Hidden Markov Models Using Hierarchical Dirichlet Process

Nishant Rawat

Department of Computer Science

Arizona State University

Tempe, AZ

nrawat@asu.edu

*Abstract*—Hidden Markov Models is a statistical model which is one of the more popular model used for sequential and temporal data. It can estimate the parameters and can efficiently do the inference while handling real world applications. Hierarchical Dirichlet Process is a Bayesian approach to the modeling of grouped data. Each group is related to a mixture model analogous with Dirichlet process mixture models and defines a nonparametric prior. We are trying to study Hidden Markov Models using prior information that is generated by a Hierarchical Dirichlet Process.

*Index Terms*—HDP, HMM, Forward-Backward Algorithm.

# Introduction

This project involved the exploration of Hidden Markov Models using prior information. The prior information was generated by a Hierarchical Dirichlet Process (HDP). We generated 10 different HMMs using two very disparate probability vectors. We tried to learn how different or similar random HMMs created from a single probability vector could behave. Hidden Markov models are used to explore sequential or temporal data. It works well because it is simple enough to not just predict parameters but also good enough to work in real world problems. In this introduction I’ll try to give a little insight into these two

## Hidden Markov Models

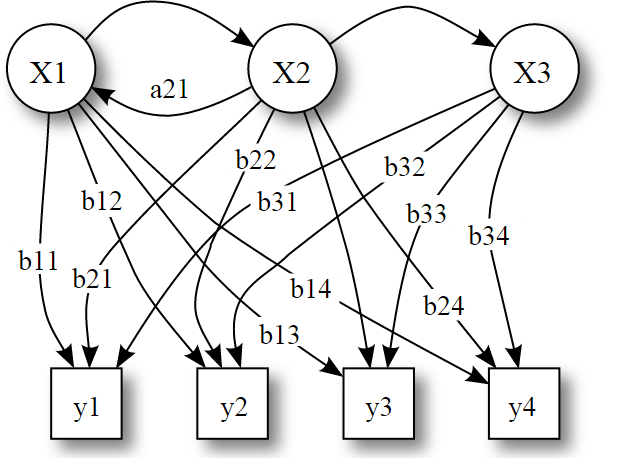
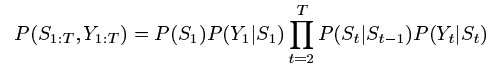
A hidden markov model is used to represent probability distribution over observation sequences. It has two properties. 

Figure 1. Probabilistic parameters of a hidden markov model

Firstly, it assumes that observations have some state generating it. These states are hidden from the observer and hence hidden markov model. It also believes that this hidden state for instance St, if given the value of state St-1 then the current state St is independent of all the states prior to t-1. It also believes that the observations at a certain time interval are independent of all the other observations and all the other states except its hidden state at that time interval. Considering the observation state to be represented by Yt and taking all these properties together, the joint distribution of a sequence of states and observations can be factored in following way:



For more information, check [2]

## Hierarchical Dirichlet Process

It was proposed in [1]. This is a nonparametric Bayesian clustering approach that clusters group data. The basic concept behind is that HDP is a nonparametric prior which allows mixture models to share components [1]. It is a distribution over a set of random probability measures. Dirichlet process is used for every group of data where each group has the same basic distribution. Thus the different groups produced share statistical strength.

# Approach And Environment

The choice for the software was easy. MATLAB stands out due to its ability to provide fast prototyping. It is often slower at execution but due to its huge library of functions, anything needed for our HMM programming could be found in MATLAB. The data that we were going to provide, i.e. 2D matrices, is also by default understood and can be quickly inputted rather than making functions to process it. It has a huge documentation which is very helpful and a very concise code but the most important factor here was the plotting tools that MATLAB provided.

We started by assuming two very disparate probability vectors. Both of them had K elements constrained to be non-negative with sum equal to one. We followed these constraints to create two mean probability vectors.

## Preparing the Environment

To prepare the environment, we installed MATLAB by using the installation steps mentioned in MATLAB documentation [here](http://www.mathworks.com/help/install/ug/install-and-activate-without-an-internet-connection.html) [5]. We also need to install an additional package called lightspeed [4]. We also installed some support packages which are needed to run lightspeed and MATLAB. These were some prerequisites that are needed for the efficient and seamless run of our study. The way to install these prerequisites is to install Microsoft Visual Studio 2010 since it has all these packages and helps skipping the tedious process of installing each and every needed package. This prepares our system for the study.

## Generate models and Observation sequences

We generated a set of 10 Hidden Markov models for each mean probability vector. We produce a set of transition probability and initial probability matrices which represent each HMM. This is used to generate a total of 20 HMMs, i.e. 20 pairs of transitions and initial probability matrices.

After the HMMs are generated, we create a state sequence for each of them depending on the initial probability vector. The initial probability value of each state represents the probability of the state sequences beginning with that particular state. The first elements of the 30 state sequences are determined based on initial probability vector. The rest of the elements in the state sequences are generated randomly. 30 state sequences with 100 states each are generated for all each of the HMMs. Assuming that the observation model is simple normal density where state k =1, 2, 3, 4 and 5, the observation is a random drawn from a normal density distribution N (k, 0.09) i.e. with mean k and variance 0.09. We generate the observations sequences from their corresponding state sequences.

## Learning Algorithm and Infer Mean HMMs

As already mentioned before, the forward-backward algorithm is used for inferences in HMMs and is one kind of many types of learning algorithm. We took 30 observation sequences in each independent HMM. These observation sequences are input to the forward backward learning algorithms and the output are HMM transition probability matrices and initial probabilities. We repeat the process for each of the 20 HMMs generated.

There are 30 observation sequences in each HMM, and 10 HMMs will be generated for a mean probability vector. We compute a mean HMM to represent all the 10 HMMs of a particular probability vector. In the end we get one mean HMM for both the probability vectors.

## Comparison

HMM models are generated by using the learning algorithms for both the mean probability vectors. These models once generated are compared with their respective mean HMMs. We computed the ratio of observation sequence probability w.r.t. to their independent HMM to the observation sequence probability w.r.t. to its mean HMM. Let us call this comparison value z.

We also compared mean HMMs. The comparison of the observation from first mean probability HMMs to mean HMM of second mean probability HMMs is denoted as R12. Similarly, the comparison of observations from second mean probability HMMs with mean HMM of first mean probability is denoted as R21.

# Forward-Backward Algorithm

Forward backward algorithm is an example of dynamic programming. Dynamic programming allows to compute some extremely non trivial things. We use forward backward algorithm to find inference in Hidden Markov Model. It is an inference algorithm which takes in sequence of observation and emission sequences and computes posterior marginal of hidden state variables. The forward backward algorithm has very important applications to both Hidden Markov Models and conditional random fields.

Following are the details of the forward backward algorithm:

1. Initializing Ai and Bj
   1. First define ξt(i,j) as the probability of being in state I at time t and state i+1 in time t+1, given the model and observation sequence.
   2. Compute ϒt(i)

ϒt(i)=



* 1. Compute initial probability πi, aij and bjk matrices





1. Using these initial values of aij and bj(k) recursively perform the following steps till aij (k)-aij (k-1) and bj(k)-bj(k-1) is less than a threshold
   1. Consider the forward variable αt(i). This is defined as

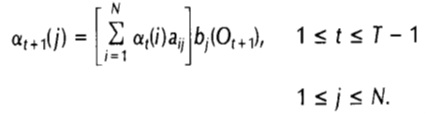
αt (i)= P(O1 O2 O3… ,qt =Si |λ)

Compute αi(k) as follows:

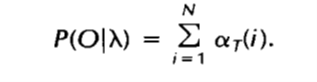
* + 1. Initialization:

αt (i)=πi bi(O1) where 1<=i<=N

* + 1. Induction:



* + 1. Termination



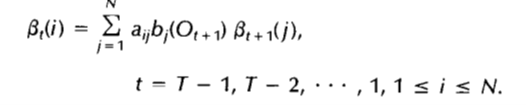
* 1. βt(i) is defined as:
     1. βt(i)=P(Ot+1 Ot+2 …. OT |qt =Si, λ)

This is the probability of the partial sequence from t+1 to the end, given state Si at time t and the model λ

* + 1. Compute βt(i) inductively as follows:
       1. Initialization:

βt(i) = 1 where 1 ≤ i ≤ N

* + - 1. Induction:



* 1. ϒij(t) is the probability of transition between ωi(t-1) to ωj(t-1). This can be defined as follows:

ϒij (t)= (αi(t-1)aijbjk βj(t))/(P(O|λ))

* 1. Now that ϒij(t) is known, we can compute ȃij and ƀik

ȃij =

ƀik=

1. We now substitute aij and bjk for the estimates and re run the step 2 of algorithm.
2. For part 4, the transition and initial probability matrix is estimated using the following:

Using the ξt(i,j) computed in Step 1.a, the transition probability matrix A and initial probability matrix πi is computed using 1.c and the mean matrix is found over the 10 HMM’s

# Results

We compared each HMM with its respective mean HMM. This leads to 20 plots, one for each HMM. Following figure 2 shows our results for it.

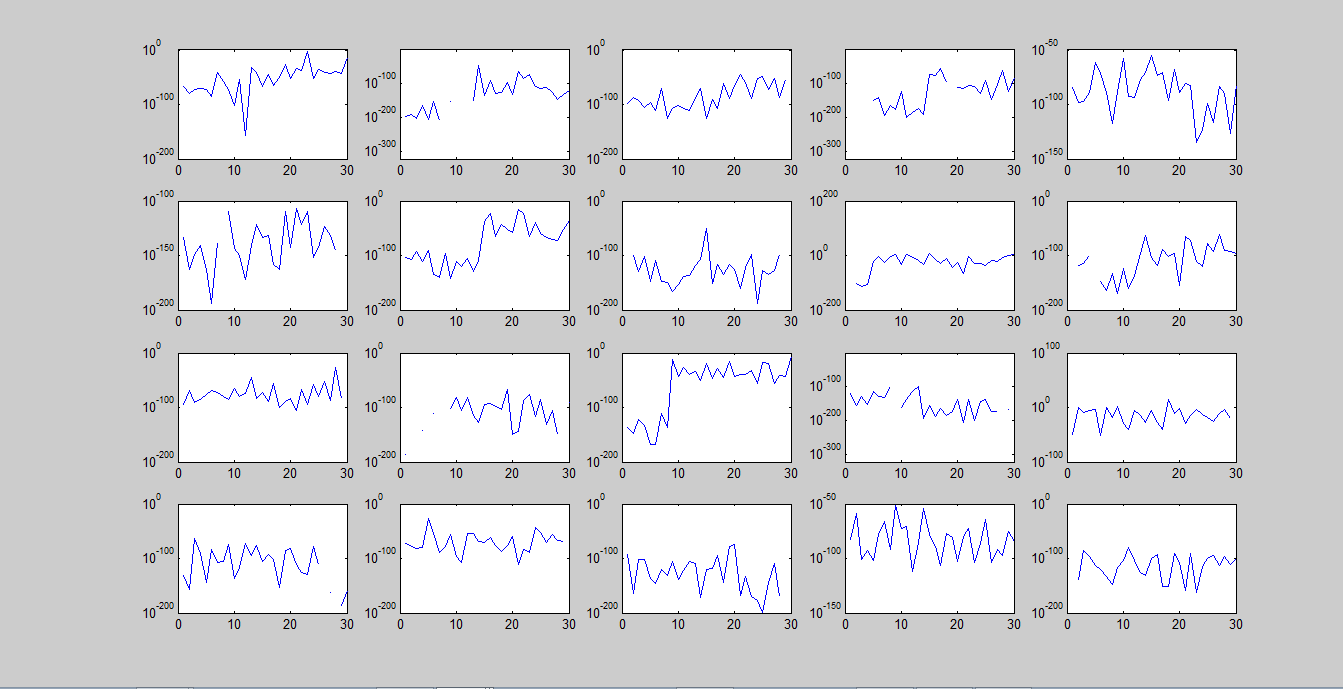


Figure 2. Independent HMM w.r.t. Mean HMM

Figure 3 shows the comparison of mean HMM of the first probability vector with respect to the mean HMM of the second probability vector.

Figure 4 shows the comparison of mean HMM of the second probability vector with respect to the mean HMM of the first probability vector.

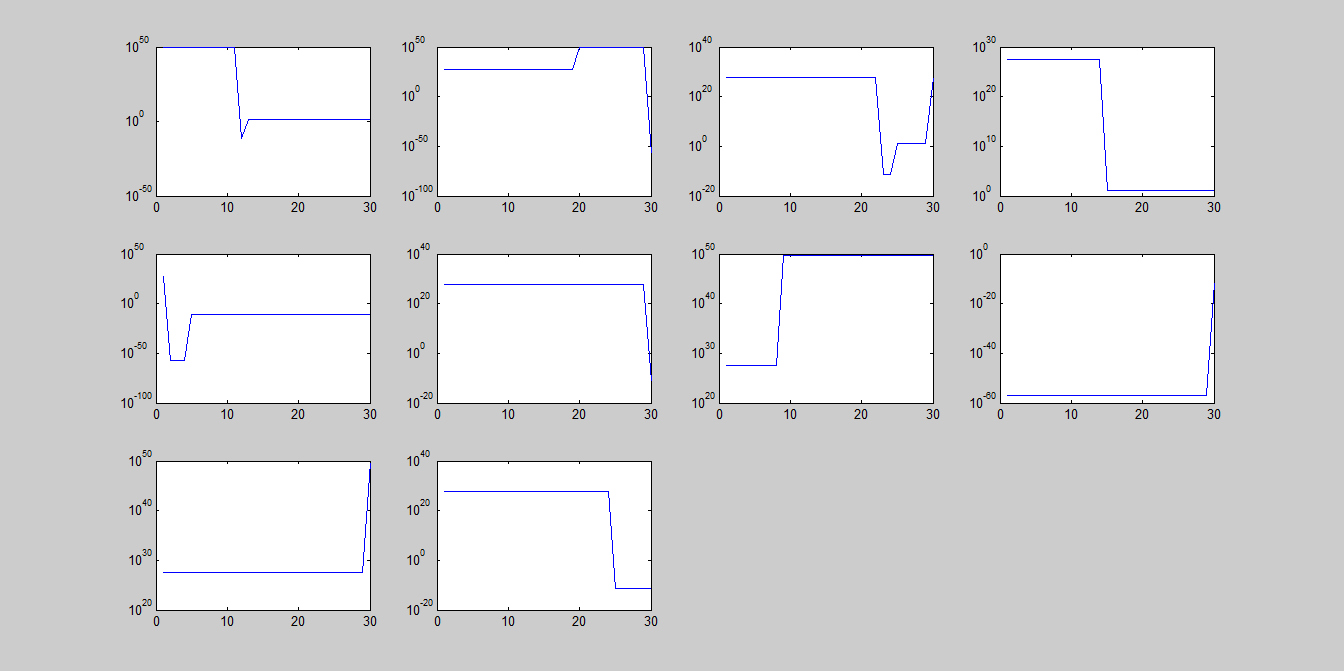


Figure 3. HMM of mean 1 w.r.t. HMM of mean 2

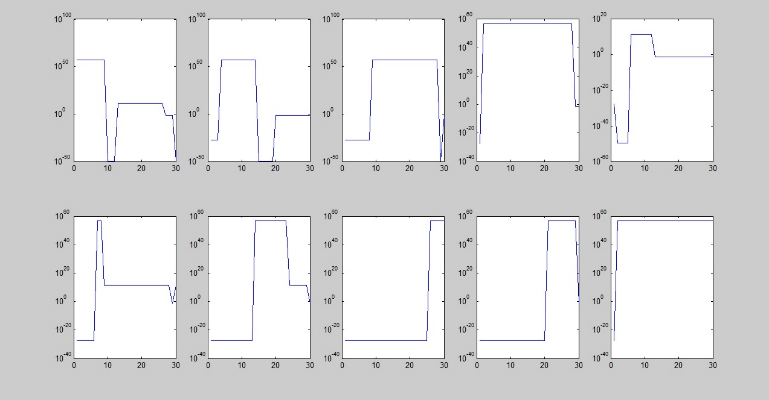


Figure 4. HMM of mean 2 w.r.t. HMM of mean 1

# Acknowledgment

I would like to thank Prof. Baoxin Li for his immense help in making this project a reality. I would also like to thank my team that comprised of Aravind Jayachandran and Asha Sasikumar. I would also like to thank Jeffrey W. Miller, whose channel mathematicalmonk was of immense help to us in understanding the hidden markov model. Thank you all.

# References

1. Y. W. Tee, M. I. Jordan, D. M. Blei and M. J. Beal, “Hierarchical Dirichlet Processes,” Nov 15, 2005
2. Lawrence R. Rabiner, “A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”
3. MathematicalMonk Channel on youtube, <https://www.youtube.com/watch?v=TPRoLreU9lA>
4. Lightspeed. <http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/>
5. MATLAB Installation Instruction http://www.mathworks.com/help/install/ug/install-and-activate-without-an-internet-connection.html
6. Z. Ghahramani, “An Introduction to Hidden Markov Models and Bayesian Networks”
7. Forward backward algorithm, Michael Collins, <http://www.cs.columbia.edu/~mcollins/fb.pdf>