Natural Language Processing
CSE 325/425

Sihong Xie

Lecture 8:

- HMM prediction and training

Predict optimal hidden states

Want to find the best sequence of POS tags for a given sentence.

- Vocabulary V
- Set of N POS tags $S = \{s_1, \ldots, s_n\}$

- Observed sentence $O = [o_1, \ldots, o_T], o_t \in V$
- Hidden states: $Q = [q_1, \ldots, q_T], q_t \in S$

All	we	gotta	do	is	go	around	the	corner
DT	PRP	VBN	VB	VBZ	VB	IN	DT	NN

- MAP (maximum a posterior) prediction

$$Q^* = \arg\max_Q \Pr(Q|O)$$

$\Leftrightarrow \arg\max_Q \Pr(O|Q)\Pr(Q)$

- Using the Bayes theorem:

$$\Pr(Q|O) = \frac{\Pr(O|Q)\Pr(Q)}{\Pr(O)}$$

(A, B, π)

Constant w.r.t. Q

- Difficulty: there are exponentially many possible sequences.

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

$u_t(j) = -v_t(j) = \min -\Pr(q_1 \cdots q_{t-1}\ q_t = j, o_1, \cdots o_t)$

$q_1 \cdots q_{t-1}$

cost of going from dummy to $q_t = j$

Predict optimal hidden states

Similar to the forward algorithm

- Use $\max(ax, ay, bx, by) = \max(a, b) \times \max(x, y)$
  instead of $ax + ay + bx + by = (a + b)(x + y)$ in the forward algorithm.
- Define the probability

DP

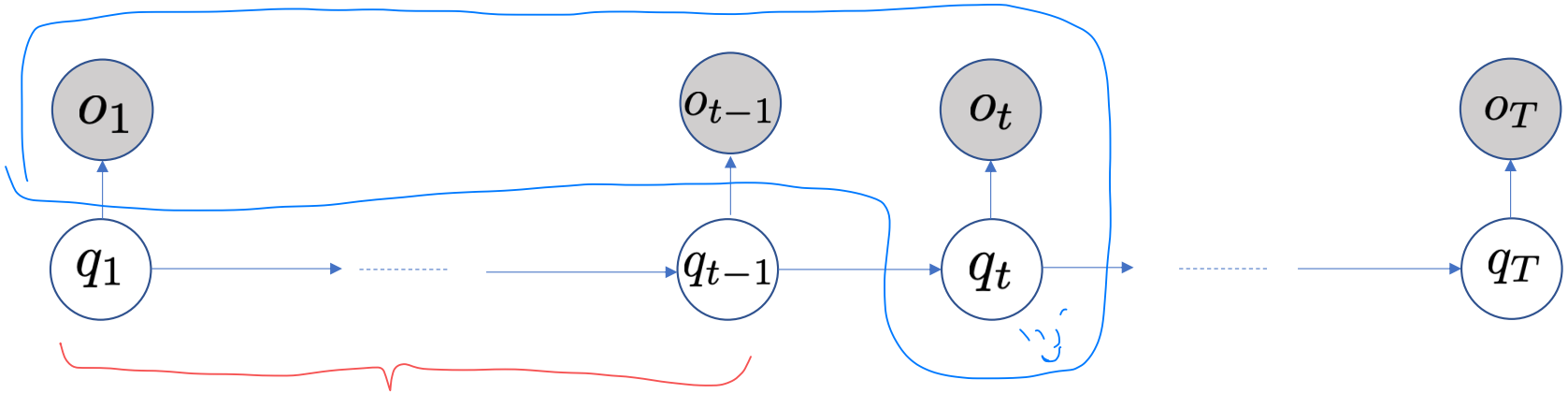
the "1 event"

$t \in \{1, \cdots, T\}$

$j \in \{1, \cdots, N\}$

$$v_t(j) = \max_{q_1, \ldots, q_{t-1}} \Pr(q_1, \ldots, q_{t-1}, q_t = j, o_1, \ldots, o_t)$$

  - Interpreted as the maximum probability of seeing POS-tag j and the observed words up to o_t .
- It is a dynamic programming algorithm and an example of a shortest path algorithm run on a trellis.



Maximized out

(marginalized out in forward)

$V_t(j)$

Predict optimal hidden states

Viterbi Algorithm: Compute the base case, the next one, and the general case

q_0 is dummy

$$v_1(i) = \max_{q_0} \Pr(q_0, q_1 = i, o_1)$$

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

π B

$$v_2(i) = \max_{q_1} \Pr(q_1, q_2 = i, o_1, o_2)$$

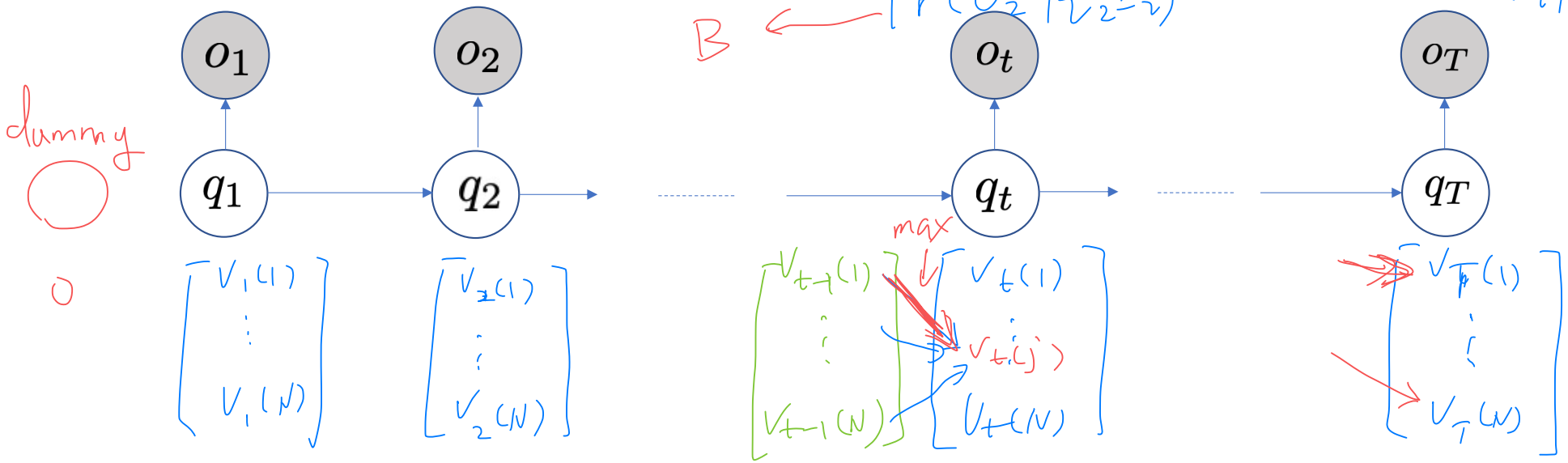
$$= \max_{q_1} \Pr(q_1 = j)\ \Pr(o_1 \mid q_1 = j)\ \Pr(q_2 = i \mid q_1 = j)\ \Pr(o_2 \mid q_2 = i)$$

$v_t(j)$ A B

$$v_t(i) = \max_{q_1 \cdots q_{t-1}} \Pr(q_1 \cdots q_{t-1}, q_t = i, o_1 \cdots o_t)$$

$$= \max_{q_{t-1}} \max_{q_1 \cdots q_{t-2}} \Pr(q_1 \cdots q_{t-2}, q_{t-1} = j, o_1 \cdots o_{t-2})\ \Pr(q_t = i \mid q_{t-1} = j)\ \Pr(o_t \mid q_t = i)$$

$v_{t-1}(j)$



$= \max_{q_T=1\ldots N} \left[\max_{q_{T-1}=1,\ldots N} \left[\max_{q_1 \cdots q_{T-2}} \Pr(q_1 \cdots q_{T-2}, o_1 \cdots o_{T-2}) \Pr(q_{T-1} | q_{T-2}) \Pr(o_{T-1} | q_{T-1}=j) \right] \times \Pr(q_T | q_{T-1}) \Pr(o_T | q_T) \right]$

$V_T(i)$ $\quad$ $V_{t-1}(j)$

Predict optimal hidden states

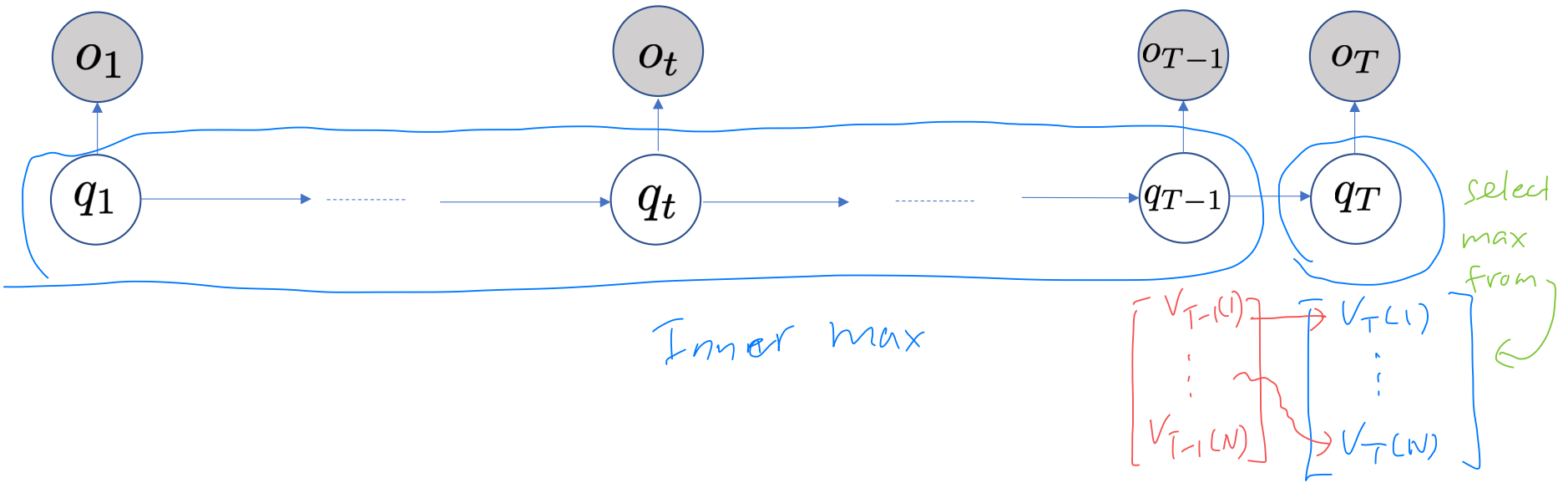
Viterbi algorithm: compute maximum probability

$$\Pr(Q^*|O) = \max_Q \Pr(Q|O)$$

$$= \max_Q \Pr(q_1 \cdots q_{T-1}, q_T\, o_1 \cdots o_{T-1}, o_T)$$

$$= \max_{q_T} \max_{q_1 \cdots q_{T-1}} \left[\Pr(q_1 \cdots q_{T-1}, o_1 \cdots o_{T-1}) \times \Pr(q_T = i | q_{T-1} = j) \times \Pr(o_T | q_T = i) \right]$$

HMM



$v_1(i) = \pi(i) \times b_{i o_1}$

Predict optimal hidden states

Viterbi algorithm: compute maximum probability and the optimal sequence (decoding)

1. Initialize $v_1(i)$ for each value i of the first hidden state q_1 .
2. for $t = 2, \ldots, T$

   for $j = 1, \ldots, N$

   compute $v_t(j) = \max_k v_{t-1}(k) a_{kj} b_j(o_t)$

   record back-pointers $p_t(j) = \arg\max_k v_{t-1}(k) a_{kj} b_j(o_t)$
3. Backtracking to find Q^*
4. Return $\Pr(Q^*|O) = \max_j v_T(j)$

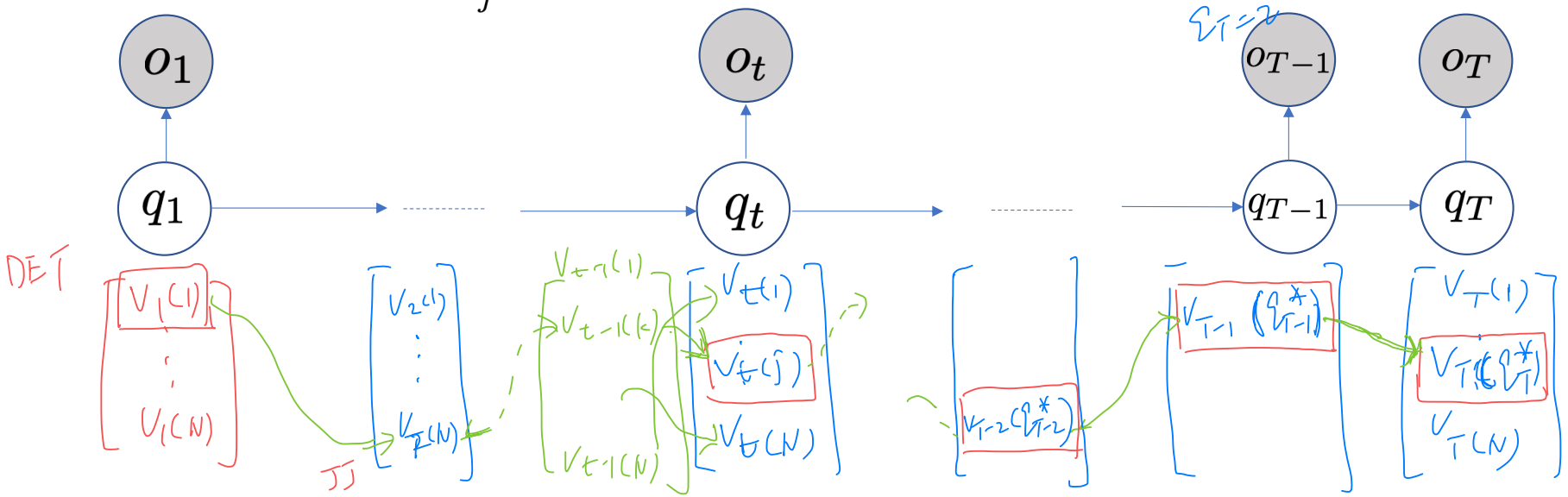
$a_{kj} = Pr(q_t = j \mid q_{t-1} = k)$

$= k^*$

$= Pr(Q^* | O)$

$q_T^* = \arg\max V_T(i)$

$q_T = i$



Predict optimal hidden states

A running example of Viterbi.

If cheating

$Pr(H) = 0.9$

- Suppose you are in a casino with a cheating dealer X.
- X flips a fair coin if X decides not to cheat and a biased coin otherwise.
- X follows a Markov chain to change between cheating and no cheating.
- You only observe a sequence of flips.
- Guess which flips are from a biased coin?