Hidden Markov Models

11-711: Algorithms for NLP

Fall 2017

Table of Contents

Outline

CarnegieMellon

Notations

CarnegieMellon

- $\mathbb{R}$: set of real numbers
- Cartesian products:
  - $\mathbb{R}^{D_1 \times D_2}$: set of matrices of size $D_1 \times D_2$, with real entries

Notations

Carnegie Mellon

- $\mathbb{R}$: set of real numbers
- Cartesian products:
  - $\mathbb{R}^{D_1 \times D_2}$: set of matrices of size $D_1 \times D_2$, with real entries
- Vectors:
  - Lower case: a, b, c, ...
  - Row major: $a \in \mathbb{R}^D$ means $a \in \mathbb{R}^{1 \times D}$

Notations

CarnegieMellon

- $\mathbb{R}$: set of real numbers
- Cartesian products:
  - $\mathbb{R}^{D_1 \times D_2}$: set of matrices of size $D_1 \times D_2$, with real entries
- Vectors:
  - Lower case: a, b, c, ...
  - Row major: $a \in \mathbb{R}^D$ means $a \in \mathbb{R}^{1 \times D}$
- Matrices:
  - Upper case: A, B, C, ...
  - Row major

Outline

Hidden Markov Model

Hidden Markov Model: Notations

Carnegie Mellon

Name	Notation	Meaning/Property

Hidden Markov Model: Notations

Carnegie Mellon

Name	Notation	Meaning/Property
State space	$Q = \{q_1, q_2, ..., q_N\}$	Set of N states
Observation space	$V = \{w_1, w_2, ..., w_V\}$	Set of V states

Hidden Markov Model: Notations

Carnegie Mellon

Name	Notation	Meaning/Property
State space	$Q = \{q_1, q_2, ..., q_N\}$	Set of N states
Observation space	$V = \{w_1, w_2, ..., w_V\}$	Set of V states
State sequence	$S = \{s_1, s_2, ..., s_T\}$	Sequence of T steps. $s_i \in Q$
Observation sequence	$O = \{o_1, o_2, ..., o_T\}$	Sequence of T steps. $o_i \in V$

Hidden Markov Model: Notations

Carnegie Mellon

Name	Notation	Meaning/Property
State space	$Q = \{q_1, q_2, ..., q_N\}$	Set of N states
Observation space	$V = \{w_1, w_2, ..., w_V\}$	Set of V states
State sequence	$S = \{s_1, s_2, ..., s_T\}$	Sequence of T steps. $s_i \in Q$
Observation sequence	$O = \{o_1, o_2, ..., o_T\}$	Sequence of T steps. $o_i \in V$
Transition probs	$A \in \mathbb{R}^{N \times N}$	Each row is a valid distribution

Valid (probability) distribution:

- $A(i, j) \geq 0$
- $\sum_{j=1}^{N} A(i, j) = 1$

Hidden Markov Model: Notations

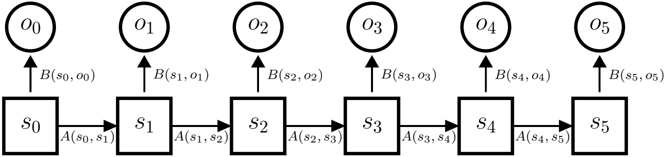
Carnegie Mellon

Name	Notation	Meaning/Property
State space	$Q = \{q_1, q_2, ..., q_N\}$	Set of N states
Observation space	$V = \{w_1, w_2, ..., w_V\}$	Set of V states
State sequence	$S = \{s_1, s_2, ..., s_T\}$	Sequence of T steps. $s_i \in Q$
Observation sequence	$O = \{o_1, o_2, ..., o_T\}$	Sequence of T steps. $o_i \in V$
Transition probs	$A \in \mathbb{R}^{N \times N}$	Each row is a valid distribution
Emission probs	$B \in \mathbb{R}^{N \times V}$	Each row is a valid distribution

Valid (probability) distribution:

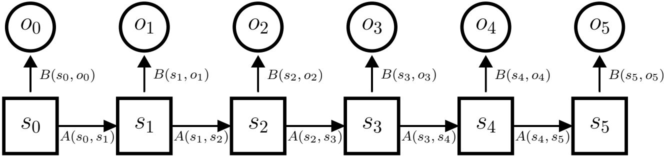
- $A(i, j) \geq 0$
- $\sum_{j=1}^{N} A(i, j) = 1$

Hidden Markov Model: Assumptions



Hidden Markov Model: Assumptions

Carnegie Mellon

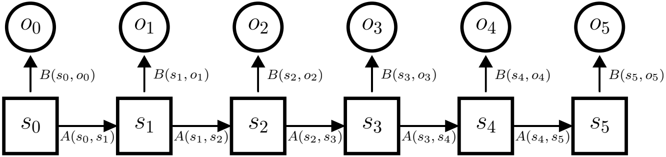


- Markov assumption

$$P(s_0, s_1, s_2, ..., s_T) = \prod_{t=1}^{T} P(s_t | s_{<t}) = \prod_{t=1}^{T} P(s_t | s_{t-1}) \stackrel{\text{def}}{=} \prod_{t=1}^{T} A(s_{t-1}, s_t)$$

Hidden Markov Model: Assumptions

Carnegie Mellon



- Markov assumption

$$P(s_0, s_1, s_2, ..., s_T) = \prod_{t=1}^{T} P(s_t | s_{<t}) = \prod_{t=1}^{T} P(s_t | s_{t-1}) \stackrel{\text{def}}{=} \prod_{t=1}^{T} A(s_{t-1}, s_t)$$

- Independent assumption

$$P(o_t | o_{<t}, s_{\leq t}) = P(o_t | s_t) \stackrel{\text{def}}{=} B(s_t, o_t)$$

More Notations

- Prior: $P(A, B)$
  - Without seeing anything, how do you believe A, B should look like?

More Notations

- Prior: $P(A, B)$
  - Without seeing anything, how do you believe A, B should look like?
- Likelihood: $P(O|A, B)$
  - Suppose you know A, B, how likely do you see O?

More Notations

Carnegie Mellon

- Prior: $P(A, B)$
  - Without seeing anything, how do you believe A, B should look like?
- Likelihood: $P(O|A, B)$
  - Suppose you know A, B, how likely do you see O?
- Posterior: $P(A, B|O)$
  - Suppose you see O, how do you believe A, B should look like?

More Notations

Carnegie Mellon

- Prior: $P(A, B)$
  - Without seeing anything, how do you believe A, B should look like?
- Likelihood: $P(O|A, B)$
  - Suppose you know A, B, how likely do you see O?
- Posterior: $P(A, B|O)$
  - Suppose you see O, how do you believe A, B should look like?
- Bayes rule

$$P(A, B|O) = \frac{P(O|A, B) \cdot P(A, B)}{P(O)}$$

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{data\_distribution}}$$

3 Questions with HMM

- Settings:
  - Prior $P(A, B)$ are uniform.
  - $P(O)$ is unknown, but O will be observed.

3 Questions with HMM

- Settings:
  - Prior $P(A, B)$ are uniform.
  - $P(O)$ is unknown, but O will be observed.
- **Question 1:** Compute the likelihood: $P(O|A, B)$

3 Questions with HMM

Carnegie Mellon

- Settings:
  - Prior $P(A, B)$ are uniform.
  - $P(O)$ is unknown, but O will be observed.
- **Question 1:** Compute the likelihood: $P(O|A, B)$
- **Question 2:** (only for HMMs) Find the hidden sequence S

$$S^* = \operatorname*{argmax}_S P(O|S, A, B)$$

3 Questions with HMM

- Settings:
  - Prior $P(A, B)$ are uniform.
  - $P(O)$ is unknown, but O will be observed.
- **Question 1:** Compute the likelihood: $P(O|A, B)$
- **Question 2:** (only for HMMs) Find the hidden sequence S

$$S^* = \operatorname*{argmax}_S P(O|S, A, B)$$

- **Question 3:** Find the posterior

$$A^*, B^* = \operatorname*{argmax}_{A,B} P(A, B|O)$$

Outline

CarnegieMellon

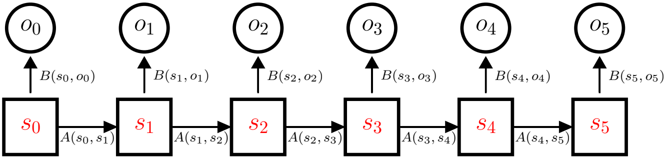
1. Notations
2. Hidden Markov Model
3. Computing the Likelihood: Forward-Pass Algorithm
4. Finding the Hidden Sequence: Viterbi Algorithm
5. Estimating Parameters: Baum-Welch Algorithm

Question

Compute the likelihood: $P(o_1, o_2, ..., o_T | A, B)$

Observation

O depends on the hidden sequence S



Compute $P(O|A, B)$ for a particular S

Carnegie Mellon

$$S = (s_1, s_2, ..., s_T)$$

$$\begin{aligned} P(O|S, A, B) &= P(o_1|s_1)P(o_2|s_2) \cdots P(o_T|s_T) \\ &= \prod_{i=1}^{T} P(o_i|s_i) \end{aligned}$$

What does Bayes say?

Carnegie Mellon

$$P(O|A,B) = \sum_S P(O|S,A,B) \cdot P(S)$$

What does Bayes say?

$$P(O|A,B) = \sum_{S} P(O|S,A,B) \cdot P(S)$$
$$= \sum_{s_1,\ldots,s_T} \left(\prod_{i=1}^{T} P(o_i|s_i) \cdot \prod_{i=1}^{T} P(s_i|s_{i-1}) \right)$$

What does Bayes say?

Carnegie Mellon

$$
\begin{aligned}
P(O|A,B) &= \sum_S P(O|S,A,B) \cdot P(S) \\
&= \sum_{s_1,\ldots,s_T} \left(\prod_{i=1}^{T} P(o_i|s_i) \cdot \prod_{i=1}^{T} P(s_i|s_{i-1}) \right) \\
&= \sum_{s_1,\ldots,s_T} \left(\prod_{i=1}^{T} B(s_i, o_i) \cdot \prod_{i=1}^{T} A(s_{i-1}, s_i) \right)
\end{aligned}
$$

What does Bayes say?

$$
\begin{aligned}
P(O|A,B) &= \sum_{S} P(O|S,A,B) \cdot P(S) \\
&= \sum_{s_1,\ldots,s_T} \left(\prod_{i=1}^{T} P(o_i|s_i) \cdot \prod_{i=1}^{T} P(s_i|s_{i-1}) \right) \\
&= \sum_{s_1,\ldots,s_T} \left(\prod_{i=1}^{T} B(s_i, o_i) \cdot \prod_{i=1}^{T} A(s_{i-1}, s_i) \right)
\end{aligned}
$$

But there are $\Theta(N^T)$ possible sequences S

Solution: Dynamic Programming

- Compute harder instances based on easier instances
- Cache easier instances

Solution: Dynamic Programming

CarnegieMellon

- $T = 1$

$$P(o_1|A,B) = \sum_{s_1} B(s_1, o_1) \cdot A(s_0, s_1)$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 1$

$$P(o_1|A,B) = \sum_{s_1} B(s_1,o_1) \cdot A(s_0,s_1)$$

- $T = 2$

$$P(o_1,o_2|A,B) = \sum_{s_2}\sum_{s_1} B(s_1,o_1)B(s_2,o_2) \cdot A(s_0,s_1)A(s_1,s_2)$$

Solution: Dynamic Programming

- $T = 1$

$$P(o_1|A,B) = \sum_{s_1} B(s_1,o_1) \cdot A(s_0,s_1)$$

- $T = 2$

$$P(o_1,o_2|A,B) = \sum_{s_2}\sum_{s_1} B(s_1,o_1)B(s_2,o_2) \cdot A(s_0,s_1)A(s_1,s_2)$$

$$= \sum_{s_2} B(s_2,o_2) \left(\sum_{s_1} B(s_1,o_1) \cdot A(s_0,s_1)A(s_1,s_2) \right)$$

Solution: Dynamic Programming

CarnegieMellon

- $T = 1$

$$P(o_1|A,B) = \sum_{s_1} B(s_1,o_1) \cdot A(s_0,s_1)$$

- $T = 2$

$$\begin{aligned} P(o_1,o_2|A,B) &= \sum_{s_2}\sum_{s_1} B(s_1,o_1)B(s_2,o_2) \cdot A(s_0,s_1)A(s_1,s_2) \\ &= \sum_{s_2} B(s_2,o_2)\left(\sum_{s_1} B(s_1,o_1) \cdot A(s_0,s_1)A(s_1,s_2)\right) \end{aligned}$$

Solution: Dynamic Programming

CarnegieMellon

- $T = 2$

$$P(o_1, o_2 | A, B) = \sum_{s_2} \sum_{s_1} B(s_1, o_1) B(s_2, o_2) \cdot A(s_0, s_1) A(s_1, s_2)$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$P(o_1, o_2|A, B) = \sum_{s_2} \sum_{s_1} B(s_1, o_1) B(s_2, o_2) \cdot A(s_0, s_1) A(s_1, s_2)$$

- $T = 3$

$$P(o_1, o_2, o_3|A, B) = \sum_{s_3} \sum_{s_2} \sum_{s_1} B(s_1, o_1) B(s_2, o_2) B(s_3, o_3) \cdot A(s_0, s_1) A(s_1, s_2) A(s_2, s_3)$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$P(o_1, o_2|A, B) = \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)$$

- $T = 3$

$$\begin{aligned} P(o_1, o_2, o_3|A, B) &= \sum_{s_3} \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2){\color{red}B(s_3, o_3)} \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \\ &= \sum_{s_3} {\color{red}B(s_3, o_3)} \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \end{aligned}$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$P(o_1, o_2|A, B) = \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)$$

- $T = 3$

$$\begin{aligned} P(o_1, o_2, o_3|A, B) &= \sum_{s_3} \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2)B(s_3, o_3) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \\ &= \sum_{s_3} B(s_3, o_3) \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \end{aligned}$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$P(o_1, o_2|A, B) = \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)$$

- $T = 3$

$$\begin{aligned} P(o_1, o_2, o_3|A, B) &= \sum_{s_3} \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2)B(s_3, o_3) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \\ &= \sum_{s_3} B(s_3, o_3) \sum_{s_2} \sum_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \end{aligned}$$

- We can catch the blue quantities

$$\boxed{f[t, s] \stackrel{\text{def}}{=} P(o_1, o_2, ..., o_t, s_t = s|A, B)}$$

Solution: Dynamic Programming

CarnegieMellon

- Caching values:

$$f[t,s] \stackrel{\text{def}}{=} P(o_1, o_2, ..., o_t, s_t = s | A, B)$$

Solution: Dynamic Programming

Carnegie Mellon

- Caching values:

$$f[t,s] \stackrel{\text{def}}{=} P(o_1, o_2, ..., o_t, s_t = s|A, B)$$

- How to compute $f[t,s]$?

$$f[t+1,s] = P(o_1, o_2, ..., o_t, o_{t+1}, s_{t+1} = s|A, B)$$

Solution: Dynamic Programming

Carnegie Mellon

- Caching values:

$$f[t, s] \stackrel{\text{def}}{=} P(o_1, o_2, ..., o_t, s_t = s | A, B)$$

- How to compute $f[t, s]$?

$$\begin{aligned} f[t+1, s] &= P(o_1, o_2, ..., o_t, o_{t+1}, s_{t+1} = s | A, B) \\ &= \sum_{s'} P(o_1, o_2, ..., o_t, s_t = s' | A, B) \cdot P(s_{t+1} = s | s_t = s') P(o_{t+1} | s) \end{aligned}$$

Solution: Dynamic Programming

CarnegieMellon

- Caching values:

$$f[t,s] \stackrel{\text{def}}{=} P(o_1, o_2, ..., o_t, s_t = s|A, B)$$

- How to compute $f[t, s]$?

$$\begin{aligned} f[t+1,s] &= P(o_1, o_2, ..., o_t, o_{t+1}, s_{t+1} = s|A, B) \\ &= \sum_{s'} {\color{red}P(o_1, o_2, ..., o_t, s_t = s'|A, B)} \cdot {\color{blue}P(s_{t+1} = s|s_t = s')}{\color{green}P(o_{t+1}|s)} \\ &= \sum_{s'} {\color{red}f[t, s']} \cdot {\color{blue}A(s', s)}{\color{green}B(s, o_{t+1})} \end{aligned}$$

Solution: Dynamic Programming

- How to compute $f[t, s]$?

$$f[t+1, s] = \sum_{s'} f[t, s'] \cdot A(s', s) B(s, o_{t+1})$$

Solution: Dynamic Programming

Carnegie Mellon

- How to compute $f[t, s]$?

$$f[t+1, s] = \sum_{s'} f[t, s'] \cdot A(s', s) B(s, o_{t+1})$$

- What can we do with $f[t, s]$?

$$P(o_1, o_2, ..., o_t | A, B) = \sum_{s} P(o_1, o_2, ..., o_t, s_t = s | A, B) \cdot P(o_t | s_t = s)$$

Solution: Dynamic Programming

Carnegie Mellon

- How to compute $f[t, s]$?

$$f[t+1, s] = \sum_{s'} f[t, s'] \cdot A(s', s) B(s, o_{t+1})$$

- What can we do with $f[t, s]$?

$$\begin{aligned} P(o_1, o_2, ..., o_t | A, B) &= \sum_{s} P(o_1, o_2, ..., o_t, s_t = s | A, B) \cdot P(o_t | s_t = s) \\ &= \sum_{s} f[t, s] \cdot B(s, o_t) \end{aligned}$$

Put Together: Forward-Pass Algorithm

Carnegie Mellon

(1) Initialize:

- For each hidden state s:

$$f[1,s] = P(o_1, s_1 = s|A, B) \leftarrow B(s, o_1) \cdot A(s_0, s)$$

(2) For $t = 2$ to T:

- For each hidden state s:

$$f[t,s] \leftarrow \sum_{s'} f[t-1, s'] \cdot A(s', s) B(s, o_t)$$

(3) Finally:

$$P(o_1, o_2, ..., o_T|A, B) \leftarrow \sum_{s} f[T, s] B(s, o_T)$$

Put Together: Forward-Pass Algorithm

(1) Initialize:

- For each hidden state s:

$$f[1, s] = P(o_1, s_1 = s|A, B) \leftarrow B(s, o_1) \cdot A(s_0, s)$$

(2) For $t = 2$ to T:

- For each hidden state s:

$$f[t, s] \leftarrow \sum_{s'} f[t-1, s'] \cdot A(s', s)B(s, o_t)$$

(3) Finally:

$$P(o_1, o_2, ..., o_T|A, B) \leftarrow \sum_{s} f[T, s]B(s, o_T)$$

Complexity: $O(T \cdot N^2)$

Outline

Question

Given O. Find the hidden sequence S

$$S^* = \operatorname*{argmax}_S P(O|S, A, B)$$

Familiar Observation

$$
\begin{aligned}
S &= (s_1, s_2, ..., s_T) \\
\Rightarrow P(O|S, A, B) &= P(o_1|s_1)P(o_2|s_2)\cdots P(o_T|s_T) \\
&= \prod_{i=1}^{T} P(o_i|s_i)
\end{aligned}
$$

Familiar Observation

$$S = (s_1, s_2, ..., s_T)$$

$$\Rightarrow P(O|S, A, B) = P(o_1|s_1)P(o_2|s_2)\cdots P(o_T|s_T)$$

$$= \prod_{i=1}^{T} P(o_i|s_i)$$

But there are $\Theta(N^T)$ possible sequences S

Solution: Dynamic Programming

Carnegie Mellon

Replace argmax with max:

- $T = 1$

$$\begin{aligned} \max P(o_1|A, B) &= \max_{s_1} P(o_1|s_1) \cdot P(s_1|s_0) \\ &= \max_{s_1} B(s_1, o_1) \cdot A(s_0, s) \end{aligned}$$

Solution: Dynamic Programming

Replace argmax with max:

- $T = 1$

$$\begin{aligned}\max P(o_1|A,B) &= \max_{s_1} P(o_1|s_1) \cdot P(s_1|s_0) \\ &= \max_{s_1} B(s_1, o_1) \cdot A(s_0, s)\end{aligned}$$

- $T = 2$

$$\max P(o_1, o_2|A,B) = \max_{s_2} \max_{s_1} B(s_1, o_1) B(s_2, o_2) \cdot A(s_0, s_1) A(s_1, s_2)$$

Solution: Dynamic Programming

Carnegie Mellon

Replace argmax with max:

- $T = 1$

$$\begin{aligned}\max P(o_1|A,B) &= \max_{s_1} P(o_1|s_1)\cdot P(s_1|s_0)\\ &= \max_{s_1} B(s_1,o_1)\cdot A(s_0,s)\end{aligned}$$

- $T = 2$

$$\begin{aligned}\max P(o_1,o_2|A,B) &= \max_{s_2}\max_{s_1} B(s_1,o_1){\color{red}B(s_2,o_2)}\cdot A(s_0,s_1)A(s_1,s_2)\\ &= \max_{s_2} {\color{red}B(s_2,o_2)}\left(\max_{s_1} B(s_1,o_1)\cdot A(s_0,s_1)A(s_1,s_2)\right)\end{aligned}$$

Solution: Dynamic Programming

Replace argmax with max:

- $T = 1$

$$\begin{aligned}\max P(o_1|A,B) &= \max_{s_1} P(o_1|s_1)\cdot P(s_1|s_0)\\ &= \max_{s_1} B(s_1,o_1)\cdot A(s_0,s)\end{aligned}$$

- $T = 2$

$$\begin{aligned}\max P(o_1,o_2|A,B) &= \max_{s_2}\max_{s_1} B(s_1,o_1)B(s_2,o_2)\cdot A(s_0,s_1)A(s_1,s_2)\\ &= \max_{s_2} B(s_2,o_2)\left(\max_{s_1} B(s_1,o_1)\cdot A(s_0,s_1)A(s_1,s_2)\right)\end{aligned}$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$\max_{s_1,s_2} P(o_1, o_2|A, B) = \max_{s_2} \max_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$\max_{s_1,s_2} P(o_1, o_2 | A, B) = \max_{s_2} \max_{s_1} B(s_1, o_1) B(s_2, o_2) \cdot A(s_0, s_1) A(s_1, s_2)$$

- $T = 3$

$$\max_{s_1,s_2,s_3} P(o_1, o_2, o_3 | A, B) = \max_{s_3} \max_{s_2} \max_{s_1} B(s_1, o_1) B(s_2, o_2) B(s_3, o_3) \cdot A(s_0, s_1) A(s_1, s_2) A(s_2, s_3)$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$\max_{s_1,s_2} P(o_1, o_2|A, B) = \max_{s_2} \max_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)$$

- $T = 3$

$$\begin{aligned}\max_{s_1,s_2,s_3} P(o_1, o_2, o_3|A, B) &= \max_{s_3} \max_{s_2} \max_{s_1} B(s_1, o_1)B(s_2, o_2){\color{red}B(s_3, o_3)} \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \\ &= \max_{s_3} {\color{red}B(s_3, o_3)} \max_{s_2} \max_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3)\end{aligned}$$

Solution: Dynamic Programming

Carnegie Mellon

- $T = 2$

$$\max_{s_1,s_2} P(o_1, o_2|A, B) = \max_{s_2} \max_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)$$

- $T = 3$

$$\begin{aligned}\max_{s_1,s_2,s_3} P(o_1, o_2, o_3|A, B) &= \max_{s_3} \max_{s_2} \max_{s_1} B(s_1, o_1)B(s_2, o_2)B(s_3, o_3) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3) \\ &= \max_{s_3} B(s_3, o_3) \max_{s_2} \max_{s_1} B(s_1, o_1)B(s_2, o_2) \cdot A(s_0, s_1)A(s_1, s_2)A(s_2, s_3)\end{aligned}$$

Solution: Dynamic Programming

CarnegieMellon

- $T = 2$

$$\max_{s_1,s_2} P(o_1, o_2|A,B) = \max_{s_2} \max_{s_1} B(s_1,o_1)B(s_2,o_2) \cdot A(s_0,s_1)A(s_1,s_2)$$

- $T = 3$

$$\begin{aligned}\max_{s_1,s_2,s_3} P(o_1,o_2,o_3|A,B) &= \max_{s_3} \max_{s_2} \max_{s_1} B(s_1,o_1)B(s_2,o_2)B(s_3,o_3) \cdot A(s_0,s_1)A(s_1,s_2)A(s_2,s_3) \\ &= \max_{s_3} B(s_3,o_3) \max_{s_2} \max_{s_1} B(s_1,o_1)B(s_2,o_2) \cdot A(s_0,s_1)A(s_1,s_2)A(s_2,s_3)\end{aligned}$$

- We can catch the blue quantities

$$\boxed{g[t,s] \stackrel{\text{def}}{=} \max_{s_1,s_2,\ldots,s_{t-1}} P(o_1,o_2,\ldots,o_t,s_t=s|A,B)}$$

Solution: Dynamic Programming

Carnegie Mellon

- Caching values:

$$g[t, s] \stackrel{\text{def}}{=} \max_{s_1 \dots s_{t-1}} P(o_1, o_2, ..., o_t, s_t = s | A, B)$$

Solution: Dynamic Programming

- Caching values:

$$g[t,s] \stackrel{\text{def}}{=} \max_{s_1 \dots s_{t-1}} P(o_1, o_2, ..., o_t, s_t = s | A, B)$$

- How to compute $g[t,s]$? Same!

$$g[t+1,s] = \max_{s_1,\dots,s_t} P(o_1, o_2, ..., o_t, o_{t+1}, s_{t+1} = s | A, B)$$

Solution: Dynamic Programming

Carnegie Mellon

- Caching values:

$$g[t,s] \stackrel{\text{def}}{=} \max_{s_1 \dots s_{t-1}} P(o_1, o_2, ..., o_t, s_t = s | A, B)$$

- How to compute $g[t,s]$? Same!

$$\begin{aligned} g[t+1,s] &= \max_{s_1,\dots,s_t} P(o_1, o_2, ..., o_t, o_{t+1}, s_{t+1} = s | A, B) \\ &= \max_{s'} P(o_1, o_2, ..., o_t, s_t = s' | A, B) \cdot P(s_{t+1} = s | s_t = s') P(o_{t+1} | s) \end{aligned}$$

Solution: Dynamic Programming

- Caching values:

$$g[t, s] \stackrel{\text{def}}{=} \max_{s_1 \dots s_{t-1}} P(o_1, o_2, ..., o_t, s_t = s | A, B)$$

- How to compute $g[t, s]$? Same!

$$\begin{aligned} g[t+1, s] &= \max_{s_1, \dots, s_t} P(o_1, o_2, ..., o_t, o_{t+1}, s_{t+1} = s | A, B) \\ &= \max_{s'} P(o_1, o_2, ..., o_t, s_t = s' | A, B) \cdot P(s_{t+1} = s | s_t = s') P(o_{t+1} | s) \\ &= \max_{s'} g[t, s'] \cdot A(s', s) B(s, o_{t+1}) \end{aligned}$$

Solution: Dynamic Programming

- How to compute $g[t, s]$?

$$g[t+1, s] = \max_{s'} g[t, s'] \cdot A(s', s) B(s, o_{t+1})$$

Solution: Dynamic Programming

Carnegie Mellon

- How to compute $g[t, s]$?

$$g[t+1, s] = \max_{s'} g[t, s'] \cdot A(s', s) B(s, o_{t+1})$$

- What can we do with $g[t, s]$? Same!

$$\max_{s_1, \ldots, s_t} P(o_1, o_2, \ldots, o_t | A, B) = \max_{s} P(o_1, o_2, \ldots, o_t, s_t = s | A, B) \cdot P(o_t | s_t = s)$$

Solution: Dynamic Programming

- How to compute $g[t, s]$?

$$g[t+1, s] = \max_{s'} g[t, s'] \cdot A(s', s) B(s, o_{t+1})$$

- What can we do with $g[t, s]$? Same!

$$\max_{s_1, \ldots, s_t} P(o_1, o_2, \ldots, o_t | A, B) = \max_{s} P(o_1, o_2, \ldots, o_t, s_t = s | A, B) \cdot P(o_t | s_t = s)$$
$$= \max_{s} g[t, s] \cdot B(s, o_t)$$

Put Together: *almost* Viterbi Algorithm

Carnegie Mellon

(1) Initialize:

- For each hidden state s:

$$g[1, s] \leftarrow B(s, o_1) \cdot A(s_0, s)$$

(2) For $t = 2$ to T:

- For each hidden state s:

$$g[t, s] \leftarrow \max_{s'} g[t-1, s'] \cdot A(s', s) B(s, o_t)$$

(3) Finally:

$$\max_{s_1 \ldots s_T} P(o_1, o_2, ..., o_T | A, B) \leftarrow \max_{s} g[T, s] B(s, o_T)$$

Put Together: *almost* Viterbi Algorithm

(1) Initialize:

- For each hidden state s:

$$g[1, s] \leftarrow B(s, o_1) \cdot A(s_0, s)$$

(2) For $t = 2$ to T:

- For each hidden state s:

$$g[t, s] \leftarrow \max_{s'} g[t-1, s'] \cdot A(s', s) B(s, o_t)$$

(3) Finally:

$$\max_{s_1 \ldots s_T} P(o_1, o_2, ..., o_T | A, B) \leftarrow \max_{s} g[T, s] B(s, o_T)$$

We want argmax , not max !!!

+ traceback: Viterbi Algorithm

(1) Initialize:

- For each hidden state s:

$$g[1, s] \leftarrow B(s, o_1) \cdot A(s_0, s)$$

(2) For $t = 2$ to T:

- For each hidden state s:

$$g[t, s] \leftarrow \max_{s'} g[t-1, s'] \cdot A(s', s) B(s, o_t)$$

$$h[t, s] \leftarrow \operatorname*{argmax}_{s'} g[t-1, s'] \cdot A(s', s) B(s, o_t)$$

(3) Follow $h[t, s]$ to find s_T^*, s_{T-1}^*, ..., s_1^*.

+ traceback: Viterbi Algorithm

Carnegie Mellon

(1) Initialize:

- For each hidden state s:

$$g[1, s] \leftarrow B(s, o_1) \cdot A(s_0, s)$$

(2) For $t = 2$ to T:

- For each hidden state s:

$$g[t, s] \leftarrow \max_{s'} g[t-1, s'] \cdot A(s', s) B(s, o_t)$$

$$h[t, s] \leftarrow \operatorname*{argmax}_{s'} g[t-1, s'] \cdot A(s', s) B(s, o_t)$$

(3) Follow $h[t, s]$ to find s^*_T, s^*_{T-1}, ..., s^*_1.

Complexity: $O(T \cdot N^2)$

Outline

CarnegieMellon

Question

Find the posterior

$$A^*, B^* = \underset{A,B}{\operatorname{argmax}}\, P(A, B|O)$$