Lecture 15
Inference in Hidden Markov Models Part 2

Alice Gao

November 2, 2021

Contents

1 Learning Goals

By the end of the lecture, you should be able to

- Calculate the smoothing probability for a time step in a hidden Markov model.
- Describe the justification for a step in the derivation of the smoothing formulas.
- Describe the forward-backward algorithm

2 The Smoothing Task

Let's recall the definition of the smoothing task. Let the first time step be 0. Suppose that today is day t-1. Smoothing asks the following question: Given the observations from day 0 to t-1, what is the probability that I am in a particular state on dad k in the past? Mathematically, what is the probability of S_k given $o_{0:(t-1)}$?

$$P(S_k|o_{0:t-1}), 0 \leq k \leq t-1$$

Smoothing is useful since new observations can help us derive more accurate estimates of the states in the past.

Similar to filtering, smoothing is not calculating a single probability. Since S_k can be true or false. We are computing a distribution containing two probabilities.

2.1 The Smoothing Formulas

To calculate a smoothed probability, we will make use of forward and backward recursion.

Example:

$$\begin{aligned}
&P(S_k|o_{0:t-1}) \\
&= \alpha\, P(S_k|o_{0:k})\, P(o_{(k+1):(t-1)}|S_k) \\
&= \alpha\, f_{0:k}\, b_{(k+1):(t-1)}
\end{aligned}$$

First, we'll take the probability and write it as a normalized product of two probabilities. Don't worry about understanding the formula now. I'll explain it a bit later.

The first probability is equivalent to the filtering message for day k, which is $f_{0:k}$. We already know how to calculate it using forward recursion.

We will calculate the second probability using backward recursion. To do this, let's define the second probability as a message for b for backward recursion. The subscript (k+1) to (t-1) indicates the sequence of observations in the probability. Backward recursion passes the message b from time step k = t - 1 back to k = 0.

Let's look at the formulas for backward recursion.

Example: Base case:

$$b_{t:(t-1)} = \vec{1}. \tag{1}$$

For k = t-1, we have the base case. In this case, the message is $b_{t:t-1}$. This message is equal to $P(o_{t:t-1}|S_{t-1}$. The sequence in the subscript doesn't make much sense. How can we have a sequence of time steps starting from t and ending at t-1? This is impossible. Because of this, we interpret this sequence of time steps as an empty sequence. Given an empty sequence of observations, we define the probabilities to be 1's. This is why the base case is a vector of 1's.

Example: Recursive case:

$$b_{(k+1):(t-1)} = \sum_{s_{k+1}} P(o_{k+1}|s_{k+1}) * b_{(k+2):(t-1)} * P(s_{k+1}|S_k), \text{ or} \tag{2}$$

$$P(o_{(k+1):(t-1)}|S_k) = \sum_{s_{k+1}} P(o_{k+1}|s_{k+1}) * P(o_{(k+2):(t-1)}|s_{k+1}) * P(s_{k+1}|S_k). \tag{3}$$

If k is an integer from 0 to t-2, we have the recursive case.

We are given the message $b_{(k+2):(t-1)}$. We want to calculate the message $b_{(k+1):(t-1)}$. Note that we are going backward in time, from day k+2 to day k+1.

The message is a summation. Each term in the summation is a product of three terms. The first term comes from the sensor model. The last term comes from the transition model.

3 Smoothing Calculations

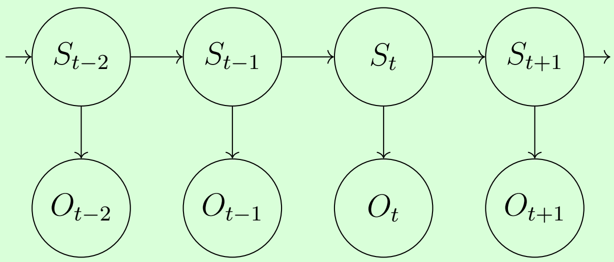
Let's go through some examples of calculating smoothed probabilities.

Problem:
Consider the umbrella story. Assume that 3 days have passed and the director carried an umbrella every day. We want to calculate the probability that it rained on day 0 ($P(S_0|o_{0:2})$) and the probability that it rained on day 1 ($P(S_1|o_{0:2})$).
The umbrella model is given below.

$P(s_0) = 0.5$

$P(s_t|s_{t-1}) = 0.7$
$P(s_t|\neg s_{t-1}) = 0.3$

$P(o_t|s_t) = 0.9$
$P(o_t|\neg s_t) = 0.2$



Solution:

Let's calculate the probability that it rained on day 1 first.

To use the smoothing formula, we should figure out the values of k and t. By comparing our probability with the probability in the formula, we can see that $k = 1$ and $t = 3$. We have gathered 3 observations and we are looking to estimate the state on day 1.

Next, let's write the probability as a normalized product of two messages.

$$\begin{aligned} & P(S_1|o_{0:2}) \\ & = \alpha \, f_{0:1} \, b_{2:2} \end{aligned}$$

The first message is $f_{0:1}$ - the probability of the state on day 1 given the observations from the first two days. We already calculated this using forward recursion.

$$_{0:0} = \langle 0.883, 0.117 \rangle. \tag{4}$$

The second message is $b_{2:2}$. We will calculate this message using backward recursion.

Problem: Next, let's calculate $b_{2:2} = P(o_{2:2}|S_1)$ using backward recursion.

Solution: Let's calculate $b_{1:1}$. Since the sequence in the subscript is not empty, so this is not the base case. We need to apply the formula for the recursive case. Let's take the recursive formula and plug in $k = 1$ and $t = 3$.

$$\begin{aligned} b_{2:2} &= P(o_{2:2}|S_1) \\ &= \sum_{s_2} P(o_2|s_2) * b_{3:2} * P(s_2|S_1) \end{aligned}$$

Next, let's rewrite the middle term from the message form back to a conditional probability. Writing the term as a probability makes it easier to expand the summation later.

$$\begin{aligned} &= \sum_{s_2} P(o_2|s_2) * b_{3:2} * P(s_2|S_1) \\ &= \sum_{s_2} P(o_2|s_2) * P(o_{3:2}|s_2) * P(s_2|S_1) \end{aligned}$$

Each term in the summation is a product of three probabilities. Recall that the middle probability is a b message from the previous step of the backward recursion. The sequence of observations in this b message is empty since it starts at day 3 and ends at day 2. This is the base case and we have defined each probability in the tuple to be 1.

Next, let's look at the last term in the product. S_1 could be true or false. Let's write the last term as a tuple of two probabilities, one for $s_1 = t$ and the other one for $s_1 = f$.

$$\begin{aligned} &= \sum_{s_2} P(o_2|s_2) * P(o_{3:2}|s_2) * P(s_2|S_1) \\ &= \sum_{s_2} P(o_2|s_2) * P(o_{3:2}|s_2) * \langle P(s_2|s_1), P(s_2|\neg s_1)\rangle \end{aligned}$$

Finally, let's write the sum over S_2 explicitly into two terms, one for $s_2 = t$ and the other one for $s_2 = f$

$$\begin{aligned} &= \sum_{s_2} P(o_2|s_2) * P(o_{3:2}|s_2) * \langle P(s_2|s_1), P(s_2|\neg s_1)\rangle \\ &= \Big(P(o_2|s_2) * P(o_{3:2}|s_2) * \langle P(s_2|s_1), P(s_2|\neg s_1)\rangle \\ &\quad + P(o_2|\neg s_2) * P(o_{3:2}|\neg s_2) * \langle P(\neg s_2|s_1), P(\neg s_2|\neg s_1)\rangle \Big) \end{aligned}$$

At this point, the formula contains all small letters. We are ready to plug in the numbers.

$$
\begin{aligned}
b_{2:2} &= P(o_{2:2}|S_1) \\
&= \Big(P(o_2|s_2) * P(o_{3:2}|s_2) * \langle P(s_2|s_1), P(s_2|\neg s_1)\rangle \\
&\quad + P(o_2|\neg s_2) * P(o_{3:2}|\neg s_2) * \langle P(\neg s_2|s_1), P(\neg s_2|\neg s_1)\rangle \Big) \\
&= \Big(0.9 * 1 * \langle 0.7, 0.3\rangle + 0.2 * 1 * \langle 0.3, 0.7\rangle \Big) \\
&= (0.9 * \langle 0.7, 0.3\rangle + 0.2 * \langle 0.3, 0.7\rangle) \\
&= (\langle 0.63, 0.27\rangle + \langle 0.06, 0.14\rangle) \\
&= \langle 0.69, 0.41\rangle
\end{aligned}
$$

The first term in each product comes from the sensor model. The last term in each product comes from the transition model. The middle term in each product is the b value in the base case. So we have a 1 in each product.

When we multiply a number with a tuple, we multiply the number with every probability in the tuple.

Problem: Calculate $P(S_1|o_{0:2})$.

Solution: We have calculated the message $b_{2:2}$ using backward recursion. We are now ready to derive the smoothed probability.

Recall the smoothing formula. The smoothed probability is a normalized product of two messages, one from forward recursion and one from backward recursion.

$$
\begin{aligned}
& P(S_1|o_{0:2}) \\
&= \alpha\, P(S_1|o_{0:1}) * P(o_{2:2}|S_1) \\
&= \alpha\, f_{0:1} * b_{2:2} \\
&= \alpha \langle 0.883, 0.117\rangle * \langle 0.69, 0.41\rangle \\
&= \alpha \langle 0.6093, 0.0480\rangle \\
&= \langle 0.927, 0.073\rangle
\end{aligned}
$$

We need to multiply the forward recursion message and the backward recursion message together. This is an element-wises multiplication again. The final step is to normalize the product.

Here is our final answer. Given that the director brought the umbrella on both days, the probability that it rained on day 0 is quite high, around 88%.

Problem: Next, let's calculate $P(S_0|o_{0:2})$.

Solution:

We can use the same procedure to calculate the probability that it rained on day 0 given the three observations.

First, we compare the probability and our formula to determine that $k = 0$ and $t = 3$.

Similar to the previous example, we've already calculated the f message through forward recursion. We need to calculate the b message through backward recursion and then calculate a normalized product of the two messages.

First, we will calculate the message $b_{1:2}$ using backward recursion. In each product, the middle term corresponds to the component in the previous message $b_{2:2}$. The main difference between this smoothing example and the previous smoothing example is that the middle term in each product is not 1 since it is not the base case of backward recursion.

$$
\begin{aligned}
& b_{1:2} = P(o_{1:2}|S_0) \\
& = (P(o_1|s_1) * P(o_{2:2}|s_1) * \langle P(s_1|s_0), P(s_1|\neg s_0)\rangle \\
& \quad + P(o_1|\neg s_1) * P(o_{2:2}|\neg s_1) * \langle P(\neg s_1|s_0), P(\neg s_1|\neg s_0)\rangle) \\
& = (0.9 * 0.69 * \langle 0.7, 0.3\rangle + 0.2 * 0.41 * \langle 0.3, 0.7\rangle) \\
& = \langle 0.4593, 0.2437\rangle
\end{aligned}
$$

Next, we will calculate the smoothed probability by multiplying the two messages together through an element-wise multiplication and normalizing the product.

$$
\begin{aligned}
& P(S_0|o_{0:2}) \\
& = \alpha\, f_{0:0} * b_{1:2} \\
& = \alpha\langle 0.818, 0.182\rangle * \langle 0.4593, 0.2437\rangle \\
& = \langle 0.894, 0.106\rangle
\end{aligned}
$$

4 Smoothing Derivations

The derivations for the smoothing formulas look even longer than those for the filtering formulas. Don't worry about understanding all the formulas now. I will explain the derivation step by step. However, once we break it down and look at it step by step, you will see that each step can be justified by one of the five reasons that I discussed previously.

For each step, please try to choose a justification yourself first. You will have five options as before: Bayes' rule, re-writing the expression, the chain rule or the product rule, the Markov assumption, and the sum rule.

4.1 Derivations Part 1

How can we derive the formula for $P(S_k|o_{0:(t-1)}), 0 \leq k < t-1$?

$$\begin{aligned}
&P(S_k|o_{0:(t-1)}) \\
&= P(S_k|o_{(k+1):(t-1)} \wedge o_{0:k}) \\
&= \alpha\, P(S_k|o_{0:k})\, P(o_{(k+1):(t-1)}|S_k \wedge o_{0:k}) \\
&= \alpha\, P(S_k|o_{0:k})\, P(o_{(k+1):(t-1)}|S_k) \\
&= \alpha f_{0:k} b_{(k+1):(t-1)}
\end{aligned}$$

Problem:
Step 1: What is the justification for the step below?

$$\begin{aligned}
&P(S_k|o_{0:(t-1)}) \\
&= P(S_k|o_{(k+1):(t-1)} \wedge o_{0:k})
\end{aligned}$$

(A) Bayes' rule
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

Solution: The correct answer is (B), re-writing the expression. We split the sequence of observations into two parts. The second term contains the observations up to k. The first term contains the observations from k+1 to the end.

$$\begin{aligned}
&P(S_k|o_{0:(t-1)}) \\
&= P(S_k|o_{(k+1):(t-1)} \wedge o_{0:k})
\end{aligned}$$

If you imagine that you are in day k, then the two sequences correspond to past observations from the start to day k, and future observations from day k+1 to the end.

Problem:
Step 2: What is the justification for the step below?

$$
\begin{aligned}
&= P(S_k | o_{(k+1):(t-1)} \land o_{0:k}) \\
&= \alpha\, P(S_k | o_{0:k})\, P(o_{(k+1):(t-1)} | S_k \land o_{0:k})
\end{aligned}
$$

(A) Bayes' rule
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

Solution: The correct answer is (A) Bayes's rule. It's easier to see this when you cross out $o_{0:k}$ since it appears in all three terms.

We switched the places of S_k and $o_{(k+1):(t-1)}$ using Bayes' rule. This is convenient since the observation sequence comes after the state on day k. It is natural to think about how the state on day k affects the observations in its future from day k+1 to the end.

$$
\begin{aligned}
&= P(S_k | o_{(k+1):(t-1)} \land \cancel{o_{0:k}}) \\
&= \alpha\, P(S_k | \cancel{o_{0:k}})\, P(o_{(k+1):(t-1)} | S_k \land \cancel{o_{0:k}})
\end{aligned}
$$

By the way, crossing out the last term doesn't mean that we can cancel it. I want you to disregard the last term so that it is easier to see the pattern in the remaining parts. This step is a version of the Bayes' rule where every term conditions on the same term. Here is a practice problem for you. Try proving the general equation yourself. You should be able to prove it with the basic rules of probability.

$$P(A|B \land C) = \alpha P(A|C) P(B|A \land C)$$

Problem:
Step 3: What is the justification for the step below?

$$
\begin{aligned}
&= \alpha\, P(S_k | o_{0:k})\, P(o_{(k+1):(t-1)} | S_k \land o_{0:k}) \\
&= \alpha\, P(S_k | o_{0:k})\, P(o_{(k+1):(t-1)} | S_k)
\end{aligned}
$$

(A) Bayes' rule

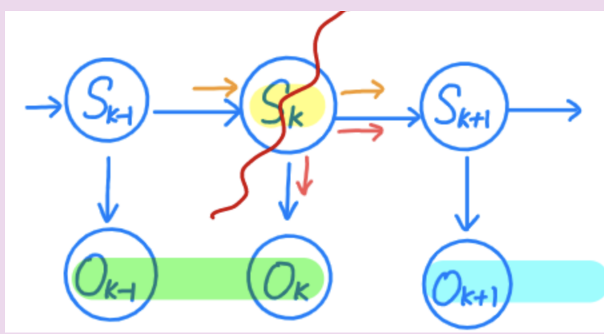
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

Solution: The correct answer is (D) The Markov assumption. The step removes $o_{0:k}$ from the second term.

The Markov assumption is basically a conditional independence relationship. In this case, given S_k, the state on day k, the future observations are independent of the past observations. That is, any observation up to day k is independent of any other observation from day k+1 onward, given the state S_k.

$$= \alpha\, P(S_k | o_{0:k})\, P(o_{(k+1):(t-1)} | S_k \wedge o_{0:k})$$
$$= \alpha\, P(S_k | o_{0:k})\, P(o_{(k+1):(t-1)} | S_k)$$

You can verify this relationship by applying d-separation on the Bayesian network. Suppose that we are in day k right now. The path between any past observation and any future observation must go through S_k. If we are considering O_k, then the two red arrows are both pointing away from S_k. If we are considering any other past observation, the two orange arrows are pointing in the same direction. Either way, if we observe S_k, it is as if we are cutting the chain at S_k, making the past observations independent of the future observations.



4.2 Derivations Part 2

Let's look at the second part of the derivations.

$$
\begin{aligned}
& P(o_{(k+1):(t-1)} | S_k) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} \wedge s_{(k+1)} | S_k) && (5) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)} \wedge S_k) * P(s_{(k+1)} | S_k) && (6) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k) && (7) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1)} \wedge o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k) && (8) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1)} | s_{(k+1)}) * P(o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k) && (9)
\end{aligned}
$$

In these steps, we derived the recursive formula for backward recursion. We are going backward in time. Given the probability of an observation sequence starting on day k+2, we can calculate the probability of an observation sequence starting on day k+1.

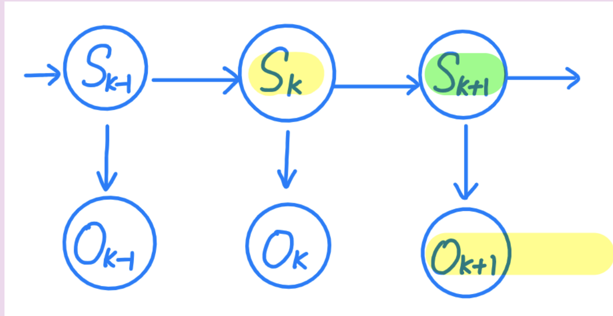
Problem:
Step 1: What is the justification for the step below?

$$
\begin{aligned}
& P(o_{(k+1):(t-1)} | S_k) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} \wedge s_{(k+1)} | S_k)
\end{aligned}
$$

(A) Bayes' rule
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

Solution: The correct answer is (E) The sum rule. We used the sum rule to introduce s_{k+1}. Introducing S_{k+1} is convenient since it is a bridge between the state on day k and the observations from day k+1 onward.

$$
\begin{aligned}
& P(o_{(k+1):(t-1)} | S_k) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} \wedge s_{(k+1)} | S_k)
\end{aligned}
$$



Problem:
Step 2: What is the justification for the step below?

$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} \wedge s_{(k+1)} | S_k)$$
$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)} \wedge S_k) P(s_{(k+1)} | S_k)$$

(A) Bayes' rule
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

Solution: The correct answer is (C) The chain rule or the product rule. This is easier to see if we cross out the last variable S_k in every term. We used the product rule to write the probability as a product of two probabilities.

$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} \wedge s_{(k+1)} | \cancel{S_k})$$
$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)} \wedge \cancel{S_k}) P(s_{(k+1)} | \cancel{S_k})$$

Problem:
Step 3: What is the justification for the step below?

$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)} \wedge S_k) P(s_{(k+1)} | S_k)$$
$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)}) P(s_{(k+1)} | S_k)$$

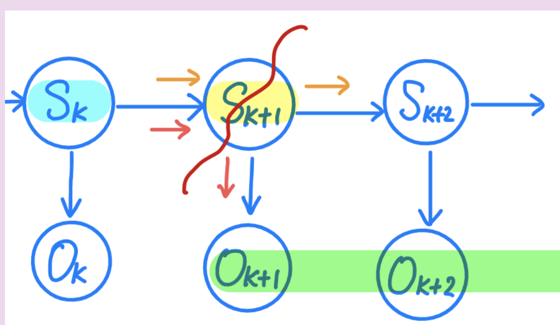
(A) Bayes' rule
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

Solution: The correct answer is (D) The Markov assumption. This step removes the value S_k from the first term.

This is another conditional independence relationship. Given the state on day k+1, S_{k+1}, all the future observations are independent of the past state on day k, S_k.

$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)} \wedge S_k) P(s_{(k+1)} | S_k)$$
$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)}) P(s_{(k+1)} | S_k)$$

You can verify this by applying d-separation on the Bayesian network. The path between S_k and any future observation has to go through S_{k+1}. Whether we are considering the observation on k+1 or any observation after that, the two arrows both point in the same direction. Therefore, if we observe S_{k+1}, it is as if we cut the chain at S_{k+1}, making the state on day k, S_k, independent of any future observation.



Problem:
Step 4: What is the justification for the step below?

$$= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k)$$
$$= \sum_{s_{(k+1)}} P(o_{(k+1)} \wedge o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k)$$

(A) Bayes' rule
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

Solution: The correct answer is (B), re-writing the expression.

$$
\begin{aligned}
&= \sum_{s_{(k+1)}} P(o_{(k+1):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1)} \land o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k)
\end{aligned}
$$

We split the sequence of observations from day k+1 to the end into two parts. The first term contains the observation on day k+1 only. The second term contains the observations from day k+2 to the end.

Problem:
Step 5: What is the justification for the step below?

$$
\begin{aligned}
&= \sum_{s_{(k+1)}} P(o_{(k+1)} \land o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k) \\
&= \sum_{s_{(k+1)}} P(o_{(k+1)} | s_{(k+1)}) * P(o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k)
\end{aligned}
$$

(A) Bayes' rule
(B) Re-writing the expression
(C) The chain/product rule
(D) The Markov assumption
(E) The sum rule

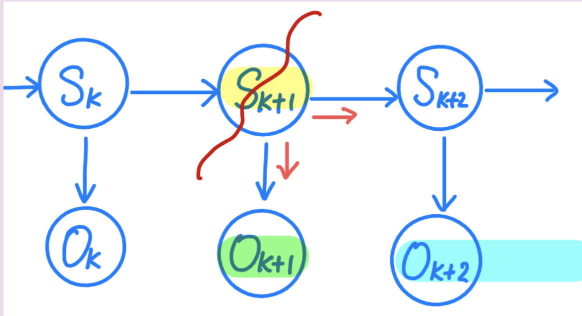
Solution: The correct answer is (D) The Markov assumption.

This is another conditional independence relationship. Given the state on day k+1, S_{k+1}, the observation on day k+1, O_{k+1} is independent of all the future observations from day k+2 to the end.

$$= \sum_{s_{(k+1)}} P(o_{(k+1)} \wedge o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k)$$

$$= \sum_{s_{(k+1)}} P(o_{(k+1)} | s_{(k+1)}) * P(o_{(k+2):(t-1)} | s_{(k+1)}) * P(s_{(k+1)} | S_k)$$

You can verify this by applying d-separation on the Bayesian network. The path between O_{k+1} and any future observation has to go through S_{k+1}. The two arrows are pointing away from S_{k+1}. If we observe S_{k+1}, it is as if we cut the chain at S_{k+1}, making the observation on day k+1, O_{k+1}, independent of any future observation.



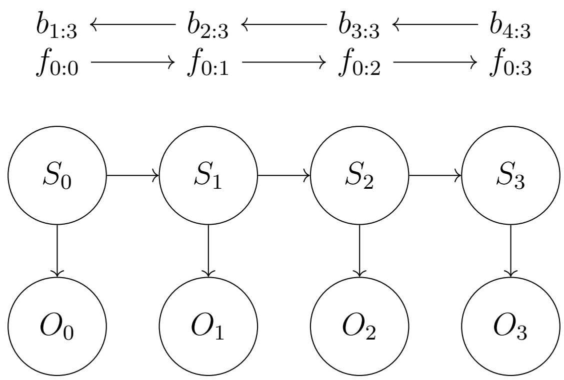
5 The Forward-Backward Algorithm

Now that we understood forward recursion and backward recursion, let's put them together in the forward-backward algorithm. I'll explain the algorithm using a hidden Markov model having 4 time steps. The algorithm works for a hidden Markov model with any finite number of time steps.

The purpose of the forward-backward algorithm is to calculate the smoothed probability for each time step. For our model, we want to calculate these four probabilities.

If we treat this model as a generic Bayesian network, we have to calculate each of the four probabilities separately, for example, by using the variable elimination algorithm. This approach is inefficient since there is no way for us to re-use any intermediate calculation results. For every probability, we have to start the process from scratch.

The forward-backward algorithm allows us to calculate the probabilities more efficiently through two passes through the network. Let's take a look.



$$P(S_0|o_{0:3}) = \alpha f_{0:0} b_{1:3} \quad (10)$$
$$P(S_1|o_{0:3}) = \alpha f_{0:1} b_{2:3} \quad (11)$$
$$P(S_2|o_{0:3}) = \alpha f_{0:2} b_{3:3} \quad (12)$$
$$P(S_3|o_{0:3}) = \alpha f_{0:3} b_{4:3} \quad (13)$$

We will first perform a forward pass using forward recursion. Start from time 0 and go forward in time. At each time step k, calculate the message $f_{0:k}$ and store the value. After the forward pass, we have four values from $f_{0:0}$ to $f_{0:3}$.

Next, we will perform a backward pass using backward recursion. Start from the last time step and go backward in time. At each time step k, we will calculate the message $b_{(k+1):(t-1)}$. Then, we can combine the stored message f and the message b to derive the smoothed probability at each time step.