CSI 758 Spring 2017 HW#7

Bruce Goldfeder

May 12, 2017

1 Genbank Files

I parsed the file *bacteria.tar* importing the file *genbank.py* to read these files. The following code shows the function to read and parse the data.

```
def readGenesIn(fname):
   # Read in the bacteria file and parse the dna
   data = gb. ReadFile ( fname )
   dna = gb. ParseDNA ( data )
   return data,dna
```

2 Gathering Data

The data was split into the three lists of codons. Each string will includes 30 characters before an ATG and 20 characters after an ATG. So the string is 53 characters long.

- Training STARTS
- Non Training STARTS
- NONSTARTS

Sample Data from STARTS:

```
['tatcaagccgaatcattaatcatcccgtttatgtcctatctaatcgctgtggt',
 'atcgacaaatacataaggaattataaccaaatgacaacgactctccaacagcg',
 'gagaattaatccatcttcgatagaggaattatgggggaagaacctgtgccggc']
```

3 Log Odds Matrices

The transition count matrix from section 2 was converted using separate functions into a probability matrix, then an odds matrix, and finally to a log odds matrix. It was necessary to convert some of the -inf and inf in locations 29,30, and 31 Matrices into zeros (no added value) as these were the 'atg' codon sequences. The following is an example output:

```
array([[ 0.36600297,  0.00404041,  0.221307  ,  0.01401761],
       [-0.37401598,  0.0104713 , -0.22809581, -0.33095389],
       [-0.09722895,  0.04512044,  0.02164133, -0.36942017],
       [-0.02917549,  0.03578911,  0.131412  ,  0.28881074]])
```

4 Scoring and Distributions

Scores were calculated for each string and resulted in the following sets of mean and standard deviation:

```
>>> mu
[6.5060524302520939, 6.6114734370816048, -0.054443821792290 66]
>>> sd
[3.2988927562978123, 3.2252953083010922, 3.1072899967618159]
>>>
```

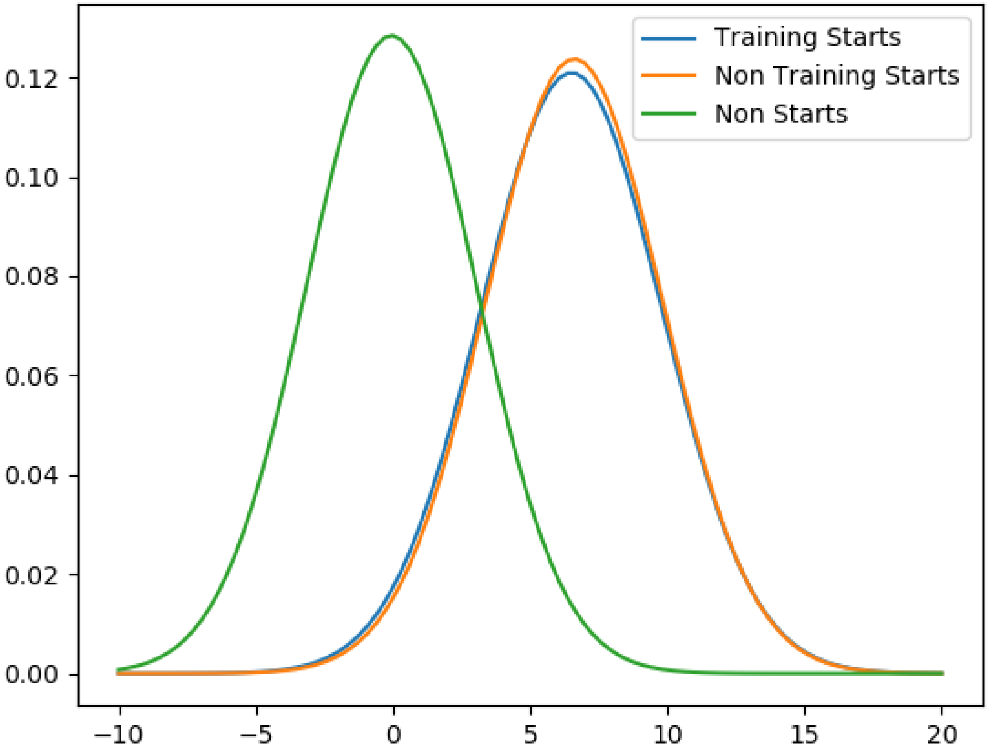

Figure 1: Gaussian Curves defined by the means and standard deviations of the three sets of Strings.

5 Determination of Importance

The methodology I employed was to search through each of the log odds 4x4 matrices and find the largest values. Sorting this data and getting the largest numbers will define which transitions are the most important. The output of this function run against the log odds from the training set and sorted for the top five produced:

```
[(1.3862943611198906, (0, 0), 29),
 (1.3862943611198906, (0, 3), 30),
 (1.3862943611198906, (3, 2), 31),
 (0.71401665916040435, (0, 2), 20),
 (0.69921042997100324, (3, 3), 45)]
```

The five most important transitions in determining START versus NON START are (29,30,31,20,45).

6 New and Improved HMM

Included the function *makeImportant(M4,hiVals)* to zero out all of the transition matrices except for the 5 most important.

7 New and Improved Distributions

Scores were calculated for each string and resulted in the following sets of mean and standard deviation:

```
>>> mu
[4.3457973035730033, 4.3970828806911397, 0.2494919690927252]
>>> sd
[0.60538290374635617, 0.59233582079962266, 0.876846908520996]
>>>
```

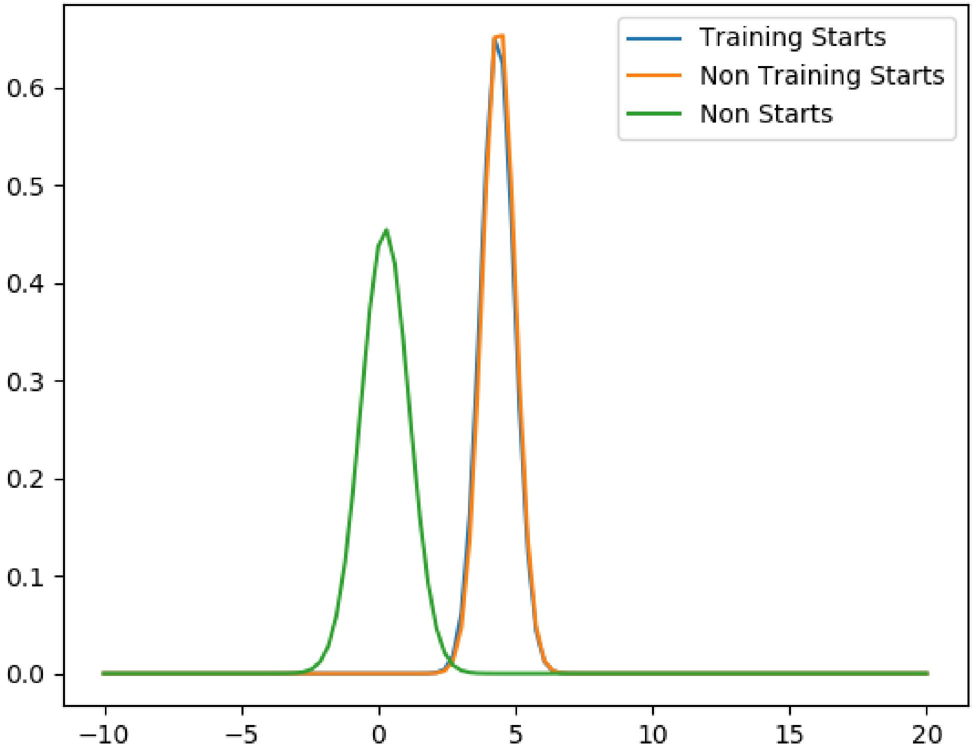

Figure 2: Top Five Important Filtered codons - Gaussian Curves defined by the means and standard deviations of the three sets of Strings.

8 Analysis

The second graph provides a clear delineation between the two states with better separation of the full Gaussian curve and almost no overlap with much lower variance than the first graph. The Hidden Markov Model contains the summation of all 52 transition probabilities (using log odds matrix). The analysis of the second experiment is based on the fact that not all of these transitions have the same weight as they each have different log odds. Selecting the top five values and its effect is described below:

Using standard notation for Hidden Markov Model this can be explained.
A = State Transition Probabilities
B = Observation Probability Matrix
π = Initial State Distribution
$\mathcal{O} = (\mathcal{O}_0, \mathcal{O}_1, ...\mathcal{O}_{52})$

The calculated state transition probabilities (log odds matrix) calculated over all 52 observations is: $\Sigma_{T=0}^{52} A_T$ As we can see in the data not all of the transitions are positively correlated to our START training set. The summation over this entire set causes an averaging bringing the graphs closer together and a loss of precision. Reducing this to the five transitions with the highest positive correlation to creates a much better analysis: $\Sigma_{i=0}^{5} A_i$ where $i \in A_{(TopFiveHighest)}$

This is made obvious in the comparison of the two graphs.