

**MODELLING SEQUENCES OF OBSERVABLE OUTCOMES OF
A COIN TOSS EXPERIMENT
USING THE DISCRETE HMM TOOLKIT**

Program #6

EE 8993: Speech Processing

Audrey Le

Professor: Dr. Joseph Picone

July 27, 1998



1. PROBLEM DESCRIPTION

In this project we are to use the discrete HMM toolkit to build a model that best characterizes the following sequences of outcomes from a coin toss experiment (Table 1). No apriori information about the nature of the coin (or multiple coins) used in the tossing experiment or how the coin is tossed is given, except the result of each coin flip. Using only the observable outcomes, we are to find the best set of parameters for the model. Once the best model is found, it is used to compute the probability of the sequences given in Table 2.

No. of Trials	Outcome of a Coin-Toss Experiment
1	HHTTTHHHHTTTTTTHHHHHHTTTTTTHHHHHHHHTTTTTTTTHHHHHHHHTTTTTTH
2	HHHHHTTTTTTHHHHTTTHHTHHHTHHHHHHHHHHHTHHHHHTHHHHHHHHHHHTHHHTHHHH
3	HHHHHHHHHTHHHHHHHHHTTHHHHHHTHHHTHHHHHTHHHHHHHTHHHHHHHTHHHHHTHTHHHH
4	HHHHHHHTTTHHHHHHHHHHHHHHHHTHHHHHTTTHHHHHHHHHHTTTHHHHHHHHHHHHTTH
5	HHHHHHHHHHHTTTHHTHHHHHHHHHHHTHHHHHTTTHHHHTHHHTHHHHHTHTHHHHHTHTHHHH
6	THTHHHHHHHHHTHTHHHHHHHTHTHHHHHTHTTTHHHHHHHHHHHHTHTHHHHHHHHHH
7	HTHHHHHHHHHHHHHHHTHHHHHHHHHHHHHTHHHHHHHHHTTTHHHHHHTHHHHHTHHHHHH
8	HHHHHTHHHTTTHHHHHHHHHHTHHHHHHHHHHHTTTHHHHTTTHHHHHHHHHHHHHHTTH

Table 1: Training data for the HMM.

Seq. No.	
1	HTHTHTHTHTHTHTHTHT
2	HHHHHHHHHHHHHHHHHHHH
3	TTTTTTTTTTTTTTTTTTTT

Table 2: Test data for the HMM.

2. INTRODUCTION

Often we are given a sequence of observable symbols and are asked to build a model to explain and characterize these observable symbols. The reason that we want to find such a model is that if found, the model can be used to identify or recognize other sequences of observations. A Hidden Markov Model, commonly known as HMM, handles situation as described previously quite successfully and efficiently.

So what is an HMM? An HMM is a doubly stochastic process where an underlying stochastic process is not known. An HMM is characterized by the following elements:

- it has a finite number of states.
- it has a set of transition probabilities among the states.

- it has a finite number of output symbols when a transition is made.
- it has an output probability distribution.
- it has an initial state distribution.

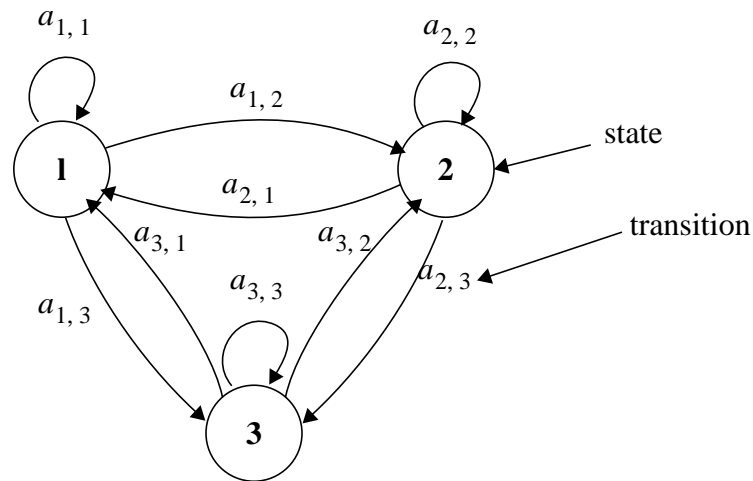


Figure 1: Elements of an HMM.

There are three basic problems when dealing with HMMs. The first problem is how to evaluate a model given a sequence of observations. We want to know how well the model performs on that sequence. From this we can choose the best model among the competing models. The second problem deals with how to uncover the hidden part of the model or the state sequence that is used to produce the output. The third problem is to decide on the size of the model. Often this last problem is the most complicated because many time we have little prior information about the observed symbols, and without of prior information choosing the optimal set of parameters is difficult and could result in trial and error before settling on the appropriate model.

3. EXPERIMENTS AND RESULTS

In this project we attempted to find the optimal model that best characterizes the set of sequences of output from a coin tossing experiment. In approaching this problem, we had no apriori information about how the sequences of observables were obtained. We resorted to trial by error. We varied three arguments: 1) the number of iterations was varied from 10 to 10,000 iterations in step of ten, 2) the number of states was varied from 2 to 100 states in step of one, and 3) the number of output symbols was varied from 2 to 60 in step of one. The limits on the three arguments were chosen at random. The last argument, the number of output symbols was chosen with the idea that it was possible that 60 coins were tossed at the same time. With these limits, we trained 6,000,000 models using the Viterbi algorithm and 6,000,000 models using the Baum-Welch algorithm.

Because the size of the training data was so small, most of the models built were statistically unreliable and were discarded. According to [2], we needed at least twenty data points to make a sound estimation of one parameter. Since we had 480 observed outputs and we needed twenty outputs for each parameter, our model could only have at most twenty-four parameters. For a fully connected HMM, the number of parameters is given by the equation below:

$$N = s^2(s + 1) \tag{1}$$

where N is the number of parameters and s is the number of states. So for our case, to get any sensible model, the maximum number of states we could have is three. However, just in case, we carried the experiment up to four states.

At least we knew the number of states needed to be around two to four so that limited the search space somewhat. We still had no clue about how many output symbols should we use. For the first experiment, we started out with two symbols. The results are given in Figure 2 and Figure 3 for the Viterbi and Baum-Welch algorithm respectively.

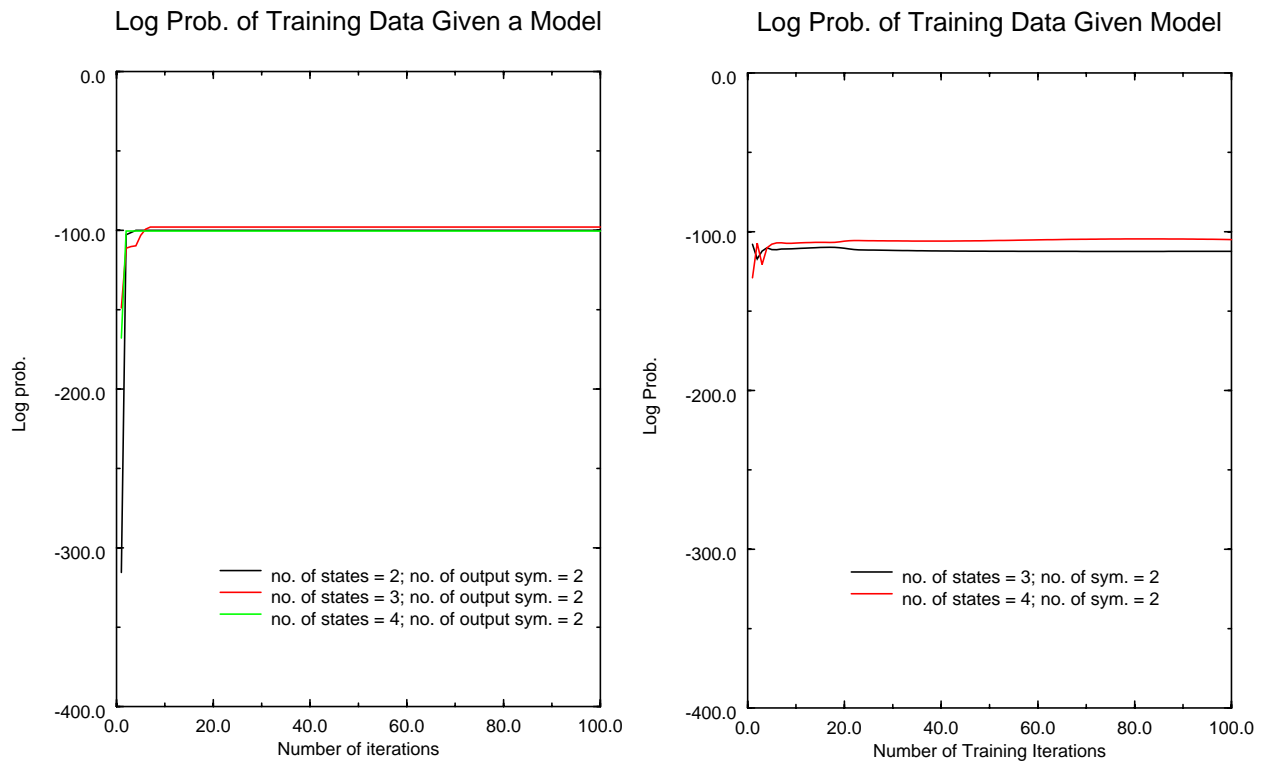


Figure 2: Log probability of various HMM models as a function of training iterations using two output symbols per state with Viterbi algorithm.

Figure 3: Log probability of various HMM models as a function of training iterations using two output symbols per state with Baum-Welch algorithm.

We noted that for the models trained with Viterbi the log probability is non-decreasing with each iteration. It either increases or stays constant but never decreases. Those trained with Baum-Welch behaved a bit differently. Although the log probability is generally non-decreasing as seen in those trained with Viterbi, it fluctuates once in a while. It is possible that numerical precision and error in estimating the underlying distributions causes this behavior.

From the five models built, the best model is a three state two output symbol model training with Viterbi with a log probability of -97.9579.

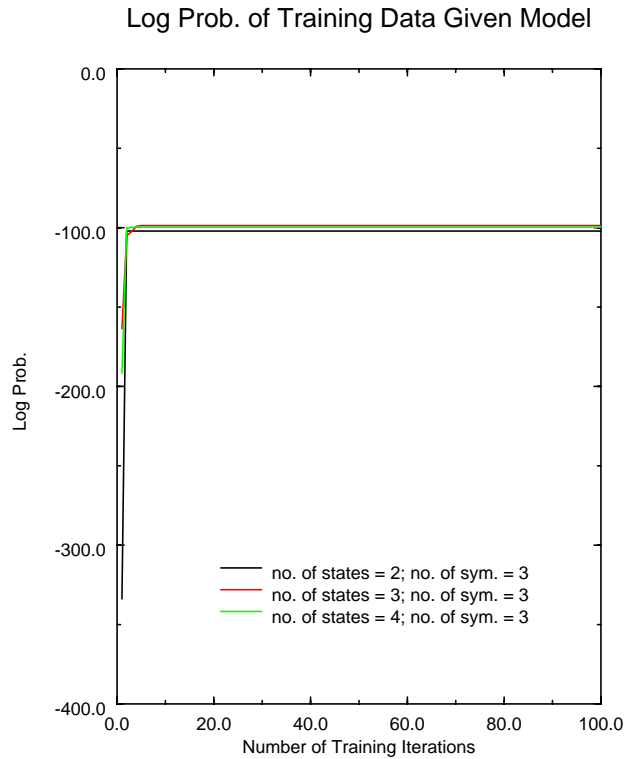


Figure 4: Log probability of various HMM models as a function of training iterations using three output symbols per state with Viterbi algorithm.

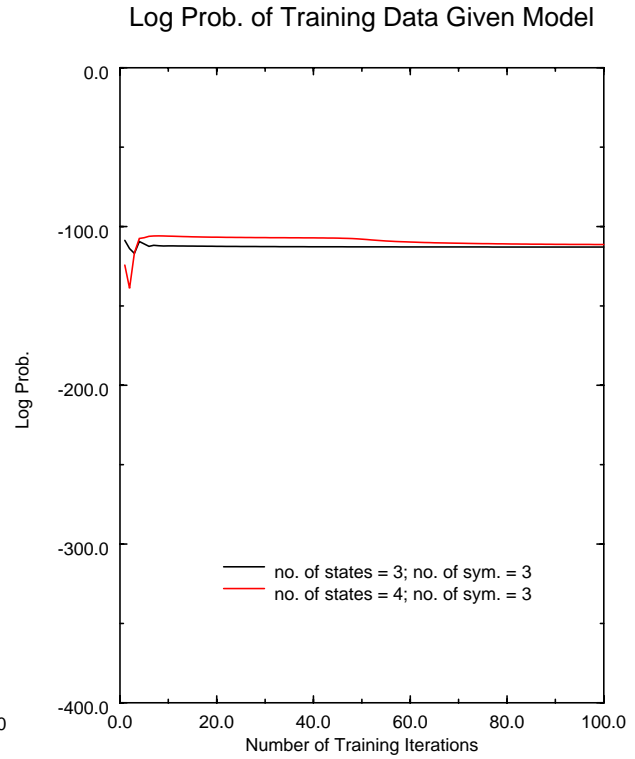


Figure 5: Log probability of various HMM models as a function of training iterations using three output symbols per state with Baum-Welch algorithm.

For the second experiment, we used the same parameters as described in the first experiment except that we increased the number of output symbols from two to three. The results are given in Figure 4 and Figure 5. We noticed the same the results of this experiment follow the trend of the first experiment. However, the log probabilities of these models are lower than those obtained in the first experiment. The best model, a three state three output symbol model with log probability of -98.646400, is still lower than the best model in the first experiment.

Since there is no increase in the probability as the number of output symbols increases, we stopped and picked the three state two output symbol model using Viterbi as the optimal model that characterizes our training data. We used this model and evaluated it on our train and test data, and the results are given in Table 3 and Table 4. Both tables show the model's performance with various testing modes on different kinds of data.

Model	Testing Mode	
	Viterbi	Baum-Welch
Viterbi	-97.9579	-124.7057

Table 3: Raw scores of our best model's performance on training data.

Model	Testing Modes	
	Viterbi	Baum-Welch
Viterbi	-20.1899	-28.7742

Table 4: Raw scores of our best model's performance on test data.

From the scores above, one would comment that model did very badly on the training data and very well on the test data. This is opposite to what is expected because usually the model does well on what it has been taught and not so well on things that it has not been taught. It is possible that the test data is too easy, and thus model performed very well. However, this is not the case for the widely differences in the scores. The reason is due to the size differences in the training and test sets. Thus, to make the scores comparable we normalized them with respect to the training data, and the normalized scores are given in Table 5 and Table 6.

Model	Testing Modes	
	Viterbi	Baum-Welch
Viterbi	-97.9579	-124.7057

Table 5: Normalized scores of our best model's performance on training data.

Model	Testing Modes	
	Viterbi	Baum-Welch
Viterbi	-161.5192	-230.1936

Table 6: Normalized scores of our best model's performance on test data.

From the normalized scores, we can see that the model performed very poorly on the test data. One reason could be that the size of our training data is not large enough to yield a statistically reliable model. Another reason is that our test data do not mirror the training data. Whatever the case may be, additional data and more experiments must be performed to find the reason for the model's poor performance with the test data.

4. CONCLUSIONS

We have presented here the results of our experiments of finding the best model to represent a sequence of observable outputs using the HMM toolkit. We have found that our best model is a three state two output symbol model. We have found that our best model performed poorly on the test data. We concluded that additional data and experiments are needed to determine the reason for the poor performance.

5. REFERENCES

- [1] J. Picone, "ECE 8993 Speech Processing Course Notes," Mississippi State Univ., Starkville, MS, 1998.

[2] L. Rabiner and B. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, January 1986.