Homework 3: Hidden Markov Models

Luis A. Valbuena Reyes

***Abstract*— In this document, we present the solution for homework 3. Trying to follow the guidelines for writing papers, we develop each the requirements of the assignment on Sections II, III, and IV.**

I. INTRODUCTION

In this document, we extend the Expectation Maximization approach to consider time, into what is known as hidden Markov models (HMM). Speech recognition is one of some applications in which a group sounds are collected and the goal is to find out the words that were spoken. The sounds are perceived and the words related to those sounds are the hidden variables[1] and because the structure of language, we know of relationships among words: there are some words that come after others and in contrast, there are words that don't have any relation among them. This embeds the Markovian property into the problem, that is, given a sequence of transitions among states (words) $Z_1, Z_2, \ldots, Z_n, Z_{n+1}, \ldots$, we have that

$$\mathbb{P}(Z_{n+1}|Z_n, \ldots, Z_2, Z_1) = \mathbb{P}(Z_{n+1}|Z_n),$$

which means that the next state Z_{n+1} on the sequence only depend on the current state Z_n and it is not required to keep track of all the states prior to Z_n. The transitions occur at equally spaced periods of time so we are studying a discrete Markov chain.

In our case, we are studying a signal that is corrupted with additive noise and we plan to model it with a HMM. We characterize a HMM with the following parameters (taken from [1]):

- M The number of states. The states are hidden but we can assume a particular number of them.
- $\mathbf{P}$ Transition probability matrix. The position P_{ij} relates the probabilty of a transition from state $Z(i)_n$ to state $Z(j)_{n+1}$. The sumation of the element in each of the rows of P are equal to 1.
- π The initial distribution.

We will denote the states as $Z(i)_n$ for $i \in \{1, 2, \ldots, M\}$. The observation measured is X_n where the subscript indicates the time n. Even though the states $Z(i)_n$ have the Markovian property, the observations are independent among them, that is

$$\mathbb{P}(X_{n+1}|Z(j)_{n+1}Z(i)_n) = \mathbb{P}(X_{n+1}|Z(j)_n),$$

that is, the observation X_{n+1} only depends of state $Z(*)_{n+1}$.

Luis Valbuena is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, 87131-0001, {lavalbuenar@unm.edu}

[1]The words are hidden inside the sounds

Although we are assuming a particular number of states M, we need to now the parameters μ_i and Σ_i for the state $Z(i)$ as well as matrix $\mathbf{P}$ and the distribution π. In the next section, we present how to find these values.

II. THEORY

Based on the material found in [1] and [2]. Because of the iterative nature of the procedure behind HMM's, we can conveniently translate the problem posed at the end of Section I into the following three problems

- Given some sequence of observed values $\mathbf{X}_i = X_1, X_2, \ldots, X_n, X_{n+1}, ...X_T$ and some preliminary values for π, $\mathbf{P}$, μ_i and Σ_i for $Z(i)_n$. What is the probability of our HMM model to generate that sequence $\mathbf{X}_i$? See Sections II-A, and II-B, and II-C.
- Given some sequence of observed values $\mathbf{X}_i = X_1, X_2, \ldots, X_n, X_{n+1}, ...X_T$ and some preliminary values for π, $\mathbf{P}$, μ_i and Σ_i for $Z(i)_n$. How do we choose a sequence of states $\mathbf{Z}_i = Z(*)_1, Z(*)_2, \ldots, Z(*)_n, Z(*)_{n+1}, ...$ that has the best measure of reproducing $\mathbf{X}_i$? See Sections II-C and II-E.
- How can we modify π, $\mathbf{P}$, μ_i and Σ_i for $Z(i)_n$ so we maximize the values calculated in the two previous questions? See Section II-D.

In the following Sections we explain the idea behind each Step as well as the mathematical expressions used Section III

A. Forwards Step

The calculation $\mathbb{P}(\mathbf{X}_i)$ given some previous model parameters would be

$$\mathbb{P}(\mathbf{X}_i) = \sum_{\text{all } \mathbf{Z}} \mathbb{P}(\mathbf{X}_i|\mathbf{Z})\mathbb{P}(\mathbf{Z}) \tag{1}$$

where all the possible sequences $\mathbf{Z}$ are the same length as $\mathbf{X}_i$. However, the number of operations involved is overwhelming. An optimal calculation consists of looking at a local perspective by considering a particular state $Z(i)_n$ at time n and looking forward and backwards in time for recursion expressions. When looking forward in time, we want to calculate the probability of all the possible states in time $n-1$ that can visit $Z(i)_n$ on the next transition. We regard the forwards step recursively as

$$\boldsymbol{\alpha}_n \propto \boldsymbol{\psi}_n(\mathbf{P}^T \odot \boldsymbol{\alpha}_{n-1}),$$

where $\psi(i)_n = \mathcal{N}(X_n|\mu_i, \Sigma_i)$ and $\odot$ is the Hadamard product. Each $\boldsymbol{\alpha}_n$ needs to be normalized and $\boldsymbol{\alpha}(i)_1 = \frac{1}{M}$.

B. Backwards Step

Following the previous discussion, we also need to go backwards in time and we do that by calculating the probability of all the possible states in time $n+1$ that can be visited from $Z(i)_n$ on the next transition. That is the backwards term and is expressed as

$$\boldsymbol{\beta}_n = \mathbf{P}(\boldsymbol{\psi}_{n+1} \odot \boldsymbol{\beta}_{n+1}).$$

Note that $\boldsymbol{\beta}_n$ need not be normalized. However, the initial condition is $\boldsymbol{\beta}(i)_T = 1$.

C. Forwards-Backwards Step

The optimal equivalent to Equation 1 is given by the Forwards-Backwards step as

$$\boldsymbol{\gamma}_n \propto (\boldsymbol{\alpha}_n \odot \boldsymbol{\beta}_n),$$

where $\boldsymbol{\gamma}_n$ in normalized.

Evolution of parameters $\boldsymbol{\alpha}_n$, $\boldsymbol{\beta}_n$, and $\boldsymbol{\gamma}_n$ can be seen on Fig. 2, Fig. 3, and Fig. 4.

D. The Baum-Welch Algorithm

The Baum-Welch Algorithm is the Expectation Maximization for HMM. First, we need to compute matrix $\boldsymbol{\xi}_{n,n+1}$ as

$$\boldsymbol{\xi}_{n,n+1} \propto P \odot (\boldsymbol{\alpha}_n(\boldsymbol{\psi}_{n+1} \odot \boldsymbol{\beta}_{n+1})^T).$$

Then, the Expectation step involves the following calculations

$$\begin{aligned}
\mathbb{E}[N_k^1] &= \sum_{i=1}^{M} \gamma_{i,1}(k), \\
\mathbb{E}[N_j] &= \sum_{i=1}^{M} \sum_{n=1}^{T} \gamma_{i,n}(j), \\
\mathbb{E}[N_{ik}] &= \sum_{i=1}^{M} \sum_{n=1}^{T} \xi_{i,n}(j,k),
\end{aligned}$$

where the notation changes slightly: the subindex i refers to a realization $\mathbf{X}_i$ and the original notation $\xi_{n,n+1}$ changes to $\xi_{i,n}$ to make reference to $\mathbf{X}_i$.

In the Maximization step we compute

$$\begin{aligned}
P_{j,k} &= \frac{\mathbb{E}[N_{ik}]}{\sum_{k'} \mathbb{E}[N_{ik'}]}, \\
\boldsymbol{\pi} &= \frac{\mathbb{E}[N_k^1]}{N}, \\
\mu_k &= \frac{\sum_{i=1}^{M} \sum_{n=1}^{T} \gamma_{i,n}(k) X_{i,n}}{\mathbb{E}[N_k]}, \\
\Sigma_k &= \frac{\sum_{i=1}^{M} \sum_{n=1}^{T} \gamma_{i,n}(k) X_{i,n} X_{i,n}^T - \mu_k \mu_k^T}{\mathbb{E}[N_k]}
\end{aligned}$$

where N is the number of realizations.

It is suggested in [2] for the initialization stage to neglect time and execute an Expectation Maximization with Gaussian Mixture Models or K-means on the observed data and then use these values to start the Baum-Welch algorithm.

E. The Viterbi Algorithm

The Viterbi Algorithm is a recursive expression intended to establish a sequence on states $\mathbf{Z}_i$ given a sequence of observations $\mathbf{X}_i$. The recursive expression is given by

$$\delta_n(j) = \max_i \{\delta_{n-1}(i)\mathbf{P}_{ij}\psi(i)_n\},$$

with the initial condition

$$\delta_1(j) = \pi_j \psi_1.$$

Another criteria to determine the best $\mathbf{Z}_i$ is using γ, i.e., $Z_n^* = \max_i \{\gamma(i)_n\}$.

III. Experiments

We can think of the artificially generated data as an arriving signal which contains noise. The signal has three states i.e., $\mu_1 = 1$, $\mu_2 = 2$, and $\mu_3 = 3$ and the standard deviation of the noise is $\sigma = 0.3$. The observed variable is the signal with the noise. The data is generated using the script given in the assignment. The transition probability matrix of the Markov chain is:

$$P = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.3 & 0.1 & 0.6 \end{bmatrix}.$$

A preliminary estimation on matrix P is done by counting the transitions on the generated signal, the result is given by

$$P_{est1} = \begin{bmatrix} 0.7895 & 0.1066 & 0.1039 \\ 0.2085 & 0.4583 & 0.3331 \\ 0.3119 & 0.1047 & 0.5834 \end{bmatrix}.$$

A realization of the observed variable and the state is presented on Fig. 1(a) and Fig. 1(b).

In the execution of our experiments we conducted 80 realizations. For the realization of Fig. 1(a) and Fig. 1(b), the evolution of the forwards term are depicted in Fig. 2(a), Fig. 2(b), and Fig. 2(c); while the backwards term is presented in Fig. 3(a), Fig. 3(b), and Fig. 3(c). Finally graphs for the forwards-backwards terms is depicted in Fig. 4(a), Fig. 4(b), and Fig. 4(c).

Consider the region $60 \leq n \leq 66$ on the horizontal axis of all the figures. Note on Fig. 1(b) that the signal stays on the state 1 for that period and Fig. 2(a) and Fig. 4(a) coincide to point their maximum over that region too.

Now consider the region $15 \leq n \leq 18$, where the state is 2. Now Fig. 2(a) and Fig. 4(a) show that region to be minimal while Fig. 2(b) and Fig. 4(b) coincide to point their maximum over that region. A similar reasoning for state 3 can be drawn on the region $55 \leq n \leq 59$.

Unfortunately, these characteristics are not clearly present on the backwards term displayed on Fig. 3. This suggest a serious implementation problem that cascades through the Baum-Welch algorithm. The final results for the transition probability matrix and the stationary distribution are given by

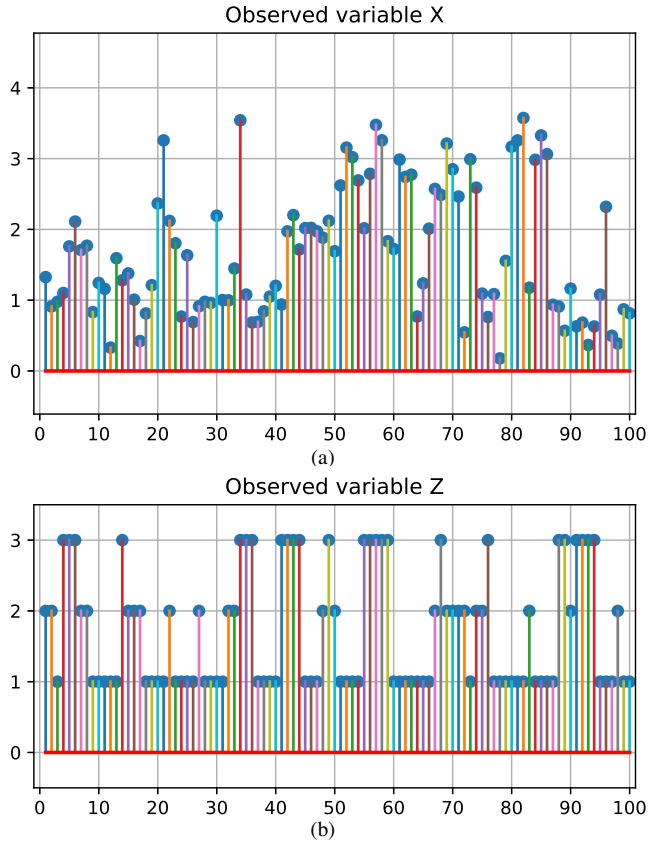


Fig. 1. A realization of the artificially generated data (a) Observed variable, (b) Real state.

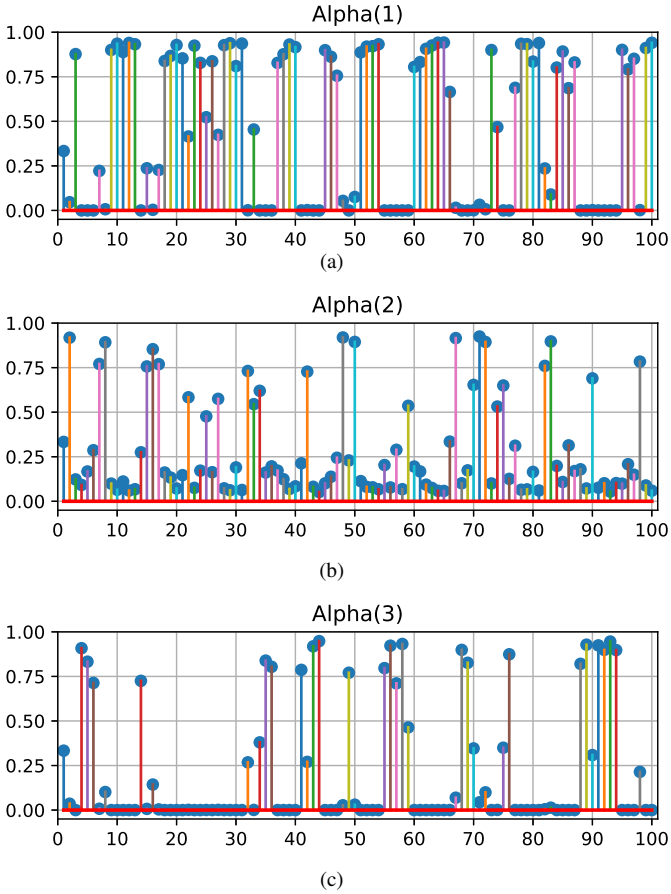


Fig. 2. Forwards term for states (a) 1, (b) 2, (c) 3.

$$P_{HMM} = \begin{bmatrix} 0.5104 & 0.2878 & 0.2017 \\ 0.4146 & 0.3479 & 0.2374 \\ 0.4983 & 0.2369 & 0.2647 \end{bmatrix},$$

$$\boldsymbol{\pi}_{HMM} = \begin{bmatrix} 0.5247 \\ 0.2497 \\ 0.2256 \end{bmatrix}.$$

Even though a fixed number of iterations were commanded without relying on the calculation of the log-likelihood, it might be possible that we are not letting the algorithm run for long enough. However, by inspecting Fig. 5 we can see that means and sigmas converge within the preestablished number of iterations.

IV. Conclusion

We have presented the methodology employ to study HMM on a academic example of a corrupted signal. We presented the theory behind the modeling of such phenomenon using HMM's. We attempted to support our claims with simulation results approximating the parameters of the artificially generated data. However, Fig. 3 suggest that the backwards expression is not correctly implemented. As a consequence, the estimation done on the maximization step are corrupted.

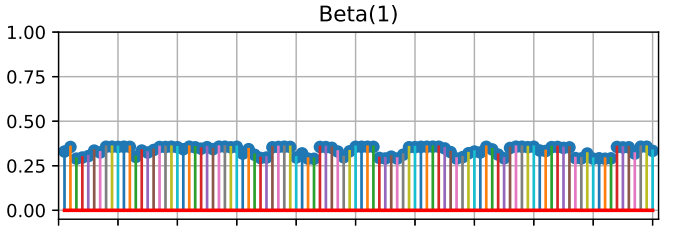
We implemented the forwards, backwards and forwards-backwards steps using both index notation as well as matrix format with no improvement.

There were suspicions about not letting the algorithm execute enough iterations but the convergence shown in Fig. 5 indicates otherwise.

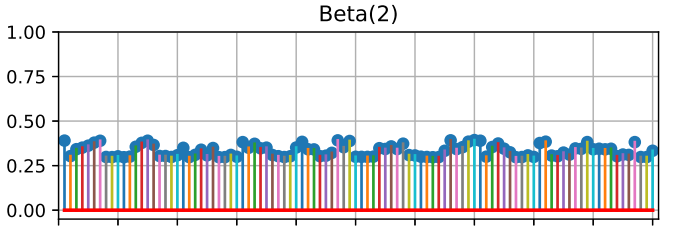
References

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1987.

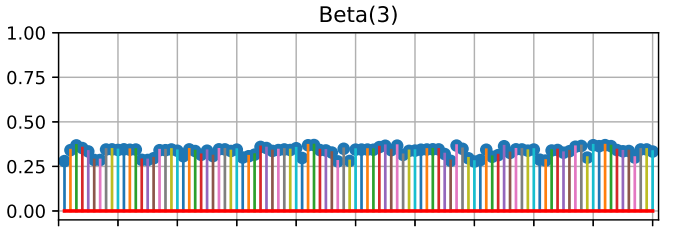
[2] K. P. Murphy, *Machine Learning a probabilistic perspective*. The MIT Press, 2012.



(a)

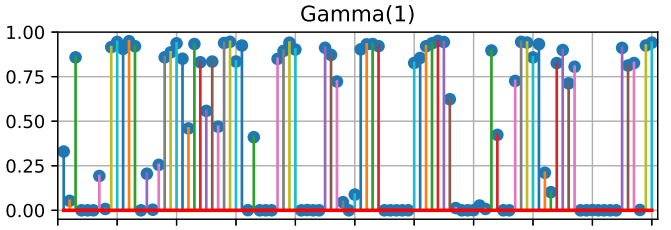


(b)

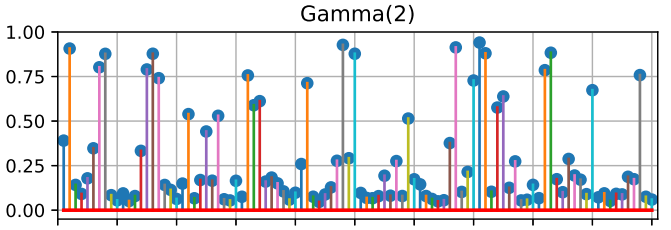


(c)

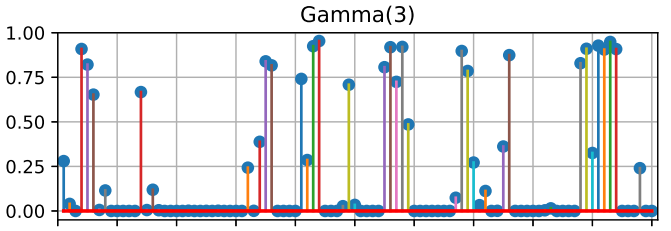
Fig. 3. Backwards term for states (a) 1, (b) 2, (c) 3.



(a)

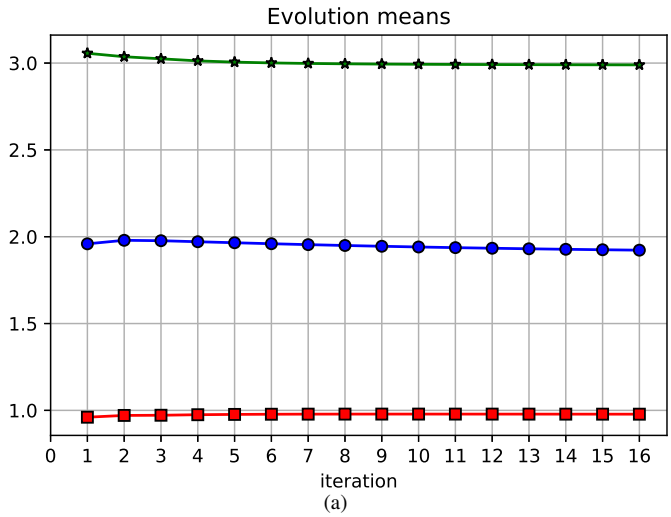


(b)

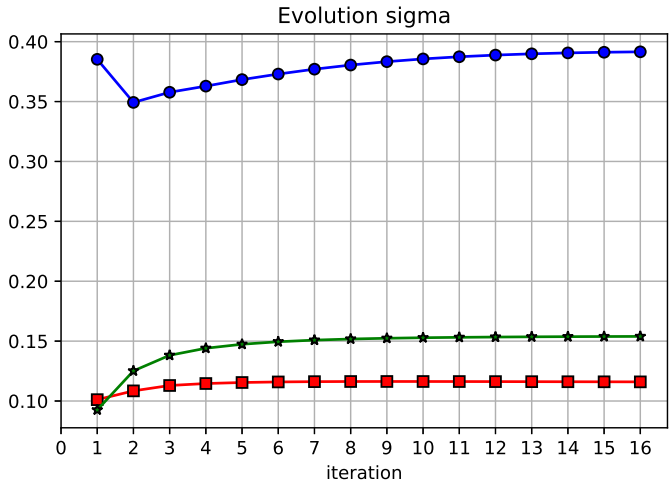


(c)

Fig. 4. Forwards-Backwards for states (a) 1, (b) 2, (c) 3.



(a)



(b)

Fig. 5. Forwards-Backwards for states (a) 1, (b) 2.