INSTITUTE FOR SIGNAL AND INFORMATION PROCESSING

homework #6

Hidden Markov Model

EE 8993: Fundamentals of Speech Recognition

December 6, 1998

submitted to:

Dr. Joseph Picone

submitted by:

Suresh Balakrishnama

Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University
MS 39762, USA
Email: balakris@isip.msstate.edu

1. Introduction

A Hidden Markov Model is a stochastic finite state automation machine used to model a speech utterance which may be a word, a subword unit, a complete sentence or paragraph. It is characterized by the number of states, the number of distinct distribution, the state transition probability distribution, the output probability distribution, and the initial state distribution . But instead of relying on the previous observations to determine what the output probabilities is, which state the hidden Markov model will use for a given input is not immediately obvious from looking at the sequence, and thus the state is "hidden" in the model. Hidden Markov model is a well-studied concept of probability theory that deals with a class of random processes that incorporate a minimum amount of memory without actually being memoryless.

2. Problem Description

A sequence of coin toss is observed and the following is given as the data:

```
HHTTTHHHHTTTTTHHHHHHTTTTTTTHHHHHHHHTTTTTTTTTHHHHHHHHTTTTTTTH
HHHHHTTTTTHHHHTTTHHTHHTTHHHHHHHHHHTHHHHHTHHHHHHHHHHTHHHTHHHH
HHHHHHHHHTHHHHHHHHHTTHHHHHHTHHHTHHHHHTHHHHHHHHTHHHHHHTHTHHHH
HHHHHHHTTHHHHHHHHHHHHHHHTHHHHHTTHHHHHHHHHHTTHHHHHHHHHHHHHTTH
HHHHHHHHHHHTTHHHTHHHHHHHHHHHTHHHHHHTTHHHHHTHHHTHHHHHTHTHHHHH
THTHHHHHHHHHTHTHHHHHHTHTHHHHHTHTTHHHHHHHHHHHHHTHTHHHHHHHHHHH
HTHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHHTHHHHHHHHHHTTHHHHHHHTHHHHHH
HHHHHTHHHHTHHHHHHHHHHTHHHHHHHHHHHHTTHHHHTTHHHHHHHHHHHHHHTTHH
```

The objective of this experiment is to use the discrete Hidden Markov Model tool to find the best model for the training data. With this model, we are to compute the probabilities of observing the following sequences:

```
HTHTHTHTHTHTHTHTHTHT
HHHHHHHHHHHHHHHHHHHH
TTTTTTTTTTTTTTTTTTTT
```

Note that the HMM tool allows the user to specify the number of states to use in the model, the number of symbols to use in the model, and the number of passes on the training data to use. Further, the system allows user to select training and testing on Baum Welch or Viterbi algorithm. From the training data, we only observe two different symbols, namely H (head) and T (tail). However, we do not know how many coins were tossed at the same time to generate the training data. Therefore, to find the best training model, we will need to perform experiments on using different number of states, number of passes, and algorithm.

3. Theory of HMM

HMM is used to model words in small-vocabulary system and to model subword units like phones for large-vocabulary systems. A given HMM is more likely to produce observation strings that would be observed from real utterances of its associated word. In training phase, the HMM is taught the statistical make up of the observation strings for its dedicated word. In recognition phase, given an incoming observation string, it imagined that one of the existing HMMs produced the observation string. The word associated with the HMM of highest likelihood is declared to be the recognized work. Figure 1 illustrates a typical HMM with six states, the states are labeled by integers. HMM generates observation sequences by jumping from state to state and emitting an observation with each jump. There are two different forms of HMM - acoustic processing HMM and language processing HMM. Acoustic processing HMM emits an observation upon arrival at each successive rate and also one observation from the initial state. Language processing HMM emits an observation during the transition.

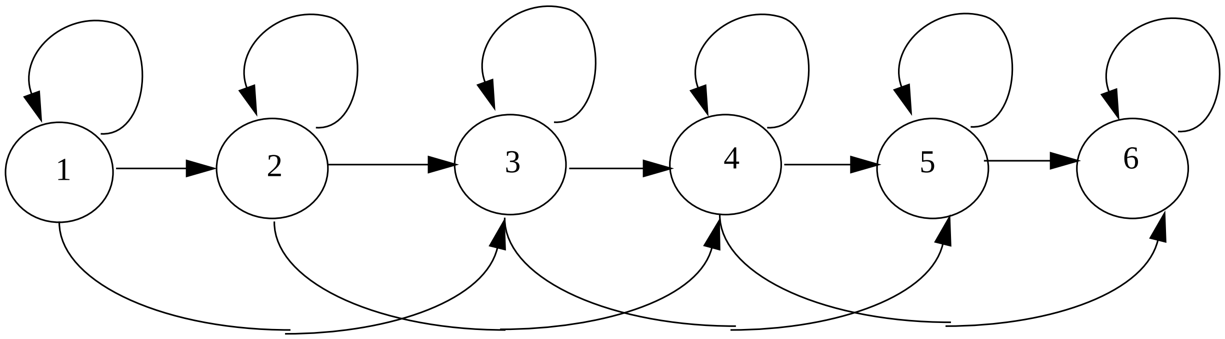

Figure 1. Hidden Markov Model with six states

4. Implementation and Results

The discrete HMM is restricted to the production of a finite set of discrete observations. Prior to training any of the HMMs for the individual observation, a set of (continuous) observation vectors form a large training data is used to derive the codebook. Any observation vector used for either training or recognition is quantized using this codebook. Our problem involves finding the best model for the testing data and the number of observation vectors used for training are two - Head (H) and Tail (T). Since each of the training sequence has 60 observations the number of states ranges from 2 to 64. The other parameter required for training is the number of passes and everytime the HMM tool is run the probability associated with each state are marginally adjusted so as to converge to a final value which produces the best model. The number of states has to be fixed to a value once the probability starts converging to a definite value. The training and testing for obtaining the best model is done using Baum Welch and Viterbi algorithm.

1. Given the training sequence, viterbi algorithm is used for training and testing to find the number of states that gives the best model for the training data. The best model is used for evaluation on the test data using viterbi algorithm. The results are illustrated in Table 1.

Rank	Number of states	Training probabilities	Testing probabilities
1	3	-97.957870	-20.189884
2	12	-98.083191	-22.700104
3	14	-98.386215	-19.659321
4	26	-98.636002	-19.685520

Table 1: Number of states, training and testing probabilities for the best training model using viterbi

2. Next step involves training the data using viterbi algorithm on 2 to 64 number of states and finding the best model for the testing data. The same models are used to evaluate the testing data with Baum Welch algorithm. The results are illustrated in Table 2.

Rank	Number of states	Training probabilities	Testing probabilities
1	2	-99.996658	-14.600151
2	33	-100.267494	-14.650858
3	42	-100.828857	-14.704206
4	61	-101.222977	-14.753456

Table 2: Number of states, training and testing probabilities for the best testing model using viterbi algorithm

3. Given the training sequence, the data is again trained using viterbi algorithm to find the number of states and the number of model. The best models are evaluated using Baum-Welch algorithm on the testing data. The results are illustrated in Table 3

Rank	Number of states	Training probabilities	Testing probabilities
1	2	-99.996658	-14.600151
2	33	-100.267494	-14.650858
3	42	-100.828857	-14.704206
4	61	-101.222977	-14.753456

Table 3: Number of states, training and testing probabilities for the best training model using viterbi algorithm of training and baum welch algorithm for testing

4. The data is trained using Viterbi algorithm on 2 to 64 number of states and the best models for the testing data using Baum Welch are found. Note that the model that is best for the testing data might not yield good results for the training data. The results are shown in Table 4.

Rank	Number of states	Training probabilities	Testing probabilities
1	55	-99.772392	-9.103511
2	61	-101.222977	-9.226512
3	42	-100.828857	-9.319737
4	58	-102.787437	-9.987990

Table 4: Number of states, training and testing probabilities for the best testing model using viterbi algorithm of training and baum welch algorithm for testing

5. Conclusions

The testing probabilities seem to be very low because the model obtained from the training data is not significant model. The training data is more biased either on “H” or “T”. The results show that the training probabilities and testing probabilities are not proportional.

6. References

[1] J.Deller, J.G.Proakis and J.Hansen, “Discrete-time processing of speech signals”, *Macmillan Publishing Company,* New York, USA.

[2] F. Jelinek, Statistical Methods for Speech Recognition, *The MIT Press*, Cambridge, Massachusetts, USA.