CPS363: Introduction to Bioinformatics
Homework 5 HMM and Viterbi Algorithm
(30 Points)

Files to turn in: source code files, output files or screenshots of your output, a readme file about how to run your program, and a report with your results and plotted graphs.

In this assignment you will implement the Viterbi algorithm and evaluate its effectiveness by experimenting with the dishonor casino's problem.

First you will need to write a function called `genDieNumbers(n)` which can simulate the dealer to generate a sequence of *n* throws using the strategy in Figure 1. The function also remembers which type of die is used for each number.

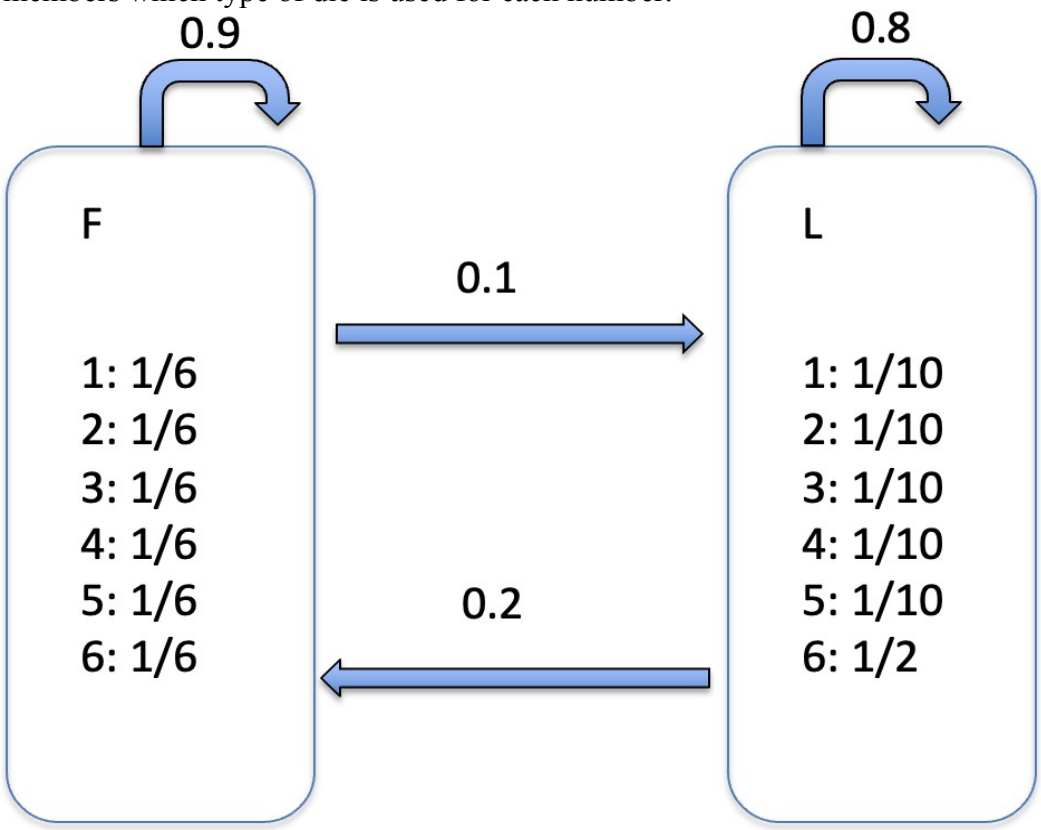

Figure 1

Then you need to write another function which can take the sequence of die numbers generated by the function above and then use the Viterbi algorithm to decode the type of dies used by the dealer for each number. Your program will generate the output in a similar format as that in Figure 2 (notice that the actual prediction and output may be quite different). Fully tested your implementation before move on. Provide at least three output examples such as Figure 2 to show whether your function works.

Figure 2

In order to see the performance of the prediction, we first need to sum up True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). In this example, we treat F's as positives and L's as negatives. For example, if there are 100 numbers generates by the fair die, and 100 numbers generates by the loaded die, among which 80 numbers generated by the fair die are predicted as being generated by the fair die (i.e., TP is 80), 20 numbers generated by the fair die are predicted as being generated by the loaded die (i.e., FN is 20), 70 numbers generated by the loaded die are predicted as being generated by the loaded die (i.e., TN is 70), and 30 numbers generated by the loaded die are predicted as being generated by the fair die (i.e., FP is 30). Then we can calculate statistics such as accuracy and MCC using the following formulas.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}}$$

Next write a main program which evaluates the effectiveness of the Viterbi algorithm for various input sizes using the following structure.

For *n* from 100 to 2,000 with step size 100 :

Do it 10 times:

Use the function genDieNumbers(n) to simulate the dealer and generate a sequence of *n* numbers.

Decode and predict what type of dies is used for each number.

Compare the predictions with the actual to find out the prediction performance in terms of accuracy and MCC.

Get the average accuracy and MCC for the input size *n*.

Then plot the accuracy vs. size and MCC vs. size in two graphs.

Submit your program, readme file about how to compile and run your code (with examples), three sample outputs (either in text file or screenshot) showing the predictions by Viterbi algorithm for sequences of numbers (similar as in Figure 2), and a PDF report with two well annotated graphs (i.e., accuracy vs. size, and MCC vs. size).