[Lecture] Hidden Markov Model (HMM)

Jae Yun JUN KIM*

August 20, 2019

<u>References</u>: Wikipedia, P. K. Biswas' lecture notes, Il-Chul Moon's lecture notes, Pieter Abbeel's slides, and Bert Huang's slides.

1 Motivation

The **Hidden Markov Model (HMM)** can be used to reason over time or space. Often, we want to reason about a sequence of observations:

- speech recognition
- robot localization
- user attention
- medical monitoring

To do this, we need to introduce time (or space) into our models. Recall that so far, we considered the data points being independent from each other. But, this is no longer true for the today topic. The problems that we will deal with today can be modeled as a **Dynamic Bayesian Network**.

1.1 Markov models

A **Bayesian Network (BN)** (a.k.a. Bayes Network, Belief Network, Bayes(ian) Model or Probabilistic Directed Acyclic Graphical Model) is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).

For simplicity, we will assume that the past and the future are independent given the present. This assumption is known as the **first order Markov property**. With this property, the BN that we deal with has a *chain structure*, and it is also known as the **Markov Model**.

The *nodes* in the BN represent random variables, while the *edges* represent the *conditional dependencies* between the involved nodes. The nodes that are not connected represent random variables that are *conditionally independent.*

Each node is associated with a probability function that takes (as input) a particular ste of values for the node's parent variables and gives (as output) the probability (or the probability distribution) of the variable represented by the node.

*ECE Paris Graduate School of Engineering, 37 quai de Grenelle 75015 Paris, France; jae-yun.jun-kim@ece.fr

On the other hand, each node is assumed to be identically distributed (i.e., stationary), and it represents the state. Further, the *model parameters* are the transitional probabilities (or dynamics), which specify how the state evolves over time.

1.2 Hidden Markov models (HMMs)

In Markov models, the state z is directly visible to the observer (and, therefore, the state transition probability are the only parameters).

In hidden Markov models, the state z is not directly visible, but only the output x is visible, which depends on the state.

Hence, an HMM can be considered as a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other.

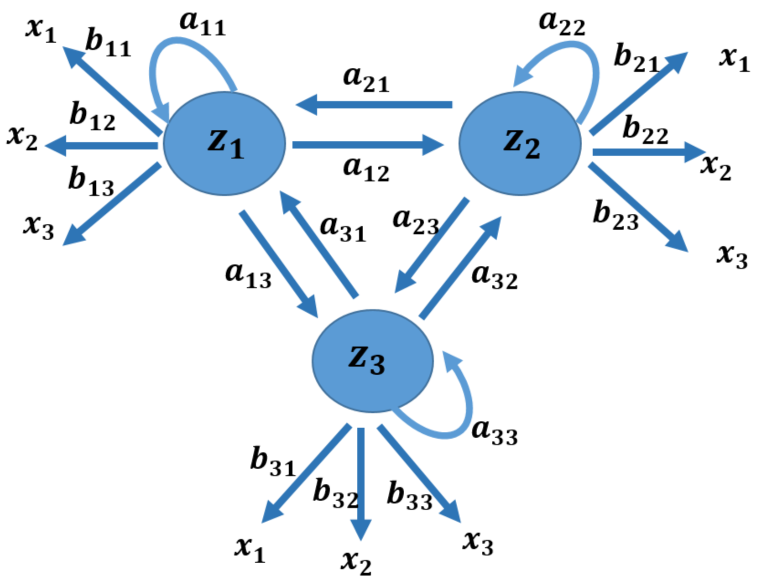

Figure 1: Hidden Markov Model 1

In HMM, there also exists a special hidden state called the **absorbing state** (a.k.a., final state, accepting state, or receiving state). Other states can transition to this one, but once the current state is this absorbing state, we can no longer transition to any other state. In addition, this accepting state emits only one visible symbol. Hence, we have

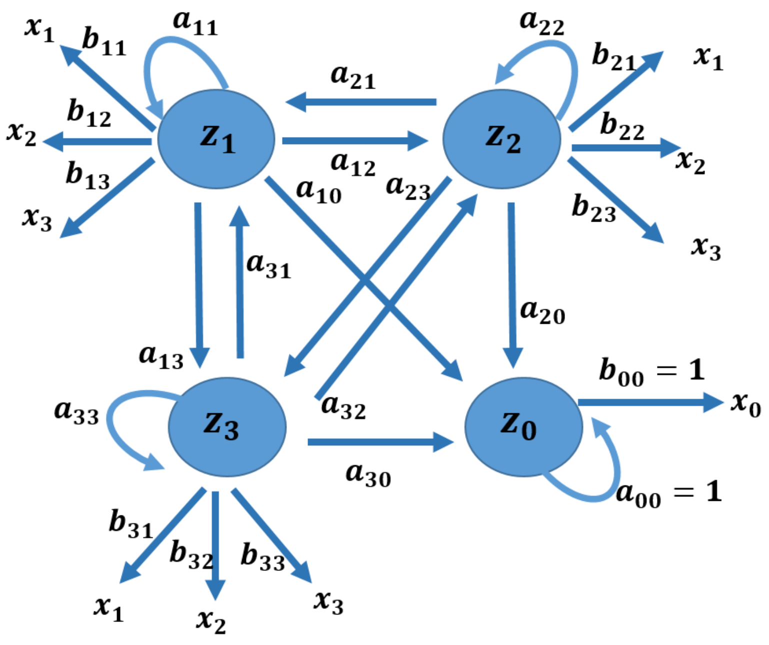

Figure 2: Hidden Markov Model 2

1.3 The problems that can be solved using the HMM

The HMM can be used mainly to solve the following three types of problems:

1. **Evaluation problem**: This problem consists of finding the answer to the question of *what is the probability that a model has generated a sequence of visible states.*

2. **Decoding problem**: This problem consists of finding the answer to the question of *what is the most probable sequence of hidden states through which the model has made the transitions to generate the given sequence of visible states.*

3. **HMM learning problem**: This problem consists of finding the state transition probabilities and the state emission probabilities, for a given coarse structure of an HMM (that is given a number of sequences of hidden states and visible states).

2 Evaluation problem

Given an HMM M and a sequence of T observable (visible) states x, compute $P(x|M)$. Knowing that Z is all possible sequences of T hidden (latent or invisible) states, we can say that

$$P(x|M) = \sum_{r=1}^{r_{max}} P(x|z_r)P(z_r), \tag{1}$$

where r_{max} is the number of all possible sequences of Z, and

$$z_r = \{z_r(1), z_r(2), \cdots, z_r(T)\} \tag{2}$$

is r-th sequence of T hidden states.
If now we denote N as the total number of possible values that a hidden state can take, then $r_{max} = N^T$.

On the other hand, one can estimate

$$P(z_r) = \prod_{t=1}^{T} P(z_r(t)|z_r(t-1)), \tag{3}$$

where $P(z_r)$ is the probability of a particular sequence of z, and $P(z_r(t)|z_r(t-1))$ is a particular **transition probability**.

Similarly, one can compute

$$P(x|z_r) = \prod_{t=1}^{T} P(x(t)|z_r(t)), \tag{4}$$

where $P(x|z_r)$ is the probability that the sequence of observable states x is generated from the sequence of hidden states z_r, and $P(x(t)|z_r(t))$ is a particular **emission probability**.

Hence,

$$P(x|M) = \sum_{r=1}^{r_{max}} \prod_{t=1}^{T} P(x(t)|z_r(t))P(z_r(t)|z_r(t-1)). \tag{5}$$

However, this computation is of order $O(N^T T)$, which is too complex.

Instead, one can use a recursive algorithm to answer the question of what is the probability that a hidden state takes a particular value at a particular time instant for a given sequence of observable states.

Consider the following figure

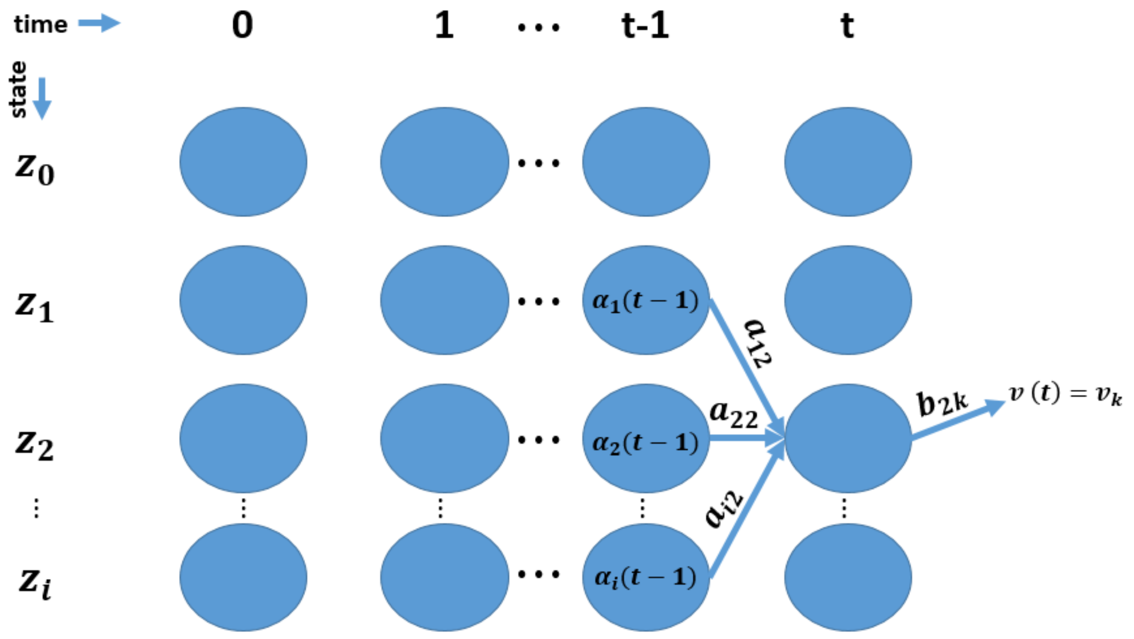

Figure 3: HMM diagram 1

If $x(1) = x_0$, then one has the transition $z_1 \rightarrow z_0$ with probability 1. But, if $x(1) \neq x_0$, then one needs to compute the transition probabilities. Suppose that we are at z_2 at time instant t and that the hidden state value is z_i at $t-1$ with probability $\alpha_i(t-1)$. Previously, we saw that the probability to transition from $z_i(t-1)$ to $z_2(t)$ is a_{i2}. On the other hand, we know the sequence of observable states x. If we now assume that $x(t) = x_k$, then the emission probability at time t will be b_{2k}. Hence, we are now able to compute the probability that the hidden state is z_2 at time t by summing all the products between $\alpha_j(t-1)$ and a_{j2} (where $j = \{0, 1, \cdots, N\}$) and by multiplying afterwards with b_{2k}.

Using this idea, we can define a recurrent algorithm to compute $P(x|M)$. For this purpose we first need to compute $\alpha_j(t)$, where $\alpha_j(t)$ is the probability of $z(t) = z_j$ at time t after having generated the first t numbers of observable states given in the sequence x.

Then, we can define $\alpha_j(t)$ as

$$\alpha_j(t) = \begin{cases} 0, & t = 0 \text{ and } j \neq \text{ initial state}, \\ 1, & t = 0 \text{ and } j = \text{ initial state}, \\ \left(\sum_{i=0}^{N} \alpha_i(t-1) a_{ij} \right) b_{jkx(t)}, & \text{otherwise.} \end{cases} \tag{6}$$

Now, we are ready to define the recurrent algorithm to compute the probability that a given sequence of observable states x is generated from a HMM M. This algorithm is known as the **HMM forward algorithm**.

Algorithm 1: The HMM forward algorithm

```
1 Given: ${a_{ij}}$, ${b_{jk}}$, $x$;
2 Initialize: $\forall j$, $\alpha_j(0)$;
3 for $t = 1 : T$ do
4     for $j = 0 : N$ do
5         $\alpha_j(t) \leftarrow b_{jkx(t)} \sum_{i=0}^{N} \alpha_i(t-1) a_{ij}$;
6 $P(x|M) \leftarrow \alpha_0(T)$;
7 return $P(x|M)$;
```

Hence, the goal of the HMM forward algorithm is to find out what is the probability that a given sequence of visible states are generated from an HMM M.

And, by resolving this problem, we have also computed $\alpha_j(t)$, for $j = \{0, 1, \cdots, N\}$ and $t = \{1, \cdots, T\}$.

2.1 Example 1

Suppose that a hidden state can take the following values: z_0, z_1, z_2, z_3. Suppose also that an observable state can take values: x_0, x_1, x_2, x_3, x_4.
Then, assume also that both the transition and emission probabilities are given:

$$\{a_{ij}\} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.7 & 0.1 & 0.1 & 0.1 \end{bmatrix}, \qquad \{b_{jk}\} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}. \tag{7}$$

On the other hand, we also know that

$$\forall i, \sum_{j=0}^{N} a_{ij} = 1; \qquad \forall j, \sum_{k=0}^{C} b_{jk} = 1. \tag{8}$$

Now, let us proceed to compute the probability of a given sequence of visible states from an HMM M.

Suppose that the following information is given: $x = \{x_1, x_3, x_2, x_0\}$, $z(0) = z_1$, with transition and emission probabilities given in (**??**).
Then, let us compute $P(x|M)$ using the *HMM forward algorithm* introduced in Algorithm **??** as follows.

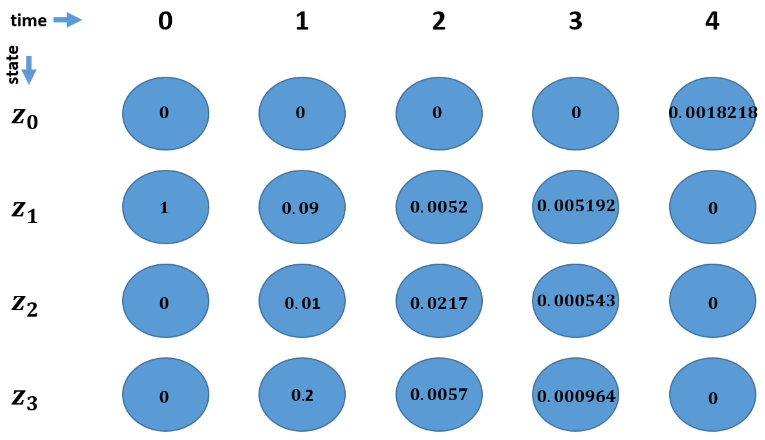

Figure 4: Example 1: the HMM forward algorithm

Hence,

$$P(x|M) = \alpha_0(4) = 0.0018218. \tag{9}$$

The above value is the probability that our model HMM M generated for $x = \{x_1, x_3, x_2, x_0\}$, where M is defined by $\{a_{ij}\}$, $\{b_{jk}\}$ and $\alpha_j(0)$.

3 Decoding problem

This problem consists of answering the question of what is the most probable sequence of hidden states z that would have generated a given sequence of observable (visible) states x.

Coming back to the previous example, we have

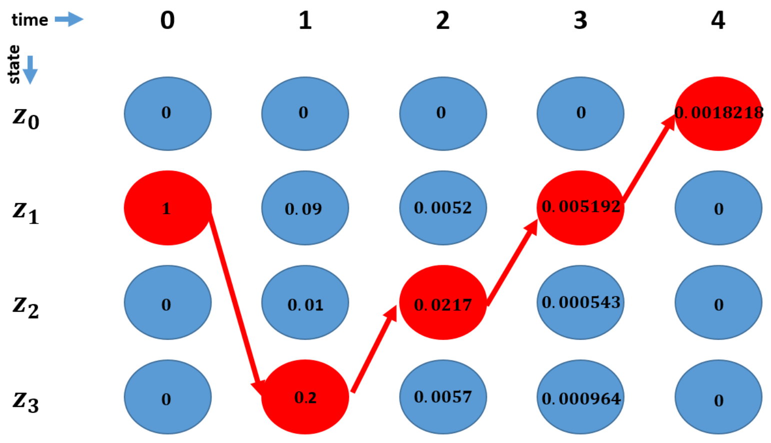

Figure 5: Example 1: Decoding problem

Hence, for each time step, we only consider the hidden state that is the most probable. Then, the sequence of states through which the model transits while generating x is: z_1, z_3, z_2, z_1, z_0.

3.1 Definition of the decoding problem and approach

The decoding problem can be defined as, for a given sequence of visible states x, find the most probable sequence of hidden states.

Algorithm 2: Algorithm for decoding problem

```
1 Given: ${a_{ij}}$, ${b_{jk}}$, $x$;
2 Initialize: $\alpha_j(0)$ and $Path \leftarrow \emptyset$;
3 for $t = 1 : T$ do
4     for $j = 0 : N$ do
5         $\alpha_j(t) \leftarrow b_{jkx(t)} \sum_{i=0}^{N} \alpha_i(t-1) a_{ij}$;
6     $j' \leftarrow \arg\max_j \alpha_j(t)$;
7     Append $z_{j'}$ to $Path$;
8 return $Path$;
```

In this way, we have found the most probable sequence of hidden states through which the machine makes the transitions while generating the sequence of the visible states x.

4 Learning problem

This problem consists of finding the state transition probabilities and the state emission probabilities, for a given coarse structure of an HMM (that is given a number of sequences of hidden states and visible states).

Recall that the *HMM forward algorithm* allowed us to compute $\alpha_j(t)$, which is the probability that the hidden state will have z_j value at time step t after having generated the first t number of visible states.

Now, we will see the **HMM backward algorithm** that will allow us to compute the probability that the model will be at hidden state z_j at time step t and that will generate the remaining part of the sequence of visible states.

4.1 Example 2

Suppose that we have the following sequence of observable states: $x = \{x_1, x_3, x_1, x_5, x_7, x_2, x_0\}$.

Recall that $\alpha_3(4)$ represents the probability that the hidden state is z_3 at time instant 4, after generating the first 4 visible states: x_1, x_3, x_1, x_5.

Now define $\beta_3(4)$ as the probability that the model will be at state z_3 at time instant 4 and that will generate the remaining visible states: x_7, x_2, x_0.

In other words, $\beta_i(t)$ is the probability that the moidel will be in $z_i(t)$ and that will generate the remainder of the given target sequence x (i.e., from $x(t+1)$ to $x(T)$).

Let us define then,

$$\beta_i(t) = \begin{cases} 0, & t = T \text{ and } z_i(t) \neq z_0, \\ 1, & t = T \text{ and } z_i(t) = z_0, \\ \displaystyle\sum_{j=0}^{N} \beta_j(t+1) a_{ij} b_{jkx(t+1)}, & \text{otherwise.} \end{cases} \tag{10}$$

Let us now define the **HMM backward algorithm** as follows:

Algorithm 3: The HMM backward algorithm

```
1 Given: $\{a_{ij}\}$, $\{b_{jk}\}$, $x$;
2 Initialize: $\forall i$, $\beta_i(T)$;
3 for $t = (T-1) : -1 : 0$ do
4     for $i = 0 : N$ do
5         $\beta_i(t) \leftarrow \sum_{j=0}^{N} \beta_j(t+1) a_{ij} b_{jkx(t+1)}$;
6 $P(x|M) \leftarrow \beta_{i'}(0)$;
7 return $P(x|M)$;
```

Now, let us use both the forward and backward algorithms to learn $\{a_{ij}\}$ and $\{b_{jk}\}$.
We observe that both $\alpha_j(t)$ and $\beta_i(t)$ use $\{a_{ij}\}$ and $\{b_{jk}\}$, but we do not know their values. For this reason, let us define $\gamma_{ij}(t)$, which represents the transition probability from $z_i(t-1)$ to $z_j(t)$ for a particular sequence of observable states x. Hence,

$$\gamma_{ij}(t) = \alpha_i(t-1) a_{ij} b_{jk} \beta_j(t). \tag{11}$$

Now, this definition can be used to estimate $\{a_{ij}\}$ and $\{b_{jk}\}$.

Note that the **expected number of transitions from $z_i(t-1)$ to $z_j(t)$ at any time in the sequence x** is

$$\sum_{t=1}^{T} \gamma_{ij}(t). \tag{12}$$

On the other hand, the **total expected number of transitions from $z_i(t-1)$ to any state** is

$$\sum_{t=1}^{T} \sum_{l=0}^{N} \gamma_{il}(t). \tag{13}$$

Then, using the above two expected numbers, we can estimate $\{a_{ij}\}$ and $\{b_{jk}\}$ as follows:

$$\hat{a}_{ij} = \frac{\displaystyle\sum_{t=1}^{T} \gamma_{ij}(t)}{\displaystyle\sum_{t=1}^{T} \sum_{l=0}^{N} \gamma_{il}(t)}, \quad \hat{b}_{jk} = \frac{\displaystyle\sum_{t=1}^{T} \sum_{l=0}^{N} \gamma_{jl}(t) \mathbb{I}\{x(t) = x_k\}}{\displaystyle\sum_{t=1}^{T} \sum_{l=0}^{N} \gamma_{jl}(t)}, \tag{14}$$

where $i = \{0, 1, \cdots, N\}$, $j = \{0, 1, \cdots, N\}$, and $k = \{0, 1, \cdots, C\}$.

This algorithm is known as the **Forward-Backward algorithm** or as the **Baum-Welch algorithm**, which can be detailed as

Algorithm 4: Baum-Welch algorithm

```
1 Given: $x$;
2 Initialize: $\{\hat{a}_{ij}\}$, $\{\hat{b}_{jk}\}$;
3 while not converged do
4     Estimate $\alpha_j(t)$ and $\beta_i(t)$;
5     Estimate $\gamma_{ij}(t)$;
6     Update $\{\hat{a}_{ij}\}$, $\{\hat{b}_{jk}\}$;
7 return $\{\hat{a}_{ij}\}$, $\{\hat{b}_{jk}\}$;
```