CS185C Midterm 2

Hidden Markov Model

Jordan Conragan, Brett Dispoto

May 3, 2020

I. Introduction

This is an implementation of a Hidden Markov Model written in Java by Brett Dispoto and Jordan Conragan. We had previously tried to implement it in C++. However, due to hidden bugs in the implementation of Problem 3, it didn't work, and we were forced to scrap it.

II. The Program

The Hidden Markov Model class and main method has been bundled into HiddenMarkovModel.java. The Hidden Markov Model class runs on Java 1.8 or later. For the main/testing program to work, there must be a folder named corpus.dos in the parent directory of the folder that contains HiddenmarkovModel.java.

a. Loading the Dataset

The model is trained on 50,000 observations taken from the Brown Corpus. Due to the method used to open and read from the files, the order of the observations read from the files is not guaranteed. This is the most likely reason why the results of the test are slightly different than the results generated by Mark Stamp.

b. Initial Matrices

The initial values for A, B, and Pi (not shown below) were hard coded into the program to match the values used my Mark Stamp. The following figure shows these values. The displayed Observation Matrix is displayed as the transpose of the real matrix and along each row is the letter that it represents.

```
TransitionMatrix:
0.47468, 0.52532,
0.51656, 0.48344,
ObservationMatrix:
a: 0.0374, 0.0391
b: 0.0341, 0.0354
c: 0.0345, 0.0354
d: 0.0383, 0.0391
e: 0.0378, 0.0358
f: 0.0392, 0.0363
g: 0.0369, 0.0405
h: 0.0341, 0.0354
i: 0.0387, 0.0382
j: 0.0406, 0.0391
k: 0.0374, 0.0349
l: 0.0397, 0.0372
m: 0.0355, 0.0354
n: 0.0374, 0.0391
o: 0.0406, 0.0340
p: 0.0360, 0.0340
q: 0.0364, 0.0382
r: 0.0341, 0.0368
s: 0.0406, 0.0405
t: 0.0355, 0.0344
u: 0.0392, 0.0354
v: 0.0406, 0.0396
w: 0.0345, 0.0382
x: 0.0360, 0.0372
y: 0.0341, 0.0377
z: 0.0341, 0.0396
 : 0.0341, 0.0340
```

III. The Program

The results, which can be seen in the figure below, show great success. After 100 iterations, the values of the A and B matrix closely match the values generated by Stamp. This means that the algorithm works, and experiments for Ensemble Learning can begin in earnest.

```
Training done, Iterations = 100
finished trainign HMM
TransitionMatrix:
0.28828012231920175, 0.7117198776807784,
0.7788054998462627, 0.22119450015370787,
ObservationMatrix:
a: 0.1443, 0.0000
b: 0.0000, 0.0228
c: 0.0013, 0.0547
d: 0.0001, 0.0681
e: 0.2215, 0.0000
f: 0.0000, 0.0352
g: 0.0059, 0.0221
h: 0.0000, 0.0721
i: 0.1273, 0.0000
j: 0.0000, 0.0036
k: 0.0024, 0.0065
l: 0.0000, 0.0719
m: 0.0000, 0.0382
n: 0.0000, 0.1128
o: 0.1363, 0.0000
p: 0.0030, 0.0337
q: 0.0000, 0.0015
r: 0.0000, 0.1005
s: 0.0002, 0.1084
t: 0.0282, 0.1263
u: 0.0465, 0.0000
v: 0.0000, 0.0160
w: 0.0000, 0.0227
x: 0.0000, 0.0044
y: 0.0082, 0.0181
z: 0.0000, 0.0011
 : 0.2747, 0.0593
```