ECE 8540

Lab 7 - Viterbi Algorithm

Tim Driscoll

November 23, 2021

1 Introduction

This lab report concerns the problem of using a Hidden Markov model (HMM) to develop a code that executes the Viterbi algorithm. The HMM is composed of two different DNA states, H and L. The H state represents high GC content and the L state represents low GC content. From each state there is a different probability for each of the possible DNA nucleotide (A, C, G, T) to occur. The model also provides details on the probability of state transitions. The Viterbi algorithm is a method that allows for the most probable path of a sequence to be computed. In this example a DNA sequence was given as the input and the most probable path of states that produced this sequence was computed. The HMM model in combination with the Viterbi algorithm allows us to efficiently determine the most probable state path that produced a specific sequence of DNA. This method can be applied to a variety of different problems if a state model is available. This method is a large improvement when compared to previous methods for determining the most probable path. Previous methods were very computationally expensive and required large amounts of time to compute and produce a final output. The Viterbi algorithm reduces the number of computations and produces accurate probability predictions.

2 Methods

In order to determine the path of maximum probability the probabilities in the model needed to be defined. The prior probabilities were {0.5, 0.5}. The state transition probabilities for state H were {0.5, 0.5} and for state L were {0.4, 0.6}. Within each state the possible DNA nucleotide {A, C, G, T} could be observed with a corresponding probability of {0.2, 0.3, 0.3, 0.2} for state H and {0.3, 0.2, 0.2, 0.3} for state L.

After defining all the probabilities in the model the Viterbi calculations could be completed. It is important to note that the calculations were done using the sums of log_2 of probabilities. This method takes the log_2 of each of the probabilities to adjust them from decimal values. Probabilities are then solved for by taking the sum instead of the product. By performing this calculation the probabilities do not become extremely small with large test sequences. This method will be further observed by the outputted probabilities. Since log_2 of all the probabilities was taken the final values will reflect numbers in the range of -25.66 to -2.74, where the larger numbers represent higher probabilities. The algorithm was run by iterating through each DNA nucleotide in the input sequence and finding the highest probability for the nucleotide to be emitted by each state. The following steps were completed to implement the algorithm:

1. Calculate the probability that the first nucleotide is emitted by state H and state L.

Store the results in the probability table. These probabilities can be calculated using the following equations:

$$p_H(1) = -1 + p_{NucleotideH} \tag{1}$$

$$p_L(1) = -1 + p_{NucleotideL} \tag{2}$$

The -1 in the above equations represent the prior probabilities for each state, which is the same for both. The variable $p_{NucleotideH}$ represents the probability that the given nucleotide {A, C, G, T} is emitted in the H state. The variable $p_{NucleotideL}$ represents the probability that the given nucleotide {A, C, G, T} is emitted in the L state. These equations are only used in the first iteration of the algorithm.

2. For the remaining nucleotide in the sequence the following equations are used to calculate the maximum probabilities that the given nucleotide is emitted by each state.

$$p_H(i) = p_{NucleotideH} + max(p_H(i-1) + p_{HH}, p_L(i-1) + p_{LH}) \tag{3}$$

$$p_L(i) = p_{NucleotideL} + max(p_H(i-1) + p_{HL}, p_L(i-1) + p_{LL}) \tag{4}$$

In equations 3 and 4 p_{HH} represents the probability that state remains in state H and p_{HL} represents the probability of a state transition from H to L. Similarly, p_{LL} represents the probability that state remains in state L and p_{LH} represents the probability of a state transition from L to H. Equations 3 and 4 compute all the possible ways that the nucleotide at position i could occur in each state, the maximum for each state is what is saved and represented in the final table.

3. Increment i by one until the probabilities for each nucleotide in the test have been calculated using equations 1-4.

4. Back track through both p_H and p_L comparing the probabilities at each index. Select the maximum probability from the two states and record the corresponding state letter (H or L) at the specific index. After the backtracking is complete a most probable state sequence is produced. In the case where the probability from each state is equivalent at a given index the L state is selected for the most probable state sequence.

2.1 MATLAB Implementation

Once it was determined how the Viterbi algorithm functioned the calculations were implemented in software. The 2020a version of MATLAB on a Windows operating system was used to implement the code. All the initial probabilities were defined in the code and a for loop was used to iterate over each one of the input sequences separately. The loop was ran for each nucleotide in the DNA sequence. At each iteration the nucleotide in the sequence was determined using if statements so the proper probability could be used in the equation. If it is the first letter than equations 1 and 2 are used to determine the two maximum state probabilities, otherwise equations 3 and 4 were used. At the conclusion of this loop the max probabilities for both states at each index were calculated. A for loop was then used to run through all the probabilities in reverse order. The probabilities for each state were compared and the maximum was declared as the most probable state at that index. At the end of this loop the most probable path was determined and the algorithm was complete.

Table 1: Table of max probabilities for DNA strand 'GGCACTGAA'.

DNA Nucleotide	H State Probability	L State Probability
G	-2.74	-3.32
G	-5.47	-6.06
C	-8.21	-8.80
A	-11.53	-10.95
C	-14.01	-14.01
T	-17.33	-16.48
G	-19.54	-19.54
A	-22.86	-22.01
A	-25.66	-24.49

3 Results

For this lab the results can be observed best in a tables 1, 2, and 3. Tables 1 and 2 show the max probabilities at each data index or nucleotide in the tested sequence. As shown in both tables the probabilities are all represented by negative numbers. The numbers closer to zero represent the state with the higher probability at the specific index location. These tables were then traced backwards to produce the most probable paths shown in table 3. The path was discovered by selecting the state with higher probability for each DNA nucleotide in the sequence. Viewing tables 1, and 2 there are multiple instances where the probability is this same for both states. In this situation the L state was always chosen. Depending on the problem the algorithm can be adjusted to behave differently in the situation where state probabilities are the same. When running the Viterbi algorithm on these simple tests the run time was very quick pointing to the computational efficiency of the algorithm. Although the run time was quick in this scenario it can also be credited to the small size of the tests. The results from table 1 and row one of table 3 were verified using an external source and were confirmed to be completely accurate. The probability of the path corresponding to test one was $2^{-24.49}$ = 4.24E-8 when converted back from log_2. The probability of the path corresponding to test 2 two was $2^{-25.01}$ = 2.96E-8 when converted back from log_2.

4 Conclusion

In this lab a Hidden Markov model (HMM) was used to develop a code that successfully executes the Viterbi algorithm. The algorithm was tested on two examples where the inputs were sequences of DNA nucleotide and the outputs represented the most probable state sequence that produced the input. The results from the example were verified for accuracy using an external reference. The algorithm functioned by iteratively calculating the max probability that the current nucleotide in the sequence was emitted from each state, H and L. Tables 1 and 2 represent the outputs of the maximum probability the nucleotide was emitted from each of the states. These tables could then be backtracked through to produce the most probable state path, displayed in 3. When backtracking the max probability from the two

Table 2: Table of max probabilities for DNA strand 'TCAGCGGCT'.

DNA Nucleotide	H State Probability	L State Probability
T	-3.32	-2.74
C	-5.80	-5.80
A	-9.12	-8.27
G	-11.33	-11.33
C	-14.07	-14.39
G	-16.80	-17.39
G	-19.54	-20.12
C	-22.28	-22.86
T	-25.60	-25.01

Table 3: Summary of tested DNA sequences and corresponding state sequence.

Test #	Initial DNA Sequence	Most Probable State Sequence
1	GGCACTGAA	HHHLLLLLL
2	TCAGCGGCT	LLLLHHHHL

states was compared and the maximum was where the nucleotide was most likely emitted from. This algorithm provides a computationally efficient method to decoding the DNA sequence and finding the most probable state path. This method improves upon previously used brute force computations that were time consuming and inefficient in decoding the most probable path of a sequence. The Viterbi algorithm reduces the computational expense and provides accurate results in decoding problems.