Natural Language Processing
CSE 325/425

Sihong Xie

Lecture 7:

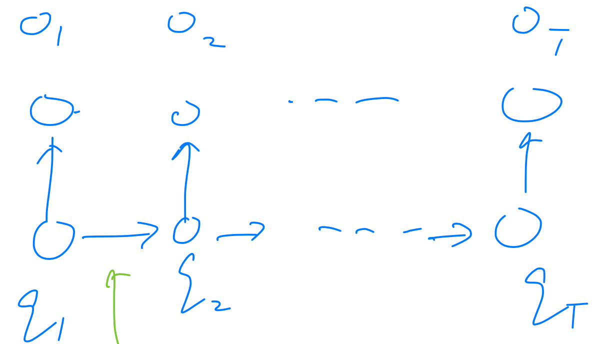
- Forward and Backard algorithm

Recall : Bigram $\Pr(O \mid \Pr(o_{t-1} \to o_t))$ ①

In the future we will talk about ②
PCFG for $\Pr(O \mid \text{PCFG})$

o_t is dependent of q_t

q_t is dependent of q_{t-1}



Likelihood of a sentence

Given: observed sentence $O = [o_1, \ldots, o_T], o_t \in V$

a fixed HMM (A, B, π)

Find: $\underline{\Pr(O|A, B, \pi)} \triangleq$ Likelihood

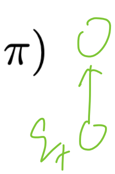
$A = N \left[\quad \right]$ N

$A_{ij} = \Pr(q_t = j \mid q_{t-1} = i)$

Useful for evaluating the validity of a sentence.

expensive

By total probability, $\Pr(O|A, B, \pi) = \sum_Q \Pr(O, Q|A, B, \pi)$



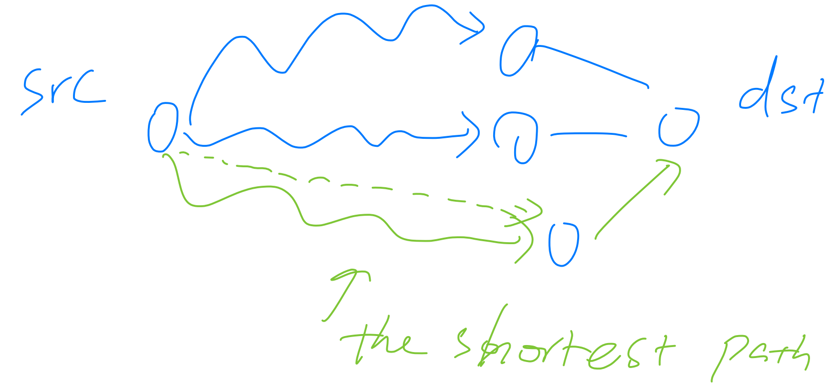
$B = N \left[\quad \right]$ $|V|$

The summation is over all possible hidden state sequences $Q = [q_1, \ldots, q_T], q_t \in S$

$\#$ of $Q = N^T$

A brute-force enumeration will have time complexity

Recall: DP example



Likelihood of a sentence

Forward algorithm: $\Rightarrow Pr(O \mid A, B, \pi)$

- It is an instance of dynamic programming algorithm:
  - solves a large problem using cached solutions to sub-problems;
  - overlapping sub-problems: can re-use solutions to sub-problems;
  - optimal structures: optimal sub-problem solutions => optimal larger problem solutions.
- It is a specific example of belief propagation or message passing algorithm for probabilistic graphical models (PGM)
  - more general PGM includes Bayesian networks and Markov random fields;
  - here we focus on a linear chain (sequence);
  - later on will discuss trees in syntactic parsing;
  - see CSE326 note for more details;

Likelihood of a sentence

Two simple identities:

- Order of product and summation can be exchanged → forward / backward

sum-product

$$ax + ay + bx + by = (a+b)(x+y)$$

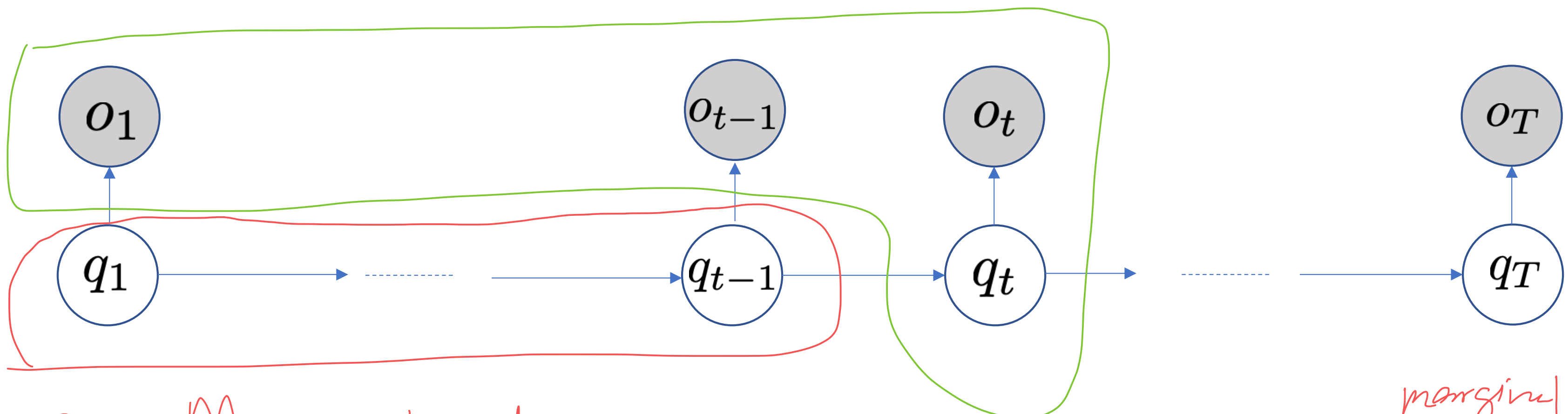
product-sum

- Order of product and max can be exchanged — viterbi

$$\max(ax, ay, bx, by) = \max(a,b) \times \max(x,y)$$

Define the *forward* probability $\alpha_t(j) = P(o_1, \ldots, o_t, q_t = j)$

$q_1 \in \{s_1, \vdots, s_N\}$



? Marginalized out

marginalized out

$O(N^{t-1})$

$$\alpha_t(j) = \sum_{q_1} \sum_{q_2} \cdots \sum_{q_{t-1}} \Pr(o_1 \cdots o_t, \underbrace{q_1, \cdots q_{t-1}}_{\text{marginalized out}}, q_t)$$

Likelihood of a sentence

Forward algorithm: a gentle start. Compute the following

Base case of the recursion

$$\alpha_1(i) = \boxed{Pr(o_1, q_1 = i)}$$

$$= Pr(q_1 = i)\, Pr(o_1 \mid q_1 = i)$$

$$= \pi(i) \times b_{i o_1} \qquad o_1 \in \{1 \cdots V\}$$

i-th row o_1-th col

word index in V

$$\alpha_2(i) \triangleq Pr(o_1, o_2, q_2 = i)$$

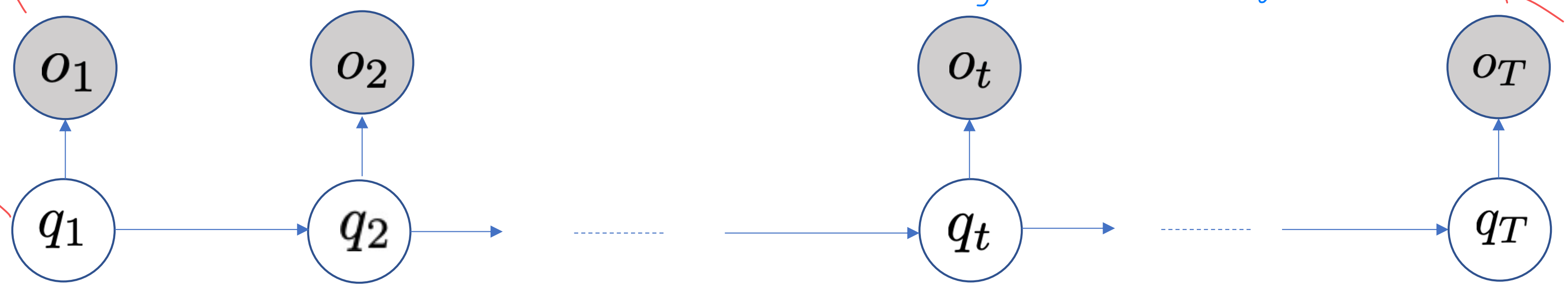
$$= \sum_j Pr(o_1, o_2, \boxed{q_1 = j}, q_2 = i)$$

R.V. marginalized out

$$= \sum_j \boxed{Pr(o_1, q_1 = j)} \times Pr(q_2 = i \mid q_1 = j) \times Pr(o_2 \mid q_2 = i)$$

$$= \sum_j \alpha_1(j) \times a_{ji} \times b_{i o_2}$$

$o_2 \in \{1, \ldots, V\}$



pos-tag index

$Pr(o_1, q_1 = i)$, $i = 1, \ldots, N$, N = # of pos-tags

$$\begin{bmatrix} \alpha_1(1) \\ \alpha_1(2) \\ \vdots \\ \alpha_1(N) \end{bmatrix} \quad \begin{bmatrix} \alpha_2(1) \\ \alpha_2(2) \\ \vdots \\ \alpha_2(N) \end{bmatrix}$$

$$\begin{bmatrix} \alpha_t(1) \\ \vdots \\ \alpha_t(N) \end{bmatrix}$$

General Recursion.

$$\alpha_t(i) = \sum_j \alpha_{t-1}(j) \times a_{ji} \times b_{i o_t}$$

Likelihood of a sentence

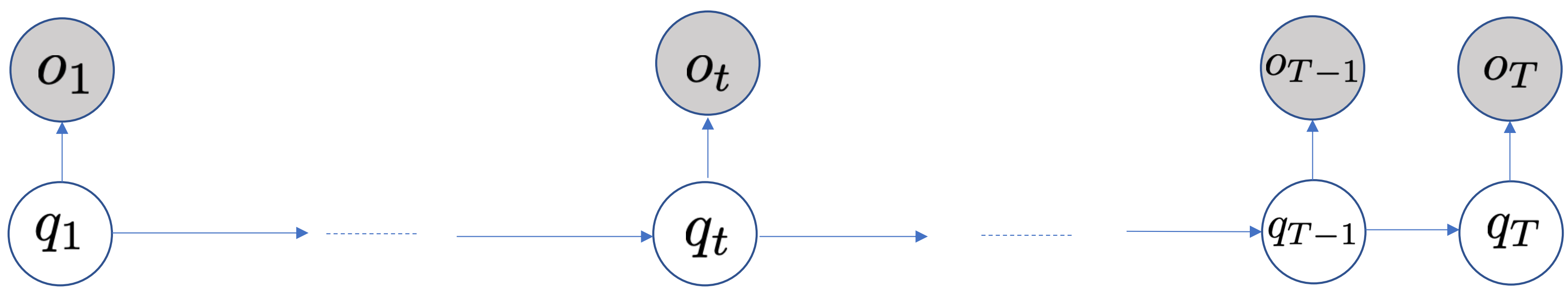
Forward algorithm: compute the likelihood of an entire sequence

$$\Pr(O) = \sum_j P(O, q_T = j) = \sum_j \boxed{\alpha_T(j)}$$

$$= \sum_j \sum_i \boxed{\alpha_{T-1}(i)} \times a_{ij} \times b_{i o_{T-1}}$$

$$= \sum_j \sum_i \left[\sum_k \boxed{\alpha_{T-2}(k)} \times a_{kj} \times b_{k o_{T-2}} \right] \times a_{ij} \times b_{i o_{T-1}}$$

$$\vdots$$



Sum-product:

$O(N^T)$

$$= \sum_{i_T} \sum_{i_{T-1}} \sum_{i_{T-2}} \cdots \sum_{i_1} \left[\pi(i_1) b_{i_1 o_1} \times a_{i_1 i_2} \times b_{i_2 o_2} \times a_{i_2 i_3} \times \cdots a_{i_{T-1} i_T} b_{i_T o_T} \right]$$

$$\begin{bmatrix} \alpha_T(1) \\ \vdots \\ \alpha_T(N) \end{bmatrix}$$

Likelihood of a sentence

Forward algorithm: compute the likelihood of an entire sequence

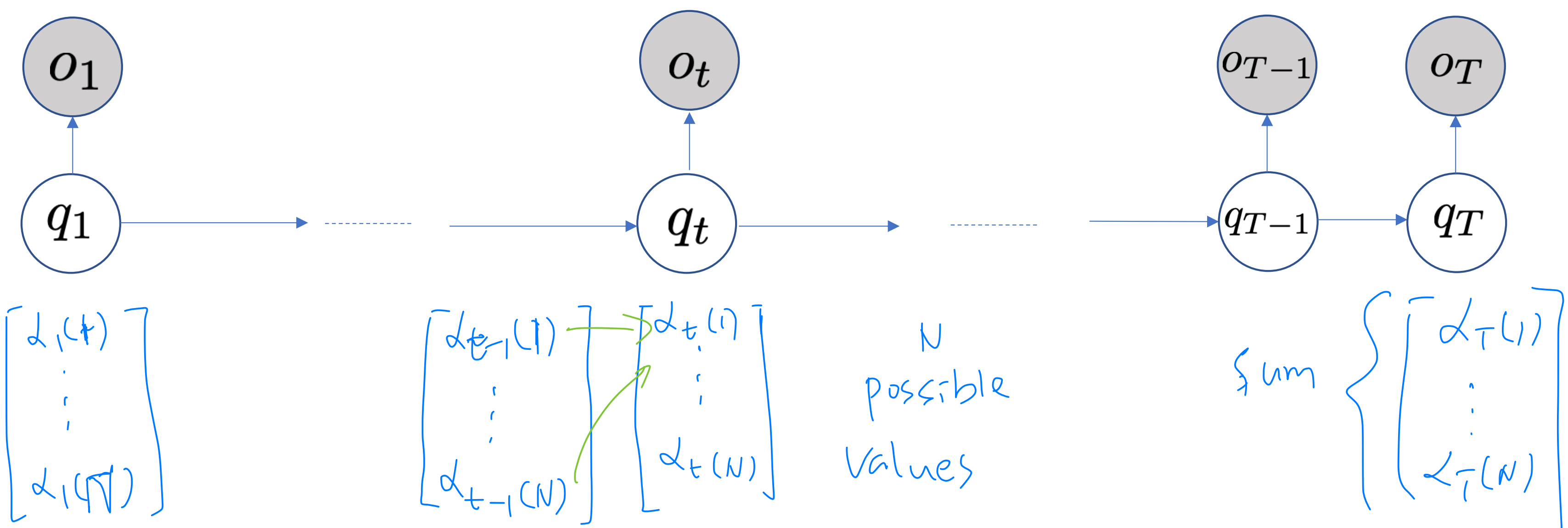
O(N) 1. Initialize $\alpha_1(i)$ for each value i of the first hidden state q_1 . $\alpha_1(i) = \pi(i)\, b_{i o_1}$

O(T) 2. for $t = 2, \dots, T$ # go through word locations

O(N) for $j = 1, \dots, N$ # go through pos-tags of the t-th location

O(NT) compute $\alpha_t(j) = \sum_k \alpha_{t-1}(k) a_{kj} b_j(o_t)$ ← B matrix ← A matrix ← previous α array } All are given or calculated.

O(N) 3. Return $\Pr(O) = \sum_j P(O, q_T = j) = \sum_j \alpha_T(j)$



Likelihood of a sentence

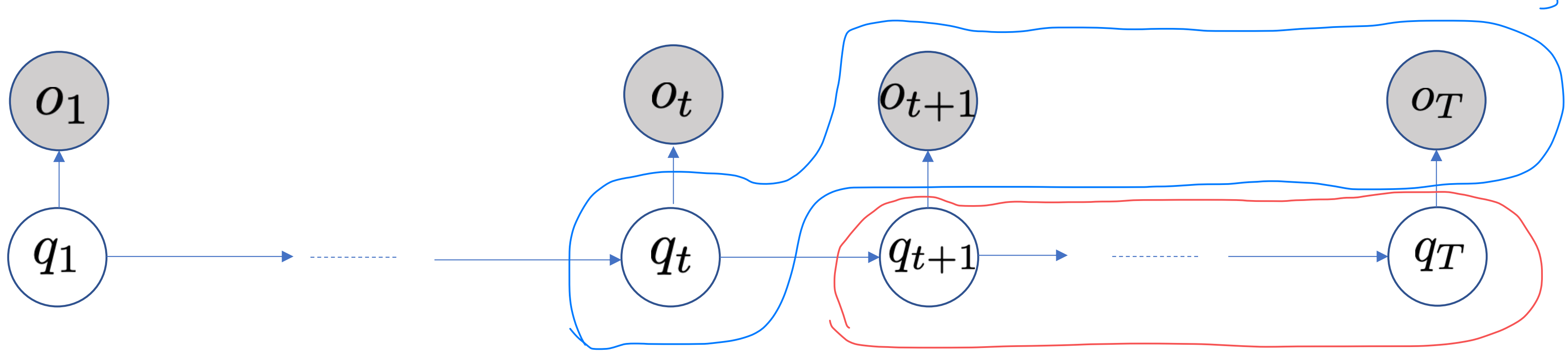
Given: observed sentence $O = [o_1, \ldots, o_T], o_t \in V$

a fixed HMM (A, B, π)

Find: $\Pr(O|A, B, \pi)$

An alternative way to evaluate $\Pr(O|A, B, \pi) = \sum_{Q} \Pr(O, Q|A, B, \pi)$

Define the *backward* probability $\beta_t(j) = P(o_{t+1}, \ldots, o_T | q_t = j)$ $j \in \{1, \ldots, N\}$



marginalized out!

Likelihood of a sentence

Backward algorithm:

Base case: $\boxed{\beta_T(i)} = \Pr(o_{T+1} \cdots o_T \mid q_T = i)$

$= \Pr(\phi \mid q_T = i)$

$= \boxed{1}$

$\beta_{T-1}(i) \triangleq \Pr(o_T \mid q_{T-1} = i)$

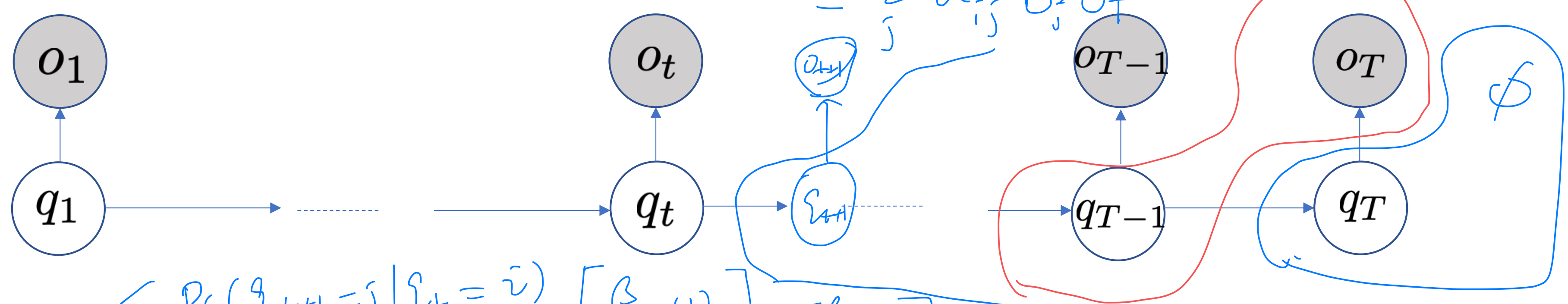
$= \sum_j \Pr(q_T = j \mid q_{T-1} = i)$

$\times \Pr(o_T \mid q_T = j)$

$\times \boxed{\Pr(o_{T+1} \cdots o_T \mid q_T = j)}$

$= \sum_j a_{ij}\, b_{j, o_T}$

General Recursion



$\beta_t(i) = \sum_j \Pr(q_{t+1} = j \mid q_t = i)$

$\times \Pr(o_{t+1} \mid q_{t+1} = j)$

$\times \beta_{t+1}(j)$

$\begin{bmatrix} \beta_t(1) \\ \vdots \\ \beta_t(N) \end{bmatrix} \quad \begin{bmatrix} \beta_{t+1}(1) \\ \vdots \\ \beta_{t+1}(N) \end{bmatrix}$

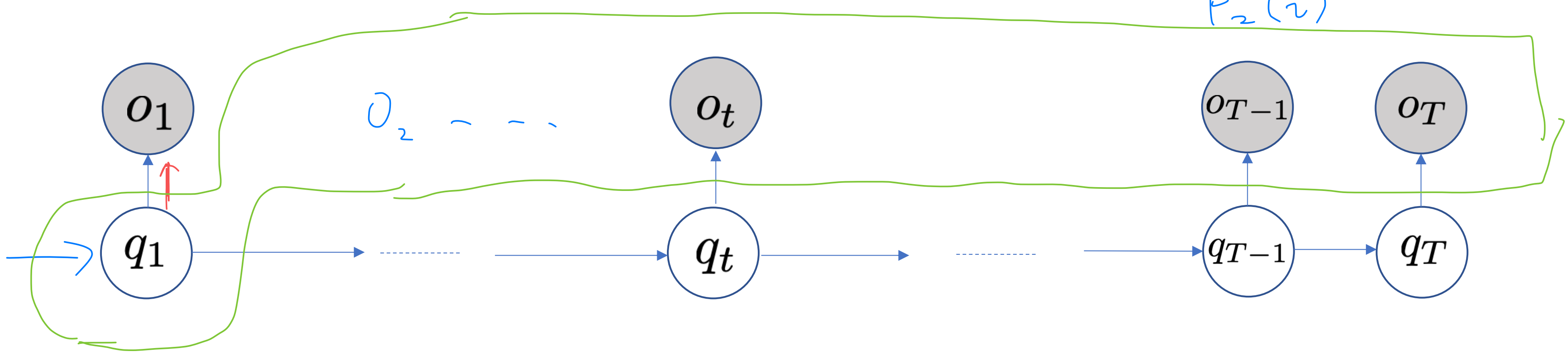
Likelihood of a sentence

Backward algorithm: compute the likelihood of an entire sequence

$$\Pr(O) = \sum_j P(O, q_1 = j) = \sum_j \pi(j) \boxed{\Pr(o_1 \mid q_1 = j)} \overbrace{\boxed{\Pr(o_2 \cdots o_T \mid q_1 = j)}}^{= \beta_1(j)}$$

$$= \sum_j \pi(j)\, b_{j o_1} \boxed{\beta_1(j)}$$

$$= \sum_j \pi(j)\, b_{j o_1} \Big[\sum_i \Pr(q_2 = i \mid q_1 = j) \underbrace{\Pr(o_2 \mid q_2 = i)}_{\beta_2(i)} \Big]$$



$O(N^T)$

$$= \sum_{i_1} \sum_{i_2} \cdots \sum_{i_T} \pi(i_1)\, b_{i_1 o_1} a_{i_1 i_2} b_{i_2 o_2} \cdots a_{i_{T-1} i_T} b_{i_T o_T}$$

Given

$\pi\,[\quad]$ $A\,[\qquad]$

$B\,[\qquad]$

Likelihood of a sentence

MLE Alg

to estimate π, A, B

$Pr(\text{corpus} \mid A, B, \pi)$

Backward algorithm: compute the likelihood of an entire sequence

1. Initialize $\beta_T(i)$ for each value i of the last hidden state q_T .
2. for $t = T-1, \ldots, 1$

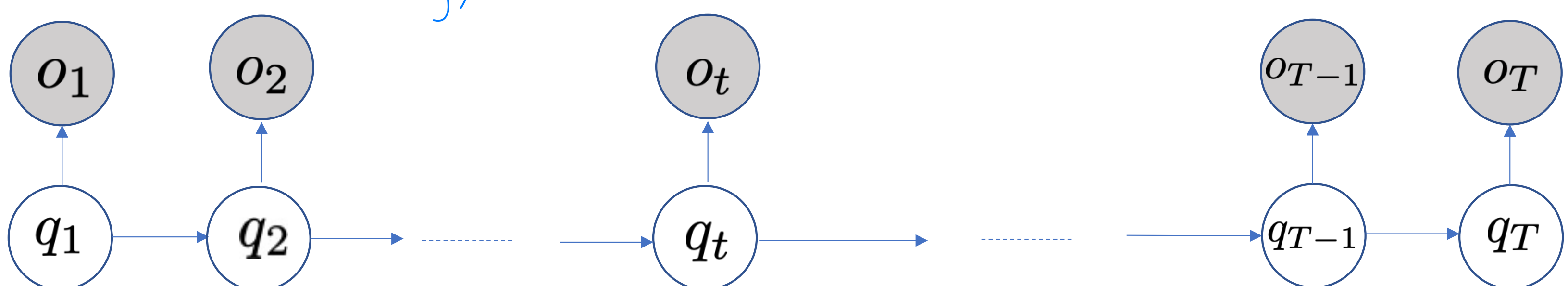
   for $j = 1, \ldots, N$

   compute $\beta_t(j) = \sum_k \beta_{t+1}(k) a_{jk} b_j(o_t)$

   j-th row & o_1-th column

   B matrix

3. Return $\Pr(O) = \sum_j \underbrace{P(q_1 = j)}_{\pi(j)} b_j(o_1) \beta_1(j)$



const

$\begin{bmatrix} \beta_1(1) \\ \vdots \\ \beta_1(N) \end{bmatrix}$ $\begin{bmatrix} \beta_{T-1}(1) \\ \vdots \\ \beta_{T-1}(N) \end{bmatrix}$ $\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

$\beta_T(i) = 1$

Summary

Given a sentence $O = [o_1, \ldots, o_T], o_t \in V$

and HMM parameters (A, B, π)

Compute $\Pr(O|A, B, \pi)$

Forward algorithm

- start from the first POS tag and go forward.
- end with the last POS tag and find sentence likelihood.

Backward algorithm

- Start from the last POS tag and go backward.
- end with the first POS tag and find sentence likelihood.