**HIDDEN MARKOV MODEL**

# A Term Project on

**Introduction to Stochastic Processes (IME 625A)**

## By

**ARIJIT GANGULY**

**(Roll Number: 17114005) DEEPAK SINGH ATTRI**

**(Roll Number:17114007) HIMANSHU RAHANGDALE**

**(Roll Number-17114010)**



**To The**

**DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING INDIAN INSTITUTE OF TECHNOLOGY, KANPUR**

**NOVEMBER 2017**

**INDEX**

|  |  |
| --- | --- |
| **Contents** | **Page No.** |
| 1. Introduction to Hidden Markov Model | 3 |
| 2. Formal Definition | 3 |
| 3. Central Issues In Hmm | 4 |
| 4. Evaluation Problem | 4 |
| 5. The Forward Algorithm | 5 |
| 6. Decoding Problem- The Viterbi Algorithm | 6 |
| 7. Learning Problem- The Forward Backward Algorithm | 6 |
| 8. Appendix | 9 |
| 9. References | 10 |

**INTRODUCTION TO HIDDEN MARKOV MODEL**

A Markov chain is useful for computing a probability for a sequence of events that we can observe in the world. But in many cases, the events we are interested in may not be directly observable in the world. A Hidden Markov Model (HMM) talks about both observed events and hidden events that we think of as causal factors in our probabilistic model. ( Daniel Jurafsky & James H. Martin ( 2013 ))

**Hidden Markov Model** (**HMM**) is a [statistical](https://en.wikipedia.org/wiki/Statistical_model) [Markov model](https://en.wikipedia.org/wiki/Markov_model) in which the system being modeled is assumed to be a Markov process with unobserved (*i.e. hidden*) states. Hidden Markov models are especially known for their application in [reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning) and [temporal](https://en.wikipedia.org/wiki/Time) [pattern](https://en.wikipedia.org/wiki/Pattern_recognition) [recognition](https://en.wikipedia.org/wiki/Pattern_recognition) such as [speech,](https://en.wikipedia.org/wiki/Speech_recognition) [handwriting,](https://en.wikipedia.org/wiki/Handwriting_recognition) [gesture recognition,](https://en.wikipedia.org/wiki/Gesture_recognition) [part-of-speech tagging,](https://en.wikipedia.org/wiki/Part-of-speech_tagging) musical score following, [partial discharges](https://en.wikipedia.org/wiki/Partial_discharge) and [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics).( Hidden Markov model (2018))

## FORMAL DEFINITION-

A HMM is specified by the following components: W = 1, 2 … N a set of N hidden states.

V = v1; v2; …… vT a sequence of T observations (observed states).

A = {aij : i to j } **transition probability** matrix A is the probability of moving from state i to

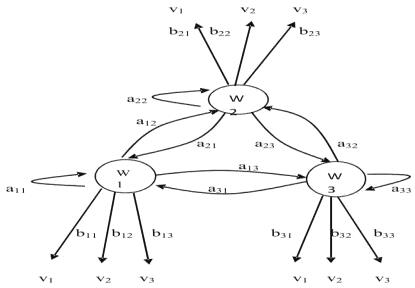
state j, such that for all i.

j=1

B = {bjk} a sequence of observation likelihoods, also called **emission probabilities**, each expressing the probability of an observation vk being generated from a state j such that

for all J.

Absorbing state- o is the final state where the machine ends up and it emits only 1 signal vo from this state.



## CENTRAL ISSUES IN HMM

Now, given such a hidden Markov model you find that in hidden Markov models, there are three central issues which need to be addressed, so what are the central issues.

* Evaluation Problem or Likelihood Problem
* Decoding Problem
* Learning Problem

## EVALUATION PROBLEM

Our first problem is to compute the likelihood of a particular observation sequence.

Computing Likelihood: Given an **HMM = (W, V, aij, bjk)** and an observation sequence VT, determine the likelihood P(VT |  ) or what is the probability that this sequence VT is generated by .

This can be done by a Crude approach and a better Recursive algorithm.But first we will discuss the crude approach.

In the Crude Approach we find the required probability using the following formula-

Where is one of the possible sequence of hidden states and it is given by-

𝑟

and t=1,2….T are the discrete time steps.

Now if there are N number of hidden states , then maximum number of possible sequences will be of the order of NT. In order to calculate the probability of a particular sequence, we use

Given that we know a particular sequence has occurred, the probability of getting the observed

pattern is found by-

So the final formula boils down to

But the order of complexity of this process is NT\*T which is huge. This is why we move onto the Forward Algorithm.

## THE FORWARD ALGORITHM

In the forward algorithm we must know the value of αj(t) which is the probability of being in state j after seeing the first t observations, given the HMM . The probability distribution of αj(t) is given by-

αj(t) = 0 t = 0 & j ≠ initial state

= 1 t = 0 & j = initial state

= [ ]otherwise

i(t-1) is the previous forward path probability from the previous time step aij the transition probability from previous state i to current state j

the state observation likelihood of the observation symbol v(t) given the current state j. The recursive algorithm is-

1. Initialization: At time step t=0, initialize the values of aij and .We also know the

sequence VT and the initial state αj(0).

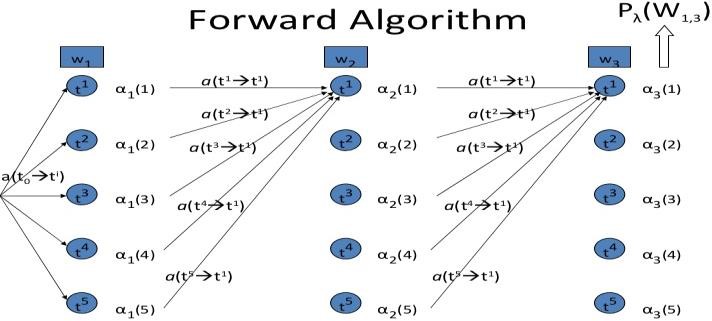
1. Increment: For tt+1

αj(t) =

𝑖=1

until t=T

1. Return which is nothing but αO(T)
2. End



**Trellis Diagram**

## LEARNING PROBLEM- THE FORWARD BACKWARD ALGORITHM

We turn to the third problem for HMMs: learning the parameters of an HMM, that is, the A={aij} and B={bjk} matrices. Formally-

Learning: Given an large number of observation sequences VT and the set of possible states in the HMM, we need to learn the HMM parameters A and B. The input to such a learning algorithm would be an unlabeled sequence of observations V and a vocabulary of potential hidden states W.

The standard algorithm for HMM training is the forward-backward, or Baum Welch algorithm. The algorithm will let us train both the transition probabilities A and the emission probabilities B of the HMM. But before that we need to know about **backward probability**.

i(t) = The probability that the model will be in state i(t) and will generate remaining of the given target sequence VT i.e from v(t+1) to v(T).

The probability distribution of (t) is given by-

(t) = 0 ωi(t) ≠ ω0 and t=T

= 1 ωi(t) = ω0 and t=T

= otherwise

otherwise

Now in order to compute we perform the following algorithm.

* 1. Initialize i(t) aio ; 1  i  N
  2. Recursion : For tt-1

Until t=1

* 1. Return which is nothing but .
  2. End

## Forward Backward Algorithm

Having computed the forward and backward probabilities we need to use them to build an algorthim to estimate the transition probability and emission probability matrix A and B.

1. For this we define

In this expression the numerator term gives us the probability of transition from i(t-1) to

j(t) for a particular sequence VT. While the denominator term gives the probability of getting the sequence VT in all possible ways.

1. Initially we should choose the values of aij and bjk arbitrarily.
2. Expected number of transitions from i(t-1) to j(t) at anytime in the sequence VT is

given by

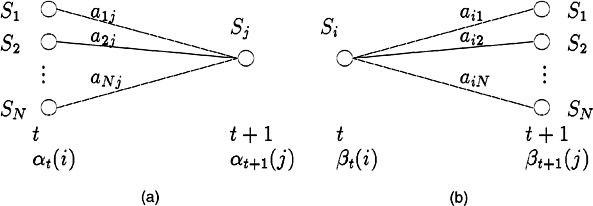
1. Total expected number of transitions from i is given by
2. The estimated value of aij is â ij = (expected number of transitions from state i to state j) divided by the ( expected number of transitions from state i)
3. The estimated value of bjk is b̂ jk= (expected number of times in state j and observing symbol vk ) divided by (expected number of times in state j).

only for those states where v(t)=vk

1. Now we need to use the estimated values â ij and b̂ jk to calculate αi(t − 1) , βj(t − 1)

and (t − 1).

1. Again we will use these values to re-estimate the parameters â ij and bjk. This process should be repeated until convergence or atleast when the values are below a specified threshold.



Although in principle the forward-backward algorithm can do completely unsupervised learning of the A and B parameters, in practice the initial conditions are very important. For this reason the algorithm is often given extra information. For example, for speech recognition, in practice the HMM structure is often set by hand, and only the emission (B) and (non-zero) A transition probabilities are trained from a set of observation sequence.

## APPENDIX

**Forecasting future states using the trained HMM**

After we train our Hidden Markov Model using the dataset of observed sequences, We

will get the transition and emission probabilities A and B. For the next future state T+1, we will calculate the forward probabilities αj(T+1) of every hidden states in the next step T+1.Also we will calculate the emission probability of each observed pattern from each of the hidden states in time step T+1.Then we need to formulate the maximum likelihood function that will give the likelihood value for the future state.

## REFERENCES

Daniel Jurafsky & James H. Martin **(** 2013 **) “**Speech and Language Processing **-**An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition” (2nd edition , page no :122-141) Pearson

Hidden Markov model (2018).

Retrieved from https://en.wikipedia.org/wiki/Hidden\_Markov\_model

Nptel Course- Pattern Recognition and applications – Hidden Markov Model . Retrieved from<http://www.nptel.ac.in/courses/117105101/38>