CSI 758 Spring 2017 HW#7

Bruce Goldfeder

May 11, 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
[5.0586087659703542, 4.9073895502794871, 0.14383108480365919]
>>> sd
[3.9307047856499771, 3.7499019160152116, 3.8040217591338523]
>>>
```

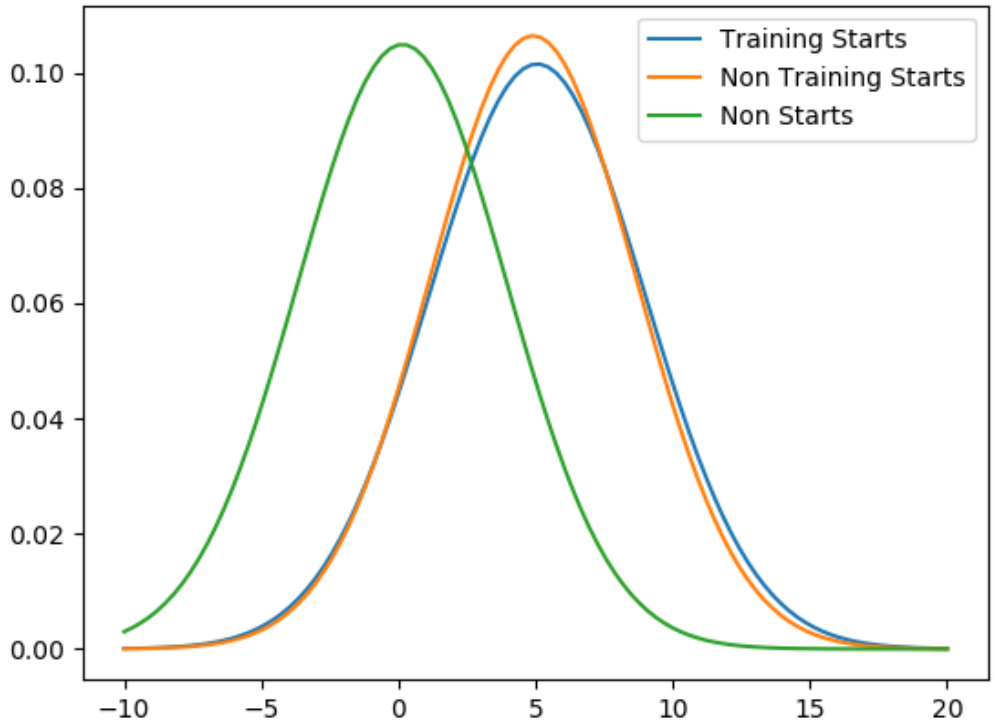

Figure 1: Gaussian Curves defined by the means and standard deviations of the three sets of Strings.

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:

```
[(1.5244862684074811, (3, 0), 29),
 (0.94644377004484259, (0, 3), 33),
 (0.93232141214778641, (0, 3), 45),
 (0.92004976291045781, (0, 3), 36),
 (0.85991398319637202, (2, 3), 22)]
```

The five most important transitions in determining START versus NON START are (29,33,45,36,22).

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
[2.1463238053508933, 2.1110450365101974, 0.11623067751567343]
>>> sd
[1.1479035772626049, 1.138241007135119, 1.3273777785097298]
>>>
```

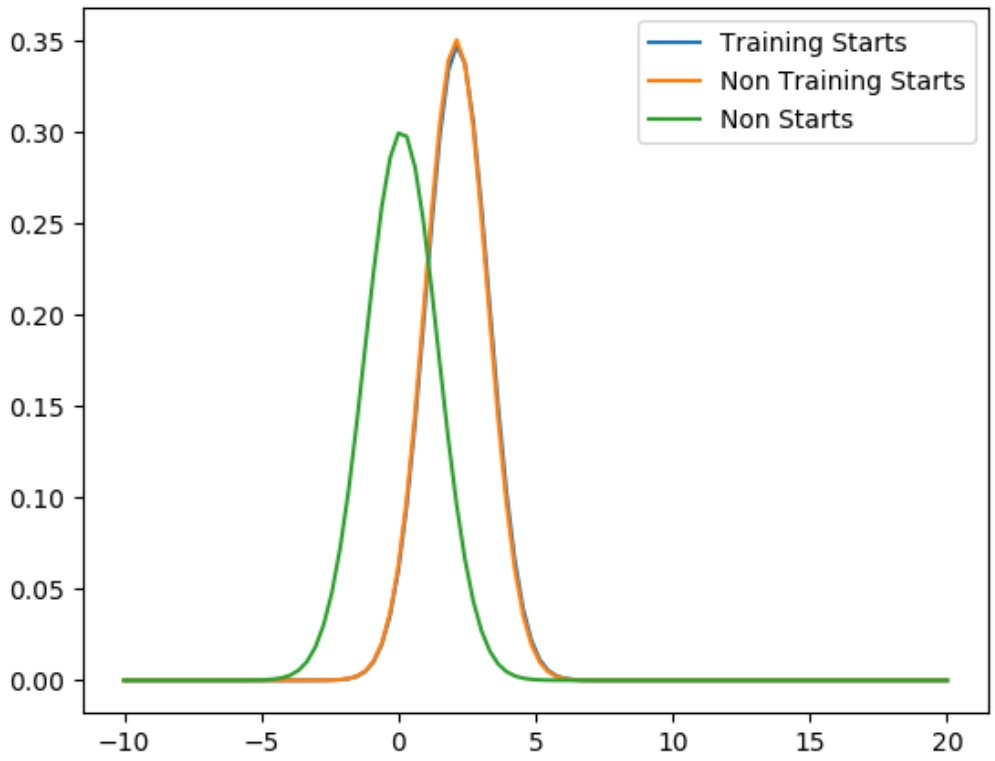

Figure 2: Top Five Important Filtered codons - Gaussian Curves defined by the means and standard deviations of the three sets of Strings.

8 Analysis

The second graph provides more area of separation than the first graph. Also, the standard deviation is better on the second graph (accounting for the delta in mean). An interesting note is that while the means for the start graphs varied significantly the mean for the non standard was reduced only slightly. The μ_2/mu_1 for Training Starts and Non Training Starts is .42 and .43 respectively however the Non Starts ratio of means was .81. This indicates to me that the five important codons selected are a very strong

indicator for defining Non Starts. The thought process of looking at the log odds is that this is a straight summation. The largest values would have the largest impact on making this a separator. I could also have selected the largest negative numbers for a counter-indicator and factored that into my assessment which would have provided better separation and accuracy. This effect seems to have reduced the overfitting of non-essential data elements similar to the PCA reducing dimensions.