Program #4A: Learning Parameter Values from Training Data

Overview: We are developing automatic spelling recognition software to help people in the region using the one-dimensional keyboard. The software is tended to help automatically correct the typos by these people. For simplicity, let's assume the values of the degenerate parameters deg_{sp} and deg_{kb} are both fixed to 2 for all the people in the region using the one-dimensional keyboard. However, the values of Pr_{hit}, Pr_{miss}, Pr_{repeat}, or Pr_{moveOn} vary in the range of 0 to 1 depending on the individual users. For each user, the software has a training stage that requires the user to type the entire Biola vision statements and it records the result in a file. The software then tries to learn the values of Pr_{hit}, Pr_{miss}, Pr_{repeat}, or Pr_{moveOn} of the user based on the result in the file.

A brute-force approach: Assuming the Biola vision statements is stored in a text file ***BiolaVision.txt*** while the result of typing the vision statements is stored in ***CorruptedBiolaVision.txt***. Since Pr_{miss} is simply 1-Pr_{hit} and Pr_{moveOn} is simply 1-Pr_{repeat}, what we need to do is to find good parameter values for Pr_{hit} and Pr_{repeat} in the range of [0,1] such that the probability *Pr(**CorruptedBiolaVision.txt | BiolaVision.txt, X**)* of generating the corrupted document ***CorruptedBiolaVision*** when typing the Biola vision statements is maximized. This can be viewed as a search problem over a 2-dimensional search space (the square area with [0,0], [0,1], [1,0] and [1,1] as the four corners of the square) for the best parameter values for Pr_{hit} and Pr_{repeat}. A brute-force approach is to try all the combinations of values for Pr_{hit} and Pr_{repeat} with fininte granularity such as an increment of 0.01 each time and having (i) Pr_{hit} range from 0.01, 0.02, 0.03, …, to 0.99 and (ii) Pr_{repeat} range from 0.01, 0.02, 0.03, …, to 0.99, 1. For each combination of values for Pr_{hit} and Pr_{repeat}, you can determine the logarithm of *Pr(**CorruptedBiolaVision.txt | BiolaVision.txt, X**)* based on the implementation from Programming #3B. In the end, you'll pick the pair of values for Pr_{hit} and Pr_{repeat} that maximizes the logarithm of *Pr(**CorruptedBiolaVision.txt | BiolaVision.txt, X**)* and set them as the parameter values for Pr_{hit} and Pr_{repeat} (and also set Pr_{miss} to 1-Pr_{hit} and Pr_{moveOn} to

1-Pr_{repeat}).

Your programming task: Implement the brute-force approach above and provide **an option** ***L*** which will (i) ask the user to provide the file names of the files storing the Biola vision statements and the corrupted result respectively, (ii) search to find the best parameter values for Pr_{hit} and Pr_{repeat} , and (iii) set the vlaues of Pr_{hit} and Pr_{repeat} (and also Pr_{miss} and Pr_{moveOn}) accordingly.

Use your program to do the following Experiment 4A:
(i) Consider Homework #1 in the beginning of the semester again. For each of the 8 text files A, B, C, D, E, F, G and H respectively, do the following.

- use option ***L*** in your program to learn the best parameter values based on the contents of file as the corrupted result from typing the Biola vision statements,
- **record** the parameter values in the self-evaluation report, and
- compare the parameter values just learned to the parameter values of Johnny, Winnie, Manny, and Cathy and **record** the person with the closest parameter values.

(ii) Compare your results above with those in **Experiment 3B.** Report whether there is any difference regarding the persons identified as authors of these 8 documents.

Submission: **(i)** Compress your entire Program 4A folder into a zip file and upload it through Biola Canvas. **(ii)** Carefully fill out this self-evaluation report, including the findings from Experiment 4A above. Upload the report through Biola Canvas.