CS 6375 Machine Learning, 2015 Spring

Homework 4

Total points: 100 points

Part II. Programming: HMM and Viterbi decoding [35pts, Due 03/20/2015]

Assume the HMM model parameters are stored in the file according to the following format:

# of states (let's say *N*)
Initial state probabilities (*N* values here)
Transition probabilities (This will contain *N***N* values in the transition matrix. The values are row-based.)
# of output symbols (let's say *M*)
Output alphabet (*M* values here. They can be discrete numbers or strings for the observations)
Output distributions (This will contain *N***M* values, *M* values for probability mass function for each state, one by one.)

Note that everything is space separated.

For example, the following is an HMM parameter file using the format above:

```
3
0.3 0.3 0.4
0.80 0.19 0.01 0.10 0.80 0.10 0.01 0.19 0.80
2
a c
0.7 0.3 0.5 0.5 0.3 0.7
```

Implement the Viterbi decoding algorithm to find the most likely state sequence for a given observation sequence. You will get the observation sequences from a test file. Each line corresponds to one sequence (space separated between observation symbols). For example,

```
c c c c c c c c c c c c c c a c a c a a a a a a a a
a c a c c c c c c c a c c c a a a
```

You can find the two files from the course homework page:
www.hlt.utdallas.edu/~yangl/cs6375/homework/hw4/
model and test.dat.

Requirement for your program:

Your program should take only two arguments: one is the model file, and the other is the file containing sequences of observation. You can write output to standard out, i.e., the most likely state sequence for each of the observation sequences in the test file.

There should be no graphical user interface (GUI). Any program that does not conform to the above specification will receive no credit.

Grading Criteria

The programming portion of this assignment will be graded based on both correctness and documentation.

Correctness. 30 points will be based on the correctness of your program.

Documentation. 5 points will be based on the documentation accompanying your source code. We expect each source file to contain a paragraph or two at the beginning to describe the contents of that file. The main program should describe the functionality of the program: the type of input it expects, the type of output it produces, and the function that it performs. The data structures used in the program must also be clearly described. The code should be modular. Do provide in-line comments to explain any code that is unusual or potentially confusing.

What to Submit

Programming part:

Your should submit via **eLearning** (i) your source code, (ii) a README file that contains instructions for **compiling** and **running** your program (as well as the platform (Windows/Linux/Solaris) on which you developed your program).

Again, you will receive **zero credit** for your program if (1) we cannot figure out how to run your program from your README file or (2) your program takes more than 2 arguments.

Written part:

Please use a separate file for the written portion of the homework when you submit via eLearning. You can also turn in your written part to the TA or me.