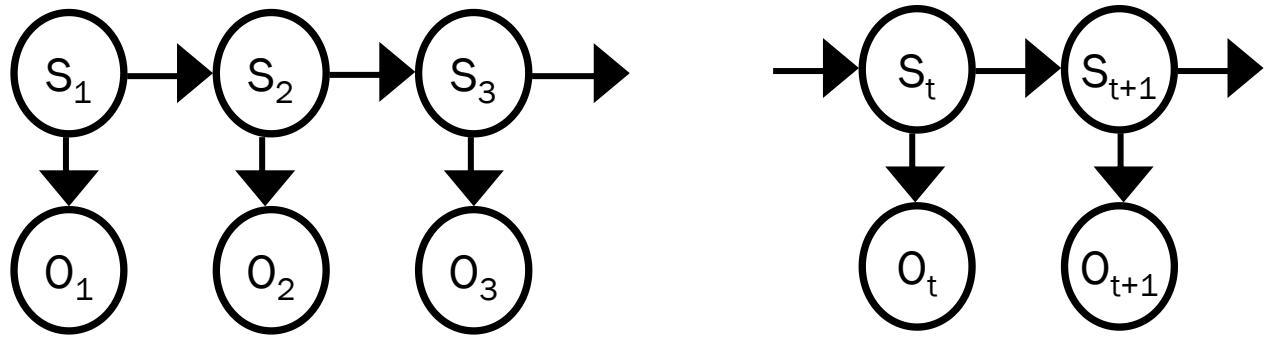
FUDAN SOE SUMMER SCHOOL INTRODUCTION TO AI

Probabilistic Reasoning and Decision Making

Day 11 – Hidden Markov Model (HMM)

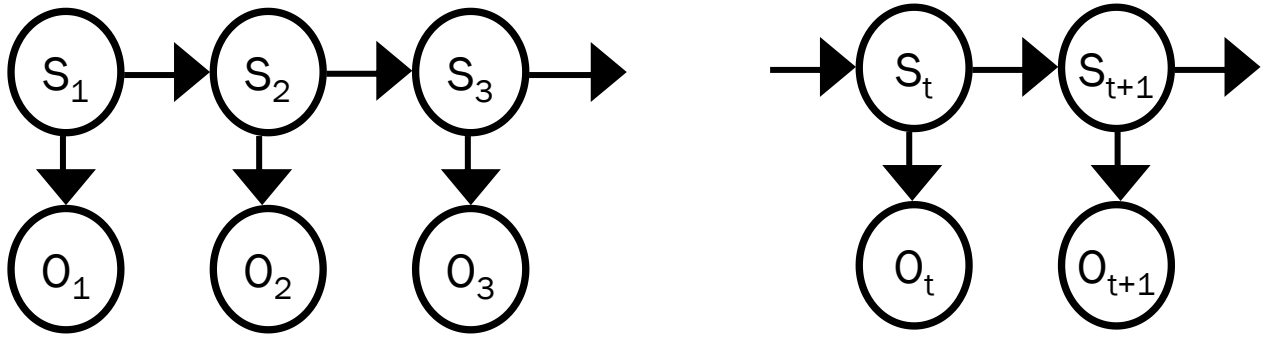
HMM Key Questions



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

1. How to compute $P(o_1, o_2, o_3, \ldots o_T)$
2. How to compute the most likely state sequence given observations
3. How to update beliefs for real-time monitoring
4. How to learn HMMs from data

How to compute $P(o_1, o_2, o_3, \ldots o_T)$ efficiently using the "forward algorithm"



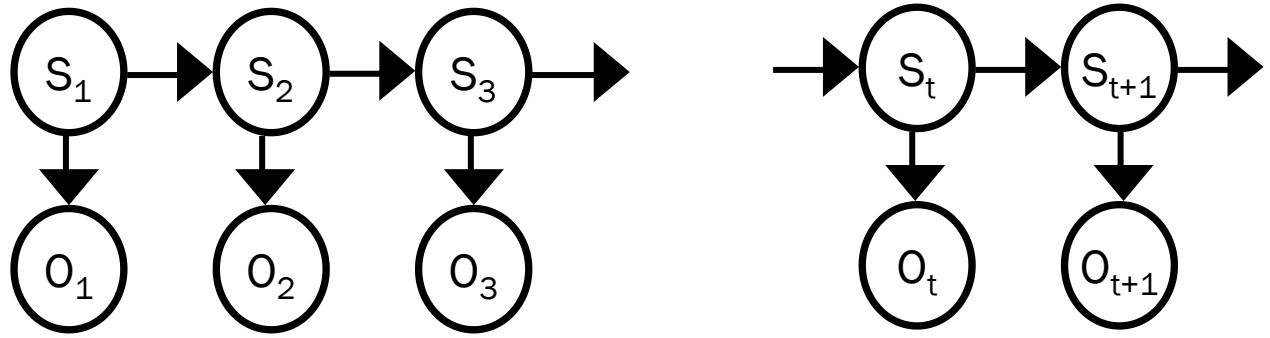
$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

Let $\alpha_{it} \leftarrow P(o_1, \ldots o_{t,}, s_t = i)$

$$P(o_1, o_2, o_3, \ldots o_T) = \sum_{i=1}^{n} P(o_1, \ldots o_T, s_T = i) = \sum_{i=1}^{n} \alpha_{iT}$$

$\alpha_{jt+1} =$

How to compute $P(o_1, o_2, o_3, \ldots o_T)$ efficiently using the "forward algorithm"



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

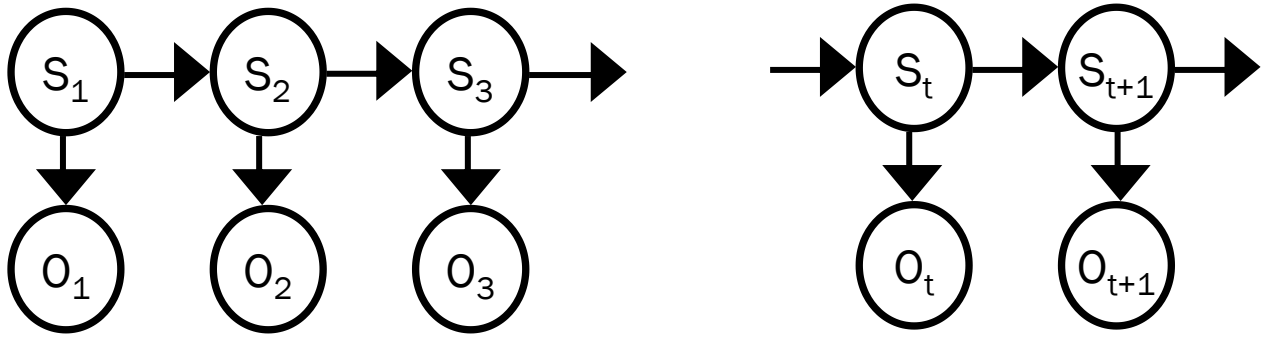
Let $\alpha_{it} \leftarrow P(o_1, \ldots o_{t,}, s_t = i)$

$$P(o_1, o_2, o_3, \ldots o_T) = \sum_{i=1}^{n} P(o_1, \ldots o_T, s_T = i) = \sum_{i=1}^{n} \alpha_{iT}$$

$$\alpha_{jt+1} = \sum_{i=1}^{n} \alpha_{it} a_{ij} b_{jo_{t+1}}$$

$$\alpha_{i1} =$$

How to compute $P(o_1, o_2, o_3, \ldots o_T)$ efficiently using the "forward algorithm"



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

Let $\alpha_{it} \leftarrow P(o_1, \ldots o_{t,}, s_t = i)$

$$P(o_1, o_2, o_3, \ldots o_T) = \sum_{i=1}^{n} P(o_1, \ldots o_T, s_T = i) = \sum_{i=1}^{n} \alpha_{iT}$$

$$\alpha_{jt+1} = \sum_{i=1}^{n} \alpha_{it} a_{ij} b_{jo_{t+1}}$$

$$\alpha_{i1} = \pi_i b_{io_1}$$

Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

Use the forward algorithm to find the probability that, if the DJ plays only two songs, the audience will dance to both of them.

Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

to two types: slow
robability that the

rds is $\frac{1}{3}$. If she plays
t the selection of ev-
mediately before it.

dance or not to
. If she plays a
er the audience

$a_{ij} = f \begin{bmatrix} f & s \\ \frac{2}{3} & \frac{1}{3} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix}$

$b_{i_{0_k}} = f \begin{bmatrix} d & nd \\ \frac{4}{5} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$

$\pi_f = \frac{3}{5}$ $\pi_s = \frac{2}{5}$

t

1	2
$\frac{12}{25}$	.376
$\frac{1}{5}$	

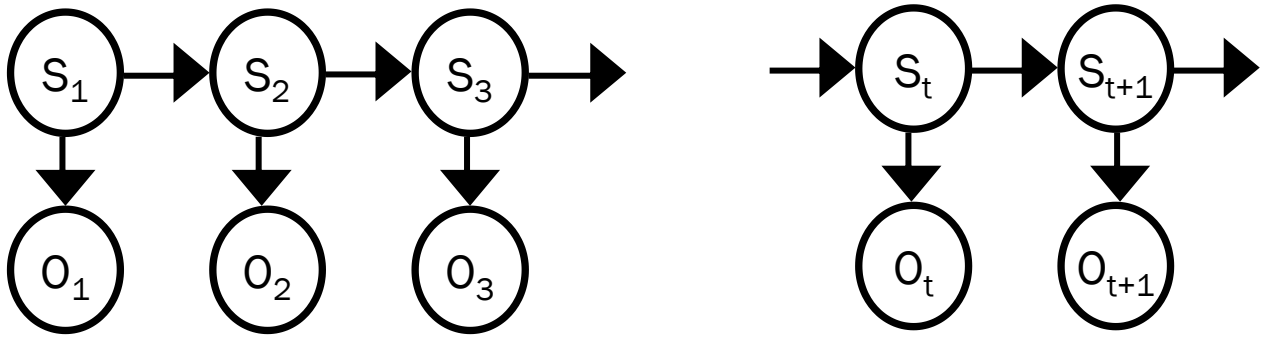
Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

Use the forward algorithm and the probabilities you computed for the previous problem to find the probability that the audience danced during the first two songs, did not dance during the third song, and the third song was fast.

How to compute $\arg\max_{\vec{s}}\left(P(s_1 \ldots s_T | o_1 \ldots o_T)\right)$

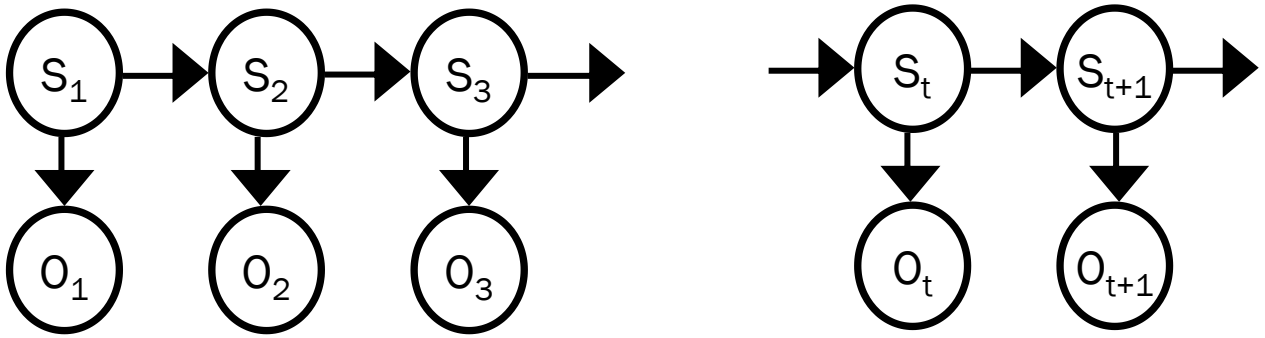


$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

Is the following equality true? (and why or why not?) A. Yes B. No

$$\arg\max_{\vec{s}}\left(P(s_1 \ldots s_T | o_1 \ldots o_T)\right) = \arg\max_{\vec{s}}\left(P(s_1 \ldots s_T, o_1 \ldots o_T)\right)$$

How to compute $\arg\max_{\vec{s}}\left(P(s_1 \dots s_T | o_1 \dots o_T)\right)$



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

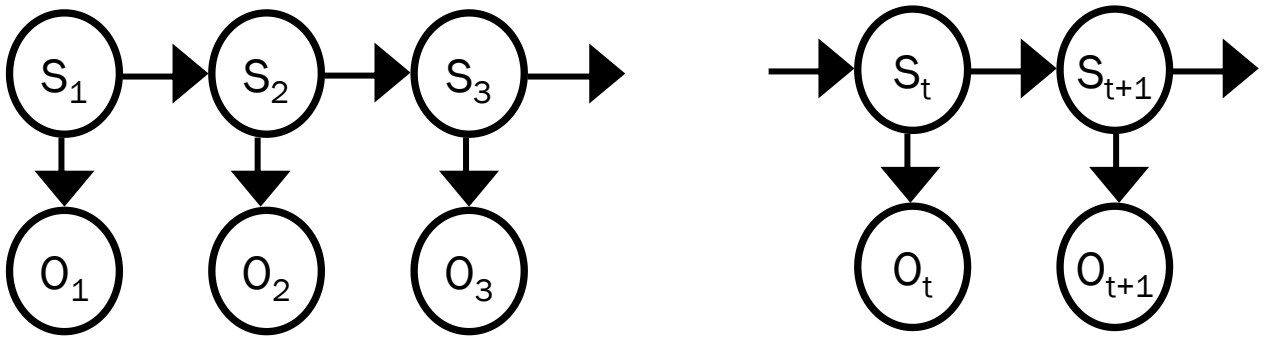
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}}\left(P(s_1 \dots s_T, o_1 \dots o_T)\right)$$

Define $\ell_{it}{}^* = \max_{s_1 \dots s_{t-1}} \log P(s_1 \dots s_{t-1}, s_t = i, o_1 \dots o_t)$

What does $\ell_{it}{}^*$ mean, in English?

A. The log probability of the most likely set of states $s_1 \dots s_{t-1}, s_t$ given observations $o_1 \dots o_t$
B. The state that is most likely at time t
C. The log probability of the most likely set of states $s_1 \dots s_{t-1}, s_t$ that ends in state $s_t = i$ and explains observations $o_1 \dots o_t$
D. The log probability of only state $s_t = i$ that explains observations $o_1 \dots o_t$

How to compute $\arg\max_{\vec{s}}\left(P(s_1 \dots s_T | o_1 \dots o_T)\right)$



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

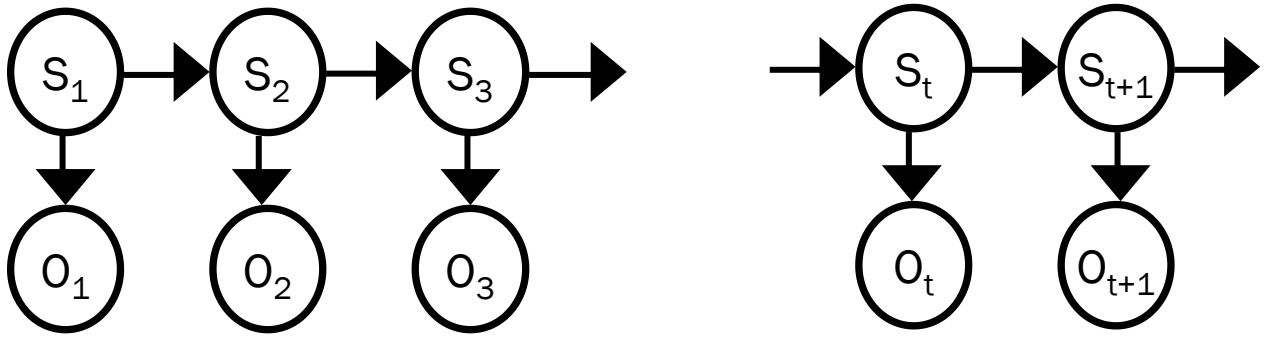
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}}\left(P(s_1 \dots s_T, o_1 \dots o_T)\right)$$

Define $\ell_{it}{}^* = \max_{s_1 \dots s_{t-1}} \log P(s_1 \dots s_{t-1}, s_t = i, o_1 \dots o_t)$

Viterbi algorithm:

1. Fill in an n x T table with the values of $\ell_{it}{}^*$ from left to right. Fill in a second table to keep track of best i's at each time t at the same time.
2. Backtrack through the second table from right to left to determine which state *i* at time *t* led to the best outcome at time *t+1*

	1	2	…	t-1	t	t+1	…	T
1								
…								
n								

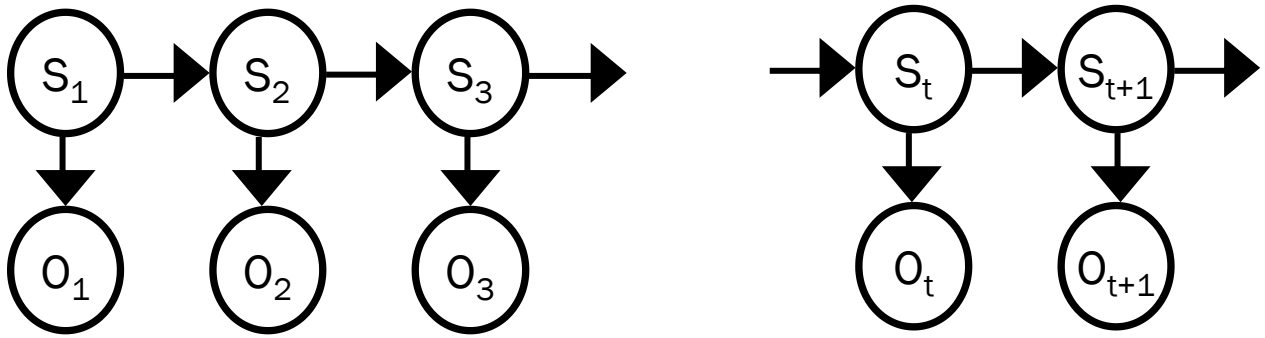


$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}} \left(P(s_1 \dots s_T, o_1 \dots o_T) \right)$$

Define $\ell_{it}{}^* = \max_{s_1 \dots s_{t-1}} \log P(s_1 \dots s_{t-1}, s_t = i, o_1 \dots o_t)$

$\ell_{j,t+1}{}^* =$



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

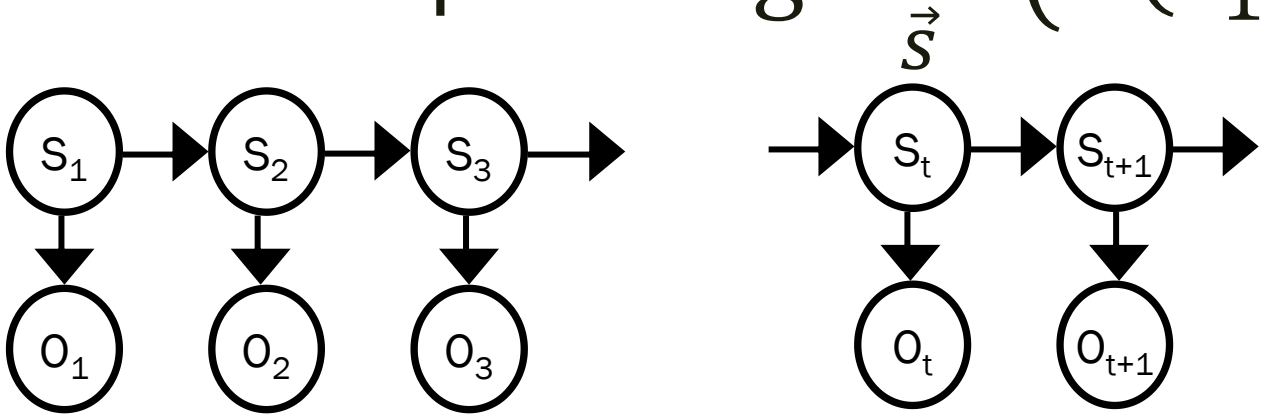
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}} \left(P(s_1 \dots s_T, o_1 \dots o_T) \right)$$

Define $\ell_{it}{}^* = \max_{s_1 \dots s_{t-1}} \log P(s_1 \dots s_{t-1}, s_t = i, o_1 \dots o_t)$

$$\ell_{j,t+1}{}^* = \max_i \{ \ell_{it}{}^* + \log a_{ij} \} + \log b_{jo_{t+1}}$$

$$\ell_{i1}{}^*$$

How to compute $\arg\max_{\vec{s}}\left(P(s_1 \dots s_T | o_1 \dots o_T)\right)$



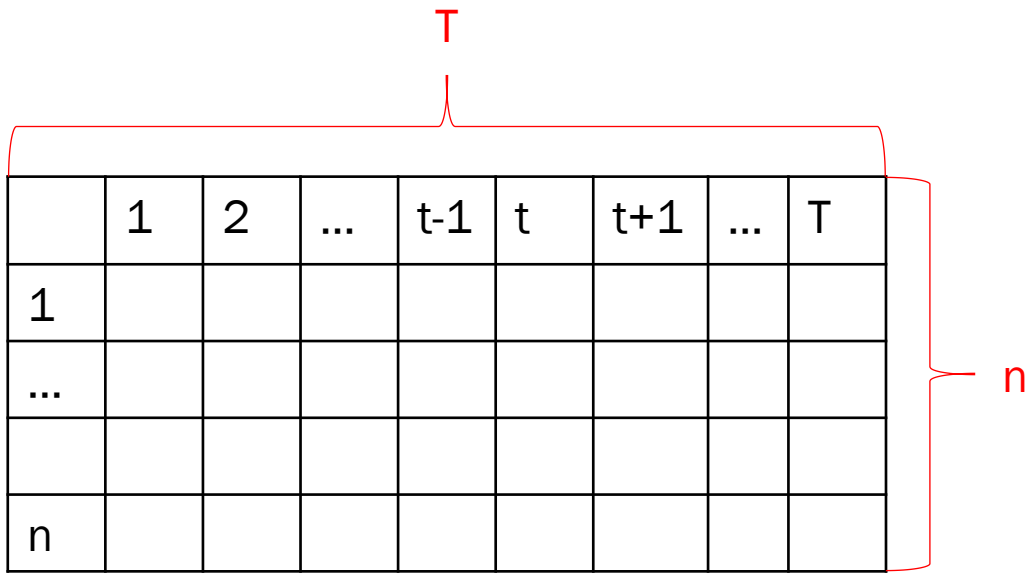
$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{ik} = P(O_t = k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

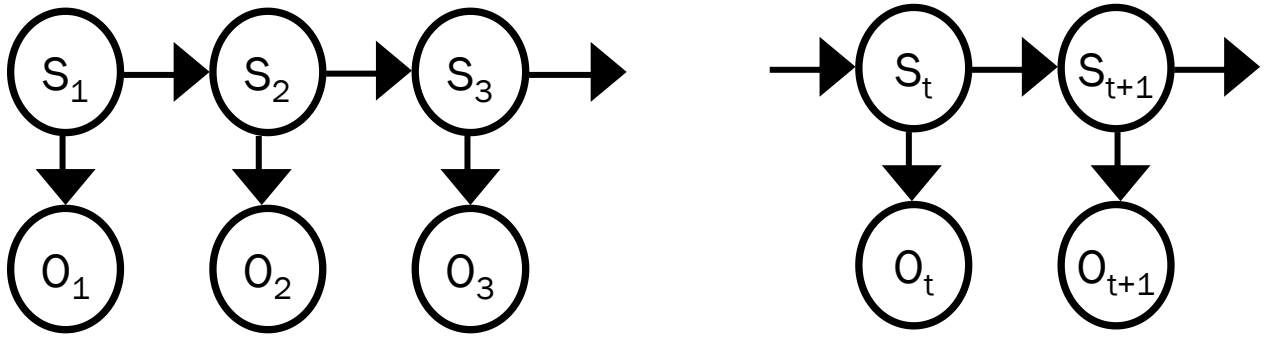
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}}\left(P(s_1 \dots s_T, o_1 \dots o_T)\right)$$

Define $\ell_{it}{}^* = \max_{s_1 \dots s_{t-1}} \log P(s_1 \dots s_{t-1}, s_t = i, o_1 \dots o_t)$

$$\ell_{i1}{}^* = \log[\pi_i b_{io_1}]$$

$$\ell_{jt+1}{}^* = \max_i \{\ell_{it}{}^* + \log a_{ij}\} + \log b_{jo_{t+1}}$$





$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

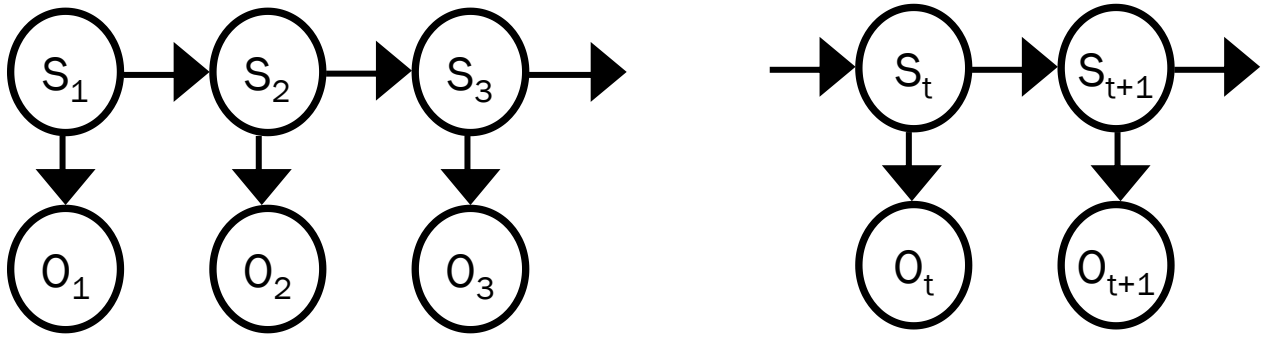
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}} \left(P(s_1 \dots s_T, o_1 \dots o_T) \right)$$

Define $\ell_{it}{}^* = \max_{s_1 \dots s_{t-1}} \log P(s_1 \dots s_{t-1}, s_t = i, o_1 \dots o_t)$

	1	2	…	t-1	t	t+1	…	T
1								
…								
n								

Viterbi algorithm:

1. Fill in an n x T table with the values of $\ell_{it}{}^*$ from left to right. Fill in a second table to keep track of best i's at each time t at the same time.
2. Backtrack through the second table from right to left to determine which state *i* at time *t* led to the best outcome at time *t+1*



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}} \left(P(s_1 \dots s_T, o_1 \dots o_T) \right)$$

Define $\ell_{it}{}^* = \max_{s_1 \dots s_{t-1}} \log P(s_1 \dots s_{t-1}, s_t = i, o_1 \dots o_t)$

	1	2	…	t-1	t	t+1	…	T
1								
…								
n								

Viterbi algorithm:

1. Fill in an n x T table with the values of $\ell_{it}{}^*$ from left to right. Fill in a second table to keep track of best i's at each time t at the same time.
2. Backtrack through the second table from right to left to determine which state *i* at time *t* led to the best outcome at time *t+1*

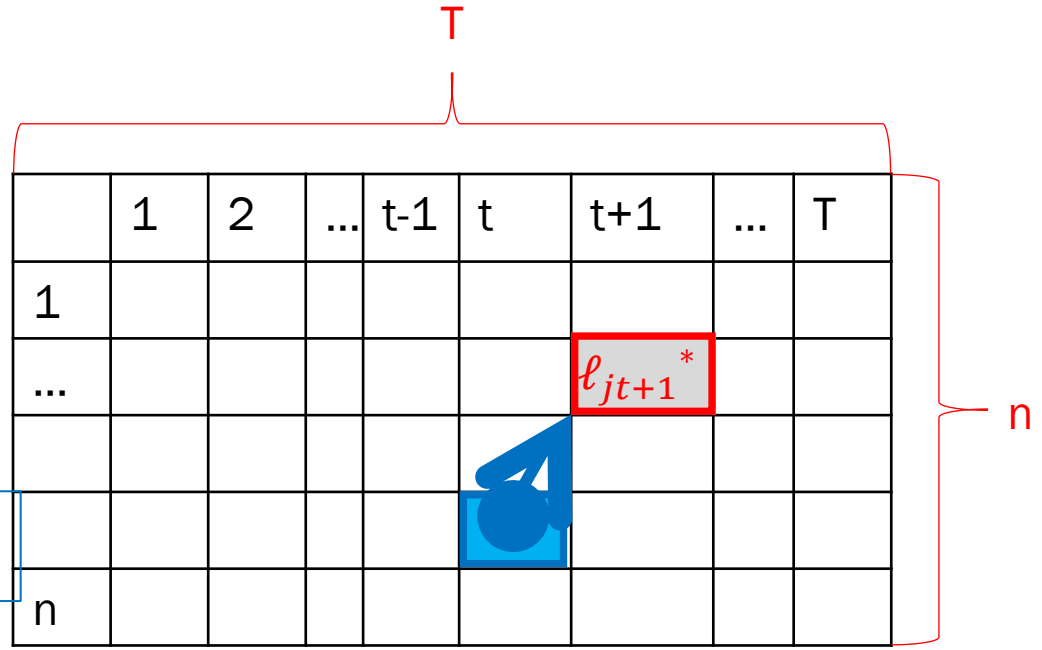
Viterbi alg. to find $(s_1^*, s_2^*, \ldots, s_T^*) = \arg\max_{\vec{s}}\left(P(s_1 \ldots s_T, o_1 \ldots o_T)\right)$

Define $\ell_{it}^* = \max_{s_1 \ldots s_{t-1}} \log P(s_1 \ldots s_{t-1}, s_t = i, o_1 \ldots o_t)$

$$\ell_{i1}^* = \log[\pi_i b_{io_1}]$$

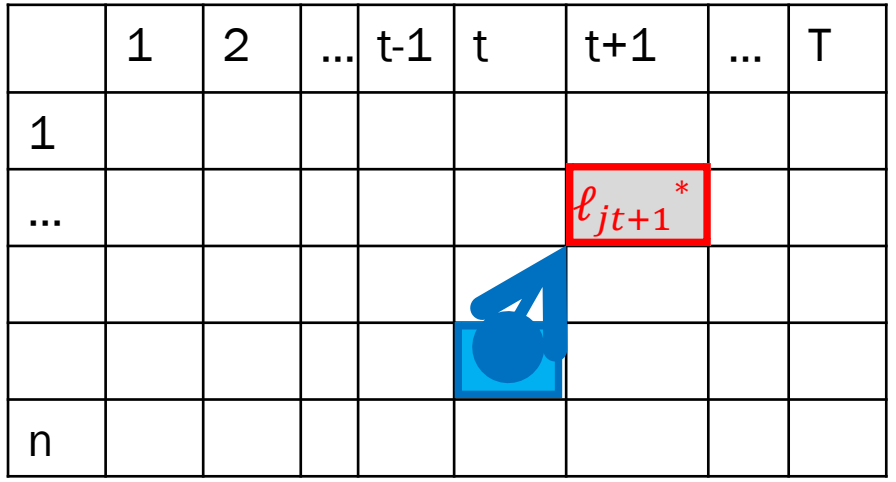
$$\ell_{jt+1}^* = \max_i\{\ell_{it}^* + \log a_{ij}\} + \log b_{jo_{t+1}}$$

$\operatorname{argmax}_i\{\ell_{it}^* + \log a_{ij}\}$



Consider the value in the shaded box. How does its value depend on the values from the previous column (at time t)?

A. It is a weighted sum of all of the values from the previous column
B. It uses only one of the values from the previous column
C. It does not use any of the values from the previous column



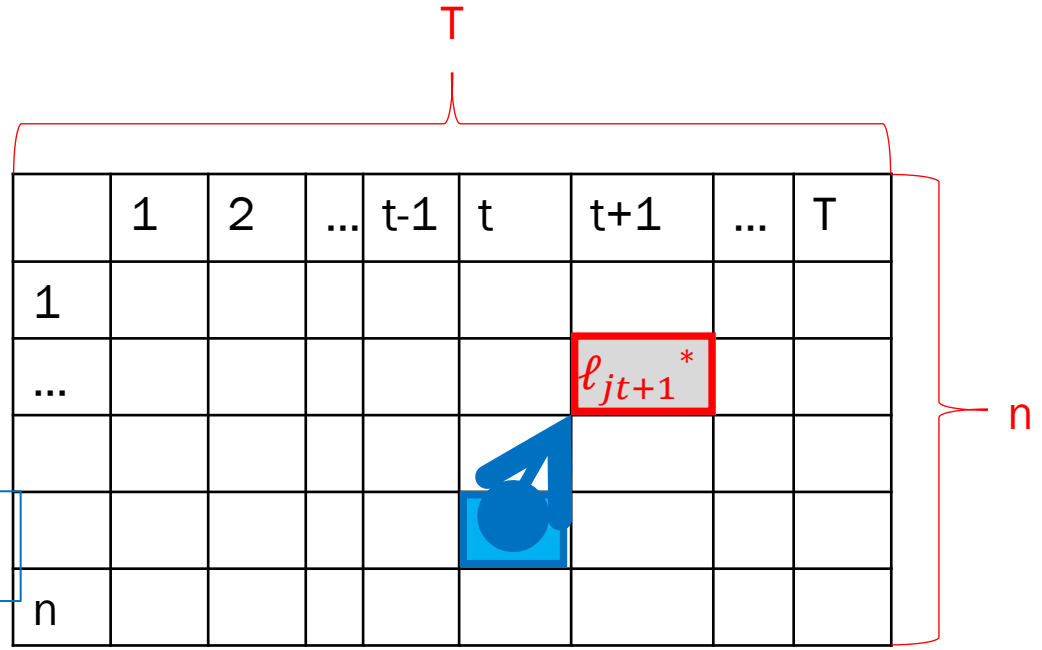
Viterbi alg. to find $(s_1^*, s_2^*, \ldots, s_T^*) = \arg\max_{\vec{s}}\left(P(s_1 \ldots s_T, o_1 \ldots o_T)\right)$

Define $\ell_{it}^* = \max_{s_1 \ldots s_{t-1}} \log P(s_1 \ldots s_{t-1}, s_t = i, o_1 \ldots o_t)$

$$\ell_{i1}^* = \log[\pi_i b_{io_1}]$$

$$\ell_{jt+1}^* = \max_i\{\ell_{it}^* + \log a_{ij}\} + \log b_{jo_{t+1}}$$

$$\Phi_{jt+1} \leftarrow \operatorname{argmax}_i\{\ell_{it}^* + \log a_{ij}\}$$



	1	2	...	t-1	t	t+1	...	T
1								
...						Φ_{jt+1}		
n								

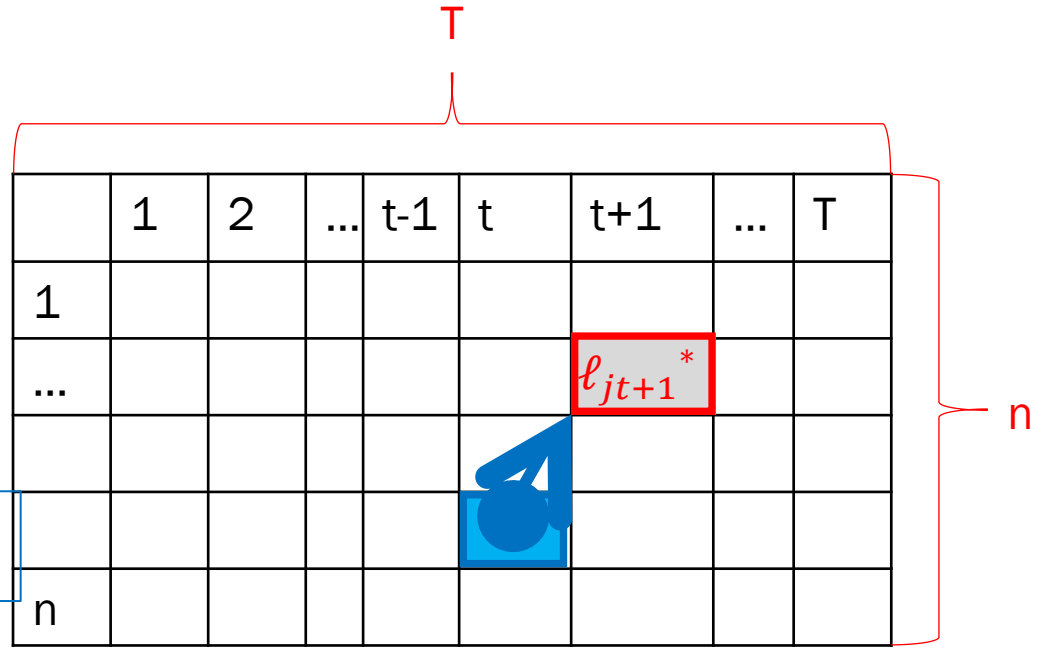
Viterbi alg. to find $(s_1{}^*, s_2{}^*, \ldots, s_T{}^*) = \arg\max_{\vec{s}}\left(P(s_1 \ldots s_T, o_1 \ldots o_T)\right)$

Define $\ell_{it}{}^* = \max_{s_1 \ldots s_{t-1}} \log P(s_1 \ldots s_{t-1}, s_t = i, o_1 \ldots o_t)$

$$\ell_{i1}{}^* = \log[\pi_i b_{io_1}]$$

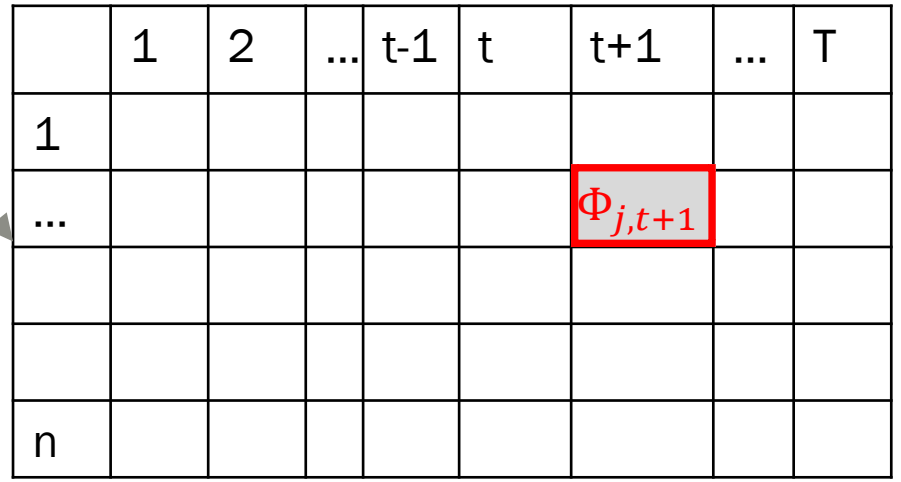
$$\ell_{jt+1}{}^* = \max_i\{\ell_{it}{}^* + \log a_{ij}\} + \log b_{jo_{t+1}}$$

Define $\Phi_{jt+1} = \operatorname{argmax}_i\{\ell_{it}{}^* + \log a_{ij}\}$



TWO tables:

1. Table of $\ell_{i1}{}^*$, filled in from left to right
2. Table of $\Phi_{i,t}$, filled in from left to right at the same time (column 1 will be empty)



Viterbi alg. to find $(s_1{}^*, s_2{}^*, \ldots, s_T{}^*) = \arg\max_{\vec{s}}\left(P(s_1 \ldots s_T, o_1 \ldots o_T)\right)$

Define $\ell_{it}{}^* = \max_{s_1 \ldots s_{t-1}} \log P(s_1 \ldots s_{t-1}, s_t = i, o_1 \ldots o_t)$

	1	2	…	t-1	t	t+1	…	T
1								
…						$\ell_{jt+1}{}^*$		
n								

Define $\Phi_{j,t+1} = \underset{i}{\text{argmax}}\{\ell_{it}{}^* + \log a_{ij}\}$

	1	2	…	t-1	t	t+1	…	T
1								
…						$\Phi_{j,t+1}$		
n								

Once both tables are filled in with values, how can you find $(s_1{}^*, s_2{}^*, \ldots, s_T{}^*)$?

A. Tracing a path from left to right (forward: starting at column 1) in the $\ell_{it}{}^*$ table
B. Tracing a path from left to right (forward: starting at column 1) in the $\Phi_{j,t}$ table
C. Tracing a path from right to left (backward: starting at column T) in the $\ell_{it}{}^*$ table
D. Tracing a path from right to left (backward: starting at column T) in the $\Phi_{j,t}$ table

Viterbi alg. to find $(s_1^*, s_2^*, \ldots, s_T^*) = \arg\max_{\vec{s}} \left(P(s_1 \ldots s_T, o_1 \ldots o_T) \right)$

Define $\ell_{it}^* = \max_{s_1 \ldots s_{t-1}} \log P(s_1 \ldots s_{t-1}, s_t = i, o_1 \ldots o_t)$

	1	2	…	t-1	t	t+1	…	T
1								
…						ℓ_{jt+1}^*		
n								

Define $\Phi_{j,t+1} = \underset{i}{\text{argmax}} \{ \ell_{it}^* + \log a_{ij} \}$

	1	2	…	t-1	t	t+1	…	T
1								
…						$\Phi_{j,t+1}$		
n								

To recover $(s_1^*, s_2^*, \ldots, s_T^*)$:

Viterbi alg. to find $(s_1^*, s_2^*, \ldots, s_T^*) = \arg\max_{\vec{s}} \left(P(s_1 \ldots s_T, o_1 \ldots o_T) \right)$

Define $\ell_{it}^* = \max_{s_1 \ldots s_{t-1}} \log P(s_1 \ldots s_{t-1}, s_t = i, o_1 \ldots o_t)$

	1	2	…	t-1	t	t+1	…	T
1								
…						ℓ_{jt+1}^*		
n								

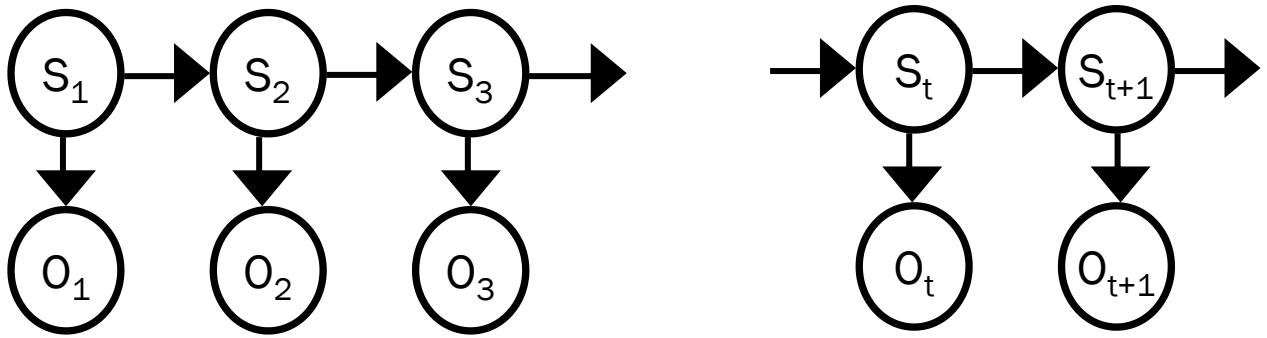
Define $\Phi_{j,t+1} = \operatorname{argmax}_i \{\ell_{it}^* + \log a_{ij}\}$

	1	2	…	t-1	t	t+1	…	T
1								
…						$\Phi_{j,t+1}$		
n								

Complete Viterbi algorithm:

1. Create two n x T tables
2. Fill in both tables from leftmost column (t=1) to rightmost column (t=T) [using recursive formula for ℓ_{jt+1}^* and $\Phi_{j,t+1}$
3. Recover path:
    1. Let $s_T^* = \operatorname{argmax}_i \ell_{it}^*$
    2. For t=T-1 to 1, $s_t^* = \Phi_{s_{t+1}^*, t+1}$

HMM Key Questions



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

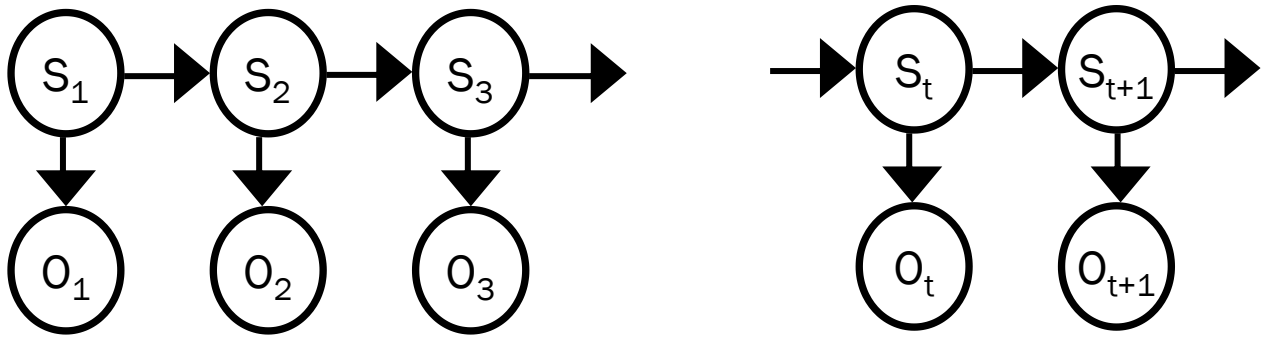
$S_t \in \{1,2,\dots,n\}$
$O_t \in \{1,2,\dots,m\}$

1. How to compute $P(o_1, o_2, o_3, \dots o_T)$ ← done
2. How to compute the most likely state sequence given observations ← done

3. How to update beliefs for real-time monitoring

4. How to learn HMMs from data

Calculating $P(S_t = j|o_1, \dots, o_t)$

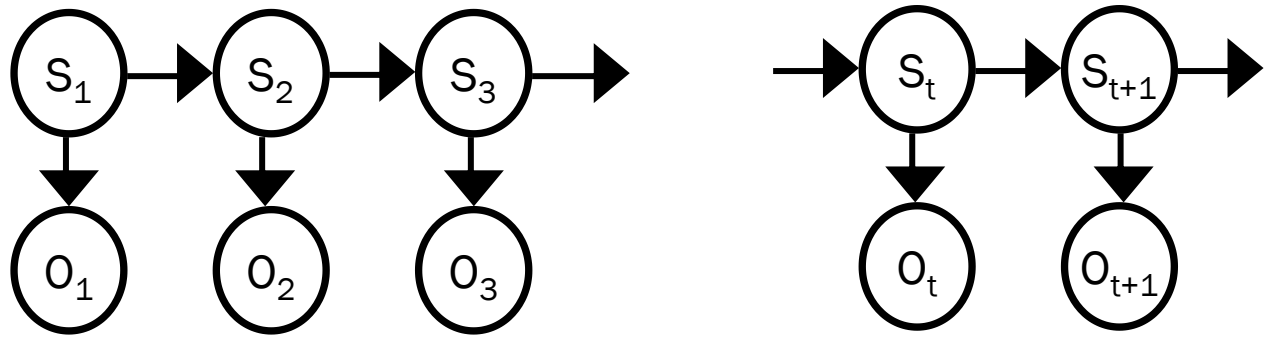


$a_{ij} = P(S_t = j|S_{t-1} = i)$
$b_{io_k} = P(O_t = o_k|S_t = i)$
$\pi_i = P(S_1 = i)$

$S_t \in \{1,2, \dots, n\}$
$O_t \in \{1,2, \dots, m\}$

True (A) or False (B): $P(S_t = j|o_1, \dots, o_t) = P(S_t = j|o_t)$

Calculating $P(S_t = j|o_1, \dots, o_t)$

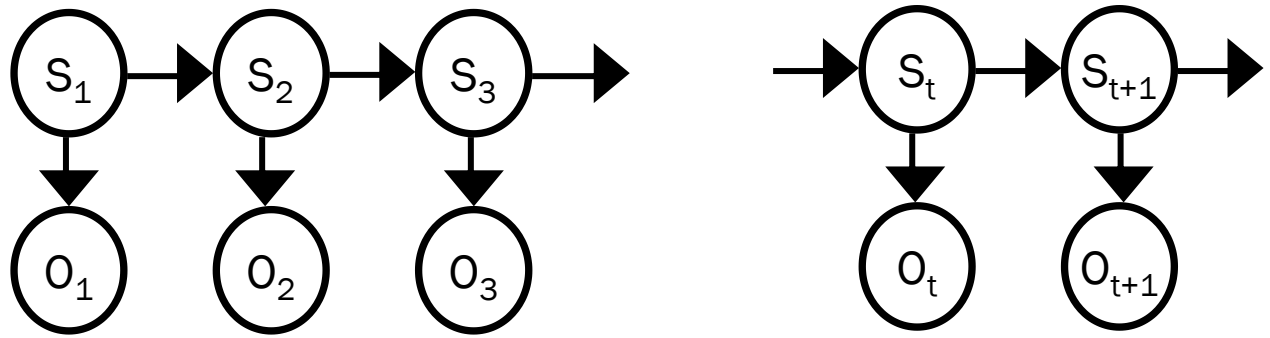


$$a_{ij} = P(S_t = j|S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k|S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$S_t \in \{1,2, \dots, n\}$
$O_t \in \{1,2, \dots, m\}$

$q_{j,t} = P(S_t = j|o_1, \dots, o_{t-1}, o_t)$

Calculating $P(S_t = j|o_1, \dots, o_t)$

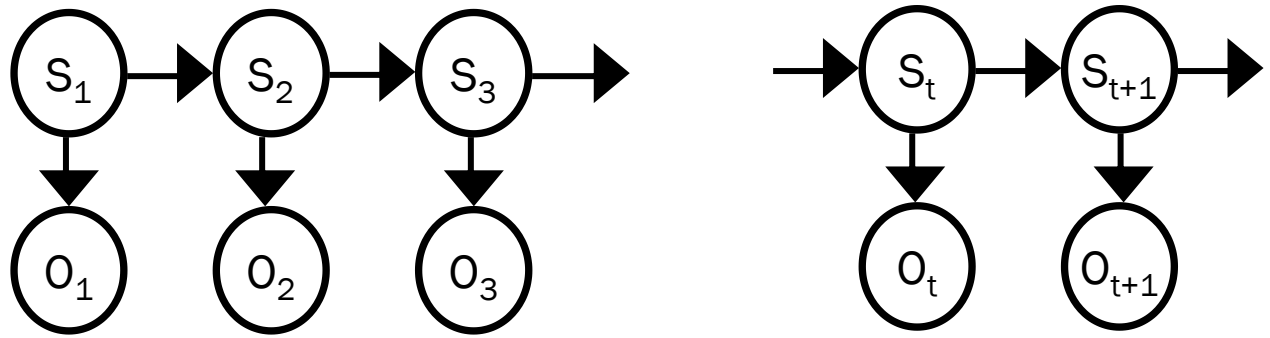


$$a_{ij} = P(S_t = j|S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k|S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$S_t \in \{1,2, \dots, n\}$
$O_t \in \{1,2, \dots, m\}$

$q_{j,t} = P(S_t = j|o_1, \dots, o_{t-1}, o_t)$

HMM Key Questions

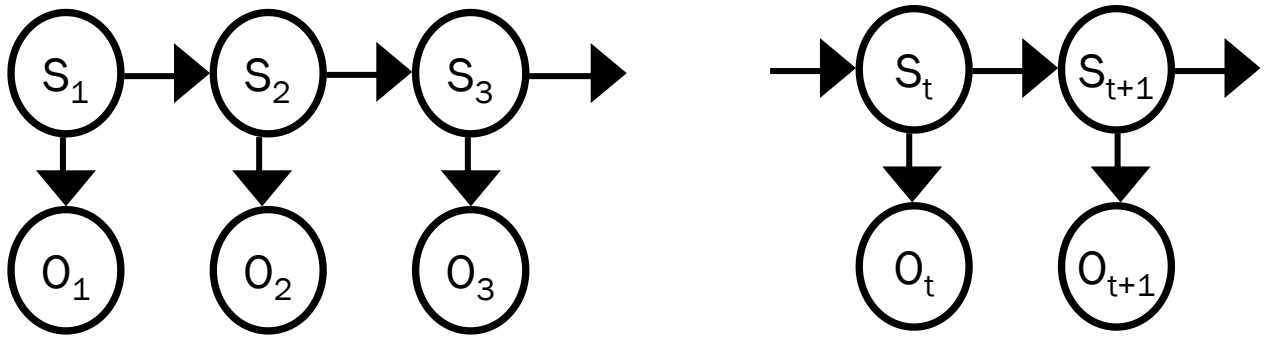


$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{ik} = P(O_t = k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$S_t \in \{1,2,\dots,n\}$
$O_t \in \{1,2,\dots,m\}$

1. How to compute $P(o_1, o_2, o_3, \dots o_T)$ ← Done
2. How to compute the most likely state sequence given observations ← Done
3. How to update beliefs for real-time monitoring ← Done
4. How to learn HMMs from data

Learning in HMMs

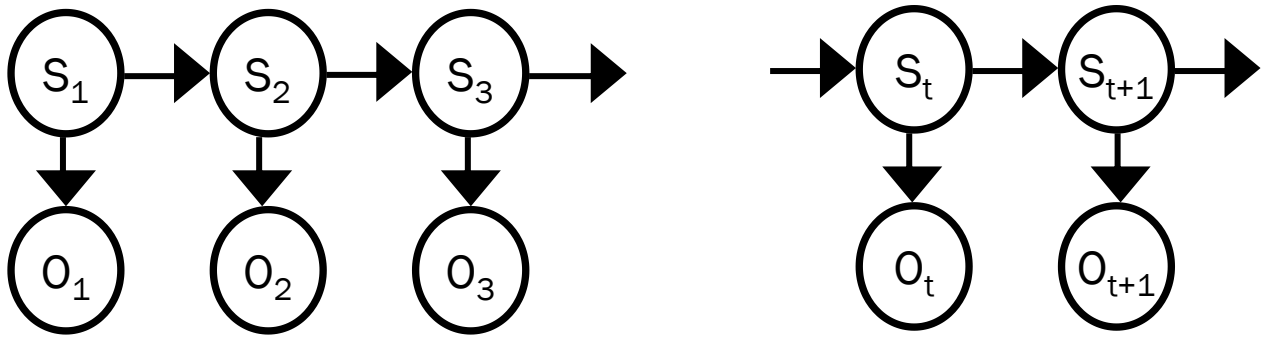


$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{ik} = P(O_t = k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$S_t \in \{1,2,\dots,n\}$
$O_t \in \{1,2,\dots,m\}$

- Given: Sequence(s) of observations: $\{o_1, o_2, \dots, o_T\}$
- Goal: Estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(o_1, o_2, o_3, \dots o_T)$
- Assume: Known and fixed n
- Approach?

Learning in HMMs



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{ik} = P(O_t = k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$S_t \in \{1,2,\dots,n\}$
$O_t \in \{1,2,\dots,m\}$

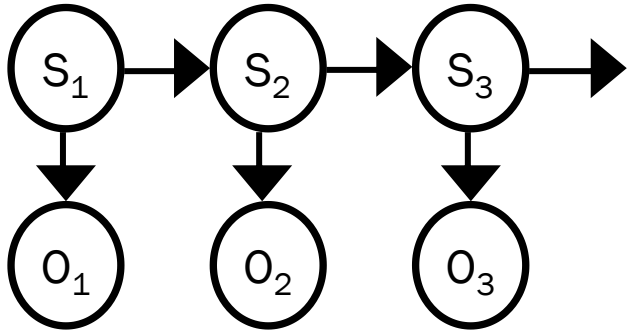
$$\pi_i \leftarrow P(S_1 = i | o_1, o_2, \dots, o_T)$$

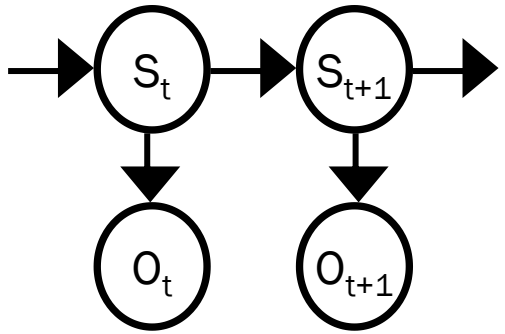
$$a_{ij} \leftarrow \frac{\sum_{t=1}^{T} P(S_t = i, S_{t+1} = j | o_1, o_2, \dots, o_T)}{\sum_{t=1}^{T} P(S_t = i | o_1, o_2, \dots, o_T)}$$

$$b_{ik} \leftarrow \frac{\sum_{t=1}^{T} I(o_t, k) P(S_t = i | o_1, o_2, \dots, o_T)}{\sum_{t=1}^{T} P(S_t = i | o_1, o_2, \dots, o_T)}$$

Explain how these update rules are just standard EM applied to HMMs.

How to compute $P(S_t = i \mid o_1, o_2, \ldots, o_T)$, etc





$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{ik} = P(O_t = k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$S_t \in \{1,2, \ldots, n\}$
$O_t \in \{1,2, \ldots, m\}$

$$\pi_i \leftarrow P(S_1 = i | o_1, o_2, \ldots, o_T)$$

$$a_{ij} \leftarrow \frac{\sum_{t=1}^{T} P(S_t = i, S_{t+1} = j | o_1, o_2, \ldots, o_T)}{\sum_{t=1}^{T} P(S_t = i \mid o_1, o_2, \ldots, o_T)}$$

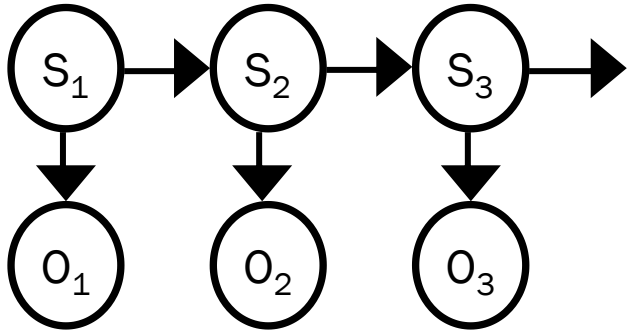
$$b_{ik} \leftarrow \frac{\sum_{t=1}^{T} I(o_t, k) P(S_t = i | o_1, o_2, \ldots, o_T)}{\sum_{t=1}^{T} P(S_t = i \mid o_1, o_2, \ldots, o_T)}$$

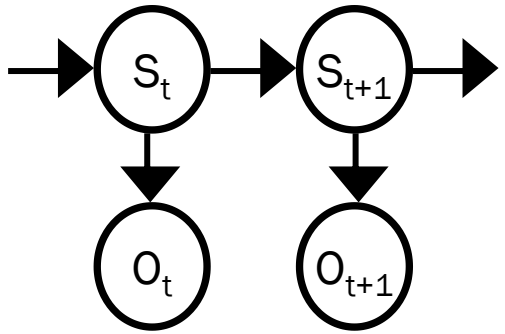
Recall $\alpha_{it} \leftarrow P(o_1, \ldots o_{t,}, S_t = i)$

$$\alpha_{i1} = \pi_i b_{io_1}$$
$$\alpha_{jt+1} = \sum_{i=1}^{n} \alpha_{it} a_{ij} b_{jo_{t+1}}$$

True (A) or False (B): $P(S_t = i \mid o_1, o_2, \ldots, o_T) = \frac{\alpha_{it}}{P(o_1, o_2, \ldots, o_T)}$

How to compute $P(S_t = i \mid o_1, o_2, \ldots, o_T)$, etc





$a_{ij} = P(S_t = j | S_{t-1} = i)$
$b_{ik} = P(O_t = k | S_t = i)$
$\pi_i = P(S_1 = i)$

$S_t \in \{1,2, \ldots, n\}$
$O_t \in \{1,2, \ldots, m\}$

Recall $\alpha_{it} \leftarrow P(o_1, \ldots o_{t,}, S_t = i)$

$\alpha_{i1} = \pi_i b_{io_1}$

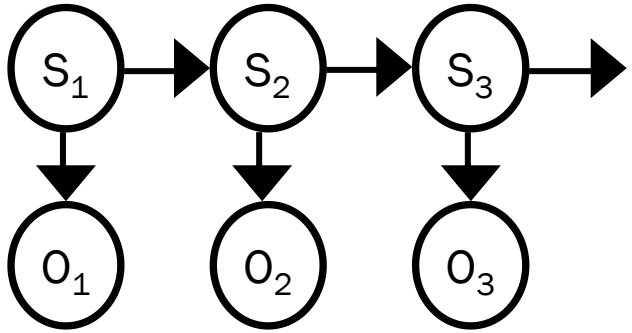
$\alpha_{jt+1} = \sum_{i=1}^{n} \alpha_{it} a_{ij} b_{jo_{t+1}}$

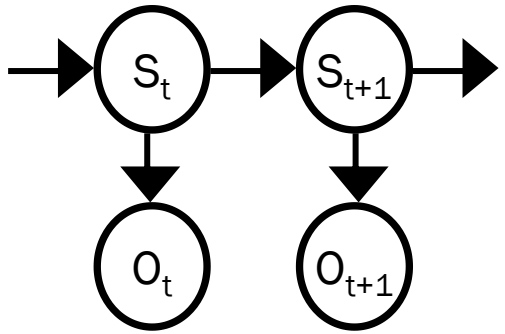
$\beta_{it} \leftarrow P(o_{t+1}, \ldots o_T \mid S_t = i)$

Which of the following is a possible base case in calculating β_{it}?

A. β_{11}
B. β_{i1}
C. β_{iT}
D. β_{nt}

How to compute $P(S_t = i \mid o_1, o_2, \ldots, o_T)$, etc





$a_{ij} = P(S_t = j | S_{t-1} = i)$
$b_{ik} = P(O_t = k | S_t = i)$
$\pi_i = P(S_1 = i)$

$S_t \in \{1,2, \ldots, n\}$
$O_t \in \{1,2, \ldots, m\}$

Recall $\alpha_{it} \leftarrow P(o_1, \ldots o_{t,}, S_t = i)$

$\alpha_{i1} = \pi_i b_{io_1}$

$\alpha_{jt+1} = \sum_{i=1}^{n} \alpha_{it} a_{ij} b_{jo_{t+1}}$

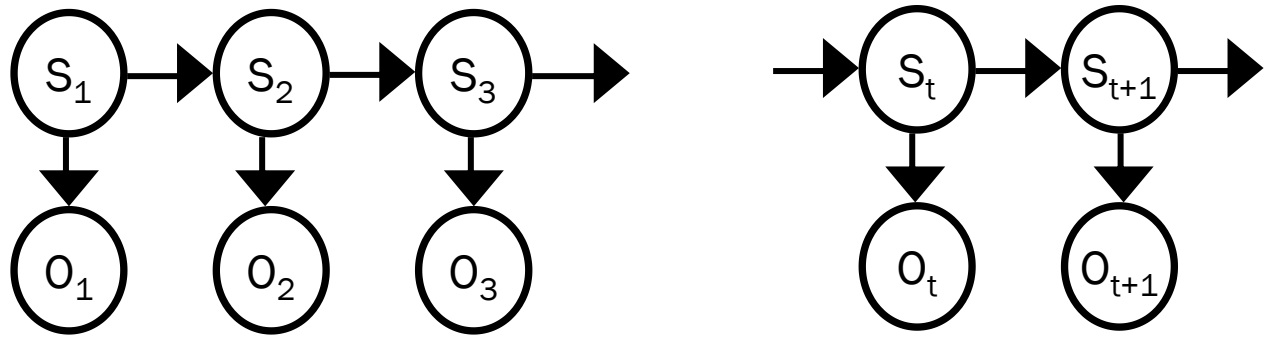
$\beta_{it} \leftarrow P(o_{t+1}, \ldots o_T \mid S_t = i)$

$\beta_{i,T-1} = \sum_{j=1}^{n} a_{ij} b_{jo_T}$

Consider $P(o_{t+2}, \ldots, o_T | o_{t+1}, S_t = i, S_{t+1} = j)$.
Which terms on the right hand side of the conditional can be ignored?

A. None of them
B. o_{t+1}
C. $S_t = i$
D. $S_{t+1} = j$
E. More than one of them

The "forward-backward" algorithm for learning



$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{ik} = P(O_t = k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$S_t \in \{1,2,\dots,n\}$
$O_t \in \{1,2,\dots,m\}$

Forward:
Fill in table of
$\alpha_{it} \leftarrow P(o_1, \dots o_{t,}, S_t = i)$
from left to right

$$\alpha_{i1} = \pi_i b_{io_1}$$
$$\alpha_{jt+1} = \sum_{i=1}^{n} \alpha_{it} a_{ij} b_{jo_{t+1}}$$

Backward:
Fill in table of
$\beta_{it} \leftarrow P(o_{t+1}, \dots o_T | S_t = i)$
from right to left

$$\beta_{i,T-1} = \sum_{j=1}^{n} a_{ij} b_{jo_T}$$
$$\beta_{it} = \sum_{j=1}^{n} a_{ij} b_{jo_{t+1}} \beta_{j,t+1}$$

Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

Use the forward algorithm to find the probability that, if the DJ plays only two songs, the audience will dance to both of them.

Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

Use the forward algorithm and the probabilities you computed for the previous problem to find the probability that the audience danced during the first two songs, did not dance during the third song, and the third song was fast.

Assume there's a DJ playing songs at a party, and the songs she plays can be grouped into two types: slow songs and fast songs. The probability that the first song she plays is slow is $\frac{2}{5}$ (so the probability that the first song is fast is $\frac{3}{5}$).

If she plays a fast song, the probability that she plays a slow song immediately afterwards is $\frac{1}{3}$. If she plays a slow song, the probability that she follows it with a slow song is $\frac{1}{4}$. We will assume that the selection of every song is conditionally independent of every single song before it except the one song immediately before it.

For each song the DJ plays, the audience can have one of two reactions to it: either to dance or not to dance. If the DJ plays a fast song, the probability that the audience will dance to it is $\frac{4}{5}$. If she plays a slow song, the probability that the audience will dance to it is $\frac{1}{2}$. We will assume that whether the audience dances or not depends only on whether the current song is fast or slow.

Assume that the DJ plays 10 songs in total, and the 8th song is slow. Use the backward algorithm to find the probability that the audience won't dance during the 9th and 10th songs.