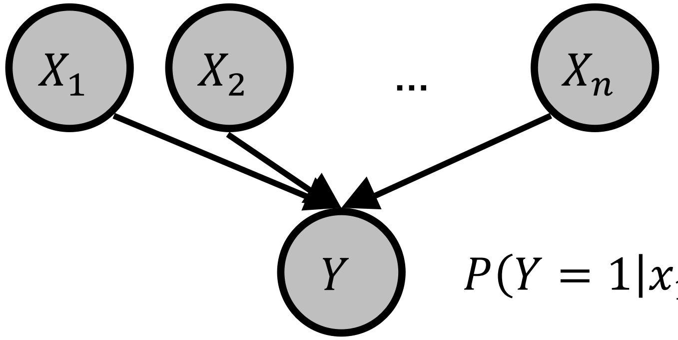
FUDAN SOE SUMMER SCHOOL INTRODUCTION TO AI

Probabilistic Reasoning and Decision Making

Day 10 – EM for Noisy-OR, Hidden Markov Model (HMM)

EM for Learning Noisy-OR model

All variables are binary



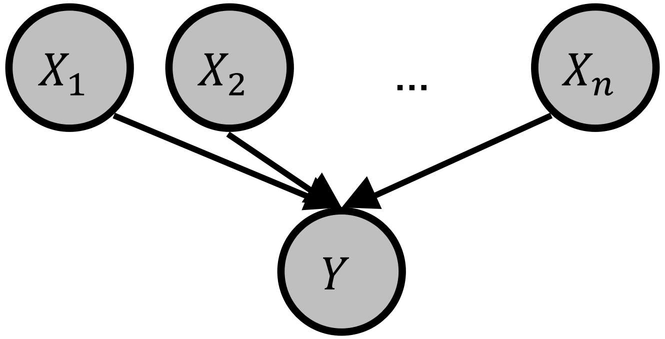
Complete the formula below:

$P(Y = 1|x_1, x_2, \dots, x_n) =$

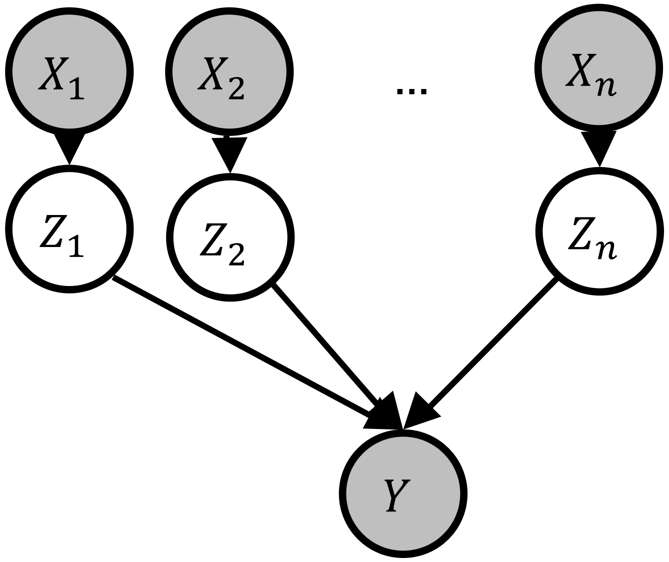
Problem: From *complete* data $\left\{\left(x_1{}^{(t)}, x_n{}^{(t)}, \dots, x_n{}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{T}$, estimate $p_i \in [0,1]$.

EM for Learning Noisy-OR: Alternate model

Problem: From (complete) data $\left\{\left(x_1{}^{(t)}, x_n{}^{(t)}, \dots, x_n{}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{T}$, estimate $p_i \in [0,1]$.



$$P(Y = 1 | x_1, x_2, \dots, x_n) = 1 - \prod_{i=1}^{n} (1 - p_i)^{x_i}$$

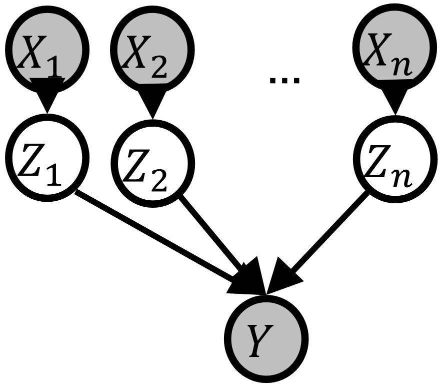


EM for Learning Noisy-OR: Alternate model

Problem: From (complete) data $\left\{\left(x_1{}^{(t)}, x_n{}^{(t)}, \ldots, x_n{}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{T}$, estimate $p_i \in [0,1]$.

$P(Z_i = 0 | X_i = 0) = 1$

$P(Z_i = 1 | X_i = 1) = p_i$



$P(Y|\vec{z}) = OR(\vec{z})$

What is $P(Y = 1 | x_1, x_2, \ldots, x_n)$ in this alternate model?

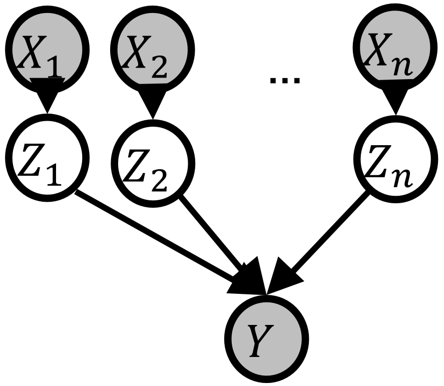
First show that $P(Y = 1|\vec{x}) = \sum_{\vec{z}} P(Y = 1|\vec{z}) P(\vec{z}|\vec{x})$

EM for Learning Noisy-OR: Alternate model

Problem: From (complete) data $\{(x_1{}^{(t)}, x_n{}^{(t)}, \ldots, x_n{}^{(t)}, y^{(t)})\}_{t=1}^{T}$, estimate $p_i \in [0,1]$.

$P(Z_i = 0|X_i = 0) = 1$

$P(Z_i = 1|X_i = 1) = p_i$



$P(Y|\vec{z}) = OR(\vec{z})$

What is $P(Y = 1|x_1, x_2, \ldots, x_n)$ in this alternate model?

$$P(Y = 1|\vec{x}) = \sum_{\vec{z}} \boxed{P(Y = 1|\vec{z})P(\vec{z}|\vec{x})}$$

When is the term in the red box 0?

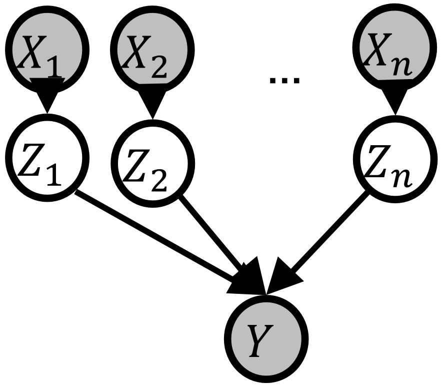
A. When at least one z_i is 0
B. When all z_is are 0
C. You can't tell from the information given

EM for Learning Noisy-OR: Alternate model

Problem: From (complete) data $\left\{\left(x_1{}^{(t)}, x_n{}^{(t)}, \dots, x_n{}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{T}$, estimate $p_i \in [0,1]$.

$P(Z_i = 0|X_i = 0) = 1$

$P(Z_i = 1|X_i = 1) = p_i$



$P(Y|\vec{z}) = OR(\vec{z})$

<u>What is $P(Y = 1|x_1, x_2, \dots, x_n)$ in this alternate model?</u>

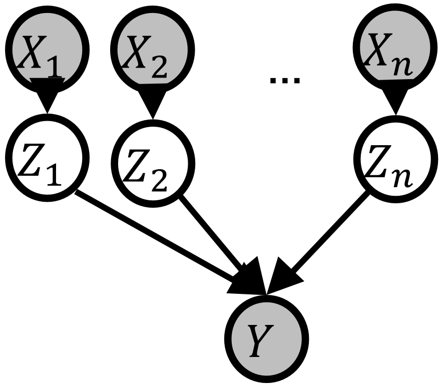
$$P(Y = 1|\vec{x}) = \sum_{\vec{z}} P(Y = 1|\vec{z}) P(\vec{z}|\vec{x})$$

EM for Learning Noisy-OR: Alternate model

Problem: From (complete) data $\left\{\left(x_1{}^{(t)}, x_n{}^{(t)}, \dots, x_n{}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{T}$, estimate $p_i \in [0,1]$.

$P(Z_i = 0 | X_i = 0) = 1$

$P(Z_i = 1 | X_i = 1) = p_i$



$P(Y|\vec{z}) = OR(\vec{z})$

What is $P(Y = 1 | x_1, x_2, \dots, x_n)$ in this alternate model?

$$P(Y = 1|\vec{x}) = \sum_{\vec{z}} P(Y = 1|\vec{z}) P(\vec{z}|\vec{x}) = \sum_{\vec{z} \neq 0} P(\vec{z}|\vec{x}) =$$

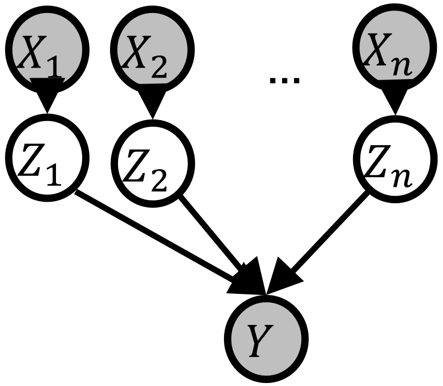
A. $P(\vec{z} = 0|\vec{x})$
B. $1 - P(\vec{z} = 0|\vec{x})$
C. $P(\vec{z}|\vec{x} = 0)$
D. $1 - P(\vec{z}|\vec{x} = 0)$
E. None of these

EM for Learning Noisy-OR: Alternate model

Problem: From (complete) data $\left\{\left(x_1{}^{(t)}, x_n{}^{(t)}, \dots, x_n{}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{T}$, estimate $p_i \in [0,1]$.

$P(Z_i = 0|X_i = 0) = 1$

$P(Z_i = 1|X_i = 1) = p_i$



$P(Y|\vec{z}) = OR(\vec{z})$

<u>What is $P(Y = 1|x_1, x_2, \dots, x_n)$ in this alternate model?</u>

$$P(Y = 1|\vec{x}) = \sum_{\vec{z}} P(Y = 1|\vec{z}) P(\vec{z}|\vec{x}) = \sum_{\vec{z} \neq 0} P(\vec{z}|\vec{x}) = 1 - P(\vec{z} = 0|\vec{x})$$

Learning $P(Z_i = 1|X_i = 1) = p_i$ using EM

- If we had fully observed data:

Learning $P(Z_i = 1 | X_i = 1) = p_i$ using EM

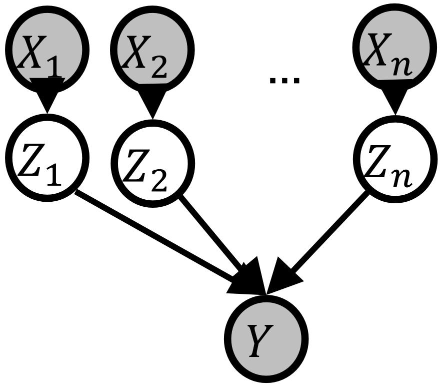
$$\widehat{count}(Z_i = 1, X_i = 1) = \sum_{t=1}^{T} P\left(Z_i = 1, X_i = 1 \middle| \vec{x}^{(t)}, y^{(t)}\right)$$

Learning $P(Z_i = 1|X_i = 1) = p_i$ using EM: Calculate $P\left(Z_i = 1\middle|\vec{x}^{(t)}, y^{(t)}\right)$ with current params

$P(Z_i = 0|X_i = 0) = 1$

$P(Z_i = 1|X_i = 1) = p_i$

$$\widehat{count}(Z_i = 1, X_i = 1) = \sum_{t=1}^{T} P\left(Z_i = 1, X_i = 1\middle|\vec{x}^{(t)}, y^{(t)}\right) = \sum_{t=1}^{T} x_i^{(t)} P\left(Z_i = 1 \middle|\vec{x}^{(t)}, y^{(t)}\right)$$



$P(Y|\vec{z}) = OR(\vec{z})$

$$P\left(Z_i = 1 \middle|\vec{x}^{(t)}, y^{(t)}\right) = \frac{P\left(Y = y^{(t)}\middle|Z_i = 1\right) P\left(Z_i = 1\middle|\vec{x}^{(t)}\right)}{P\left(Y = y^{(t)}\middle|\vec{x}^{(t)}\right)}$$

Which of the following is NOT a true statement?

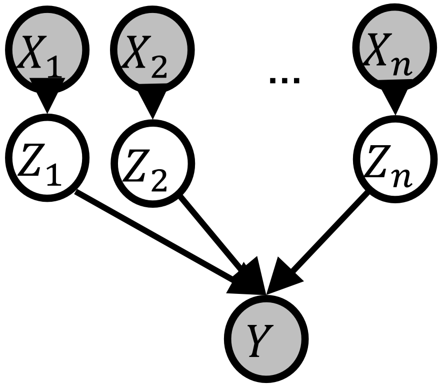
A. $P\left(Y = y^{(t)}\middle|Z_i = 1\right) = 0$ when $y^{(t)} = 0$
B. $P\left(Y = y^{(t)}\middle|Z_i = 1\right) = 1$ when $y^{(t)} = 1$
C. $P\left(Z_i = 1\middle|\vec{x}^{(t)}\right) = 1$ when $x_i^{(t)} = 1$
D. $P\left(Z_i = 1\middle|\vec{x}^{(t)}\right) = 0$ when $x_i^{(t)} = 0$
E. More than one of the above is NOT true

Learning $P(Z_i = 1|X_i = 1) = p_i$ using EM: Calculate $P\left(Z_i = 1\middle|\vec{x}^{(t)}, y^{(t)}\right)$ with current params

$P(Z_i = 0|X_i = 0) = 1$

$P(Z_i = 1|X_i = 1) = p_i$

$$\widehat{count}(Z_i = 1, X_i = 1) = \sum_{t=1}^{T} P\left(Z_i = 1, X_i = 1\middle|\vec{x}^{(t)}, y^{(t)}\right) = \sum_{t=1}^{T} x_i^{(t)} P\left(Z_i = 1\ \middle|\vec{x}^{(t)}, y^{(t)}\right)$$



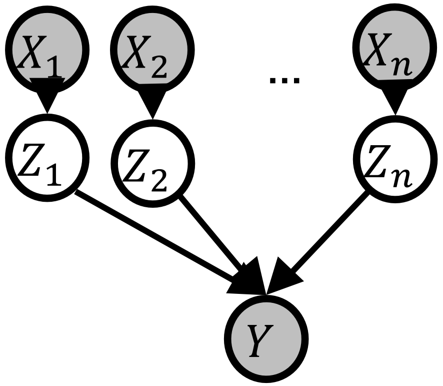
$$P\left(Z_i = 1\ \middle|\vec{x}^{(t)}, y^{(t)}\right) = \frac{P\left(Y = y^{(t)}\middle|Z_i = 1\right) P\left(Z_i = 1\middle|\vec{x}^{(t)}\right)}{P\left(Y = y^{(t)}\middle|\vec{x}^{(t)}\right)}$$

$P(Y|\vec{z}) = OR(\vec{z})$

Learning $P(Z_i = 1|X_i = 1) = p_i$ using EM: Update rule

$P(Z_i = 0|X_i = 0) = 1$

$P(Z_i = 1|X_i = 1) = p_i$



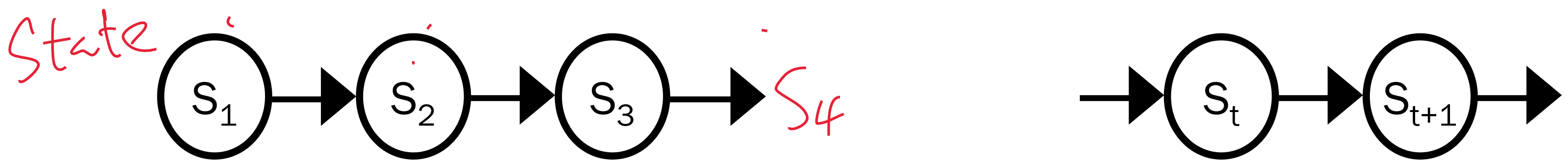
$P(Y|\vec{z}) = OR(\vec{z})$

$$\widehat{count}(Z_i = 1, X_i = 1) = \sum_{t=1}^{T}\left[\frac{p_i y^{(t)} x_i^{(t)}}{1 - \prod_{i=1}^{n}(1 - p_i)^{x_i^{(t)}}}\right]$$

$$p_i \leftarrow \frac{\widehat{count}(Z_i = 1, X_i = 1)}{\widehat{count}(X_i = 1)} = \frac{p_i}{T_i}\sum_{t=1}^{T}\left[\frac{y^{(t)} x_i^{(t)}}{1 - \prod_{i=1}^{n}(1 - p_i)^{x_i^{(t)}}}\right]$$

Markov chains (Bayes Net)

(1st order)



<u>Two simplifying assumptions:</u>

1. Finite Context

$$P(S_t \mid S_1 \cdots S_{t-1}) = P(S_t \mid S_{t-1})$$

(1st order)
makes CPTs very small

2. Position Invariance

$$P(S_{t+1} = j \mid S_t = i) = P(S_{t+k+1} = j \mid S_{t+k} = i)$$

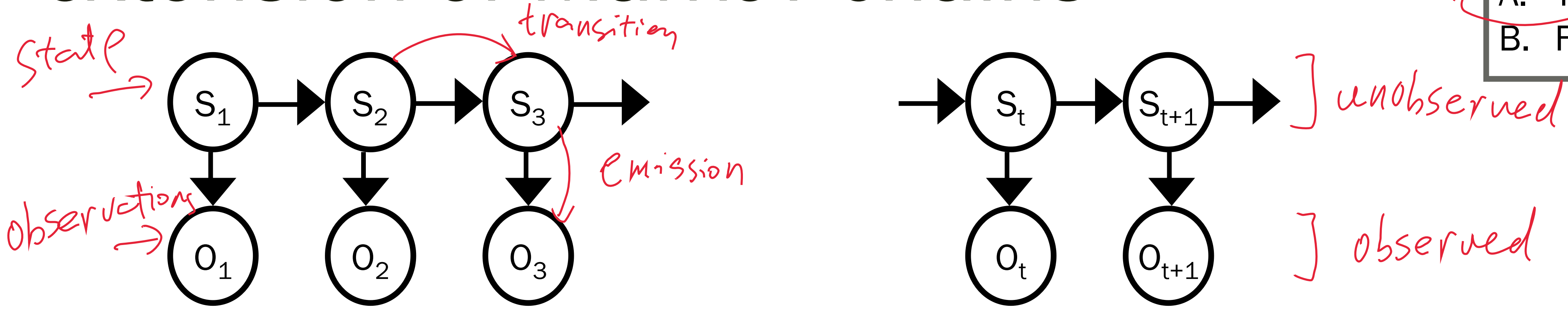
(save a lot of CPTs)

Hidden Markov Models (HMMs): An extension of Markov chains

Bayes Net

True or False: An HMM is a polytree.

A. True
B. False



finite context

$$P(S_t \mid S_1 \cdots S_{t-1}) = P(S_t \mid S_{t-1})$$

$$P(O_t \mid S_1 \cdots S_t) = P(O_t \mid S_t)$$

Assumptions

position invariant

$$P(S_{t+1} \mid S_t) = P(S_{t+k+1} \mid S_{t+k})$$

$$P(O_t \mid S_t) = P(O_{t+k} \mid S_{t+k})$$

CPTs are matrices

HMM examples

- HMMs for language: If the hidden states are words, what might the observations be?

  States AI → is → fun → but → math → heavy

  Sound waves

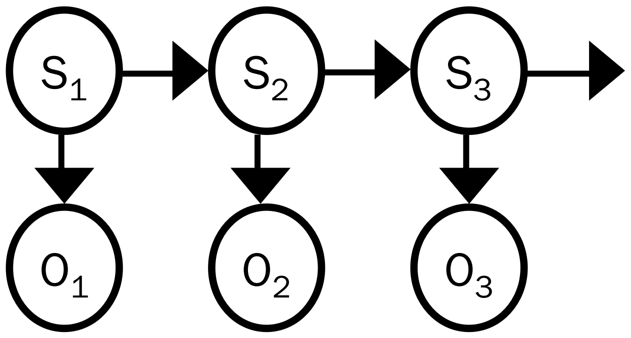
- HMMs for Robotics: If the hidden states are the robot's orientation and location, what might the observations be?

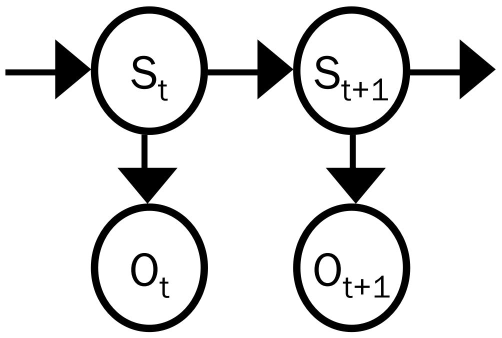
- HMMs for biology: if the hidden states are status of genes in DNA, what might the observations be?

  State : gene or not

  observed : GATC sequence

Conditional Independence in an HMM

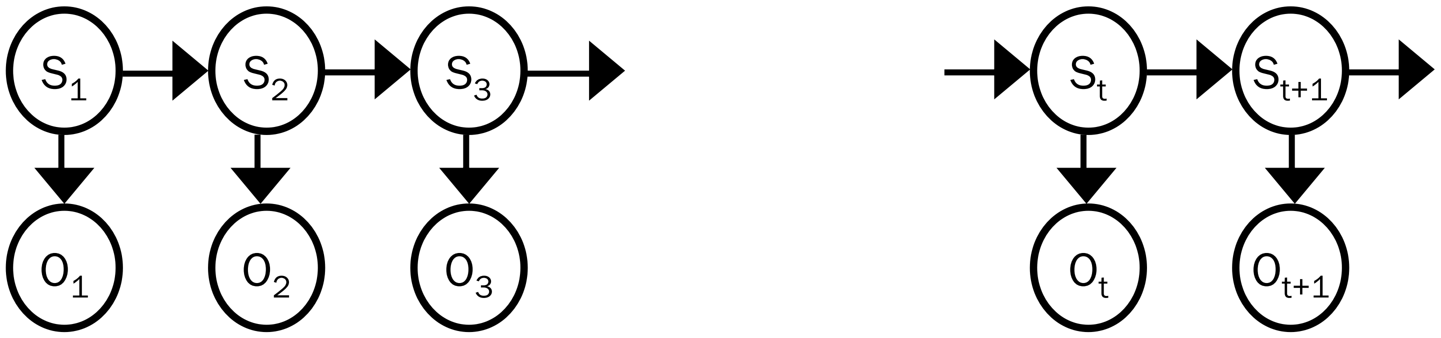




Is the statement below true (A) or false (B)?

$$P(S_t|S_{t-1}) = P(S_t|S_{t-1}, O_t)$$

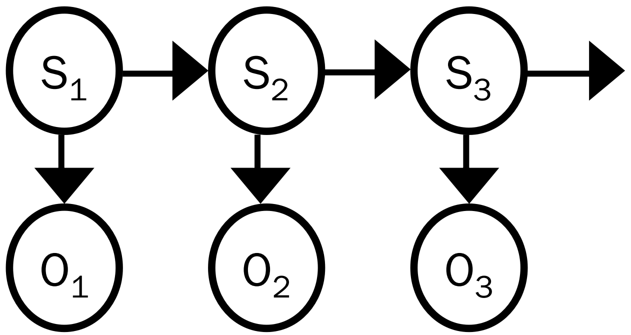
Conditional Independence in an HMM

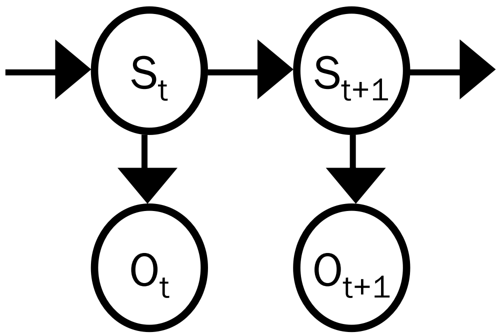


Is the statement below true (A) or false (B)?

$$P(S_2, S_3, \ldots, S_T \mid S_1) = \prod_{t=2}^{T} P(S_t \mid S_{t-1})$$

Parameters of the HMM

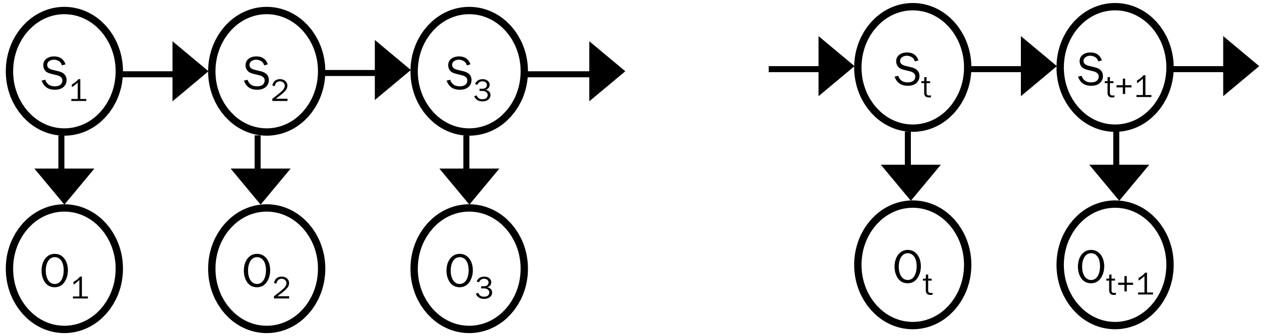




Which of the following is NOT a parameter of this model?

A. $P(S_t|S_{t+1})$
B. $P(S_1)$
C. $P(O_t|O_{t-1})$
D. $P(O_t|S_t)$
E. More than one of these is NOT a parameter of the model

HMM Key Questions



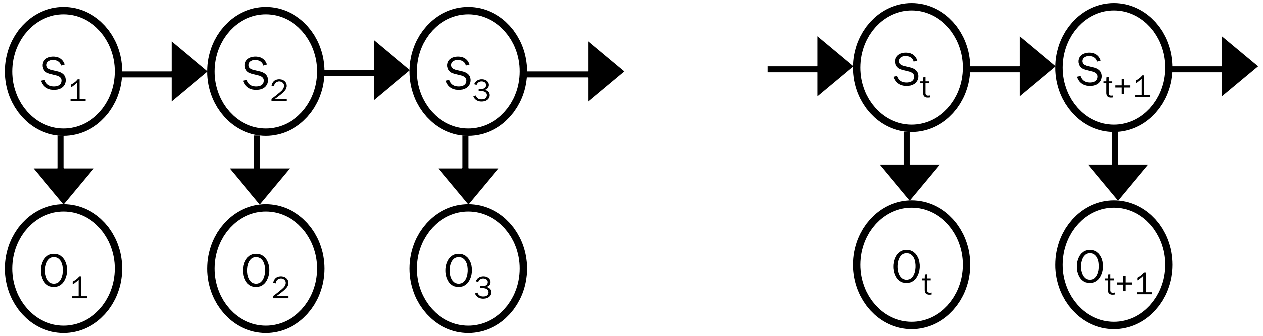
$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

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 $P(o_1, o_2, o_3, \dots o_T)$

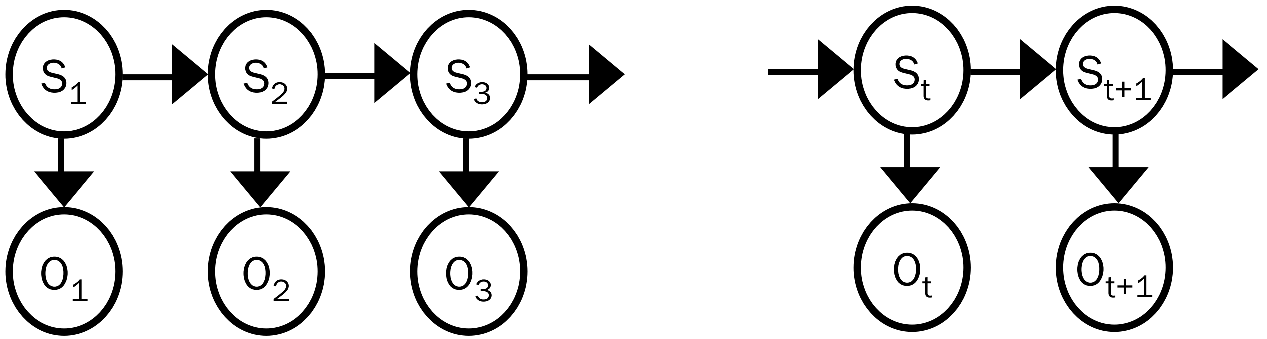


$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

$$P(o_1, o_2, o_3, \dots o_T) = \sum_{\vec{s}} P(s_1 \dots . s_T, o_1 \dots . o_T)$$

Use the structure of the HMM to factor this joint probability

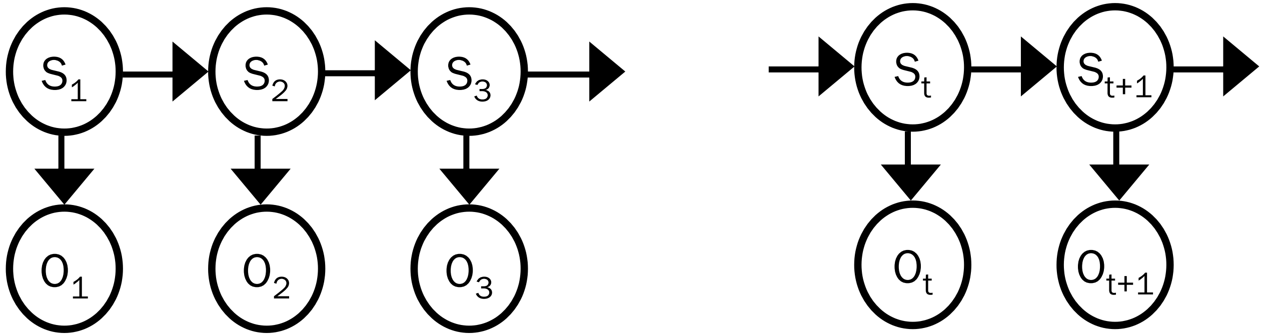
How to compute $P(o_1, o_2, o_3, \dots o_T)$ efficiently using the "forward algorithm"



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

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

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

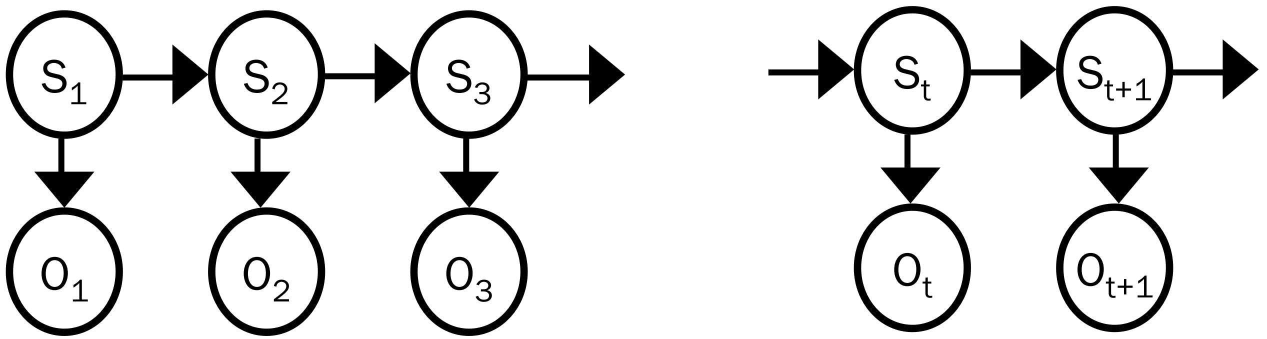


$a_{ij} = P(S_t = j | S_{t-1} = i)$
$b_{io_k} = P(O_t = o_k | S_t = i)$
$\pi_i = P(S_1 = i)$

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

$P(o_1, o_2, o_3, \ldots o_T) =$

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



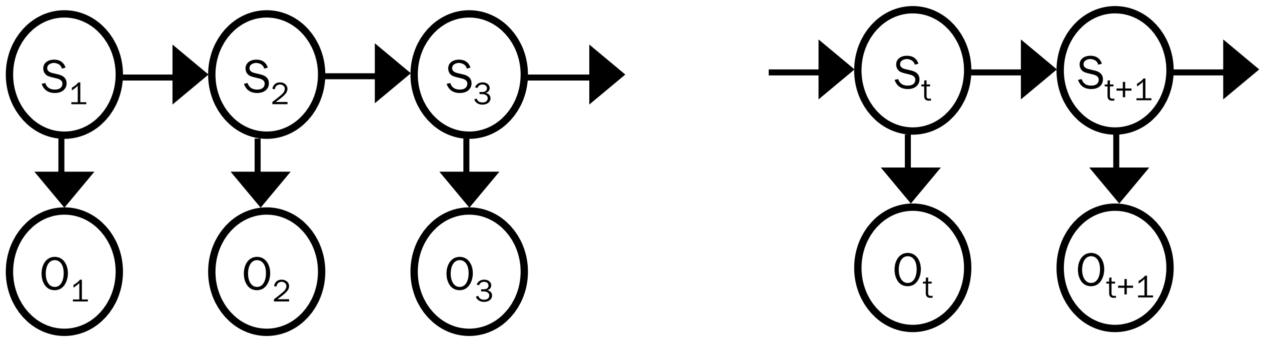
$a_{ij} = P(S_t = j | S_{t-1} = i)$
$b_{io_k} = P(O_t = o_k | S_t = i)$
$\pi_i = P(S_1 = i)$

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"



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

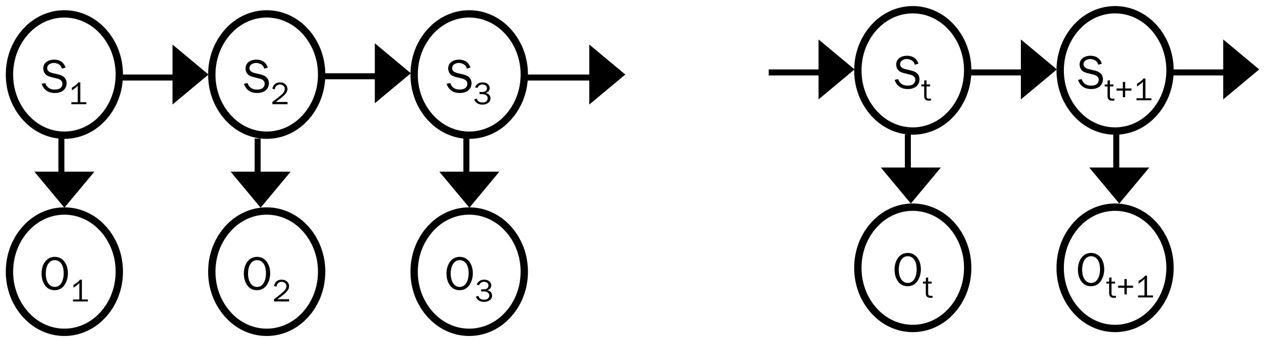
Let $\alpha_{it} \leftarrow P(o_1, \ldots o_{t,}, s_t = i)$

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

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

$$\alpha_{i1} =$$

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



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

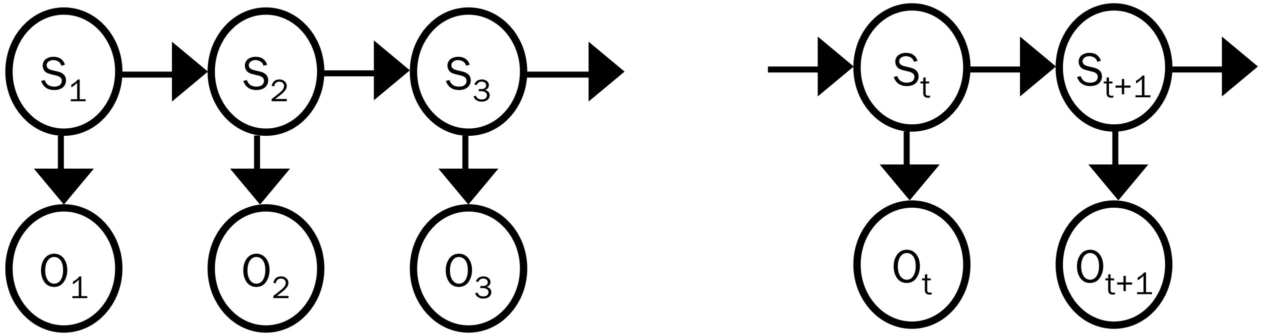
Let $\alpha_{it} \leftarrow P(o_1, \dots o_{t,}, s_t = i)$

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

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

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

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

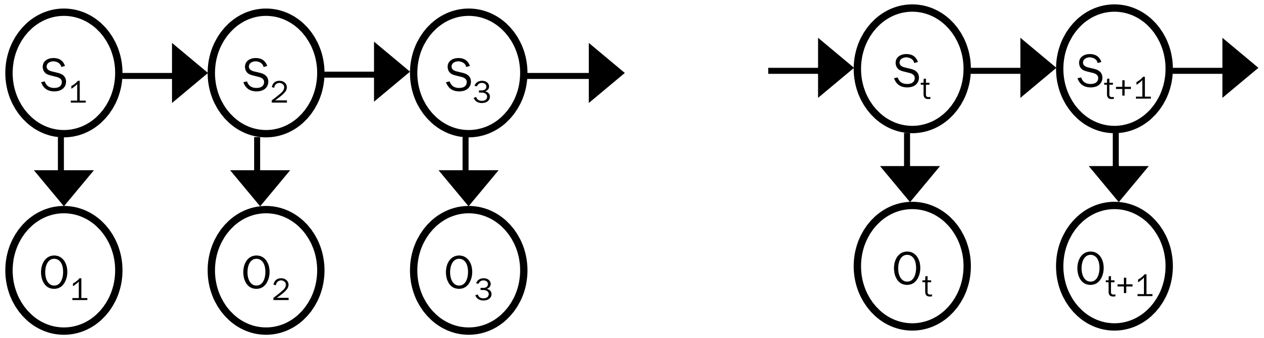


$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

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

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

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



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

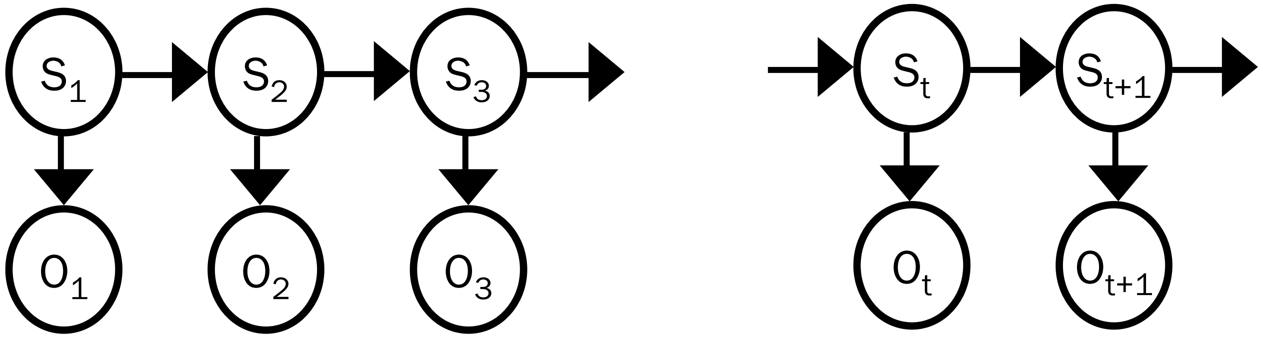
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}}\left(P(s_1 \dots s_T, o_1 \dots o_T)\right)$$

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



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

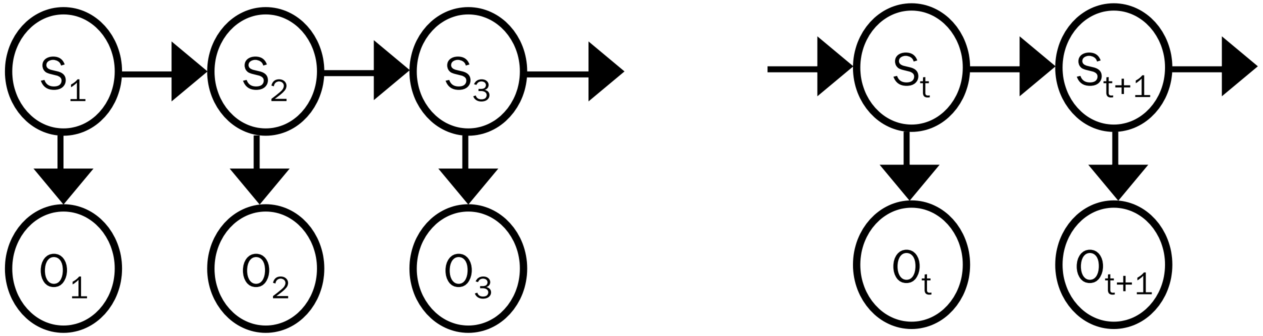
$$({s_1}^*, {s_2}^*, \dots, {s_T}^*) = \arg\max_{\vec{s}}\left(P(s_1 \dots s_T, o_1 \dots o_T)\right)$$

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								

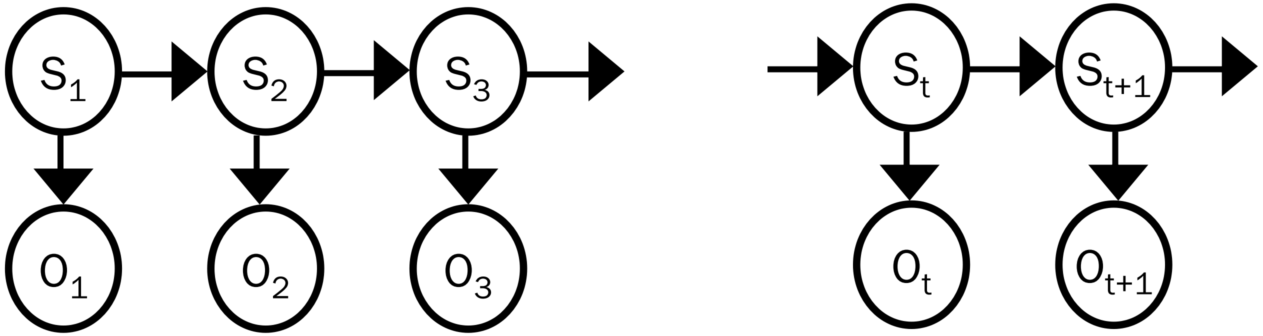


$$a_{ij} = P(S_t = j | S_{t-1} = i)$$
$$b_{io_k} = P(O_t = o_k | S_t = i)$$
$$\pi_i = P(S_1 = i)$$

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

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

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



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

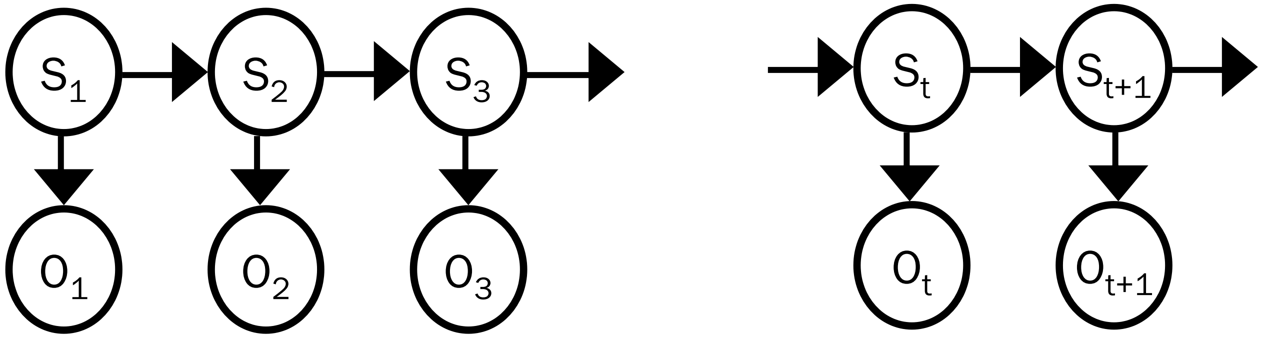
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}} \big(P(s_1 \dots s_T, o_1 \dots o_T)\big)$$

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

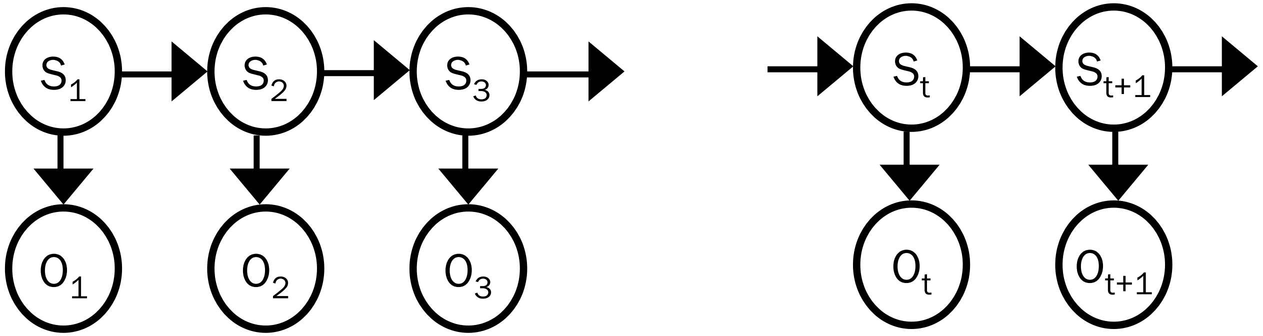
$$\ell_{i1}{}^* = \log\left[\pi_i b_{io_1}\right]$$

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

T

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

n



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

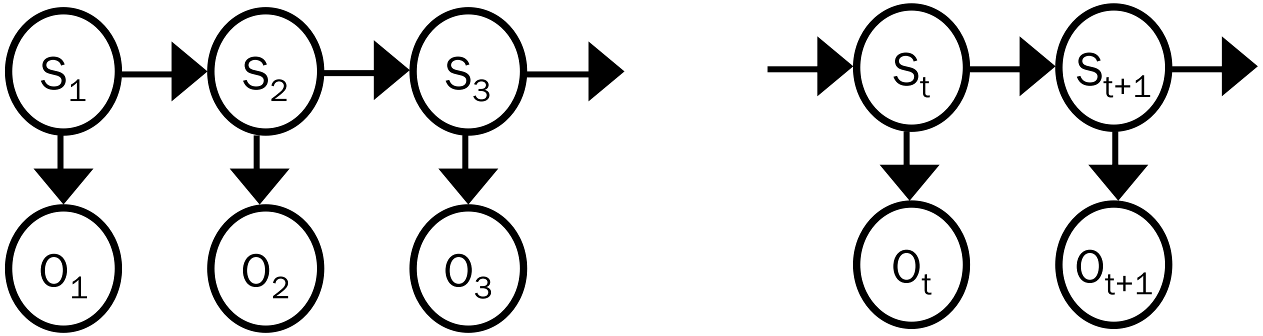
$$(s_1{}^*, s_2{}^*, \dots, s_T{}^*) = \arg\max_{\vec{s}} \Big(P(s_1 \dots s_T, o_1 \dots o_T)\Big)$$

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								



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

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

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

<u>Viterbi algorithm:</u>

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*

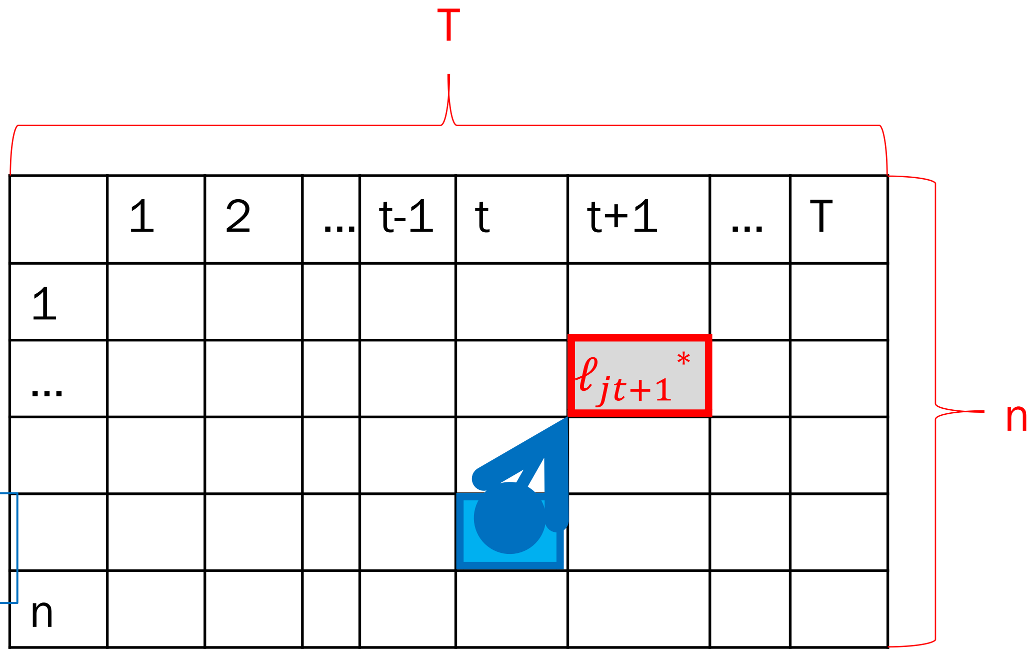
	1	2	...	t-1	t	t+1	...	T
1								
...								
n								

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

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

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

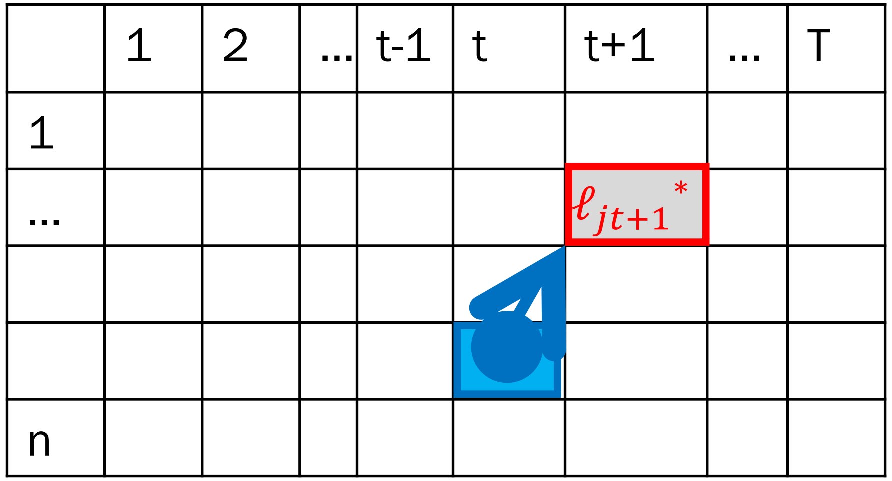
$$\ell_{jt+1}{}^* = \max_i \left\{\ell_{it}{}^* + \log a_{ij}\right\} + \log b_{jo_{t+1}}$$



$\underset{i}{\operatorname{argmax}}\left\{\ell_{it}{}^* + \log a_{ij}\right\}$

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

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

$$\ell_{i1}{}^* = \log\left[\pi_i b_{i o_1}\right]$$

$$\ell_{jt+1}{}^* = \max_i\left\{\ell_{it}{}^* + \log a_{ij}\right\} + \log b_{j o_{t+1}}$$

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

T

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

n

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

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

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

$$\ell_{i1}{}^* = \log\left[\pi_i b_{i o_1}\right]$$

$$\ell_{jt+1}{}^* = \max_i \left\{\ell_{it}{}^* + \log a_{ij}\right\} + \log b_{j o_{t+1}}$$

Define $\Phi_{jt+1} = \underset{i}{\operatorname{argmax}}\left\{\ell_{it}{}^* + \log a_{ij}\right\}$

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

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)

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

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

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								
…						$\boxed{\ell_{jt+1}{}^*}$		
n								

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

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

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

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

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								
…						$\boxed{\ell_{jt+1}{}^*}$		
n								

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

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

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

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

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								
…						$\boxed{\ell_{jt+1}{}^*}$		
n								

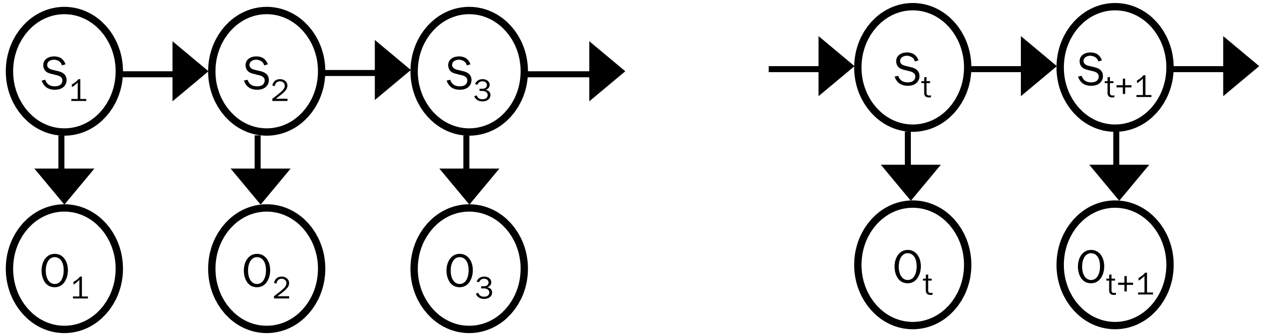
Define $\Phi_{j,t+1} = \operatorname{argmax}_i \{\ell_{it}{}^* + \log a_{ij}\}$

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

<u>Complete Viterbi algorithm:</u>

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 $\ell_{jt+1}{}^*$ 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



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

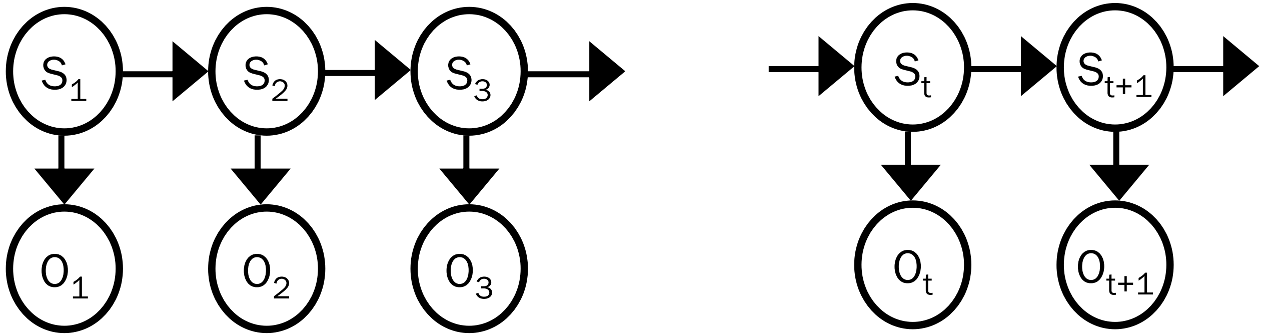
$S_t \in \{1,2,\ldots,n\}$
$O_t \in \{1,2,\ldots,m\}$

1. How to compute $P(o_1, o_2, o_3, \ldots o_T)$ ← LAST WEEK (running time?)
2. How to compute the most likely state sequence given observations ← LAST TIME (running time?)

3. How to update beliefs for real-time monitoring

4. How to learn HMMs from data

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

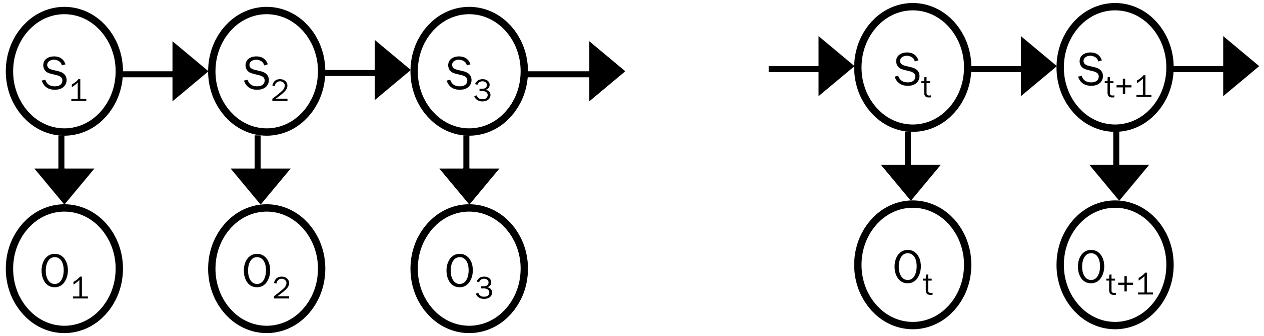


$a_{ij} = P(S_t = j|S_{t-1} = i)$
$b_{io_k} = P(O_t = o_k|S_t = i)$
$\pi_i = P(S_1 = i)$

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

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

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

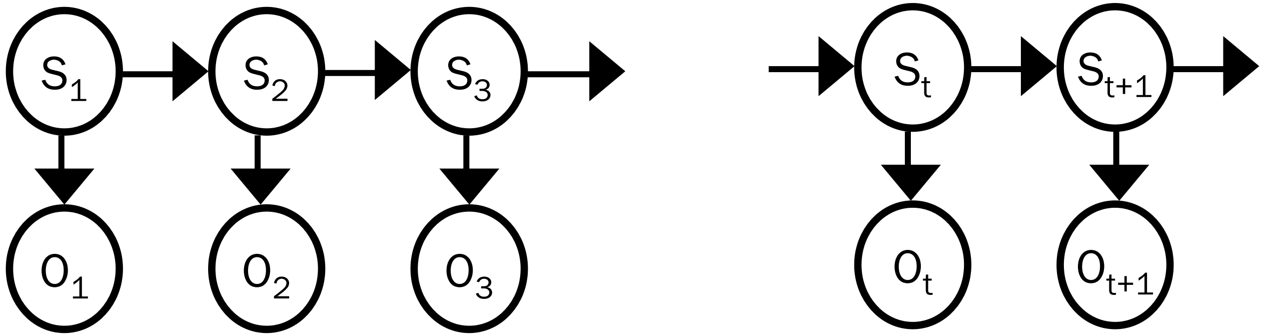


$$\boxed{\begin{array}{l} a_{ij} = P(S_t = j | S_{t-1} = i) \\ b_{io_k} = P(O_t = o_k | S_t = i) \\ \pi_i = P(S_1 = i) \end{array}}$$

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

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

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



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

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

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