Hussain Almattar

Prof. Purtee

CSC 246

May 2nd, 2021

Project 3: Sequential Data

Description of Experimental Setup

This algorithm utilizes the forward-backward algorithm to find convergence within the HMM model. I decided to go with 3 and 5 states for my main experimentation. Higher states took much longer and much more memory. Fortunately, my memory was sufficient to do up to 10 states. Notably, those took much longer; Sometimes overnight. I decided to experiment with max iterations without convergence pausing. That means that I ran through the full epochs without stopping. This may not exactly show the moment of convergence, but it shows more data through time. I chose the epsilon value to be 500 as it is closest to a middle ground among various epochs and sample sizes.

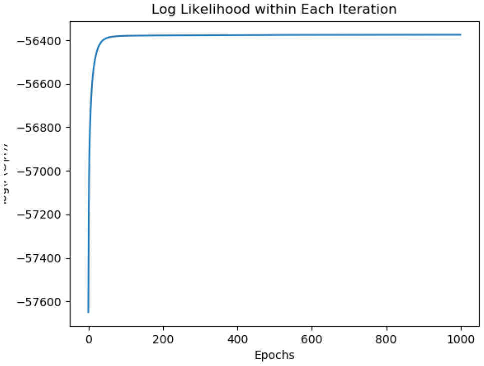

Figure 1: 1k epochs – 10 samples - 486 seconds - 5 states

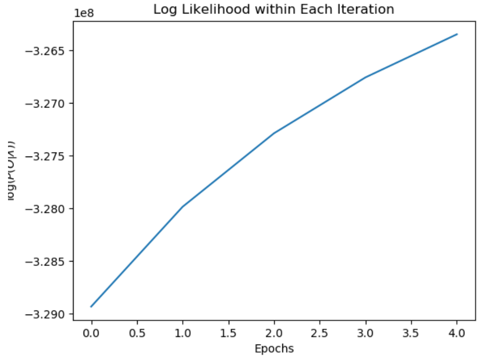

Figure 2: 5 epochs - 50k samples - 3.35 hrs - 5 states

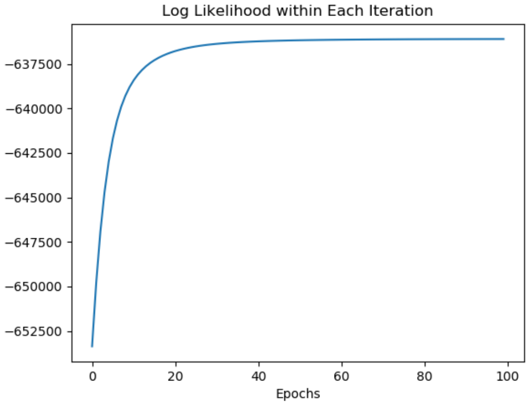

Figure 3: 100 epochs – 100 samples - 500 seconds - 5 states

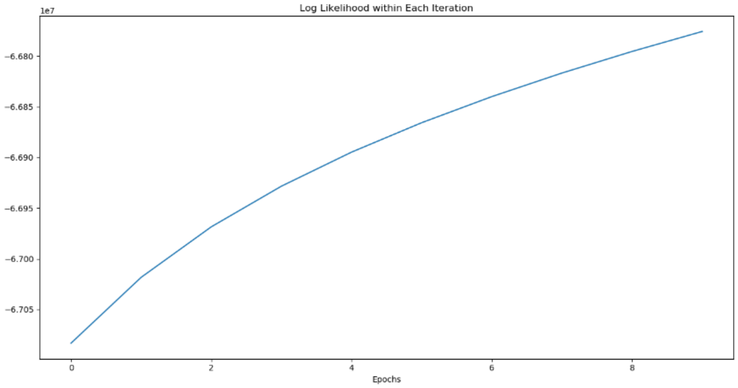

Figure 4: 10 epochs - 10k - 1.33 hrs - 5 states

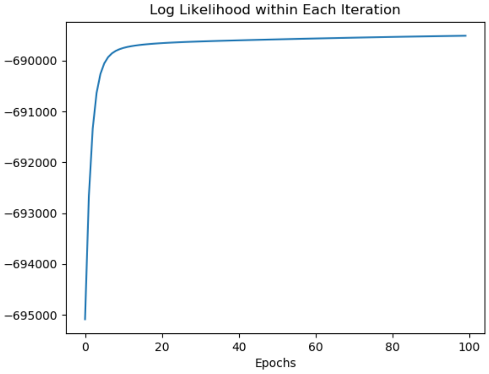

Figure 5: 100 epochs – 100 samples - 395 seconds - 3 states

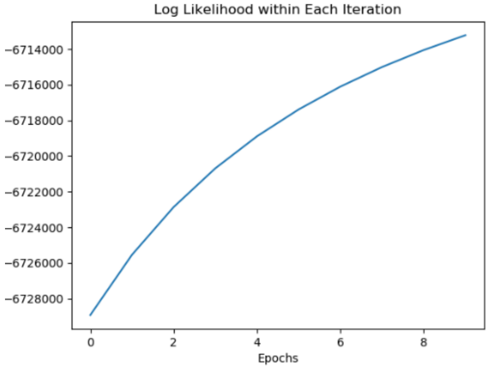

Figure 6: 10 epochs – 1000 samples - 407 seconds - 3 states

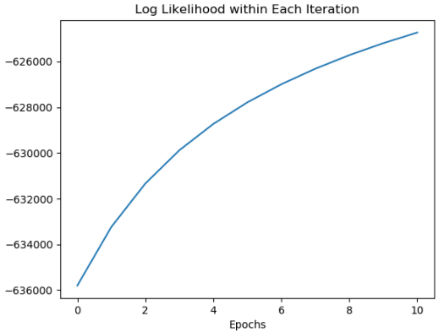

Figure 7: 12 epochs - 1000 samples - 983 seconds - 10 states

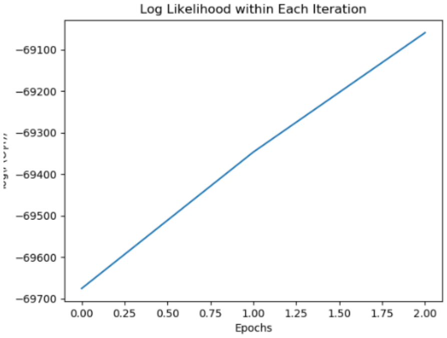

Figure 8: 2 epochs - 100 samples - 29 seconds - 10 states

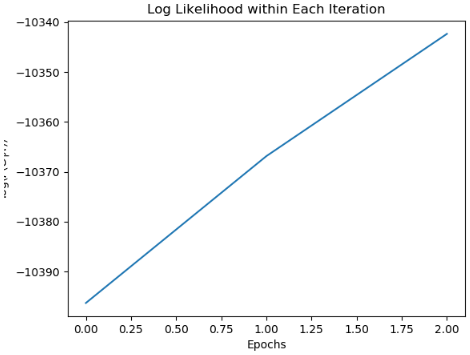

Figure 9: 4 epochs – 10 samples - 3 seconds - 10 states

Results and Discussion

One of the most notable results is that the increasing number of states consumed more memory, but also allowed for higher likelihood values. My model didn't end up being very efficient as it was very slow, and it's noticeable because of my sufficient memory. Convergence was only utilized for 10 states as those took much longer time than any. That is only due to me putting high epsilon values initially. The small samples with high epochs converge sharply as visible in 1, 3, 5. That is due to the high number of epochs that they went through. I also tested the full dataset as seen in 2 and illustrated how stiff the convergence looks due to the variant and abundant data samples.

Conclusion

To answer the questions in the experiments section:

For epsilon, many attempts were gathered that had an epsilon that was too large, which stopped the program way too soon. Also, there were ones that are too small that led to unnecessary running for the time and memory cost. The final ended up being 500. For the number of hidden states, 5 seemed the most reasonable as large amounts of data were impractical in my experimentation, sometimes even when they were 5. Having 3 states was somewhat fast, but unfair since not much memory was being used and not enough time for the model to train. 10 was testing boundaries after implementing convergence and it also took long relatively with the amount of data given. The results seemed more compute-bound as my device hasn't run out of memory, but the fastest it was doing seemed slow unfortunately. In the cases where there was convergence, I believe there were under-fitted results. On the other hand, those that did not stop after convergence were overfitted.

In conclusion, the main steps this algorithm took, were by the description in the original hmm.py file.