CSCI567 Machine Learning (Spring 2021)

Sirisha Rambhatla

University of Southern California

April 9, 2021

Outline

1. Review of last lecture

2. (Hidden) Markov models II

Logistics for Quiz 2

- There will be 5 questions.
- Only Question 1 will cover cumulative course material, i.e. all topics covered in the class.
- Question 1 will have 5 Multiple Choice Questions (MCQs) and 5 Single CQs.
- Questions 2-5 will be based on material covered after Quiz 1.

Outline

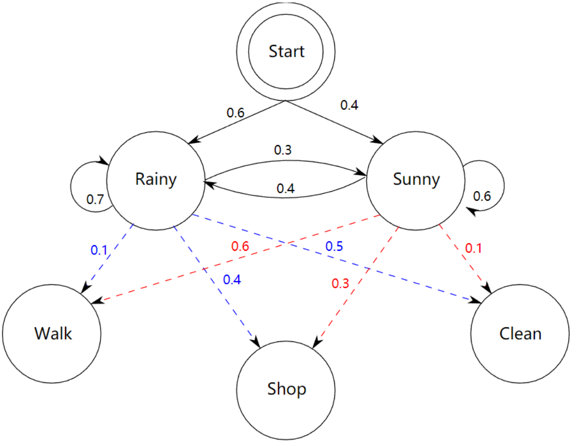
1 Review of last lecture

2 (Hidden) Markov models II

An example

picture from Wikipedia

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



Definition

A **Markov chain** is a stochastic process with **Markov property**: a sequence of random variables $Z_1, Z_2, \cdots$ s.t.

$$P(Z_{t+1} \mid Z_{1:t}) = P(Z_{t+1} \mid Z_t) \qquad \text{(Markov property)}$$

i.e. *the current state only depends on the most recent state* (notation $Z_{1:t}$ denotes the sequence $Z_1, \ldots, Z_t$).

Definition

A **Markov chain** is a stochastic process with **Markov property**: a sequence of random variables $Z_1, Z_2, \cdots$ s.t.

$$P(Z_{t+1} \mid Z_{1:t}) = P(Z_{t+1} \mid Z_t) \qquad \text{(Markov property)}$$

i.e. *the current state only depends on the most recent state* (notation $Z_{1:t}$ denotes the sequence $Z_1, \ldots, Z_t$).

We consider the following setting:

- All Z_t's take value from the same discrete set $\{1, \ldots, S\}$
- $P(Z_1 = s) = \pi_s$ **initial distribution**
- $P(Z_{t+1} = s' \mid Z_t = s) = a_{s,s'}$, known as **transition probability**
- $P(X_t = o \mid Z_t = s) = b_{s,o}$ **emission probability**
- $(\{\pi_s\}, \{a_{s,s'}\}\{b_{s,o}\}) = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ **parameters of the model**

Outline

1. Review of last lecture

2. (Hidden) Markov models II
   - Inferring HMMs
   - Learning HMMs

What can we infer about an HMM?

Knowing the parameter of an HMM, we can infer

What can we infer about an HMM?

Knowing the parameter of an HMM, we can infer

- **the probability of observing some sequence**

$$P(X_{1:T} = x_{1:T})$$

  e.g. prob. of observing Bob's activities "walk, walk, shop, clean, walk, shop, shop" for one week

What can we infer about an HMM?

Knowing the parameter of an HMM, we can infer

- **the probability of observing some sequence**

$$P(X_{1:T} = x_{1:T})$$

  e.g. prob. of observing Bob's activities "walk, walk, shop, clean, walk, shop, shop" for one week

- **the state at some point, given an observation sequence**

$$P(Z_t = s \mid X_{1:T} = x_{1:T})$$

  e.g. given Bob's activities for one week, how was the weather like on Wed?

What can we infer for a known HMM?

Knowing the parameter of an HMM, we can infer

- **the transition at some point, given an observation sequence**

$$P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T})$$

e.g. given Bob's activities for one week, how was the weather like on Wed and Thu?

What can we infer for a known HMM?

Knowing the parameter of an HMM, we can infer

- **the transition at some point, given an observation sequence**

$$P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T})$$

  e.g. given Bob's activities for one week, how was the weather like on Wed and Thu?

- **most likely hidden states path, given an observation sequence**

$$\operatorname*{argmax}_{z_{1:T}} P(Z_{1:T} = z_{1:T} \mid X_{1:T} = x_{1:T})$$

  e.g. given Bob's activities for one week, what's the most likely weather for this week?

Forward and backward messages

The key to infer all these is to compute two things:

Forward and backward messages

The key to infer all these is to compute two things:

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

$$\alpha_s(t) = P(Z_t = s, X_{1:t} = x_{1:t})$$

Forward and backward messages

The key to infer all these is to compute two things:

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

$$\alpha_s(t) = P(Z_t = s, X_{1:t} = x_{1:t})$$

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

$$\beta_s(t) = P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s)$$

Computing forward messages

Key: *establish a recursive formula*

Computing forward messages

Key: *establish a recursive formula*

$$\alpha_s(t)$$
$$= P(Z_t = s, X_{1:t} = x_{1:t})$$

Computing forward messages

Key: *establish a recursive formula*

$$
\begin{aligned}
&\alpha_s(t) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) \\
&= P(X_t = x_t \mid Z_t = s, X_{1:t-1} = x_{1:t-1}) P(Z_t = s, X_{1:t-1} = x_{1:t-1})
\end{aligned}
$$

Computing forward messages

Key: *establish a recursive formula*

$$
\begin{aligned}
&\alpha_s(t) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) \\
&= P(X_t = x_t \mid Z_t = s, X_{1:t-1} = x_{1:t-1}) P(Z_t = s, X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \sum_{s'} P(Z_t = s, Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) \qquad \text{(marginalizing)}
\end{aligned}
$$

Computing forward messages

Key: *establish a recursive formula*

$$
\begin{aligned}
&\alpha_s(t) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) \\
&= P(X_t = x_t \mid Z_t = s, X_{1:t-1} = x_{1:t-1}) P(Z_t = s, X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \sum_{s'} P(Z_t = s, Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) \qquad \text{(marginalizing)} \\
&= b_{s,x_t} \sum_{s'} P(Z_t = s | Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) P(Z_{t-1} = s', X_{1:t-1} = x_{1:t-1})
\end{aligned}
$$

Computing forward messages

Key: *establish a recursive formula*

$$
\begin{aligned}
&\alpha_s(t) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) \\
&= P(X_t = x_t \mid Z_t = s, X_{1:t-1} = x_{1:t-1}) P(Z_t = s, X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \sum_{s'} P(Z_t = s, Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) \qquad \text{(marginalizing)} \\
&= b_{s,x_t} \sum_{s'} P(Z_t = s | Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) P(Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \sum_{s'} a_{s',s} \alpha_{s'}(t-1) \qquad \text{(recursive form!)}
\end{aligned}
$$

Computing forward messages

Key: *establish a recursive formula*

$$
\begin{aligned}
&\alpha_s(t) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) \\
&= P(X_t = x_t \mid Z_t = s, X_{1:t-1} = x_{1:t-1}) P(Z_t = s, X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \sum_{s'} P(Z_t = s, Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) \qquad \text{(marginalizing)} \\
&= b_{s,x_t} \sum_{s'} P(Z_t = s | Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) P(Z_{t-1} = s', X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \sum_{s'} a_{s',s} \alpha_{s'}(t-1) \qquad \text{(recursive form!)}
\end{aligned}
$$

Base case: $\alpha_s(1) = P(Z_1 = s, X_1 = x_1) = \pi_s b_{s,x_1}$

Forward procedure

Forward procedure

For all $s \in [S]$, compute $\alpha_s(1) = \pi_s b_{s,x_1}$.

For $t = 2, \ldots, T$

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

$$\alpha_s(t) = b_{s,x_t} \sum_{s'} a_{s',s} \alpha_{s'}(t-1)$$

Forward procedure

Forward procedure

For all $s \in [S]$, compute $\alpha_s(1) = \pi_s b_{s,x_1}$.

For $t = 2, \ldots, T$

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

$$\alpha_s(t) = b_{s,x_t} \sum_{s'} a_{s',s} \alpha_{s'}(t-1)$$

It takes $O(S^2 T)$ time and $O(ST)$ space.

Computing backward messages

Again establish a recursive formula

$$\beta_s(t)$$
$$= P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s)$$

Computing backward messages

Again establish a recursive formula

$$\begin{aligned}
&\beta_s(t) \\
&= P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s) \\
&= \sum_{s'} P(X_{t+1:T} = x_{t+1:T}, Z_{t+1} = s' \mid Z_t = s) && \text{(marginalizing)}
\end{aligned}$$

Computing backward messages

Again establish a recursive formula

$$
\begin{aligned}
&\beta_s(t) \\
&= P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s) \\
&= \sum_{s'} P(X_{t+1:T} = x_{t+1:T}, Z_{t+1} = s' \mid Z_t = s) && \text{(marginalizing)} \\
&= \sum_{s'} P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s', Z_t = s)
\end{aligned}
$$

Computing backward messages

Again establish a recursive formula

$$
\begin{aligned}
&\beta_s(t) \\
&= P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s) \\
&= \sum_{s'} P(X_{t+1:T} = x_{t+1:T}, Z_{t+1} = s' \mid Z_t = s) \qquad \text{(marginalizing)} \\
&= \sum_{s'} P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s', Z_t = s) \\
&= \sum_{s'} a_{s,s'} P(X_{t+1} = x_{t+1} \mid Z_{t+1} = s') P(X_{t+2:T} = x_{t+2:T} \mid Z_{t+1} = s')
\end{aligned}
$$

Computing backward messages

Again establish a recursive formula

$$
\begin{aligned}
&\beta_s(t) \\
&= P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s) \\
&= \sum_{s'} P(X_{t+1:T} = x_{t+1:T}, Z_{t+1} = s' \mid Z_t = s) && \text{(marginalizing)} \\
&= \sum_{s'} P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s', Z_t = s) \\
&= \sum_{s'} a_{s,s'} P(X_{t+1} = x_{t+1} \mid Z_{t+1} = s') P(X_{t+2:T} = x_{t+2:T} \mid Z_{t+1} = s') \\
&= \sum_{s'} a_{s,s'} b_{s',x_{t+1}} \beta_{s'}(t+1) && \text{(\textit{recursive form!})}
\end{aligned}
$$

Computing backward messages

Again establish a recursive formula

$$
\begin{aligned}
&\beta_s(t) \\
&= P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s) \\
&= \sum_{s'} P(X_{t+1:T} = x_{t+1:T}, Z_{t+1} = s' \mid Z_t = s) && \text{(marginalizing)} \\
&= \sum_{s'} P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s', Z_t = s) \\
&= \sum_{s'} a_{s,s'} P(X_{t+1} = x_{t+1} \mid Z_{t+1} = s') P(X_{t+2:T} = x_{t+2:T} \mid Z_{t+1} = s') \\
&= \sum_{s'} a_{s,s'} b_{s',x_{t+1}} \beta_{s'}(t+1) && \text{(\textit{recursive form!})}
\end{aligned}
$$

Base case: $\beta_s(T) = 1$

Backward procedure

Backward procedure

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

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

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

$$\beta_s(t) = \sum_{s'} a_{s,s'} b_{s',x_{t+1}} \beta_{s'}(t+1)$$

Backward procedure

Backward procedure

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

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

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

$$\beta_s(t) = \sum_{s'} a_{s,s'} b_{s',x_{t+1}} \beta_{s'}(t+1)$$

Again it takes $O(S^2T)$ time and $O(ST)$ space.

Using forward and backward messages

With forward and backward messages, we can easily infer many things,

Using forward and backward messages

With forward and backward messages, we can easily infer many things, e.g.

$$\gamma_s(t) = P(Z_t = s \mid X_{1:T} = x_{1:T})$$

Using forward and backward messages

With forward and backward messages, we can easily infer many things, e.g.

$$\begin{aligned}\gamma_s(t) &= P(Z_t = s \mid X_{1:T} = x_{1:T}) \\ &\propto P(Z_t = s, X_{1:T} = x_{1:T})\end{aligned}$$

Using forward and backward messages

With forward and backward messages, we can easily infer many things, e.g.

$$\begin{aligned}
\gamma_s(t) &= P(Z_t = s \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t})
\end{aligned}$$

Using forward and backward messages

With forward and backward messages, we can easily infer many things, e.g.

$$
\begin{aligned}
\gamma_s(t) &= P(Z_t = s \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t}) \\
&= \alpha_s(t)\beta_s(t)
\end{aligned}
$$

Using forward and backward messages

With forward and backward messages, we can easily infer many things, e.g.

$$\begin{aligned}
\gamma_s(t) &= P(Z_t = s \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t}) \\
&= \alpha_s(t)\beta_s(t)
\end{aligned}$$

What constant are we omitting in "$\propto$"?

Using forward and backward messages

With forward and backward messages, we can easily infer many things, e.g.

$$\begin{aligned}
\gamma_s(t) &= P(Z_t = s \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t}) \\
&= \alpha_s(t)\beta_s(t)
\end{aligned}$$

What constant are we omitting in "$\propto$"? It is exactly

$$P(X_{1:T} = x_{1:T}) = \sum_s \alpha_s(t)\beta_s(t),$$

the probability of observing the sequence $x_{1:T}$.

Using forward and backward messages

With forward and backward messages, we can easily infer many things, e.g.

$$\begin{aligned}\gamma_s(t) &= P(Z_t = s \mid X_{1:T} = x_{1:T}) \\ &\propto P(Z_t = s, X_{1:T} = x_{1:T}) \\ &= P(Z_t = s, X_{1:t} = x_{1:t}) P(X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t}) \\ &= \alpha_s(t)\beta_s(t)\end{aligned}$$

What constant are we omitting in "$\propto$"? It is exactly

$$P(X_{1:T} = x_{1:T}) = \sum_s \alpha_s(t)\beta_s(t),$$

the probability of observing the sequence $x_{1:T}$.

This is true for any t; a good way to check correctness of your code.

Using forward and backward messages

Another example: the conditional probability of transition s to s' at time t

$$\xi_{s,s'}(t)$$
$$= P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T})$$

Using forward and backward messages

Another example: the conditional probability of transition s to s' at time t

$$
\begin{aligned}
&\xi_{s,s'}(t) \\
&= P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, Z_{t+1} = s', X_{1:T} = x_{1:T})
\end{aligned}
$$

Using forward and backward messages

Another example: the conditional probability of transition s to s' at time t

$$
\begin{aligned}
&\xi_{s,s'}(t) \\
&= P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, Z_{t+1} = s', X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(Z_{t+1} = s', X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t})
\end{aligned}
$$

Using forward and backward messages

Another example: the conditional probability of transition s to s' at time t

$$
\begin{aligned}
&\xi_{s,s'}(t) \\
&= P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, Z_{t+1} = s', X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(Z_{t+1} = s', X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t}) \\
&= \alpha_s(t) P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s')
\end{aligned}
$$

Using forward and backward messages

Another example: the conditional probability of transition s to s' at time t

$$
\begin{aligned}
&\xi_{s,s'}(t)\\
&= P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T})\\
&\propto P(Z_t = s, Z_{t+1} = s', X_{1:T} = x_{1:T})\\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(Z_{t+1} = s', X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t})\\
&= \alpha_s(t) P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s')\\
&= \alpha_s(t) a_{s,s'} P(X_{t+1} = x_{t+1} \mid Z_{t+1} = s') P(X_{t+2:T} = x_{t+2:T} \mid Z_{t+1} = s')
\end{aligned}
$$

Using forward and backward messages

Another example: the conditional probability of transition s to s' at time t

$$
\begin{aligned}
&\xi_{s,s'}(t) \\
&= P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, Z_{t+1} = s', X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(Z_{t+1} = s', X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t}) \\
&= \alpha_s(t) P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s') \\
&= \alpha_s(t) a_{s,s'} P(X_{t+1} = x_{t+1} \mid Z_{t+1} = s') P(X_{t+2:T} = x_{t+2:T} \mid Z_{t+1} = s') \\
&= \alpha_s(t) a_{s,s'} b_{s',x_{t+1}} \beta_{s'}(t+1)
\end{aligned}
$$

Using forward and backward messages

Another example: the conditional probability of transition s to s' at time t

$$
\begin{aligned}
&\xi_{s,s'}(t) \\
&= P(Z_t = s, Z_{t+1} = s' \mid X_{1:T} = x_{1:T}) \\
&\propto P(Z_t = s, Z_{t+1} = s', X_{1:T} = x_{1:T}) \\
&= P(Z_t = s, X_{1:t} = x_{1:t}) P(Z_{t+1} = s', X_{t+1:T} = x_{t+1:T} \mid Z_t = s, X_{1:t} = x_{1:t}) \\
&= \alpha_s(t) P(Z_{t+1} = s' \mid Z_t = s) P(X_{t+1:T} = x_{t+1:T} \mid Z_{t+1} = s') \\
&= \alpha_s(t) a_{s,s'} P(X_{t+1} = x_{t+1} \mid Z_{t+1} = s') P(X_{t+2:T} = x_{t+2:T} \mid Z_{t+1} = s') \\
&= \alpha_s(t) a_{s,s'} b_{s',x_{t+1}} \beta_{s'}(t+1)
\end{aligned}
$$

The normalization constant is in fact again $P(X_{1:T} = x_{1:T})$

Decoding: Finding the most likely path

Though can't use forward and backward messages directly to find the most likely path, it is very similar to the forward procedure.

Decoding: Finding the most likely path

Though can't use forward and backward messages directly to find the most likely path, it is very similar to the forward procedure. Key: compute

$$\delta_s(t) = \max_{z_{1:t-1}} P(Z_t = s, Z_{1:t-1} = z_{1:t-1}, X_{1:t} = x_{1:t})$$

the probability of the most likely path for time $1:t$ ending at state s

Computing $\delta_s(t)$

Observe

$$\delta_s(t) = \max_{z_{1:t-1}} P(Z_t = s, Z_{1:t-1} = z_{1:t-1}, X_{1:t} = x_{1:t})$$

Computing $\delta_s(t)$

Observe

$$\begin{aligned}
\delta_s(t) &= \max_{z_{1:t-1}} P(Z_t = s, Z_{1:t-1} = z_{1:t-1}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} \max_{z_{1:t-2}} P(Z_t = s, Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t} = x_{1:t})
\end{aligned}$$

Computing $\delta_s(t)$

Observe

$$
\begin{aligned}
\delta_s(t) &= \max_{z_{1:t-1}} P(Z_t = s, Z_{1:t-1} = z_{1:t-1}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} \max_{z_{1:t-2}} P(Z_t = s, Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} P(Z_t = s \mid Z_{t-1} = s') P(X_t = x_t \mid Z_t = s) \cdot \\
&\qquad\qquad \max_{z_{1:t-2}} P(Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t-1} = x_{1:t-1})
\end{aligned}
$$

Computing $\delta_s(t)$

Observe

$$
\begin{aligned}
\delta_s(t) &= \max_{z_{1:t-1}} P(Z_t = s, Z_{1:t-1} = z_{1:t-1}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} \max_{z_{1:t-2}} P(Z_t = s, Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} P(Z_t = s \mid Z_{t-1} = s') P(X_t = x_t \mid Z_t = s) \cdot \\
&\qquad\qquad \max_{z_{1:t-2}} P(Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \max_{s'} a_{s',s} \delta_{s'}(t-1) \qquad\qquad (\textit{recursive form!})
\end{aligned}
$$

Computing $\delta_s(t)$

Observe

$$\begin{aligned}
\delta_s(t) &= \max_{z_{1:t-1}} P(Z_t = s, Z_{1:t-1} = z_{1:t-1}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} \max_{z_{1:t-2}} P(Z_t = s, Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} P(Z_t = s \mid Z_{t-1} = s') P(X_t = x_t \mid Z_t = s) \cdot \\
&\qquad \max_{z_{1:t-2}} P(Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \max_{s'} a_{s',s} \delta_{s'}(t-1) && (\textit{recursive form!})
\end{aligned}$$

Base case: $\delta_s(1) = P(Z_1 = s, X_1 = x_1) = \pi_s b_{s,x_1}$

Computing $\delta_s(t)$

Observe

$$
\begin{aligned}
\delta_s(t) &= \max_{z_{1:t-1}} P(Z_t = s, Z_{1:t-1} = z_{1:t-1}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} \max_{z_{1:t-2}} P(Z_t = s, Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t} = x_{1:t}) \\
&= \max_{s'} P(Z_t = s \mid Z_{t-1} = s') P(X_t = x_t \mid Z_t = s) \cdot \\
&\qquad\qquad \max_{z_{1:t-2}} P(Z_{t-1} = s', Z_{1:t-2} = z_{1:t-2}, X_{1:t-1} = x_{1:t-1}) \\
&= b_{s,x_t} \max_{s'} a_{s',s} \delta_{s'}(t-1) \qquad\qquad (\textit{recursive form!})
\end{aligned}
$$

Base case: $\delta_s(1) = P(Z_1 = s, X_1 = x_1) = \pi_s b_{s,x_1}$

Exactly the same as forward messages except replacing "sum" by "max"!

Viterbi Algorithm (!)

Viterbi Algorithm

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

Viterbi Algorithm (!)

Viterbi Algorithm

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

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

Viterbi Algorithm (!)

Viterbi Algorithm

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

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

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

Viterbi Algorithm (!)

Viterbi Algorithm

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

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

Backtracking: let $z_T^* = \operatorname{argmax}_s \delta_s(T)$.

Viterbi Algorithm (!)

Viterbi Algorithm

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

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

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

Backtracking: let $z_T^* = \operatorname{argmax}_s \delta_s(T)$.
For each $t = T, \dots, 2$: set $z_{t-1}^* = \Delta_{z_t^*}(t)$.

Viterbi Algorithm (!)

Viterbi Algorithm

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

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

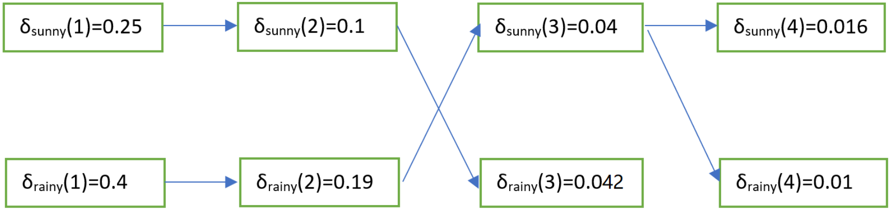
$$\Delta_s(t) = \operatorname*{argmax}_{s'} a_{s',s} \delta_{s'}(t-1).$$

Backtracking: let $z_T^* = \mathrm{argmax}_s\, \delta_s(T)$.
For each $t = T, \ldots, 2$: set $z_{t-1}^* = \Delta_{z_t^*}(t)$.

Output the most likely path $z_1^*, \ldots, z_T^*$.

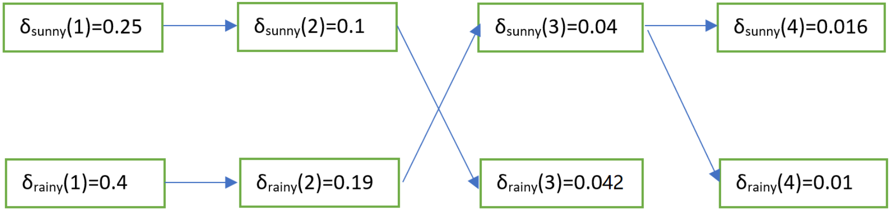
Example

Arrows represent the "argmax", i.e. $\Delta_s(t)$.



Example

Arrows represent the "argmax", i.e. $\Delta_s(t)$.



The most likely path is **"rainy, rainy, sunny, sunny"**.

Exercise 1

What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T_0}$ for some $T_0 < T$?

Exercise 1

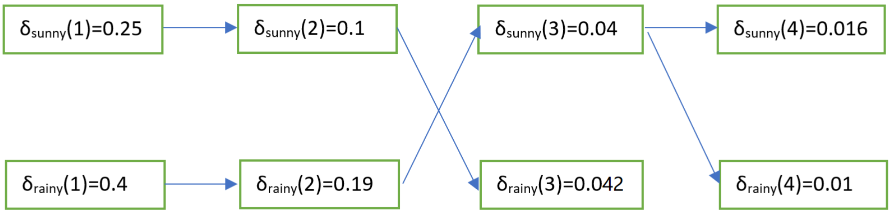
What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T_0}$ for some $T_0 < T$?

- Is it the first T_0 outputs of the Viterbi algorithm (with all data)?

Exercise 1

What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T_0}$ for some $T_0 < T$?

- Is it the first T_0 outputs of the Viterbi algorithm (with all data)?



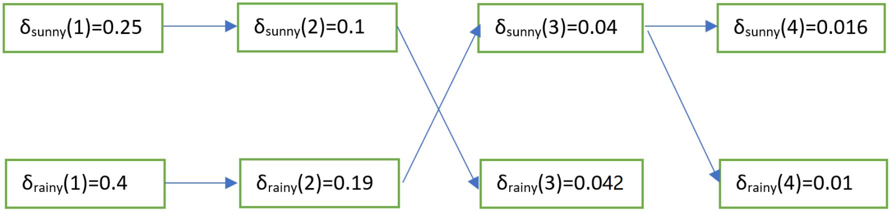
Exercise 1

What is the most likely sequence $z^*_{\color{red}{1:T_0}}$ given $x_{\color{red}{1:T_0}}$ for some $T_0 < T$?

- Is it the first T_0 outputs of the Viterbi algorithm (with all data)?

No. It should be

- $z^*_{T_0} = \operatorname{argmax}_s \delta_s(\color{red}{T_0})$
- for each $t = T_0, \ldots, 2$: $z^*_{t-1} = \Delta_{z^*_t}(t)$



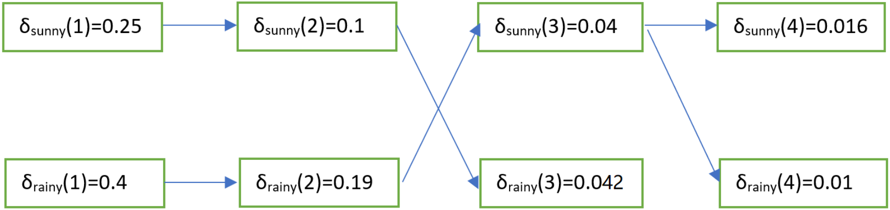
Exercise 1

What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T_0}$ for some $T_0 < T$?

- Is it the first T_0 outputs of the Viterbi algorithm (with all data)?

No. It should be

- $z^*_{T_0} = \operatorname{argmax}_s \delta_s(T_0)$
- for each $t = T_0, \ldots, 2$: $z^*_{t-1} = \Delta_{z^*_t}(t)$



The answer for $T_0 = 3$ is: **"sunny, sunny, rainy"**.

Exercise 2

What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T}$ for some $T_0 < T$?

Exercise 2

What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T}$ for some $T_0 < T$?

- Is it the same as Exercise 1?

Exercise 2

What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T}$ for some $T_0 < T$?

- Is it the same as Exercise 1?
- Is it the first T_0 outputs of the Viterbi algorithm (with all data)?

Exercise 2

What is the most likely sequence $z^*_{1:T_0}$ given $x_{1:T}$ for some $T_0 < T$?

- Is it the same as Exercise 1?
- Is it the first T_0 outputs of the Viterbi algorithm (with all data)?

Neither. It should be

- $z^*_{T_0} = \mathrm{argmax}_s\, \delta_s(T_0)\beta_s(T_0)$
- for each $t = T_0, \ldots, 2$: $z^*_{t-1} = \Delta_{z^*_t}(t)$

Exercise 2 (cont.)

Reasoning:

$$z_{T_0}^* = \operatorname*{argmax}_{s} \max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T} = x_{1:T})$$

Exercise 2 (cont.)

Reasoning:

$$
\begin{aligned}
z^*_{T_0} &= \operatorname*{argmax}_s \max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T} = x_{1:T}) \\
&= \operatorname*{argmax}_s \max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0}) \cdot \\
&\qquad P(X_{T_0+1,T} = x_{T_0+1:T} \mid Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0})
\end{aligned}
$$

Exercise 2 (cont.)

Reasoning:

$$
\begin{aligned}
z_{T_0}^* &= \operatorname*{argmax}_s \max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T} = x_{1:T}) \\
&= \operatorname*{argmax}_s \max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0}) \cdot \\
&\qquad P(X_{T_0+1,T} = x_{T_0+1:T} \mid Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0}) \\
&= \operatorname*{argmax}_s \left(\max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0}) \right) \cdot \\
&\qquad P(X_{T_0+1,T} = x_{T_0+1:T} \mid Z_{T_0} = s)
\end{aligned}
$$

Exercise 2 (cont.)

Reasoning:

$$
\begin{aligned}
z^*_{T_0} &= \operatorname*{argmax}_s \max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T} = x_{1:T}) \\
&= \operatorname*{argmax}_s \max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0}) \cdot \\
&\qquad P(X_{T_0+1,T} = x_{T_0+1:T} \mid Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0}) \\
&= \operatorname*{argmax}_s \left(\max_{z_{1:T_0-1}} P(Z_{T_0} = s, Z_{1:T_0-1} = z_{1:T_0-1}, X_{1:T_0} = x_{1:T_0}) \right) \cdot \\
&\qquad P(X_{T_0+1,T} = x_{T_0+1:T} \mid Z_{T_0} = s) \\
&= \operatorname*{argmax}_s \delta_s(T_0) \beta_s(T_0)
\end{aligned}
$$

Exercise 3

What is the most likely sequence $z^*_{1:T}$ given $x_{1:T_0}$ for some $T_0 < T$?

Exercise 3

What is the most likely sequence $z^*_{1:T}$ given $x_{1:T_0}$ for some $T_0 < T$?

- Is it the same as the Viterbi algorithm (with all data)?

Exercise 3

What is the most likely sequence $z^*_{1:T}$ given $x_{1:T_0}$ for some $T_0 < T$?

- Is it the same as the Viterbi algorithm (with all data)?
- Are the first T_0 states the same as Exercise 1?

Exercise 3

What is the most likely sequence $z^*_{1:T}$ given $x_{1:T_0}$ for some $T_0 < T$?

- Is it the same as the Viterbi algorithm (with all data)?
- Are the first T_0 states the same as Exercise 1?

Again, neither is true.

Exercise 3 (cont.)

Viterbi Algorithm with partial data $x_{1:T_0}$

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

Exercise 3 (cont.)

Viterbi Algorithm with partial data $x_{1:T_0}$

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

$$\delta_s(t) = \begin{cases} b_{s,x_t} \max_{s'} a_{s',s}\delta_{s'}(t-1) & \text{if } t \leq T_0 \\ \end{cases}$$

Exercise 3 (cont.)

Viterbi Algorithm with partial data $x_{1:T_0}$

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

$$\delta_s(t) = \begin{cases} b_{s,x_t} \max_{s'} a_{s',s}\delta_{s'}(t-1) & \text{if } t \leq T_0 \\ \max_{s'} a_{s',s}\delta_{s'}(t-1) & \text{else} \end{cases}$$

Exercise 3 (cont.)

Viterbi Algorithm with partial data $x_{1:T_0}$

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

$$\delta_s(t) = \begin{cases} b_{s,x_t} \max_{s'} a_{s',s}\delta_{s'}(t-1) & \text{if } t \leq T_0 \\ \max_{s'} a_{s',s}\delta_{s'}(t-1) & \text{else} \end{cases}$$

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

Exercise 3 (cont.)

Viterbi Algorithm with partial data $x_{1:T_0}$

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

$$\delta_s(t) = \begin{cases} b_{s,x_t} \max_{s'} a_{s',s} \delta_{s'}(t-1) & \text{if } t \leq T_0 \\ \max_{s'} a_{s',s} \delta_{s'}(t-1) & \text{else} \end{cases}$$

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

Backtracking: let $z_T^* = \operatorname{argmax}_s \delta_s(T)$.

Exercise 3 (cont.)

Viterbi Algorithm with partial data $x_{1:T_0}$

For each $s \in [S]$, compute $\delta_s(1) = \pi_s b_{s,x_1}$.

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

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

$$\delta_s(t) = \begin{cases} b_{s,x_t} \max_{s'} a_{s',s}\delta_{s'}(t-1) & \text{if } t \leq T_0 \\ \max_{s'} a_{s',s}\delta_{s'}(t-1) & \text{else} \end{cases}$$

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

Backtracking: let $z_T^* = \operatorname{argmax}_s \delta_s(T)$.
For each $t = T, \dots, 2$: set $z_{t-1}^* = \Delta_{z_t^*}(t)$.

Output the most likely path $z_1^*, \dots, z_T^*$.

Learning the parameters of an HMM

All previous inferences depend on knowing the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.

Learning the parameters of an HMM

All previous inferences depend on knowing the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.

How do we learn the parameters based on N observation sequences $x_{n,1}, \ldots, x_{n,T}$ for $n = 1, \ldots, N$?

Learning the parameters of an HMM

All previous inferences depend on knowing the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.

How do we learn the parameters based on N observation sequences $x_{n,1}, \ldots, x_{n,T}$ for $n = 1, \ldots, N$?

MLE is intractable due to the hidden variables $Z_{n,t}$'s (similar to GMMs)

Learning the parameters of an HMM

All previous inferences depend on knowing the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.

How do we learn the parameters based on N observation sequences $x_{n,1}, \ldots, x_{n,T}$ for $n = 1, \ldots, N$?

MLE is intractable due to the hidden variables $Z_{n,t}$'s (similar to GMMs)

Need to apply **EM** again! Known as the **Baum–Welch algorithm**.

Applying EM: E-Step

Recall in the E-Step we fix the parameters and find the **posterior distributions q of the hidden states** (for each sample n),

Applying EM: E-Step

Recall in the E-Step we fix the parameters and find the **posterior distributions q of the hidden states** (for each sample n), which leads to the complete log-likelihood:

$$\mathbb{E}_{z_{1:T}\sim q}\left[\ln P(Z_{1:T} = z_{1:T}, X_{1:T} = x_{1:T})\right]$$

Applying EM: E-Step

Recall in the E-Step we fix the parameters and find the **posterior distributions q of the hidden states** (for each sample n), which leads to the complete log-likelihood:

$$\mathbb{E}_{z_{1:T}\sim q}\left[\ln P(Z_{1:T}=z_{1:T}, X_{1:T}=x_{1:T})\right]$$
$$=\mathbb{E}_{z_{1:T}\sim q}\left[\ln \pi_{z_1}+\sum_{t=1}^{T-1}\ln a_{z_t,z_{t+1}}+\sum_{t=1}^{T}\ln b_{z_t,x_t}\right]$$

Applying EM: E-Step

Recall in the E-Step we fix the parameters and find the **posterior distributions q of the hidden states** (for each sample n), which leads to the complete log-likelihood:

$$
\begin{aligned}
&\mathbb{E}_{z_{1:T}\sim q}\left[\ln P(Z_{1:T}=z_{1:T}, X_{1:T}=x_{1:T})\right] \\
&= \mathbb{E}_{z_{1:T}\sim q}\left[\ln \pi_{z_1} + \sum_{t=1}^{T-1} \ln a_{z_t,z_{t+1}} + \sum_{t=1}^{T} \ln b_{z_t,x_t}\right] \\
&= \sum_{s} \gamma_s(1)\ln \pi_s + \sum_{t=1}^{T-1}\sum_{s,s'} \xi_{s,s'}(t)\ln a_{s,s'} + \sum_{t=1}^{T}\sum_{s}\gamma_s(t)\ln b_{s,x_t}
\end{aligned}
$$

We have discussed how to compute

$$
\begin{aligned}
\gamma_s(t) &= P(Z_t = s \mid X_{1:T}=x_{1:T}) \\
\xi_{s,s'}(t) &= P(Z_t = s, Z_{t+1}=s' \mid X_{1:T}=x_{1:T})
\end{aligned}
$$

Applying EM: M-Step

The maximizer of complete log-likelihood is simply doing **weighted counting** (compared to the unweighted counting on Slide 18 Lecture 21):

$$\pi_s \propto \sum_n \gamma_s^{(n)}(1) = \mathbb{E}_q \left[\textbf{\#initial states with value } s \right]$$

$$a_{s,s'} \propto \sum_n \sum_{t=1}^{T-1} \xi_{s,s'}^{(n)}(t) = \mathbb{E}_q \left[\textbf{\#transitions from } s \textbf{ to } s' \right]$$

$$b_{s,o} \propto \sum_n \sum_{t:x_t=o} \gamma_s^{(n)}(t) = \mathbb{E}_q \left[\textbf{\#state-outcome pairs } (s,o) \right]$$

Applying EM: M-Step

The maximizer of complete log-likelihood is simply doing **weighted counting** (compared to the unweighted counting on Slide 18 Lecture 21):

$$\pi_s \propto \sum_n \gamma_s^{(n)}(1) = \mathbb{E}_q\left[\ \textbf{\#initial states with value } s\right]$$

$$a_{s,s'} \propto \sum_n \sum_{t=1}^{T-1} \xi_{s,s'}^{(n)}(t) = \mathbb{E}_q\left[\ \textbf{\#transitions from } s \textbf{ to } s'\right]$$

$$b_{s,o} \propto \sum_n \sum_{t:x_t=o} \gamma_s^{(n)}(t) = \mathbb{E}_q\left[\ \textbf{\#state-outcome pairs } (s,o)\right]$$

where

$$\gamma_s^{(n)}(t) = P(Z_{n,t} = s \mid X_{n,1:T} = x_{n,1:T})$$

$$\xi_{s,s'}^{(n)}(t) = P(Z_{n,t} = s, Z_{n,t+1} = s' \mid X_{n,1:T} = x_{n,1:T})$$

Slide 18 Lecture 21: Learning the model

If we observe N state-outcome sequences: $z_{n,1}, x_{n,1}, \ldots, z_{n,T}, x_{n,T}$ for $n = 1, \ldots, N$, the MLE can again be obtained in a similar way (verify yourself):

$$\pi_s \propto \textbf{\#initial states with value } s$$
$$a_{s,s'} \propto \textbf{\#transitions from } s \textbf{ to } s'$$
$$b_{s,o} \propto \textbf{\#state-outcome pairs } (s, o)$$

Baum–Welch algorithm

Step 0 Initialize the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$

Baum–Welch algorithm

Step 0 Initialize the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$

Step 1 (E-Step) Fixing the parameters, compute forward and backward messages for all sample sequences, then use these to compute $\gamma_s^{(n)}(t)$ and $\xi_{s,s'}^{(n)}(t)$ for each n, t, s, s' (see Slides 15 and 16).

Baum–Welch algorithm

Step 0 Initialize the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$

Step 1 (E-Step) Fixing the parameters, compute forward and backward messages for all sample sequences, then use these to compute $\gamma_s^{(n)}(t)$ and $\xi_{s,s'}^{(n)}(t)$ for each n, t, s, s' (see Slides 15 and 16).

Step 2 (M-Step) Update parameters:

$$\pi_s \propto \sum_n \gamma_s^{(n)}(1), \quad a_{s,s'} \propto \sum_n \sum_{t=1}^{T-1} \xi_{s,s'}^{(n)}(t), \quad b_{s,o} \propto \sum_n \sum_{t:x_t=o} \gamma_s^{(n)}(t)$$

Baum–Welch algorithm

Step 0 Initialize the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$

Step 1 (E-Step) Fixing the parameters, compute forward and backward messages for all sample sequences, then use these to compute $\gamma_s^{(n)}(t)$ and $\xi_{s,s'}^{(n)}(t)$ for each n, t, s, s' (see Slides 15 and 16).

Step 2 (M-Step) Update parameters:

$$\pi_s \propto \sum_n \gamma_s^{(n)}(1), \quad a_{s,s'} \propto \sum_n \sum_{t=1}^{T-1} \xi_{s,s'}^{(n)}(t), \quad b_{s,o} \propto \sum_n \sum_{t:x_t=o} \gamma_s^{(n)}(t)$$

Step 3 Return to Step 1 if not converged

Summary

Very important models: **Markov chains**, **hidden Markov models**

Summary

Very important models: **Markov chains**, **hidden Markov models**

Several algorithms:

- forward and backward procedures
- inferring HMMs based on forward and backward messages
- Viterbi algorithm
- Baum–Welch algorithm

Additional Resources:

- `https://web.stanford.edu/~jurafsky/slp3/A.pdf`
- MLaPP 17.3