Machine Learning for Data Science (CS 4786)

Lecture 20: Hidden Markov Models

The text in black outlines main ideas to retain from the lecture. The text in blue give a deeper understanding of how we "derive" or get to the algorithm or method. The text in red are mathematical details for those who are interested. But is not crucial for understanding the basic workings of the method.

1 Markov Models

A stationary markov model or stationary markov chain is a classic model in probability theory where we have a a sequence of random variables, call them states, that are generated as follows. Initial state S_1 is decided by drawing from marginal distribution $P(S_1)$ to be one of K values. Next, subsequent states are drawn based on previous state using the so called transition probability $P(S_t = s_t|S_{t-1} = s_{t-1}) = T[s_{t-1}, s_t]$ where T is the $K \times K$ transition probability table where the i'th row indicates the probability of jumping from state i to each of the K different states. This probabilistic model can be represented by the following graphical model: The parameters of model are the conditional probability tables of each variable given parents which in this case are: 1. $P(S_1)$ which is a table of size $1 \times K$ indicating initial probabilities of S_1 being in one of K states, and 2. Transition probability table T of size $K \times K$.

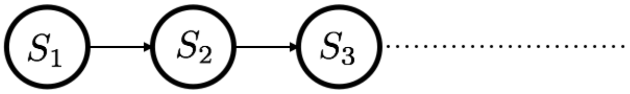

Figure 1: Mixture model

2 Hidden Markov Models

While Markov chains are fascinating topics to which whole books are dedicated to, lets quickly turn our attention to an extension called the hidden markov model. These models consist of a markov model, however the internal states S_t's are unobservable, what we observe at each step is X_t an observation generated by state S_t. Hidden markov models have great many applications. One can think of internal state as being location of a bot and observation as being sensory information we observe for bot location example, or we can think of state S_t as being phoneme we wanted to utter at the t'th time snap and X_t as the actual waveform located for that time and here inferring states is a key step in voice recognition tasks.

3 Inference in HMM

In this section, we will answer key question of inference in HMM's. Specifically questions like how do we calculate $P(S_t = k|X_1, \ldots, X_N)$, that is probability that state at time t is k given all observations $X_1, \ldots, X_N$. To this end note that

$$\begin{aligned}
&P(S_t = k|X_1, \ldots, X_N) \\
&= \frac{P(X_{t+1}, \ldots, X_N|S_t = k, X_1, \ldots, X_t)P(S_t = k, X_1, \ldots, X_t)}{P(X_1, \ldots, X_N)} && \text{by Bayes theorem} \\
&\propto P(X_{t+1}, \ldots, X_N|S_t = k, X_1, \ldots, X_t)P(S_t = k, X_1, \ldots, X_t) \\
&= P(X_{t+1}, \ldots, X_N|S_t = k, X_1, \ldots, X_t)P(X_t|S_t = k, X_1, \ldots, X_{t-1})P(S_t = k, X_1, \ldots, X_{t-1}) \\
&= P(X_{t+1}, \ldots, X_N|S_t = k, X_1, \ldots, X_t)P(X_t|S_t = k)P(S_t = k, X_1, \ldots, X_{t-1}) && \text{Local Markov} \\
&= P(X_{t+1}, \ldots, X_N|S_t = k)P(X_t|S_t = k)P(S_t = k, X_1, \ldots, X_{t-1}) && \text{Local Markov}
\end{aligned}$$

Now notice that in the above, the term $P(X_t|S_t = k)$ is a conditional probability of node given parent and is given by looking up the so called emission probability table $E[k, X_t]$. We now only need to figure out how to compute terms $P(X_{t+1}, \ldots, X_N|S_t = k)$ and $P(S_t = k, X_1, \ldots, X_{t-1})$. We will show how to compute these two tables recursively given their value for neighbors. Specifically, we will think of the term $P(S_t = k, X_1, \ldots, X_{t-1})$ as being calculated from left to right recursively (called forward pass) and the term $P(X_{t+1}, \ldots, X_N|S_t = k)$ as being computed from right to left called backward pass.

3.1 Forward Pass

We will denote the term $M_{S_{t-1} \mapsto S_t}(k) = P(S_t = k, X_1, \ldots, X_{t-1})$ and think of $M_{S_{t-1} \mapsto S_t}$ as a K dimensional vector froward message that state $t-1$ passes to state t. Note that at $t = 1$, we have no left neighbor and on this round,

$$M_{\{\} \mapsto S_1}(k) = P(S_t = k)$$

Or in other words we can think of a hallucinated message from left for first guy as simply the initial state probability which is parameter to our model. Now that we ave the initial one, let us compute the others recursively. Specifically, assume that for any $j \in \{1, \ldots, K\}$,

$M_{S_{t-2}\mapsto S_{t-1}}(j) = P(S_{t-1} = j, X_1, \ldots, X_{t-2})$. Now note that

$$\begin{aligned}
&M_{S_{t-1}\mapsto S_t}(k) \\
&= P(S_t = k, X_1, \ldots, X_{t-1}) \\
&= \sum_{j=1}^{K} P(S_t = k, S_{t-1} = j, X_1, \ldots, X_{t-1}) && \text{:(marginalization)} \\
&= \sum_{j=1}^{K} P(S_t = k|S_{t-1} = j, X_1, \ldots, X_{t-1}) P(S_{t-1} = j, X_1, \ldots, X_{t-1}) && \text{:}(P(A,B) = P(A|B)P(B)) \\
&= \sum_{j=1}^{K} P(S_t = k|S_{t-1} = j) P(S_{t-1} = j, X_1, \ldots, X_{t-1}) && \text{:(local markov)} \\
&= \sum_{j=1}^{K} P(S_t = k|S_{t-1} = j) P(X_{t-1}|S_{t-1} = j, X_1, \ldots, X_{t-2}) P(S_{t-1} = j, X_1, \ldots, X_{t-2}) && \text{:}(P(A,B) = P(A|B)P(B)) \\
&= \sum_{j=1}^{K} P(S_t = k|S_{t-1} = j) P(X_{t-1}|S_{t-1} = j) P(S_{t-1} = j, X_1, \ldots, X_{t-2}) && \text{:(local markov)} \\
&= \sum_{j=1}^{K} P(S_t = k|S_{t-1} = j) P(X_{t-1}|S_{t-1} = j) P(S_{t-1} = j, X_1, \ldots, X_{t-2}) \\
&= \sum_{j=1}^{K} T[j,k] E[j, X_{t-1}] M_{S_{t-2}\mapsto S_{t-1}}(j)
\end{aligned}$$

Thus we see that if the $t-1$'th node receives its message from $t-2$'th node, then it can compute its message to t'th node using just the message from left and the T and E tables.

4 Backward Pass

The backward pass can be done similarly. Note that $M_{S_t\mapsto S_{t-1}}(k) = P(X_{t+1}, \ldots, X_N|S_t = k)$. Now let us compute the message from node $t = N-1$ onwards, that is, note that, In this case, note that

$$\begin{aligned}
M_{S_{N-1}\mapsto S_{N-2}}(k) &= P(X_N|S_{N-1} = k) \\
&= \sum_{j=1}^{K} P(S_N = j, X_N|S_{N-1} = k) \\
&= \sum_{j=1}^{K} P(X_N|S_N = j, S_{N-1} = k) P(S_N = j|S_{N-1} = k) \\
&= \sum_{j=1}^{K} P(X_N|S_N = j) P(S_N = j|S_{N-1} = k) && X_N \text{ c.i. of } S_{N-1} \text{ given } S_N \\
&= \sum_{j=1}^{K} E[S_N = j, X_N] T[k,j]
\end{aligned}$$

Thus we see that we can compute $M_{S_{N-1} \mapsto S_{N-2}}$. Now subsequently, let us compute each message to send to left by node based on the backward message it received. To this end note that,

$$\begin{aligned}
&M_{S_t \mapsto S_{t-1}}(k) \\
&= P(X_{t+1}, \ldots, X_N | S_t = k) \\
&= \sum_{j=1}^{K} P(X_{t+1}, \ldots, X_N, S_{t+1} = j | S_t = k) && \text{(marginalization)} \\
&= \sum_{j=1}^{K} P(X_{t+1}, \ldots, X_N | S_{t+1} = j, S_t = k) P(S_{t+1} = j | S_t = k) && :(P(A, B|C) = P(A|B, C)P(B|C))) \\
&= \sum_{j=1}^{K} P(X_{t+1}, \ldots, X_N | S_{t+1} = j) P(S_{t+1} = j | S_t = k) && \text{(local markov)} \\
&= \sum_{j=1}^{K} P(X_{t+1} | S_{t+1} = j) P(X_{t+2}, \ldots, X_N | S_{t+1} = j) P(S_{t+1} = j | S_t = k) && \text{(local markov)} \\
&= \sum_{j=1}^{K} E[j, X_{t+1}] M_{S_{t+1} \mapsto S_t}(j) T[k, j]
\end{aligned}$$

Thus we see that the backward message can be computed recursively as well.

5 Inference

Now say you wanted to answer the question of what is $P(S_t = k | X_1, \ldots, X_N)$ we cn do this simply by setting:

$$P(S_t = k | X_1, \ldots, X_N) = M_{S_{t-1} \mapsto S_t}(k) \times E[k, X_t] \times M_{S_{t+1} \mapsto S_t}(k)$$

that is product of emission table entry with messages from left and right as we saw earlier.

Similarly let us say we want to infer the probability $P(S_t = s_t, S_{t-1} = s_{t-1} | X_1, \ldots, X_N)$. Note that,

$$\begin{aligned}
&P(S_t = s_t, S_{t-1} = s_{t-1}|X_1, \ldots, X_N) \\
&= \frac{P(X_1, \ldots, X_N|S_t = s_t, S_{t-1} = s_{t-1})P(S_t = s_t, S_{t-1} = s_{t-1})}{P(X_1, \ldots, X_N)} \\
&\propto P(X_1, \ldots, X_N|S_t = s_t, S_{t-1} = s_{t-1})P(S_t = s_t, S_{t-1} = s_{t-1}) \\
&= P(X_1, \ldots, X_{t-1}|S_t = s_t, S_{t-1} = s_{t-1})P(X_t, \ldots, X_N|S_t = s_t, S_{t-1} = s_{t-1})P(S_t = s_t, S_{t-1} = s_{t-1}) \\
&= P(X_1, \ldots, X_{t-1}|S_{t-1} = s_{t-1})P(X_t, \ldots, X_N|S_t = s_t)P(S_t = s_t, S_{t-1} = s_{t-1}) \qquad \text{(local markov)} \\
&= P(X_1, \ldots, X_{t-2}|S_{t-1} = s_{t-1})P(X_{t-1}|S_{t-1} = s_{t-1})P(X_{t+1}, \ldots, X_N|S_t = s_t)P(X_t|S_t = s_t) \\
&\quad \times P(S_t = s_t, S_{t-1} = s_{t-1}) \qquad \text{(local markov)} \\
&= P(X_1, \ldots, X_{t-2}|S_{t-1} = s_{t-1})P(X_{t-1}|S_{t-1} = s_{t-1})P(X_{t+1}, \ldots, X_N|S_t = s_t)P(X_t|S_t = s_t) \\
&\quad \times P(S_t = s_t|S_{t-1} = s_{t-1})P(S_{t-1} = s_{t-1}) \\
&= P(X_1, \ldots, X_{t-2}, S_{t-1} = s_{t-1})P(X_{t-1}|S_{t-1} = s_{t-1})P(X_{t+1}, \ldots, X_N|S_t = s_t)P(X_t|S_t = s_t) \\
&\quad \times P(S_t = s_t|S_{t-1} = s_{t-1}) \\
&= M_{S_{t-1} \mapsto S_{t-1}}(s_{t-1}) \times M_{S_{t+1} \mapsto S_t}(s_t) \times E[s_t, X_t] \times E[s_{t-1}, X_{t-1}] \times T[s_{t-1}, s_t]
\end{aligned}$$

Thus we see that we can compute $P(S_t = s_t, S_{t-1} = s_{t-1}|X_1, \ldots, X_N)$ as well from forward and backward messages.

6 Learning in HMM

Our go to model for learning whenever we have latent variables is the EM algorithm. The general strategy in EM algorithm is that we start randomly initialized parameter (in this case T^0 and E^0 for now assume $P(S_1)$ is given). Subsequently over multiple iterations, t, we first compute $Q^{(t)}(\text{latent}) = P(\text{latent}|\text{Observation})$. in the E-step (which is clearly and inference) followed by a M-step (maximization step) where we compute

$$(T^{(t)}, E^{(t)}) = \arg\max_{T,E} \sum_{\text{all}-\text{values}-\text{of}-\text{latent}} Q^{(t)}(\text{latent}) \log(P(\text{latent}, \text{observed}))$$

Hence for HMM this is simply,

$$(T^{(t)}, E^{(t)}) = \arg\max_{T,E} \sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} Q^{(t)}(S_1 = s_1, \ldots, S_N = s_N) \log(P_{T,E}(X_1, \ldots, X_N, S_1 = s_1, \ldots, S_N = s_n))$$

Now we use the fact that the the joint probability factors over the graph and so:

$$P(X_1, \ldots, X_N, S_1 = s_1, \ldots, S_N = s_n) = \prod_{i=1}^{N} P_{T,E}(X_i|S_i = s_i)P_{T,E}(S_i = s_i|S_{i-1} = s_{i-1}) = \prod_{i=1}^{N} E[s_i, X_i]T[s_{i-1}, s_i]$$

Hence we see that

$$\begin{aligned}
(T^{(t)}, E^{(t)}) &= \arg\max_{T,E} \sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} Q^{(t)}(S_1 = s_1, \ldots, S_N = s_N) \log(\prod_{i=1}^{N} E[s_i, X_i] T[s_{i-1}, s_i]) \\
&= \arg\max_{T,E} \sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} Q^{(t)}(S_1 = s_1, \ldots, S_N = s_N) \left(\sum_{i=1}^{N} \log(E[s_i, X_i]) + \sum_{i=1}^{N} \log(T[s_{i-1}, s_i]) \right) \\
&= \arg\max_{T,E} \sum_{i=1}^{N} \left(\sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} Q^{(t)}(S_1 = s_1, \ldots, S_N = s_N) \log(E[s_i, X_i]) \right. \\
&\qquad\qquad \left. + \sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} Q^{(t)}(S_1 = s_1, \ldots, S_N = s_N) \log(T[s_{i-1}, s_i]) \right)
\end{aligned}$$

Now note that for any i,

$$\begin{aligned}
&\sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} Q^{(t)}(S_1 = s_1, \ldots, S_N = s_N) \log(T[s_{i-1}, s_i]) \\
&\quad = \sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} P_{T^{(t-1)},E^{(t-1)}}(S_1 = s_1, \ldots, S_N = s_N | X_1, \ldots, X_N) \log(T[s_{i-1}, s_i]) \\
&\quad = \sum_{s_{i-1}=1}^{K} \sum_{s_i=1}^{K} \left(\sum_{s_1=1}^{K} \cdots \sum_{s_{i-2}=1}^{K} \sum_{s_{i+1}=1}^{K} \cdots \sum_{s_N=1}^{K} P_{T^{(t-1)},E^{(t-1)}}(S_1 = s_1, \ldots, S_N = s_N | X_1, \ldots, X_N) \right) \log(T[s_{i-1}, s_i]) \\
&\quad = \sum_{s_{i-1}=1}^{K} \sum_{s_i=1}^{K} P_{T^{(t-1)},E^{(t-1)}}(S_{i-1} = s_{i-1}, S_i = s_i | X_1, \ldots, X_N) \log(T[s_{i-1}, s_i])
\end{aligned}$$

Similarly we have that

$$\begin{aligned}
&\sum_{s_1=1}^{K} \cdots \sum_{s_N=1}^{K} Q^{(t)}(S_1 = s_1, \ldots, S_N = s_N) \log(E[s_i, X_i]) \\
&\quad = \sum_{s_i=1}^{K} P_{T^{(t-1)},E^{(t-1)}}(S_i = s_i | X_1, \ldots, X_N) \log(E[s_i, X_i])
\end{aligned}$$

Hence we conclude that the M-step in EM algorithm is given by

$$T^{(t)} = \arg\max_{T} \sum_{i=1}^{N} \sum_{s_{i-1}=1}^{K} \sum_{s_i=1}^{K} P_{T^{(t-1)},E^{(t-1)}}(S_{i-1} = s_{i-1} S_i = s_i | X_1, \ldots, X_N) \log(T[s_{i-1}, s_i])$$

and

$$E^{(t)} = \arg\max_{E} \sum_{i=1}^{N} \sum_{s_i=1}^{K} P_{T^{(t-1)},E^{(t-1)}}(S_i = s_i | X_1, \ldots, X_N) \log(E[s_i, X_i])$$

Note that from the above we can conclude that all we need from M-step is only $P_{T^{(t-1)},E^{(t-1)}}(S_i = s_i|X_1,\ldots,X_N)$ and $P_{T^{(t-1)},E^{(t-1)}}(S_i = s_i, S_{i-1} = s_{i-1}|X_1,\ldots,X_N)$ both of which we have shown how to compute in the inference section. Similar to the derivation of the M-step in the mixture of multinomial setting (see lecture notes), if we add the constraint that rows of T and E are probability tables, we can conclude that:

$$\forall u, v \quad T^{(t)}[u,v] = \frac{\sum_{i=2}^{N} P_{T^{(t-1)},E^{(T-1)}}(S_i = v, S_{i-1} = u|X_1,\ldots,X_N)}{\sum_{i=2}^{N} P_{T^{(t-1)},E^{(T-1)}}(S_{i-1} = u|X_1,\ldots,X_N)}$$

and similarly

$$\forall v, x \quad E^{(t)}[v,x] = \frac{\sum_{i:X_i=x} P_{T^{(t-1)},E^{(T-1)}}(S_i = v|X_1,\ldots,X_N)}{\sum_{i=2}^{N} P_{T^{(t-1)},E^{(T-1)}}(S_i = v|X_1,\ldots,X_N)}$$

This gives the Baum Welch algorithm where we start with random initialization of T and E and on every iteration first run forward backward algorithm, compute $P_{T^{(t-1)},E^{(T-1)}}(S_t = v|X_1,\ldots,X_N)$ and $P_{T^{(t-1)},E^{(T-1)}}(S_i = v, S_{i-1} = u|X_1,\ldots,X_N)$ using the forward and backward messages for every i in the E-step and then in M-step set the new E and T tables using above calculations.