CHAPTER 3

HIDDEN MARKOV MODEL
3.1. Background

Hidden Markov chains was primitively introduced and studied in the late 1960s and early 1970s. During the 1980s the models gains popularity. There was two reason for that firstly, the hidden Markov models provides very rich in mathematical structure and hence can form the theoretical basis for a wide range of applications. Secondly, the proper models with HMM, turned out to be highly successful M. Karlsson [46]. Some notable applications of HMM are speech recognition and bio-informatics in particular protein modeling. In this section, basics for the hidden Markov models are explained also the problems, related to modeling of HMM are outlined and solutions for these problems are also provided. Furthermore, extension of the models is also discussed with some implementation issues are considered.

3.2. HMM

Hidden Markov models (HMM) considered as an extension of Markov models to the case where the observation is a probabilistic function of the states of the system. Hence, the resulting model is a doubly embedded stochastic process, which is not necessarily observable, but can be observed through another set of stochastic processes that generate the sequence of observations.

3.3. HMM Description

L.R. Rabiner (1989) [42] suggest the description of HMM as follows:

1. $N$, the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or two sets of states of the model. Let, the individual states as $S = \{s_1, s_2, ..., s_N\}$, and the state at time $n$ as $Z_n$.
2. $M$, the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system
being modeled. Let, the individual symbols as \( V = \{v_1, v_2, ..., v_M\} \), and the symbol at time \( n \) as \( X_n \).

3. The state transition probability matrix \( P = \{P_{ij}\} \), where

\[
P_{ij} = P(Z_{n+1} = S_j | Z_n = S_i), \quad i \leq i, \ j \leq N
\]  

(3.1)

4. The observation symbol probability distribution is state \( S_j \), \( B=(b_{j}(k)} \), where

\[
b_{j}(k) = P(X_n = v_k | Z_n = s_j), \quad 1 \leq j \leq N, \ 1 \leq k \leq M
\]  

(3.2)

5. The initial state distribution \( \pi = \{\pi_i\} \) where,

\[
\pi_i = P(Z_0 = S_i), \quad 1 \leq i \leq N
\]  

(3.3)

The initial state distribution of an HMM requires specifications of the two model parameters \( (N \text{ and } M) \), specification of the observation symbols and the specification of the three probability measures \( P, B \) and \( \pi \). For convenience, we use the compact notation.

\[
\lambda = (P, B, \pi)
\]  

(3.4)

It should be noted here that the above discussion has considered only the case when the observations is characterized as discrete symbols. In principle, this is however not necessary. The symbols or outputs can be either discrete or continuous, and either scalar or vector valued. However, in all cases we need to assume that the stochastic process \( \{Z_n\} \) is a Markov chain having the property that \( X_k^- = \{X_0, ..., X_k\} \) and \( Z_k+1 = \{Z_{k+1}, Z_{k+2} ....\} \) are conditionally independent given. \( Z_k^- = \{Z_0, ..., Z_k\} \). We will, however, from now on assume that we have the case with discrete scalar symbols.

### 3.4. Problems with HMMs

In order to apply the hidden Markov models in real-world applications we need to solve three basic problems:

1. Given the observation sequence, \( X = (x_0, x_1, ..., x_T) \), and a model \( \lambda = (P, B, \pi) \) how do we efficiently compute \( P(X | \lambda) \), the probability of the observation sequence, given the model?
2. Given the observation sequence $X = (x_0, x_1, ..., x_T)$, and the model $\lambda = (P, B, \pi)$, how do we choose a corresponding state sequence $Z = (Z_0, Z_1, ..., Z_T)$, which is optimal in some meaningful sense?

3. How do we adjust the model parameters $\lambda = (P, B, \pi)$ to maximize $P(X|\lambda)$?

The problem 1, can be seen as one of scoring how well a given model matches a given observation sequence, i.e. the solution to this problem would give us a tool to choose between competing models. The problem 2, can be seen as the problem of uncovering the hidden part of the model, i.e. to find the correct state sequence. Problem 3, is the one in which we try to optimize the model parameters so as to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called a training sequence. The training problem is the most crucial one for most applications of HMMs. We will now move on to some discussion on the mathematical solutions of each of the three problems above.

### 3.4.1. Estimation of the probability of the observation sequence $X = (x_0, x_1, ..., x_T)$

The problem is to calculate the probability of the observation sequence given the model $\lambda$. It is possible to do this in a straightforward way, but this is unfortunately computationally unfeasible, even for small values of $N$ and $T$. However, there exists a more efficient procedure called the forward-backward procedure. Consider the forward variable $\alpha_n(i)$ defined as,

$$\alpha_n(i) = P(x_1, x_2, ..., x_n, Z_n = S_i | \lambda)$$  \hspace{1cm} (3.5)

i.e., the probability of the partial observation sequence, $(X_1, X_2, ..., X_n)$ until time $n$ and state $s_i$ at time $n$, given the model $\lambda$. We can here use induction for the problem. First for $n=0$, we have

$$\text{alpha}_0(i) = \pi_i b_i(X_0), \hspace{0.5cm} 1 \leq i \leq N,$$  \hspace{1cm} (3.6)

Induction leads to,
\[ \alpha_{n+1} = \sum_{i=1}^{N} \alpha_n(i) P_{ij} b_j(x_{n+1}), \quad 1 \leq i \leq N, \quad 1 \leq n \leq T - 1 \quad (3.7) \]

\[ \alpha_T(i) = P(X_1, X_2, \ldots, X_T, Z_T = S_i | \lambda) \quad (3.8) \]

it follows that,

\[ P(X | \lambda) = \sum_{i=1}^{N} \alpha_T(i), \quad (3.9) \]

Using the forward variable \( a_n(i) \) we have now solved the first problem above. (Note that this does not include any backward variable. The backward variable is actually not necessary for the solution and is therefore excluded here, but it will appear in the section 3.4.3)

### 3.4.2. Estimation of state sequence \( Z = (z_0, z_1, \ldots, z_T) \) that "best explains" the observations sequence \( X = (x_0, x_1, \ldots, x_T) \).

Unlike, problem in section 3.4.1 where an exact solution can be given, there are several possible ways of solving this problem, i.e. finding the optimal state sequence associated with the given observation sequence. The difficulty comes from the fact that there are a number of different criteria for the term "best explains". One possible "best explains" criterion is to choose the states \( Z_n \), which are individually most likely. This criterion maximizes the expected number of correct individual states, but it does not take into consideration whether the sequence of states is possible. For instance although the transition between two states is impossible i.e. \( P_{ij} = 0 \) for some \( i \) and \( j \), they may still be the most likely at the very instants. This is due to the fact that the solution of this problem simply determines the most likely state at every instant, without considering the probability of occurrence of sequences of states. The most widely used criterion is instead to find the single best state sequence, i.e. to maximize \( P(Z|X, \lambda) \), which is equivalent to maximizing \( P(Z|X, \lambda) \). An algorithm for solving this problem has been found and is called the Viterbi
algorithm. This algorithm can simply be seen as the maximum likelihood estimate. The algorithm can be summarized as follows:

To find the best state sequence, \( Z = \{Z_0, Z_1, ..., Z_T\} \), for the given observation \( X = \{X_0, X_1, ... X_T\} \), we need to define the quantity.

\[
\delta_n(S_i) = \arg \max_{Z_0, Z_1, ..., Z_{n-1}} P(Z_0, Z_1, ..., Z_n = s_i, X_0, ..., X_n | \lambda) \quad (3.10)
\]

i.e. \( \delta_n(S_i) \) is the best score (the highest probability along a single path, at time \( n \), which accounts for the first \( n+1 \) observations and ends in state \( s_i \). By induction, we have.)

\[
\delta_{n+1}(S_i) = \max_i \delta_n(S_i)P_{ij}b_j(X_{n+1}) \quad (3.11)
\]

To actually retrieve the state sequence, we need to keep track of the argument which maximized the above equation, for each \( n \) and \( j \). We do this with the array \( \Psi_n(S_j) \). The procedure for finding the best state sequence now follows as:

Initialization:

\[
\delta_0(S_i) = \pi_i b_i(X_0), \quad 1 \leq i \leq N, \quad (3.12)
\]

\[
\psi_0(s_i) = 0,
\]

Recursion:

\[
\delta_n(S_i) = \max_{1 \leq i \leq N} \delta_{n-1}(S_i)P_{ij}b_j(X_n), \quad 1 \leq n \leq T, 1 \leq j \leq N, \quad (3.13)
\]

\[
\psi_n(S_i) = \left[ \max_{1 \leq i \leq N} \delta_{n-1}(S_i)P_{ij} \right], \quad 1 \leq n \leq T, 1 \leq j \leq N,
\]

Termination:

\[
P^* = \max_{1 \leq i \leq N} [\delta_T(s_i)], \quad (3.14)
\]

\[
Z^* = \arg \max_{1 \leq i \leq N} [\delta_T(s_i)],
\]
4) State sequence backtracking:

\[ Z_n^* = \psi_{n+1}(Z_{n+1}^*), \ n = T - 1, T - 2, \ldots, 1, 0. \]  (3.15)

The best sequence according to the Viterbi algorithm is thus found as \( Z^* = (Z_0^*, Z_1^*, \ldots, Z_T^*) \). It should be noted that apart from the backtracking step the Viterbi algorithm is rather similar to the forward calculation used in section 3.4.1.

### 3.4.3. Tuning of the parameters to find the best model for given observation sequence \( O \)

The by far most difficult of the three problems is to determine a method to adjust the model parameters \( \lambda = (P, B, \pi) \) to maximize the probability of the observation sequence given the model. This problem is in fact not possible to solve using a finite observation sequence as training data, but we can choose \( \lambda = (P, B, \pi) \) such that \( P(X|\lambda) \) is locally maximized using an iterative procedure such as the Baum-Welch method. (Equivalent results will be found using the EM method.) We start of with introducing a backward variable \( \beta_n(i) \)

defined as,

\[ \beta_n(i) = P(X_n+1, X_n+2, \ldots, X_T | Z_n = s_i, \lambda) \]  (3.16)

i.e. the probability of the partial observation sequence from \( n+1 \) to the end, given the state \( s_i \) at time \( n \) and the model \( \lambda \). Again we can solve for \( \beta_n(i) \) inductively, as follows,

\[ \beta_T(i) = 1, \ 1 \leq i \leq N, \]  (3.17)

and induction leads to

\[ \beta_n(i) = \sum_{j=1}^{N} P_{ij} b_j(X_{n+1}) \beta_{n+1}(j), \ n = T - 1, T - 2, \ldots, 1, 0, 1 \leq i \leq N, \]  (3.18)

In order to describe the procedure for re-estimation of HMM parameters, we also define \( \xi_n(i, j) \), the probability of being in state \( s_i \) at time \( n \) and state \( s_j \) at time \( n + 1 \), given the model and the observation sequence, i.e.

\[ \xi_n(i, j) = P(Z_n = s_i, Z_{n+1} = s_j | X, \lambda) \]  (3.19)
From the definition of the forward and backward variables it follows that we can write $\xi_n (i, j)$ in the form,

$$\xi_n (i, j) = \frac{\alpha_n (i) P_{ij} b_j (X_{n+1}) \beta_{n+1} (j)}{P (X | \lambda)} = \frac{\alpha_n (i) P_{ij} b_j (X_{n+1}) \beta_{n+1} (j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_n (i) P_{ij} b_j (X_{n+1}) \beta_{n+1} (j)}$$

(3.20)

where the numerator is simply $P (Z_n = s_i, Z_{n+1} = s_j, X | \lambda)$ and the division by $P (X | \lambda)$ gives the desired probability measure. We also need to define $\gamma_n (i)$ as the probability of being in state $s_i$ at time $n$, given the observation sequence and the model. It follows that,

$$gamma_n (i) = \sum_{j=1}^{N} \xi_n (i, j),$$

(3.21)

If we sum $\gamma_n (i)$ over the time index up to time $T–1$ we get a quantity, which can be interpreted as the expected number of transitions made from state $s_i$. Similarly, summation of $\xi_n (i, j)$ up to time $T–1$ can be interpreted as the expected number of transitions from state $s_i$ to state $s_j$. We can also sum $\gamma_n (i)$ over the time index up to time $T$, which can be interpreted as the expected number of times in state $s_i$. Using this, we can get a method for re-estimation of the parameters in an HMM. The re-estimation formulas can be found as,

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } s_i \text{ at times } (t = 1) = \gamma_n (i)$$

(3.22)

$$P_{ij} = \frac{\text{expected number of transition from state } s_i \text{ to state } s_j}{\text{expected number of transitions from state } s_i} = \frac{\sum_{n=1}^{T} \xi_n (i, j)}{\sum_{n=1}^{T} \xi_n (i, j)}$$

(3.23)

$$b_j (k) = \frac{\text{expected number of times in state } s_j \text{ and observing symbol } v_k}{\text{expected number of times in state } s_j}$$

$$= \frac{\sum_{n=1}^{T} \gamma_n (j) \mathcal{I} (X_n = v_k)}{\sum_{n=1}^{T} \gamma_n (j)}$$

(3.24)

The re-estimation procedure now runs as follows. We define the current model as $\lambda = (P, B, \pi)$, and use that to compute the right-hand side of the above equations, which is put equal to the left-hand side. The left-hand sides are the parameters in the model and this can be used to further improve the model by repeating the procedure until a limiting point is reached.