CSE 150A-250A AI: Probabilistic Models

Lecture 13

Fall 2025

Trevor Bonjour
Department of Computer Science and Engineering
University of California, San Diego

Slides adapted from previous versions of the course (Prof. Lawrence, Prof. Alvarado, Prof Berg-Kirkpatrick)

Agenda

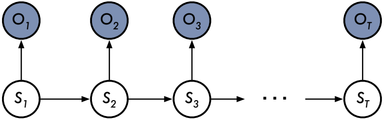
Review

Learning in HMMs

Backward Algorithm

Review

- Belief network



observations $o_t \in \{1, 2, \ldots, m\}$

states $s_t \in \{1, 2, \ldots, n\}$

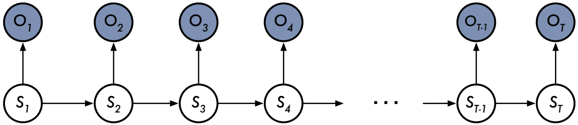
- Parameters

$$
\begin{aligned}
a_{ij} &= P(S_{t+1}=j|S_t=i) && \text{transition matrix} \\
b_{ik} &= P(O_t=k|S_t=i) && \text{emission matrix} \\
\pi_i &= P(S_1=i) && \text{initial state distribution}
\end{aligned}
$$

- Notation

  Sometimes we'll write $b_i(k) = b_{ik}$ to avoid double subscripts.

Key computations in HMMs



Inference

1. How to compute the likelihood $P(o_1, o_2, \ldots, o_T)$?
2. How to compute the most likely hidden states $\mathrm{argmax}_{\vec{s}}\, P(\vec{s}|\vec{o})$?
3. How to update beliefs by computing $P(s_t|o_1, o_2, \ldots, o_t)$?

Learning

How to estimate parameters $\{\pi_i, a_{ij}, b_{ik}\}$ that maximize the log-likelihood of observed sequences?

For a particular sequence of observations $\{o_1, o_2, \ldots, o_T\}$, define the matrix with elements:

$$\alpha_{it} = P(o_1, o_2, \ldots, o_t, S_t = i) \qquad \begin{array}{cc} n \text{ rows} & \left\{ \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1,T-1} & \alpha_{1T} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2,T-1} & \alpha_{2T} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{n,T-1} & \alpha_{nT} \end{bmatrix} \right. \end{array}$$

The forward algorithm fills in the matrix of α_{it} elements one column at a time:

$$\boxed{\begin{aligned} \alpha_{i1} &= \pi_i \, b_i(o_1) \\ \alpha_{j,t+1} &= \sum_{i=1}^{n} \alpha_{it} \, a_{ij} \, b_j(o_{t+1}) \end{aligned}}$$

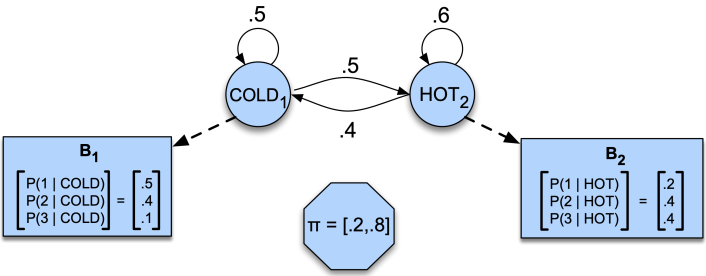
Computing the likelihood $P(o_1, o_2, \ldots, o_T)$

$$
\begin{array}{rc}
 & \begin{array}{ccccc} & & & & \textbf{Sum!} \end{array} \\
n \text{ rows} & \left\{ \begin{bmatrix}
\alpha_{11} & \alpha_{12} & \cdots & \alpha_{1,T-1} & \alpha_{1T} \\
\alpha_{21} & \alpha_{22} & \cdots & \alpha_{2,T-1} & \alpha_{2T} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{n,T-1} & \alpha_{nT}
\end{bmatrix} \right.
\end{array}
$$

$$
\begin{aligned}
P(o_1, o_2, \ldots, o_T) \\
&= \sum_{i=1}^{n} P(o_1, o_2, \ldots, o_T, s_T = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{i=1}^{n} \alpha_{iT} \qquad \boxed{\text{sum of last column}}
\end{aligned}
$$

Example[1]

- Two hidden states (Weather): $\{H, C\}$
- Observations (Ice creams): $\{1, 2, 3\}$

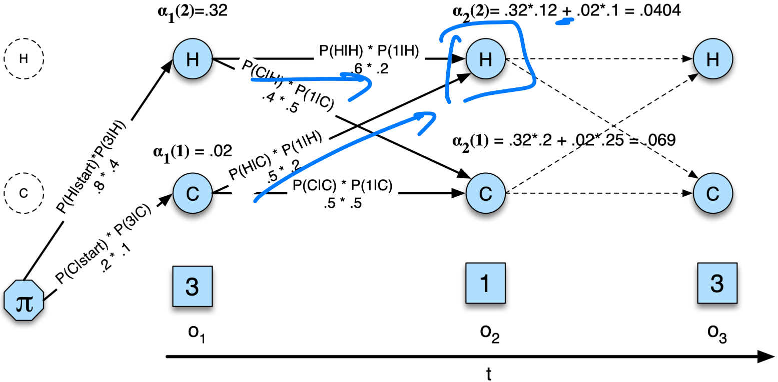


[1] Eisner, J. 2002. An interactive spreadsheet for teaching the forward-backward algorithm.

Example - Forward Algorithm

$$\alpha_{i1} = \pi_i \, b_i(o_1)$$

$$\alpha_{j,t+1} = \sum_{i=1}^{n} \alpha_{it} \, a_{ij} \, b_j(o_{t+1})$$



$$
\begin{aligned}
&\{s_1^*, s_2^*, \ldots, s_T^*\} \\
&= \operatorname{argmax}_{s_1, s_2, \ldots, s_T} P(s_1, s_2, \ldots, s_T | o_1, o_2, \ldots, o_T) \\
&= \operatorname{argmax}_{s_1, s_2, \ldots, s_T} \log P(s_1, s_2, \ldots, s_T, o_1, o_2, \ldots, o_T)
\end{aligned}
$$

For a particular sequence of observations $\{o_1, o_2, \ldots, o_T\}$, we define the following matrix:

$$
\ell_{it}^* = \max_{s_1, s_2, \ldots, s_{t-1}} \log P(s_1, s_2, \ldots, s_{t-1}, S_t = i, o_1, o_2, \ldots, o_t)
$$

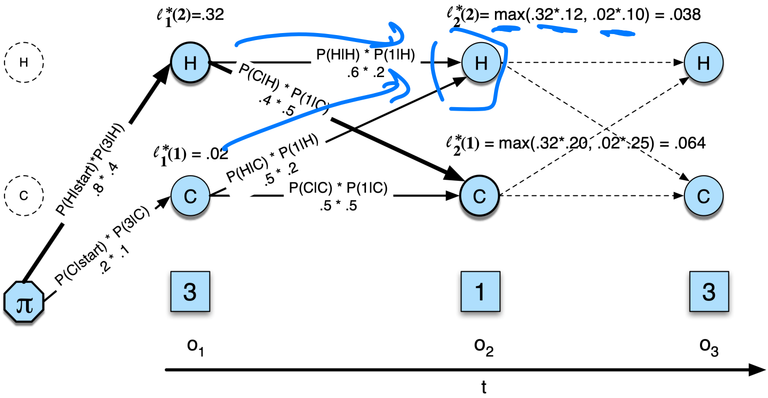
Viterbi Algorithm

We compute the matrix ℓ^* one column at a time, from left to right:

$$
\boxed{
\begin{aligned}
\ell^*_{i1} &= \log \pi_i + \log b_i(o_1) \\
\ell^*_{j,t+1} &= \max_i \left[\ell^*_{it} + \log a_{ij} \right] + \log b_j(o_{t+1})
\end{aligned}
}
$$

$$
n \text{ rows} \left\{ \begin{bmatrix}
\ell^*_{11} & \ell^*_{12} & \cdots & \ell^*_{1,T-1} & \ell^*_{1T} \\
\ell^*_{21} & \ell^*_{22} & \cdots & \ell^*_{2,T-1} & \ell^*_{2T} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\ell^*_{n1} & \ell^*_{n2} & \cdots & \ell^*_{n,T-1} & \ell^*_{nT}
\end{bmatrix} \right.
$$

Example - Viterbi (Fill ℓ^*)



max $f(x)$ → max value

- Form one more matrix:

state argmax $f(x)$

$$\Phi_{t+1}(j) = \arg\max_i \left[\ell_{it}^* + \log a_{ij} \right]$$

↳ $\boxed{\text{val of } x \_}$

- Compute the most likely states by backtracking:

$$s_T^* = \arg\max_i \left[\ell_{iT}^* \right]$$

$\in S$

Hot Cold

Max!

$$\left[\begin{array}{cccccc} \ell_{11}^* & \ell_{12}^* & \cdots & \ell_{1,T-1}^* & \ell_{1T}^* \\ \ell_{21}^* & \ell_{22}^* & \cdots & \ell_{2,T-1}^* & \ell_{2T}^* \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \ell_{n1}^* & \ell_{n2}^* & \cdots & \ell_{n,T-1}^* & \ell_{nT}^* \end{array} \right]$$

Computing $\{s_1^*, s_2^*, \ldots, s_T^*\}$

- Form one more matrix:

$$\Phi_{t+1}(j) \;=\; \arg\max_i \left[\ell_{it}^* + \log a_{ij} \right]$$

- Compute the most likely states by backtracking:

$$s_T^* \;=\; \arg\max_i \left[\ell_{iT}^* \right]$$

for $t = T-1$ **to** 1

$$s_t^* \;=\; \Phi_{t+1}(s_{t+1}^*)$$

end

$$\begin{array}{c} \hspace{14em}\text{Max!} \\ \begin{bmatrix} \ell_{11}^* & \ell_{12}^* & \cdots & \ell_{1,T-1}^* & \ell_{1T}^* \\ \ell_{21}^* & \ell_{22}^* & \cdots & \ell_{2,T-1}^* & \ell_{2T}^* \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \ell_{n1}^* & \ell_{n2}^* & \cdots & \ell_{n,T-1}^* & \ell_{nT}^* \end{bmatrix} \end{array}$$

Summary of Viterbi algorithm

- Fill ℓ^* matrix from left to right:

$$\boxed{t = 1} \qquad \ell^*_{i1} = \log \pi_i + \log b_i(o_1)$$

$$\boxed{t > 1} \qquad \ell^*_{j,t+1} = \max_i \left[\ell^*_{it} + \log a_{ij} \right] + \log b_j(o_{t+1})$$

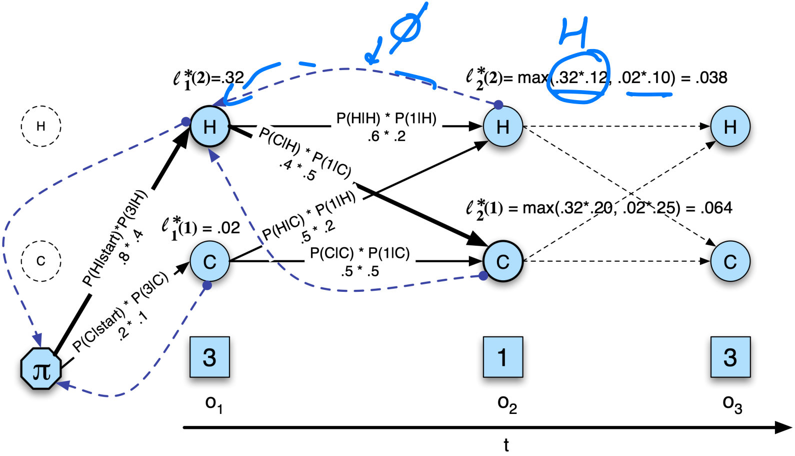
- Backtrack through ℓ^* from right to left:

$$\boxed{t = T} \qquad s^*_T = \text{argmax}_i \left[\ell^*_{iT} \right]$$

$$\boxed{t < T} \qquad s^*_t = \text{argmax}_i \left[\ell^*_{it} + \log a_{is^*_{t+1}} \right]$$

Sometimes $\{s^*_1, s^*_2, \ldots, s^*_T\}$ is called the **Viterbi path**.

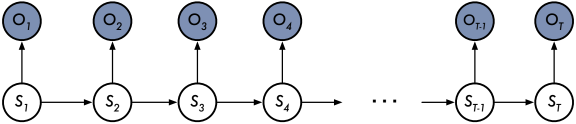
Example - Viterbi (Backtrack through ℓ^*)

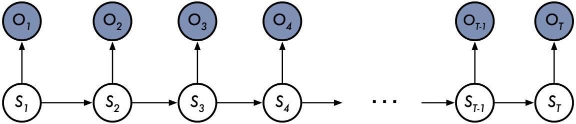


Learning in HMMs

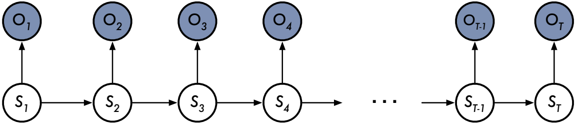
Learning in HMMs

Learning in HMMs



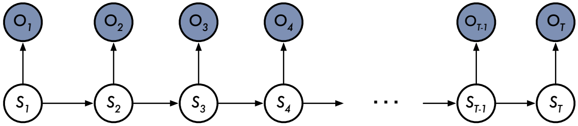


Given: one or more sequences of observations $\{o_1, o_2, \ldots, o_T\}$.



Given: one or more sequences of observations $\{o_1, o_2, \ldots, o_T\}$.
For simplicity, we'll assume just one.

Learning in HMMs

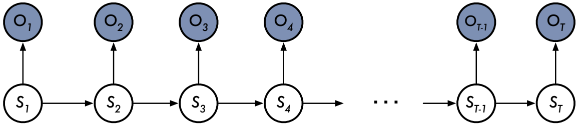


Given: one or more sequences of observations $\{o_1, o_2, \ldots, o_T\}$.

For simplicity, we'll assume just one.

Goal: estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(o_1, o_2, \ldots, o_T)$,

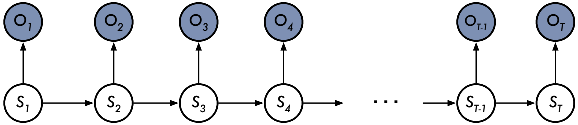
Learning in HMMs



Given: one or more sequences of observations $\{o_1, o_2, \dots, o_T\}$.
For simplicity, we'll assume just one.

Goal: estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(o_1, o_2, \dots, o_T)$,
the likelihood of the observed data.

Learning in HMMs

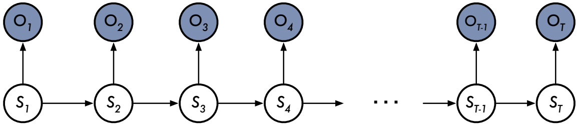


Given: one or more sequences of observations $\{o_1, o_2, \ldots, o_T\}$.
For simplicity, we'll assume just one.

Goal: estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(o_1, o_2, \ldots, o_T)$,
the likelihood of the observed data.

Assume: the cardinality n of the hidden state space is fixed.

Learning in HMMs



Given: one or more sequences of observations $\{o_1, o_2, \ldots, o_T\}$.
For simplicity, we'll assume just one.

Goal: estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(o_1, o_2, \ldots, o_T)$,
the likelihood of the observed data.

Assume: the cardinality n of the hidden state space is fixed.
$s_t \in \{1, 2, \ldots, n\}$

How do we estimate $\{\pi_i, a_{ij}, b_{ik}\}$?

How do we estimate $\{\pi_i, a_{ij}, b_{ik}\}$?

EM Algorithm!

How do we estimate $\{\pi_i, a_{ij}, b_{ik}\}$?

EM Algorithm!
How EM works in general:

To re-estimate $P(X_i = x | \mathrm{pa}_i = \pi)$ in the M-step,
we must compute $P(X_i = x, \mathrm{pa}_i = \pi | V)$ in the E-step.

EM algorithm for HMMs

EM algorithm for HMMs

- CPTs to re-estimate:

EM algorithm for HMMs

- CPTs to re-estimate:

$$\pi_i \quad = \quad P(S_1 = i)$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned} \pi_i &= P(S_1 = i) \\ a_{ij} &= P(S_{t+1} = j | S_t = i) \end{aligned}$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned} \pi_i &= P(S_1 = i) \\ a_{ij} &= P(S_{t+1} = j | S_t = i) \\ b_{ik} &= P(O_t = k | S_t = i) \end{aligned}$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}\pi_i &= P(S_1 = i)\\ a_{ij} &= P(S_{t+1} = j | S_t = i)\\ b_{ik} &= P(O_t = k | S_t = i)\end{aligned}$$

- E-step in HMMs must compute:

EM algorithm for HMMs

- CPTs to re-estimate:

$$
\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}
$$

- E-step in HMMs must compute:

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

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}\pi_i &= P(S_1=i)\\ a_{ij} &= P(S_{t+1}=j|S_t=i)\\ b_{ik} &= P(O_t=k|S_t=i)\end{aligned}$$

- E-step in HMMs must compute:

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

$$P(S_{t+1}=j, S_t=i|o_1, o_2, \ldots, o_T)$$

- CPTs to re-estimate:

$$
\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}
$$

- E-step in HMMs must compute:

$$
\begin{aligned}
P(S_1 = i | o_1, o_2, \ldots, o_T) \\
P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T) \\
P(O_t = k, S_t = i | o_1, o_2, \ldots, o_T)
\end{aligned}
$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}\pi_i &= P(S_1 = i)\\ a_{ij} &= P(S_{t+1} = j | S_t = i)\\ b_{ik} &= P(O_t = k | S_t = i)\end{aligned}$$

- E-step in HMMs must compute:

$$\begin{aligned}&P(S_1 = i | o_1, o_2, \ldots, o_T)\\ &P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T)\\ P(O_t = k, S_t = i | o_1, o_2, \ldots, o_T) &= I(o_t, k)\, P(S_t = i | o_1, o_2, \ldots, o_T)\end{aligned}$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}\pi_i &= P(S_1\!=\!i)\\ a_{ij} &= P(S_{t+1}\!=\!j|S_t\!=\!i)\\ b_{ik} &= P(O_t\!=\!k|S_t\!=\!i)\end{aligned}$$

- E-step in HMMs must compute:

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

$$P(S_{t+1}\!=\!j, S_t\!=\!i|o_1,o_2,\ldots,o_T)$$

$$P(O_t\!=\!k, S_t\!=\!i|o_1,o_2,\ldots,o_T) = I(o_t,k)\,\overbrace{P(S_t\!=\!i|o_1,o_2,\ldots,o_T)}^{\text{special case of below } (t=1)}$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned} \pi_i &= P(S_1 = i) \\ a_{ij} &= P(S_{t+1} = j | S_t = i) \\ b_{ik} &= P(O_t = k | S_t = i) \end{aligned}$$

- E-step in HMMs must compute:

$$\begin{aligned} P(S_1 = i | o_1, o_2, \ldots, o_T) & \\ P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T) & \\ P(O_t = k, S_t = i | o_1, o_2, \ldots, o_T) &= I(o_t, k) \underbrace{P(S_t = i | o_1, o_2, \ldots, o_T)}^{\text{special case of below } (t=1)} \end{aligned}$$

$\boxed{\text{How to efficiently compute these posteriors?}}$

Computing $P(S_t = i | o_1, \ldots, o_T)$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) \; =$$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)}$$

product rule

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \qquad \boxed{\text{product rule}}$$

- Numerator

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \qquad \boxed{\text{product rule}}$$

- Numerator

$P(S_t = i, o_1, o_2, \ldots, o_T)$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \quad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ = \quad & P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) \quad \boxed{\text{product rule}} \end{aligned}$$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \qquad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) \qquad \boxed{\text{product rule}} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) \qquad \boxed{\text{conditional independence}} \end{aligned}$$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)}$$

product rule

- Numerator

$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) && \text{product rule} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) && \text{conditional independence} \\ &= \alpha_{it}\, \underbrace{P(o_{t+1}, \ldots, o_T | S_t = i)} \rightarrow ? \end{aligned}$$

fwd

Backward Algorithm

We need one more matrix ...

We need one more matrix ...

Analogous to $\alpha_{it} \quad = \quad P(o_1, o_2, \ldots, o_t, S_t = i),$

We need one more matrix ...

$$\begin{aligned} \text{Analogous to } \alpha_{it} &= P(o_1, o_2, \ldots, o_t, S_t = i), \\ \text{define } \beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i). \end{aligned}$$

We need one more matrix ...

$$
\begin{aligned}
\text{Analogous to } \alpha_{it} &= P(o_1, o_2, \ldots, o_t, S_t = i), \\
\text{define } \beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T \mid S_t = i).
\end{aligned}
$$

$$
n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & \beta_{1T} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & \beta_{2T} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & \beta_{nT} \end{bmatrix} \right.
$$

We need one more matrix ...

$$
\begin{aligned}
\text{Analogous to } \alpha_{it} &= P(o_1, o_2, \ldots, o_t, S_t = i), \\
\text{define } \beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T \mid S_t = i).
\end{aligned}
$$

$$
n \text{ rows} \left\{ \begin{bmatrix}
\beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & \beta_{1T} \\
\beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & \beta_{2T} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & \beta_{nT}
\end{bmatrix} \right.
$$

Understand the **differences** between these matrices:

We need one more matrix ...

$$\begin{aligned} \text{Analogous to } \alpha_{it} &= P(o_1, o_2, \ldots, o_t, S_t = i), \\ \text{define } \beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T \mid S_t = i). \end{aligned}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & \beta_{1T} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & \beta_{2T} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & \beta_{nT} \end{bmatrix} \right.$$

Understand the **differences** between these matrices:

- α_{it} predicts observations up to and including time t.

We need one more matrix …

$$\begin{aligned} \text{Analogous to } \alpha_{it} &= P(o_1, o_2, \ldots, o_t, S_t = i), \\ \text{define } \beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i). \end{aligned}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & \beta_{1T} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & \beta_{2T} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & \beta_{nT} \end{bmatrix} \right.$$

Understand the **differences** between these matrices:

- α_{it} predicts observations up to and including time t.
- β_{it} predicts observations from **time $t+1$** to time T.

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Last column ($t = T$)

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Last column ($t = T$)

$$\beta_{iT} = P(\underline{\quad\quad} | S_T = i)$$

What does this mean?

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Last column ($t = T$)

$$\beta_{iT} \;=\; P(\underline{\quad\quad} | S_T = i)$$

What does this mean?

Note: β_{it} computes the probability of the future given $S_t = i$.

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Last column ($t = T$)

$$\beta_{iT} \;=\; P(\underline{\quad\quad} | S_T = i)$$

What does this mean?

Note: β_{it} computes the probability of the future given $S_t = i$.

But we don't see *any* observations beyond time T.

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- **Last column ($t = T$)**

$$\beta_{iT} \;=\; P(\underline{\qquad} | S_T = i)$$

What does this mean?

Note: β_{it} computes the probability of the future given $S_t = i$.

But we don't see *any* observations beyond time T.
Put another way, the future after time T is unspecified.

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- **Last column ($t = T$)**

$$\beta_{iT} \;=\; P(\,\underline{\quad\quad}\,| S_T = i)$$

What does this mean?

Note: β_{it} computes the probability of the future given $S_t = i$.

But we don't see *any* observations beyond time T.
Put another way, the future after time T is unspecified.

What is the probability of some unspecified future occurring?

- **Last column ($t = T$)**

$$\beta_{iT} \ = \ P(\underline{\quad\quad} | S_T = i)$$

What does this mean?

Note: β_{it} computes the probability of the future given $S_t = i$.

But we don't see *any* observations beyond time T.
Put another way, the future after time T is unspecified.

What is the probability of some unspecified future occurring?
By definition, we set:

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- **Last column** ($t = T$)

$$\beta_{iT} = P(\underline{\quad\quad} | S_T = i)$$

What does this mean?

Note: β_{it} computes the probability of the future given $S_t = i$.

But we don't see *any* observations beyond time T.
Put another way, the future after time T is unspecified.

What is the probability of some unspecified future occurring?
By definition, we set:

$$\boxed{\beta_{iT} = 1 \quad \textbf{for all} \quad i \in \{1, 2, \ldots, n\}}$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$$
$$=$$

β_{it}

what do I have?

$\boxed{\beta_{it-1}}$ $\boxed{\beta_{it+1}}$

$$\beta_{it+1} = P(o_{t+2} \ldots o_T | S_{t+1} = i)$$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} P(S_{t+1} | S_t = i)\, P(o_{t+1} \ldots o_T | S_t = i, S_{t+1} = j)
\end{aligned}
$$

$$P(\underline{o_{t+1}} | \cancel{S_t = i}, \underline{S_{t+1} = j}, \cancel{o_{t+2} \ldots o_T})$$

$$\cdot\, \underline{P(o_{t+2} \ldots o_T | S_t, S_{t+1})}$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot
\end{aligned}
$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot
\end{aligned}
$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) && \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot \\
&\qquad P(o_{t+2}, \ldots, o_T | S_t = i, S_{t+1} = j, o_{t+1}) \Big] && \boxed{\text{product rule}} \\
&=
\end{aligned}
$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot \\
&\qquad P(o_{t+2}, \ldots, o_T | S_t = i, S_{t+1} = j, o_{t+1}) \Big] \qquad \boxed{\text{product rule}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i)
\end{aligned}$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot \\
&\qquad P(o_{t+2}, \ldots, o_T | S_t = i, S_{t+1} = j, o_{t+1}) \Big] \qquad \boxed{\text{product rule}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \, P(o_{t+1} | S_{t+1} = j)
\end{aligned}
$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot \\
&\qquad P(o_{t+2}, \ldots, o_T | S_t = i, S_{t+1} = j, o_{t+1}) \Big] \qquad \boxed{\text{product rule}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i)\, P(o_{t+1} | S_{t+1} = j)\, P(o_{t+2}, \ldots, o_T | S_{t+1} = j) \Big] \qquad \boxed{\text{CI}} \\
&=
\end{aligned}
$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

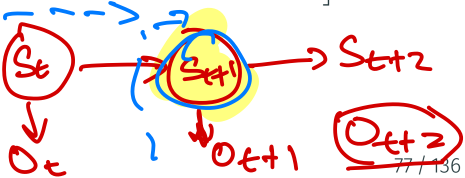
- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \qquad \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot \\
&\qquad P(o_{t+2}, \ldots, o_T | S_t = i, S_{t+1} = j, o_{t+1}) \Big] \qquad \boxed{\text{product rule}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i)\, P(o_{t+1} | S_{t+1} = j)\, P(o_{t+2}, \ldots, o_T | S_{t+1} = j) \Big] \qquad \boxed{\text{CI}} \\
&= \sum_{j=1}^{n} a_{ij}
\end{aligned}
$$

Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) && \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot \\
&\qquad \underline{P(o_{t+2}, \ldots, o_T} | \underline{S_t = i}, S_{t+1} = j, o_{t+1}) \Big] && \boxed{\text{product rule}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i)\, P(o_{t+1} | S_{t+1} = j)\, P(o_{t+2}, \ldots, o_T | S_{t+1} = j) \Big] && \boxed{\text{CI}} \\
&= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})
\end{aligned}
$$



Computing $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$

- Previous columns ($t < T$)

$$
\begin{aligned}
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) \\
&= \sum_{j=1}^{n} P(S_{t+1} = j, o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i) && \boxed{\text{marginalization}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i) \cdot \\
&\qquad P(o_{t+1} | S_t = i, S_{t+1} = j) \cdot \\
&\qquad P(o_{t+2}, \ldots, o_T | S_t = i, S_{t+1} = j, o_{t+1}) \Big] && \boxed{\text{product rule}} \\
&= \sum_{j=1}^{n} \Big[P(S_{t+1} = j | S_t = i)\, P(o_{t+1} | S_{t+1} = j)\, P(o_{t+2}, \ldots, o_T | S_{t+1} = j) \Big] && \boxed{\text{CI}} \\
&= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} && \boxed{\text{CPTs}}
\end{aligned}
$$

Backward algorithm

Backward algorithm

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

$$\boxed{\begin{aligned} \beta_{iT} &= 1 \quad \text{for} \quad i \in \{1, 2, \ldots, n\} \\ \beta_{it} &= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} \end{aligned}}$$

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

$$\boxed{\begin{aligned} \beta_{iT} &= 1 \quad \textbf{for} \quad i \in \{1, 2, \ldots, n\} \\ \beta_{it} &= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} \end{aligned}}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & \boxed{1} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & 1 \end{bmatrix} \right.$$

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

$$\boxed{\begin{aligned} \beta_{iT} &= 1 \quad \textbf{for} \quad i \in \{1, 2, \ldots, n\} \\ \beta_{it} &= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} \end{aligned}}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & 1 \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & 1 \end{bmatrix} \right.$$

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

$$\boxed{\begin{aligned} \beta_{iT} &= 1 \quad \textbf{for} \quad i \in \{1, 2, \dots, n\} \\ \beta_{it} &= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} \end{aligned}}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & 1 \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & 1 \end{bmatrix} \right.$$

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

$$\boxed{\begin{aligned} \beta_{iT} &= 1 \quad \textbf{for} \quad i \in \{1, 2, \ldots, n\} \\ \beta_{it} &= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} \end{aligned}}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & 1 \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & 1 \end{bmatrix} \right.$$

Backward algorithm

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

$$\boxed{\begin{aligned} \beta_{iT} &= 1 \quad \textbf{for} \quad i \in \{1, 2, \ldots, n\} \\ \beta_{it} &= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} \end{aligned}}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & 1 \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & 1 \end{bmatrix} \right.$$

Backward algorithm

The backward algorithm fills in the matrix of β_{it} elements one column at a time:

$$\boxed{\begin{aligned} \beta_{iT} &= 1 \quad \textbf{for} \quad i \in \{1, 2, \ldots, n\} \\ \beta_{it} &= \sum_{j=1}^{n} a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1} \end{aligned}}$$

$$n \text{ rows} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1,T-1} & 1 \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2,T-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{n,T-1} & 1 \end{bmatrix} \right.$$

Learning in HMMs - EM Algorithm

Computing $P(S_t = i | o_1, \ldots, o_T)$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)}$$ product rule

- Numerator

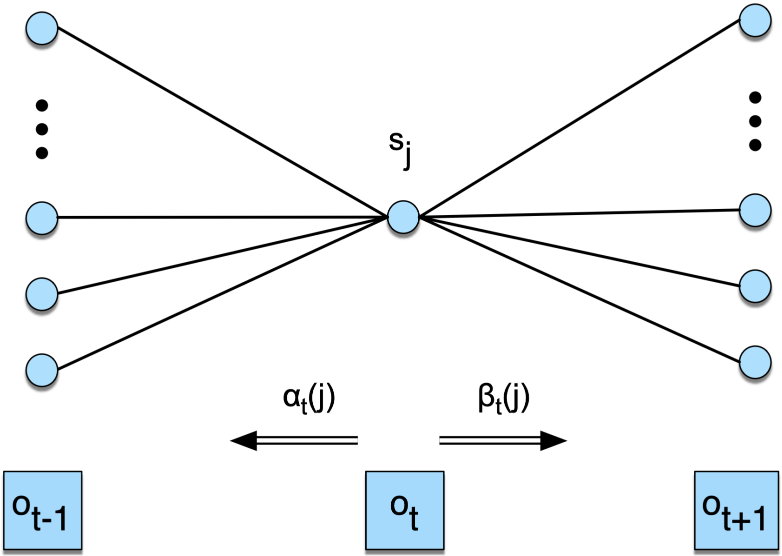
$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) && \text{product rule} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) && \text{conditional independence} \\ &= \alpha_{it}\, P(o_{t+1}, \ldots, o_T | S_t = i) \end{aligned}$$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)}$$ product rule

- Numerator

$$\begin{aligned} P(S_t = i, o_1, o_2, \ldots, o_T) \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) && \text{product rule} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) && \text{conditional independence} \\ &= \alpha_{it}\, P(o_{t+1}, \ldots, o_T | S_t = i) \\ &= \alpha_{it}\, \beta_{it} \end{aligned}$$

Visually



Computing $P(S_t = i | o_1, \ldots, o_T)$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) =$$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \qquad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ & \quad = P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) \qquad \boxed{\text{product rule}} \\ & \quad = P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) \qquad \boxed{\text{conditional independence}} \\ & \quad = \alpha_{it}\, \beta_{it} \end{aligned}$$

- Denominator

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \quad \boxed{\text{product rule}}$$

- **Numerator**

$$P(S_t = i, o_1, o_2, \ldots, o_T)$$

$$\begin{aligned} &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) && \boxed{\text{product rule}} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) && \boxed{\text{conditional independence}} \\ &= \alpha_{it}\, \beta_{it} \end{aligned}$$

- **Denominator**

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

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \qquad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) \qquad \boxed{\text{product rule}} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) \qquad \boxed{\text{conditional independence}} \\ &= \alpha_{it}\, \beta_{it} \end{aligned}$$

- Denominator

$$P(o_1, o_2, \ldots, o_T) = \sum_k P(S_t = k, o_1, o_2, \ldots, o_T) \qquad \boxed{\text{marginalization}}$$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)} \quad \boxed{\text{product rule}}$$

- **Numerator**

$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ = \quad & P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) \quad \boxed{\text{product rule}} \\ = \quad & P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) \quad \boxed{\text{conditional independence}} \\ = \quad & \alpha_{it}\, \beta_{it} \end{aligned}$$

- **Denominator**

$$\begin{aligned} P(o_1, o_2, \ldots, o_T) &= \sum_k P(S_t = k, o_1, o_2, \ldots, o_T) \quad \boxed{\text{marginalization}} \\ &= \sum_k \alpha_{kt}\, \beta_{kt} \quad \boxed{\text{by above}} \end{aligned}$$

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)}$$ product rule

- **Numerator**

$$\begin{aligned} & P(S_t = i, o_1, o_2, \ldots, o_T) \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) && \text{product rule} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) && \text{conditional independence} \\ &= \alpha_{it}\, \beta_{it} \end{aligned}$$

- **Denominator**

$$\begin{aligned} P(o_1, o_2, \ldots, o_T) &= \sum_k P(S_t = k, o_1, o_2, \ldots, o_T) && \text{marginalization} \\ &= \sum_k \alpha_{kt}\, \beta_{kt} && \text{by above} \end{aligned}$$

Note: this holds for all values of t.

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) \; =$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

Express the numerator $P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)$ in terms of α, β, and parameters of the model a, b.

$$P(S_t = i, S_{t+1} = j, o_1, \ldots, o_T) \;= ?$$

How are you progressing?

A. Not sure where to start.

B. Making progress, but not there yet.

C. I got an answer, but I am not sure if it is right. ✓ 40%

D. I finished and feel pretty confident about it.

E. I got lost and wandered off into virtual space.

Computing $P(S_t = i, S_{t+1} = j | o_1, \dots, o_T)$

$$P(S_t = i, S_{t+1} = j \mid O_1, \dots O_T)$$

$$\Rightarrow \frac{P(S_t = i, S_{t+1} = j, O_1 \dots O_T)}{P(O_1 \dots O_T)}$$

$P(O_1 \dots O_T) \rightarrow$ we know this.

$$\underbrace{P(S_t, O_1, \dots, O_t)}_{\alpha_{it}}$$

$$P(S_{t+1}, O_{t+1}, \dots, O_T \mid \quad)$$

$$P(S_{t+1} \mid \quad)$$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) \; =$$

$$\alpha_{it}$$

$$\big(P(S_t, O_1, \ldots, O_t)$$

$$P(S_{t+1}, O_{t+1}, \ldots, O_T | \;)$$

$$P(S_{t+1} | S_t) = P(S_{t+1} = j | S_t = i$$

$$P(O_{t+1} | S_{t+1} = j, S_t = i, O_1, \ldots, O_t)$$

$$P(O_{t+2} \ldots O_T | \;)$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) \; = \; \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \qquad \boxed{\text{product rule}}$$

- Numerator

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}}$$

product rule

- Numerator

$$P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}}$$

product rule

- Numerator

$$P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)$$

$$= \Big[P(o_1, \ldots, o_t, S_t = i) \cdot$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}}$$

product rule

- Numerator

$$P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)$$

$$= \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- Numerator

$$P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)$$

$$= \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot$$

$$P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned} &P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\ &= \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\ &\quad P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\ &\quad P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\text{product rule}} \end{aligned}$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned} & P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\ = \; & \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\ & \quad P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\ & \quad P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\text{product rule}} \\ = \; & P(o_1, \ldots, o_t, S_t = i) \cdot \end{aligned}$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned}
&P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\
&= \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\
&\quad P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\
&\quad P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\text{product rule}} \\
&= P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | S_t = i) \cdot
\end{aligned}$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \dots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \dots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \dots, o_T)}{\underbrace{P(o_1, o_2, \dots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned}
&P(S_t = i, S_{t+1} = j, o_1, o_2, \dots, o_T) \\
&= \Big[P(o_1, \dots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \dots, o_t, S_t = i) \cdot \\
&\quad P(o_{t+1} | o_1, \dots, o_t, S_t = i, S_{t+1} = j) \cdot \\
&\quad P(o_{t+2}, \dots, o_T | o_1, \dots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\text{product rule}} \\
&= P(o_1, \dots, o_t, S_t = i) \cdot P(S_{t+1} = j | S_t = i) \cdot \\
&\quad P(o_{t+1} | S_{t+1} = j) \cdot
\end{aligned}$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- **Numerator**

$$\begin{aligned}
&P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\
&= \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\
&\quad P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\
&\quad P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\text{product rule}} \\
&= P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | S_t = i) \cdot \\
&\quad P(o_{t+1} | S_{t+1} = j) \cdot P(o_{t+2}, \ldots, o_T | S_{t+1} = j) \quad \boxed{\text{conditional independence}}
\end{aligned}$$

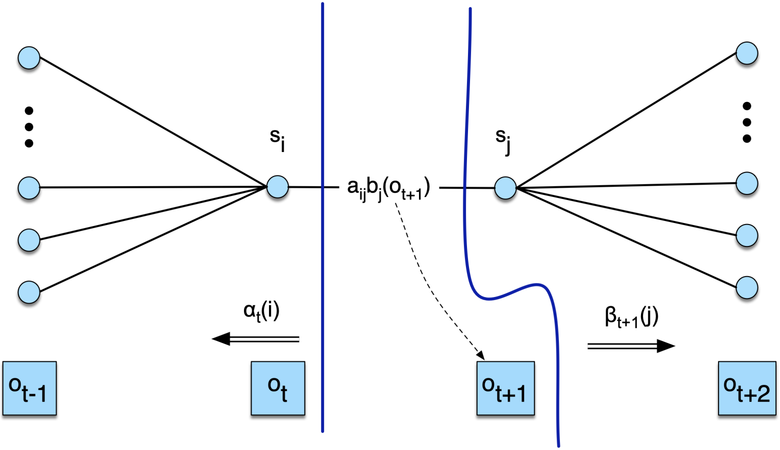
Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- Numerator

$$\begin{aligned}
&P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\
&= \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\
&\quad P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\
&\quad P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\text{product rule}} \\
&= P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | S_t = i) \cdot \\
&\quad P(o_{t+1} | S_{t+1} = j) \cdot P(o_{t+2}, \ldots, o_T | S_{t+1} = j) \quad \boxed{\text{conditional independence}} \\
&= \alpha_{it}\, a_{ij}\, b_j(o_{t+1})\, \beta_{j,t+1}
\end{aligned}$$

Visually



Forward-backward algorithm for inference in HMMs

- Summary of E-step:

$$
\boxed{
\begin{aligned}
P(S_t = i | o_1, \ldots, o_T) &= \frac{\alpha_{it}\,\beta_{it}}{\sum_j \alpha_{jt}\,\beta_{jt}} \\
P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) &= \frac{\alpha_{it}\,a_{ij}\,b_j(o_{t+1})\,\beta_{j,t+1}}{\sum_k \alpha_{kt}\,\beta_{kt}}
\end{aligned}
}
$$

EM algorithm for HMMs

EM algorithm for HMMs

- CPTs to re-estimate:

EM algorithm for HMMs

- CPTs to re-estimate:

$$\pi_i \quad = \quad P(S_1 = i)$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned} \pi_i &= P(S_1 = i) \\ a_{ij} &= P(S_{t+1} = j | S_t = i) \end{aligned}$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$
\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}
$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$
\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}
$$

- M-step updates:

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}\pi_i &= P(S_1=i)\\ a_{ij} &= P(S_{t+1}=j|S_t=i)\\ b_{ik} &= P(O_t=k|S_t=i)\end{aligned}$$

- M-step updates:

$$\pi_i \leftarrow$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}\pi_i &= P(S_1 = i)\\ a_{ij} &= P(S_{t+1} = j | S_t = i)\\ b_{ik} &= P(O_t = k | S_t = i)\end{aligned}$$

- M-step updates:

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

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}$$

- M-step updates:

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

$$a_{ij} \leftarrow$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$
\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}
$$

- M-step updates:

$$
\begin{aligned}
\pi_i &\leftarrow P(S_1 = i | o_1, o_2, \ldots, o_T) \\
a_{ij} &\leftarrow \frac{\sum_t P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)}
\end{aligned}
$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}$$

- M-step updates:

$$\begin{aligned}
\pi_i &\leftarrow P(S_1 = i | o_1, o_2, \ldots, o_T) \\
a_{ij} &\leftarrow \frac{\sum_t P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)} \\
b_{ik} &\leftarrow
\end{aligned}$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned} \pi_i &= P(S_1 = i) \\ a_{ij} &= P(S_{t+1} = j | S_t = i) \\ b_{ik} &= P(O_t = k | S_t = i) \end{aligned}$$

- M-step updates:

$$\begin{aligned} \pi_i &\leftarrow P(S_1 = i | o_1, o_2, \ldots, o_T) \\ a_{ij} &\leftarrow \frac{\sum_t P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)} \\ b_{ik} &\leftarrow \frac{\sum_t I(o_t, k)\, P(S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)} \end{aligned}$$

EM algorithm for HMMs

- CPTs to re-estimate:

$$\begin{aligned}
\pi_i &= P(S_1 = i) \\
a_{ij} &= P(S_{t+1} = j | S_t = i) \\
b_{ik} &= P(O_t = k | S_t = i)
\end{aligned}$$

- M-step updates:

$$\begin{aligned}
\pi_i &\leftarrow P(S_1 = i | o_1, o_2, \ldots, o_T) \\
a_{ij} &\leftarrow \frac{\sum_t P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)} \\
b_{ik} &\leftarrow \frac{\sum_t I(o_t, k)\, P(S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)}
\end{aligned}$$

(for one sequence of observations)

Question

What is the running time for **each** iteration of the update?

A. $O(n)$

B. $O(n^2)$

C. $O(Tn^2)$ why? fwd / bck . . .

D. $O(T^2 n^4)$

E. $O(n^T)$

Time complexity of HMM computations

T	length of observation sequence $(o_1, o_2, \ldots, o_T)$
n	cardinality of state space $s_t \in \{1, 2, \ldots, n\}$
m	cardinality of observation space $o_t \in \{1, 2, \ldots, m\}$

- **All of the following computations are $O(n^2 T)$:**

  (a) computing the likelihood $P(o_1, o_2, \ldots, o_T)$

  (b) decoding $\text{argmax}_{s_1,\ldots,s_T} P(s_1, \ldots, s_T | o_1, \ldots, o_T)$

  (c) re-estimating $\{\pi_i, a_{ij}, b_{ik}\}$ by **one update of EM**

  (d) updating beliefs $P(S_t = i | o_1, \ldots, o_t)$ **for T steps**

That's all folks!