CSCI-567: Machine Learning (Fall 2019)

Prof. Victor Adamchik

U of Southern California

Nov. 19, 2019

Outline

1. Hidden Markov Models

Outline

1. Hidden Markov Models
   - Forward and backward messages
   - Viterbi Algorithm
   - Viterbi Algorithm: Example
   - Learning HMMs

Definition

A Markov chain is a stochastic process with the **Markov property**: a sequence of random variables $X_1, X_2, \cdots, X_T$ s.t.

$$P(X_{t+1}|X_1, X_2, \cdots, X_t) = P(X_{t+1}|X_t)$$

i.e. *the current state only depends on the most recent state*.

We denote the transition and initial probabilities as

$$a_{s,s'} = P(X_{t+1} = s'|X_t = s), \quad \pi_s = P(X_1 = s)$$

Each state $X_t \in 1, 2, \ldots, S$ also "emits" some **outcome** O_t based on the following model

$$P(O_t \mid X_t = s) = b_{s,O_t} \qquad \text{(emission probability)}$$

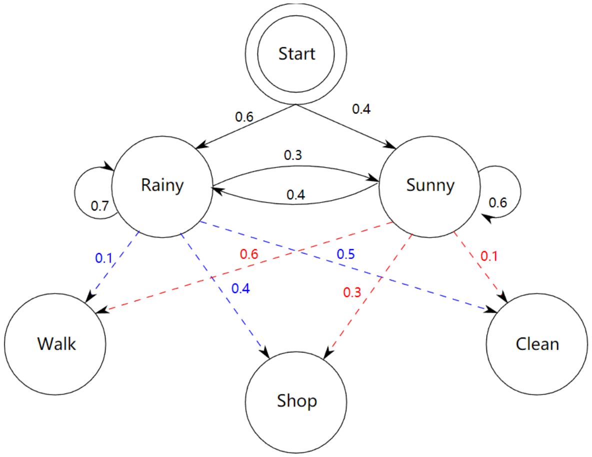
independent of anything else.

The model parameters are $(\{\pi_s\}, \{a_{s,s'}\}, \{b_{s,O_t}\}) = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.

Example

picture from Wikipedia

On each day, we observe **Bob's activity: walk, shop, or clean**, which only depends on the weather of that day.



HMM defines a joint probability

$$P(X_1, X_2, \cdots, X_T, O_1, O_2, \cdots, O_T)$$
$$= P(X_1, X_2, \cdots, X_T)\, P(O_1, O_2, \cdots, O_T \mid X_1, X_2, \cdots, X_T)$$

- Markov assumption simplifies the first term

$$P(X_1, X_2, \cdots, X_T) = P(X_1) \prod_{t=2}^{T} P(X_t \mid X_{t-1})$$

- The *independence* assumption simplifies the second term

$$P(O_1, O_2, \cdots, O_T \mid X_1, X_2, \cdots, X_T) = \prod_{t=1}^{T} P(O_t \mid X_t)$$

Namely, each O_t is conditionally independent of anything else, if conditioned on X_t.

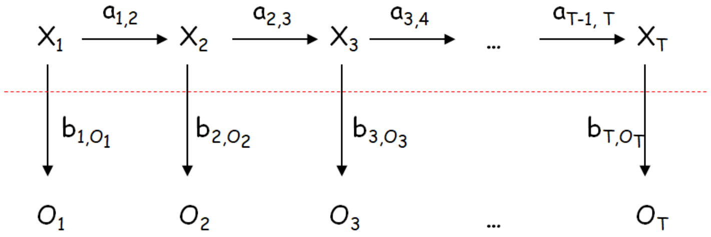
Learning the model

If we observe M state-outcome sequences: $x_{m,1}, o_{m,1}, \ldots, x_{m,T}, o_{m,T}$ for $m = 1, \ldots, M$, the MLE is again very simple (verify yourself).

However, *most often we do not observe the states!* Think about the speech recognition example. This is called **Hidden Markov Model (HMM)**.

Notice that "hidden" is referred to the states of the Markov chain, not to the parameters of the model.

A generic hidden Markov model is illustrated in this picture:



HMM problems

There are three fundamental problems that we solve:

- **Problem 1**: Scoring and evaluation

Given an observation sequence $O_1, O_2, \ldots, O_T$ and a model $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$, how to compute efficiently the probability of $P(O_1, O_2, \ldots, O_T)$?

HMM problems

There are three fundamental problems that we solve:

- **Problem 2**: Decoding (Viterbi algorithm)

  Given an observation sequence $O_1, O_2, \ldots, O_T$ and a model $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$, how do we determine the optimal corresponding state sequence $X_1, X_2, \ldots, X_T$ that best explains how the observations were generated?

HMM problems

There are three fundamental problems that we solve:

- **Problem 3**: Training

  Given an observation sequence $O_1, O_2, \ldots, O_T$, how to adjust the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ to maximize the probability of $P(O_1, O_2, \ldots, O_T)$? In the other words, find a model to best fit the observed data. we will solve this by the Baum–Welch algorithm.

Chain Rule

In all derivations we will be using the chain rule to calculate any member of the joint distribution using only conditional probabilities.

$$P(X, Y) = P(X \mid Y)\ P(Y) = P(Y \mid X)\ P(X)$$

$$P(X, Y, Z) = P(X, Y \mid Z)\ P(Z)$$

$$P(X, Y, Z) = P(X \mid Y, Z)\ P(Y \mid Z)\ P(Z)$$

Forward and backward messages

The key is to compute two things:

- **forward messages**: for each s and t

  $$\alpha_s(t) = P(X_t = s, O_{1:t} = o_{1:t})$$

  The intuition is, if we observe up to time t, what is the likelihood of the Markov chain in state s?

- **backward messages**: for each s and t

  $$\beta_s(t) = P(O_{t+1:T} = o_{t+1:T} \mid X_t = s)$$

  The interpretation is: if we are told that the Markov chain at time t is in the state s , then what are the likelihood of observing future observations from $t + 1$ to T?

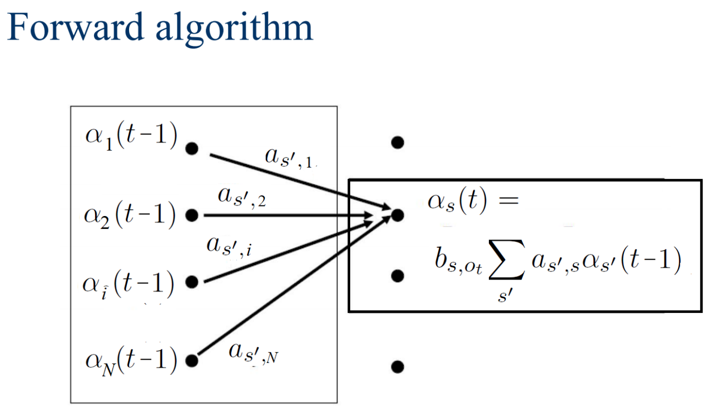
Computing forward messages

Key: *establish a recursive formula*

$$
\begin{aligned}
\alpha_s(t) &= P(X_t = s, O_{1:t}) = P(X_t = s, O_{1:t-1}, O_t) \\
&= P(O_t \mid X_t = s, O_{1:t-1}) P(X_t = s, O_{1:t-1}) \\
&= P(O_t \mid X_t = s) P(X_t = s, O_{1:t-1}) && \text{(independence)} \\
&= b_{s,o_t} \sum_{s'} P(X_t = s, X_{t-1} = s', O_{1:t-1}) && \text{(marginalizing)} \\
&= b_{s,o_t} \sum_{s'} P(X_t = s | X_{t-1} = s', O_{1:t-1}) P(X_{t-1} = s', O_{1:t-1}) \\
&= b_{s,o_t} \sum_{s'} P(X_t = s | X_{t-1} = s') P(X_{t-1} = s', O_{1:t-1}) \\
&= b_{s,o_t} \sum_{s' \in [N]} a_{s',s} \alpha_{s'}(t-1)
\end{aligned}
$$

Base case: $\alpha_s(1) = P(X_1, O_1) = P(O_1|X_1)P(X_1) = \pi_s b_{s,o_1}$

Forward procedure



Forward procedure

Forward algorithm

For all $s \in [N]$, compute $\alpha_s(1) = \pi_s b_{s,o_1}$.

For $t = 2, \ldots, T$

- for each $s \in [N]$, compute

$$\alpha_s(t) = b_{s,o_t} \sum_{s' \in [N]} a_{s',s} \, \alpha_{s'}(t-1)$$

It takes $O(N^2T)$ time and $O(NT)$ space using dynamic programming.

Oh, no, CSCI-570 again..

Computing backward messages

Again establish a recursive formula

$$
\begin{aligned}
\beta_s(t) &= P(O_{t+1:T} \mid X_t = s) = P(O_{t+1:T}, X_t = s)/P(X_t = s) = \\
&= \sum_{s'} P(O_{t+1:T}, X_{t+1} = s', X_t = s)/P(X_t = s) && \text{(marginalizing)} \\
&= \sum_{s'} P(O_{t+1:T} \mid X_{t+1} = s', X_t = s) P(X_{t+1} = s' \mid X_t = s) \\
&= \sum_{s'} a_{s,s'} P(O_{t+1:T} \mid X_{t+1} = s') = \sum_{s'} a_{s,s'} P(O_{t+1}, O_{t+2:T} \mid X_{t+1} = s') \\
&= \sum_{s'} a_{s,s'} P(O_{t+1} \mid O_{t+2:T}, X_{t+1} = s') P(O_{t+2:T} \mid X_{t+1} = s') \\
&= \sum_{s'} a_{s,s'} b_{s',o_{t+1}} \beta_{s'}(t+1)
\end{aligned}
$$

Base case: $\beta_s(T) = 1$ (prove it!)

Backward procedure

Backward algorithm

For all $s \in [N]$, set $\beta_s(T) = 1$.

For $t = T - 1, \ldots, 1$

- for each $s \in [N]$, compute

$$\beta_s(t) = \sum_{s' \in [N]} a_{s,s'}\, b_{s',o_{t+1}} \beta_{s'}(t+1)$$

Again it takes $O(N^2T)$ time and $O(NT)$ space.

Solving Problem 1

With forward messages $\alpha_s(t) = P(X_t = s, O_{1:t})$, we can compute $P(O_{1:T})$.

Indeed,

$$P(O_{1:T}) = \sum_s P(O_{1:T}, X_T = s) = \sum_s \alpha_s(T)$$

Solving Problem 2

Given the model and a sequence of observations, our goal is to find the most likely sequence of states that maximizes $P(X_{1:T}, O_{1:T})$.

This is called Viterbi decoding. We solve this using Dynamic Programming.

We define DP subproblems in the following way – the highest probable state sequence that ends at $X_t = s$ given observations $O_1, O_2, \ldots, O_t$

$$\delta_s(t) = \max_{X_{1:t-1}} P(X_{1:t-1}, X_t = s, O_{1:t})$$

In the next slide we compute $\delta_s(t)$ recursively.

Computing $\delta_s(t)$

The goal is to get a recurrence. We will use $X_t = s, X_{t-1} = s'$.

$$\begin{aligned}
\delta_s(t) &= \max_{X_{1:t-1}} P(X_t = s, X_{1:t-1}, O_{1:t}) \\
&= \max_{X_{1:t-1}} P(X_t = s, O_t, X_{1:t-1}, O_{1:t-1}) \\
&= \max_{s'} P(X_t = s, O_t \mid X_{1:t-1}, O_{1:t-1}) \max_{X_{1:t-2}} P(X_{1:t-1}, O_{1:t-1}) \\
&= \max_{s'} \delta_{s'}(t-1) P(X_t, O_t \mid X_{1:t-1}, O_{1:t-1}) \\
&= \max_{s'} \delta_{s'}(t-1) P(O_t, X_t \mid X_{1:t-1}) \\
&= \max_{s'} \delta_{s'}(t-1) P(O_t \mid X_t, X_{1:t-1}) P(X_t \mid X_{1:t-1}) \\
&= \max_{s'} \delta_{s'}(t-1) P(O_t \mid X_t) P(X_t \mid X_{t-1}) \\
&= b_{s,o_t} \max_{s'} a_{s',s} \delta_{s'}(t-1)
\end{aligned}$$

Base case: $\delta_s(1) = P(X_1 = s, O_1 = o_1) = \pi_s b_{s,o_1}$

The optimal path

Note that this only gives the optimal probability, not the optimal path itself.

$$\delta_s(t) = b_{s,o_t} \max_{s'} a_{s',s}\delta_{s'}(t-1)$$

We need to keep a track of each preceding state where the maximum occurs. Thus we create a table to record the highest-scoring state at each possible state at each time-stamp.

$$\Delta_s(t) = \operatorname*{argmax}_{s'} a_{s',s}\delta_{s'}(t-1)$$

This must remind you Dijkstra's shortest path algorithm from CS570.

Viterbi Algorithm

Viterbi Algorithm

For each $s \in [N]$, compute $\delta_s(1) = \pi_s b_{s,o_1}$.

For each $t = 2, \ldots, T$,

- for each $s \in [N]$, compute

$$\delta_s(t) = b_{s,o_t} \max_{s'} a_{s',s}\delta_{s'}(t-1)$$

$$\Delta_s(t) = \operatorname*{argmax}_{s'} a_{s',s}\delta_{s'}(t-1)$$

Backtracking: let $o^*_T = \operatorname{argmax}_s \delta_s(T)$.
For each $t = T, \ldots, 2$: set $o^*_{t-1} = \Delta_{o^*_t}(t)$.

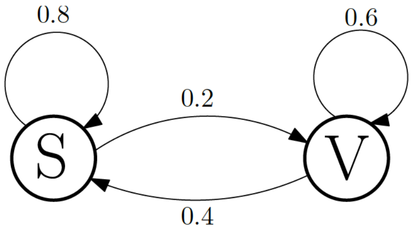
Output the most likely path $o^*_1, \ldots, o^*_T$.

Example

Consider the HMM below. In this world, every time step (say every few minutes), you can either be Studying or playing Video games. You're also either Grinning or Frowning while doing the activity.

E	$p(E\|X=S)$
grin	0.5
frown	0.5

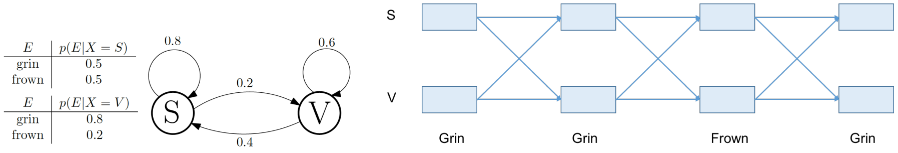
E	$p(E\|X=V)$
grin	0.8
frown	0.2



Suppose that we believe that the initial state distribution is 50/50. We observe: Grin, Grin, Frown, Grin. What is the most likely path for this sequence of observations?

$t = 1$, the initial time

$\delta_s(1) = \pi_s b_{s,o_1}$. Compute $\delta_S(1)$ and $\delta_V(1)$.

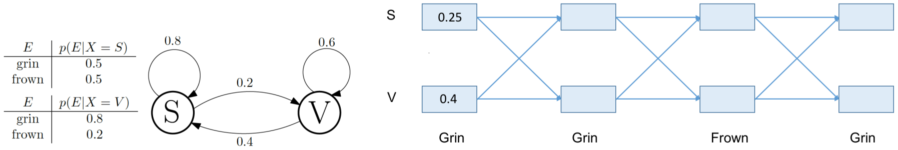


$$\delta_S(1) = P(O_1 = Grin|X_1 = S)\pi(X_1 = S) = 0.5 \times 0.5 = 0.25$$

$$\delta_V(1) = P(O_1 = Grin|X_1 = V)\pi(X_1 = V) = 0.8 \times 0.5 = 0.4$$

$t = 2$

$\delta_s(t) = b_{s,o_t} \max_{s'} a_{s',s}\delta_{s'}(t-1)$. Compute $\delta_S(2)$ and $\delta_V(2)$.



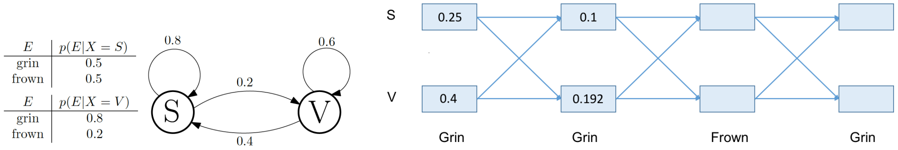
E	$p(E\|X=S)$
grin	0.5
frown	0.5

E	$p(E\|X=V)$
grin	0.8
frown	0.2

$$\begin{aligned}
\delta_S(2) = {} & P(O_2 = Grin|X_2 = S)\times \\
& \max\{P(X_2 = S|X_1 = S)\delta_S(1), P(X_2 = S|X_1 = V)\delta_V(1)\} \\
& = 0.5 \times \max\{0.8 \times 0.25, 0.4 \times 0.4\} = 0.01 \\
\delta_V(2) = {} & P(O_2 = Grin|X_2 = V)\times \\
& \max\{P(X_2 = V|X_1 = S)\delta_S(1), P(X_2 = V|X_1 = V)\delta_V(1)\} \\
& 0.8 \times \max\{0.2 \times 0.25, 0.6 \times 0.4\} = 0.192 \\
\Delta_S(2) = {} & S, \Delta_V(2) = S
\end{aligned}$$

$t = 3$

$\delta_s(t) = b_{s,o_t} \max_{s'} a_{s',s}\delta_{s'}(t-1)$. Compute $\delta_S(3)$ and $\delta_V(3)$.



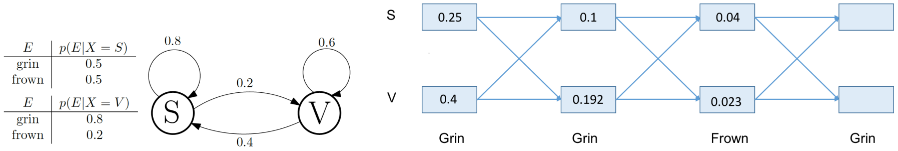
E	$p(E\|X=S)$
grin	0.5
frown	0.5

E	$p(E\|X=V)$
grin	0.8
frown	0.2

$$\begin{aligned}
\delta_S(3) = {} & P(O_3 = Frown|X_3 = S)\times \\
& \max\{P(X_3 = S|X_2 = S)\delta_S(2), P(X_3 = S|X_2 = V)\delta_V(2)\} \\
& = 0.5 \times \max\{0.8 \times 0.1, 0.4 \times 0.192\} = 0.04 \\
\delta_V(3) = {} & P(O_3 = Frown|X_3 = V)\times \\
& \max\{P(X_3 = V|X_2 = S)\delta_S(2), P(X_3 = V|X_2 = V)\delta_V(2)\} \\
& = 0.2 \times \max\{0.2 \times 0.1, 0.6 \times 0.192\} = 0.023 \\
\Delta_S(3) = {} & S, \Delta_V(3) = V
\end{aligned}$$

$t = 4$

Observation at $t = 4$ is 'Grin'



E	$p(E\|X=S)$
grin	0.5
frown	0.5

E	$p(E\|X=V)$
grin	0.8
frown	0.2

$$\begin{aligned}
\delta_S(4) = {} & P(O_4 = Grin|x_4 = S)\times \\
& \max\{P(X_4 = S|X_3 = S)\delta_S(3), P(X_4 = S|X_3 = V)\delta_V(3)\} \\
& = 0.5 \times \max\{0.8 \times 0.04, 0.4 \times 0.023\} = 0.016 \\
\delta_V(4) = {} & P(O_4 = Grin|X_4 = V)\times \\
& \max\{P(X_4 = V|X_3 = S)\delta_S(3), P(X_4 = V|X_3 = V)\delta_V(3)\} \\
& = 0.8 \times \max\{0.2 \times 0.04, 0.6 \times 0.023\} = 0.011 \\
\Delta_S(4) = {} & S, \Delta_V(4) = V
\end{aligned}$$

Then the path is $S(4) \leftarrow S(3) \leftarrow S(2) \leftarrow S(1)$. Please verify!

Problem 3

Given a sequence of observations, our goal is to adjust the model parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ to best fit the observations (to maximize the probability of $P(O_1, O_2, \dots, O_T)$).

First, we define

$$\gamma_s(t) = P(s \mid O_{1:T})$$

as a probability of being at state $X_t = s$ at time t. e.g. given Bob's activities for one week, how was the weather like on Wed?

$\gamma_s(t)$ is computed using forward and backward messages:

$$\gamma_s(t) = \frac{\alpha_s(t)\beta_s(t)}{P(O_{1:T})}$$

Here the denominator is the solution to Problem 1.

Computing $\gamma_s(t)$

Problem 3

Next, we define

$$\xi_{s,s'}(t) = P(s, s' \mid O_{1:T})$$

a probability of being at state $X_t = s$ at time t and at state $X_{t+1} = s'$ at time $t + 1$, e.g. given Bob's activities for one week, how was the weather like on Wed and Thu?

This probability is computed using forward and backward messages:

$$\xi_{s,s'}(t) = \frac{\alpha_s(t)\, a_{s,s'}\, b_{s',O_{t+1}}\, \beta_{s'}(t+1)}{P(O_{1:T})}$$

Here $\gamma_s(t)$ and $\xi_{s,s'}(t)$ are related by

$$\sum_{s'} \xi_{s,s'}(t) = \gamma_s(t)$$

Computing $\xi_{s,s'}(t)$

The Baum–Welch algorithm

The algorithm trains both the transition probabilities A and the emission probabilities B of the HMM.

The Baum–Welch algorithm (1972) is a special case of the more general Expectation-Maximization (EM) algorithm (1977).

EM is an iterative algorithm, computing an initial estimate for the probabilities, then using those estimates to computing a better estimate, and so on, iteratively improving the probabilities that it learns.

The Baum–Welch algorithm

The solution to Problem 3 can be summarized as follows:

- Initialize the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.
- Compute $\alpha_s(t), \beta_s(t), \gamma_s(t)$ and $\xi_{s,s'}(t)$.
- Update the model parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.
- If $P(O_{1:T})$ increases, goto 2.

Initialization

We initialize $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ with a best guess or randomly (uniformly)

$$\pi_s \sim 1/N, \quad a_{s,s'} \sim 1/N, \quad b_{s,O_t} \sim 1/N.$$

Note, the parameters must be row stochastic:

$$\sum_i \pi_i = 1, \quad \sum_j a_{i,j} = 1, \quad \sum_j b_{i,j} = 1.$$

Updating the model parameters

Compute new initial probability in state s by:

$$\pi_s = \gamma_s(1)$$

A new transition probability from state s to state s' is computed by:

$$a_{s,s'} = \sum_{t=1}^{T-1} \xi_{s,s'}(t) \Big/ \sum_{t=1}^{T-1} \gamma_s(t)$$

The numerator here is the expected number of transitions from state s to state s'.

The denominator is the expected number of transitions from s to any state.

Updating the model parameters

A new emission probability in state s observing $O_t = k$ is computed by:

$$b_{s,k} = \sum_{t=1}^{T} \mathbb{I}\left[O_t == k\right] \gamma_s(t) \Big/ \sum_{t=1}^{T} \gamma_s(t)$$

where $\mathbb{I}\left[x\right]$ denotes an indicator function.

The denominator here is the expected number of times the model is in state s.

The numerator is the expected number of times the model is in state s with observation $O_t = k$.

General EM algorithm

Step 0 Initialize $\theta = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.

Step 1 (E-Step) update the posterior of latent variables

$$q = P(X_t = s \mid O_{1:T}\ ; \theta)$$

and obtain expectation of complete likelihood

$$Q(\theta) = \mathbb{E}_{X_{1:T} \sim q}\left[\ln P(O_{1:T}, X_{1:T}\ ; \theta)\right]$$

Step 2 (M-Step) update the model parameter via maximization

$$\underset{\theta}{\operatorname{argmax}}\, Q(\theta)$$

Step 3 goto Step 1 if not converged

Applying EM: E-Step

In the E-Step we fix the parameters and find the posterior distributions q of the hidden states (for each sample)

$$q = P(X_t = s \mid O_{1:T}\ ; \theta) = \gamma_s(t)$$

This leads to the complete log-likelihood:

$$Q(\theta) = \mathbb{E}_{X_{1:T} \sim q}\left[\ln P(X_{1:T}, O_{1:T})\right]$$

We showed in the previous lecture that

$$\ln P(X_{1:T}, O_{1:T}) = \ln \pi_{X_1} + \sum_{t=2}^{T} \ln a_{X_{t-1}, X_t} + \sum_{t=1}^{T} \ln b_{X_t, O_t}$$

Applying EM: E-Step

It follows,

$$\begin{aligned} Q(\theta) &= \mathbb{E}_{X_{1:T} \sim q}\left[\ln \pi_{X_1} + \sum_{t=2}^{T} \ln a_{X_{t-1}, X_t} + \sum_{t=1}^{T} \ln b_{X_t, O_t}\right] \\ &= \sum_s \gamma_s(1) \ln \pi_s + \sum_{t=1}^{T-1} \sum_{s,s'} \xi_{s,s'}(t) \ln a_{s,s'} + \sum_{t=1}^{T} \sum_s \gamma_s(t) \ln b_{s,O_t} \end{aligned}$$

In the first term we are repeatedly selecting the values of X_1, so it is just the marginal expression for time $t = 1$.

In the second term we are looking over all transitions from X_{t-1} to X_t and weighting that by the corresponding probability.

In the last term we are looking at the emissions for all states and weighting each possible emission by the corresponding probability, so that is just the sum of the marginal for time t.

Applying EM: E-Step

Let us maximize the first term.

Adding the Lagrange multiplier λ, using the constraint that $\sum_s \pi_s = 1$, and setting the derivative equal to zero, we get

$$\frac{\partial}{\partial \pi_s}\left(\sum_s \gamma_s(1) \ln \pi_s + \lambda\left(1 - \sum_s \pi_s\right)\right) = 0$$

Take the derivative

$$\frac{\gamma_s(1)}{\pi_s} = \lambda$$

then use the constraint $\sum_s \pi_s = 1$, to get $\sum_s \gamma_s(1) = \lambda$.

So we get

$$\pi_s = \frac{\gamma_s(1)}{\sum_s \gamma_s(1)} = \gamma_s(1)$$

Applying EM: E-Step

Let us maximize the third term:

$$\sum_{t=1}^{T}\sum_{s}\gamma_s(t)\ln b_{s,O_t}$$

where

$$\begin{aligned}\gamma_s(t) &= P(X_t = s \mid O_{1:T})\\ &= P(X_t = s, O_t = k \mid O_{1:T}) + P(X_t = s, O_t \neq k \mid O_{1:T})\end{aligned}$$

Adding the Lagrange multiplier λ, using the constraint that $\sum_k b_{s,k} = 1$, and setting the derivative equal to zero, we get

$$\frac{\partial}{\partial b_{s,O_t}}\left(\sum_{t=1}^{T}\sum_{s}\gamma_s(t)\ln b_{s,O_t} + \lambda\left(1 - \sum_{k} b_{s,k}\right)\right) = 0$$

Applying EM: E-Step

Take the derivative to get

$$\frac{1}{b_{s,k}}\sum_{t=1}^{T} P(X_t = s, O_t = k \mid O_{1:T}) = \lambda$$

Next we sum it up over k

$$\sum_{k}\sum_{t=1}^{T} P(X_t = s, O_t = k \mid O_{1:T}) = \lambda\sum_{k} b_{s,k}$$

Using marginalization and the the constraint $\sum_k b_{s,k} = 1$, we get

$$\sum_{t=1}^{T}\gamma_s(t) = \lambda$$

Applying EM: E-Step

So, it follows

$$b_{s,k} = \frac{\sum_{t=1}^{T} P(X_t = s, O_t = k \mid O_{1:T})}{\sum_{t=1}^{T}\gamma_s(t)}$$

which could be also written as in slide 36

$$b_{s,k} = \frac{\sum_{t=1}^{T}\gamma_s(t)\mathbb{I}\left[O_t == k\right]}{\sum_{t=1}^{T}\gamma_s(t)}$$