tree that can decide what are the best conditions to play tennis.

- `iris`: contains **numerical** information about plants of three classes: iris setosa, iris virginica and iris versicolor. The attributes are petal length and width and sepal length and width. The task is to build a decision tree that can tell to which class a plant belongs to given its sepal and petal length and width.

In addition to generate a decision tree, your program must also be prepared to accept as input a file with test examples, i.e., after generating your tree, you must be able to apply your tree to new examples and be able to classify them appropriately. For example, suppose you generated a tree to the restaurant problem. Now, you can enter new examples (without any class/label) and be able to give them a proper class.

Important note: The dataset iris contains numerical values. You need to implement a way of discretizing these values in order to minimize the size of your decision tree.

What to deliver?

1. Written report

   Organization:

   Introduction
   - What is a decision tree?
   - What is it for?

   Algorithm for decision tree induction
   - search books and the internet for some algorithm
   - explain differences among them
   - discuss about the different metrics used to select attributes to place in the tree during its construction
   - Explain how the ID3 algorithm works

   Implementation
   - language
   - data structures used and justification for using them
   - organization of your code

   Results
   - report the generated trees for each dataset
   - show counters in each branch of the tree

   Final Comments and Conclusions

2. source code and how to compile, run etc, as well as the runtime environment used to run and test your program. The program must run from the command line.

   As usual, submission will be done through Moodle UP.

This work can be performed in teams of up to 2 people.