Problem 1: The Evaluation problem

<u>input</u>: HMM defined by $\lambda = \{A, B, \pi\}$ and an observation sequence $O = o_1 o_2 o_3 \ldots o_T$

<u>output</u>: the probability of observing the sequence O given the HMM λ : $P(O|\lambda)$

<u>ex</u> O: a a b c c c $\Rightarrow |O| = 6$

* if we have N states, there are N^6 possible sequences of states underlying the observation sequence O

Let's define a few fundamental probabilities:

▷ P [observing O given a fixed sequence of states Q]: $P[O|Q] = P(o_1|q_1)\, P(o_2|q_2) \ldots P(o_T|q_T)$

$$= b_{q_1}(o_1)\, b_{q_2}(o_2) \ldots b_{q_T}(o_T)$$

▷ P[having a particular sequence of states Q]: $P[Q] = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}$

▷ Joint probability of O and Q: $P[O, Q] = P[O|Q] \cdot P[Q]$

$$\Longrightarrow P[O|\lambda] = \sum_{\substack{\text{all seq}\\ \text{of states}\\ Q}} P[O|Q]\, P[Q] = \sum_{q_1 q_2 \ldots q_T} \pi_{q_1} b_{q_1}(o_1)\, a_{q_1 q_2} b_{q_2}(o_2) \ldots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

$O(N^T) \longrightarrow$ impractical

The Forward Algorithm (dynamic programming)

- <u>defn</u> α: forward variable
  - $\alpha_t(i) = P[o_1 o_2 \ldots o_t, q_t = s_i]$ = probability of observing the first t observations and being in state s_i @ time t

① <u>initialization</u>

$\alpha_1(i) = \pi_i b_i(o_1)$, $1 \le i \le N$ ⟵ compute for each state i

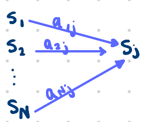
② <u>recurrence</u>

$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(o_{t+1})$ $1 \le t \le T-1$, $1 \le j \le N$

③ <u>termination</u>

$P[O|\lambda] = \sum_{i=1}^{N} \alpha_T(i)$

$O(N^2 T)$ because the α calculation iterates through all i's ($1 \le i \le N$), and we do the calculation for all states j ($1 \le j \le N$) for each time t ($1 \le t \le T-1$).



* can arrive at state S_j @ time t+1 from any state S_i ($1 \le i \le N$)

Problem 2: The decoding problem

<u>input</u>: HMM defined by $\lambda = \{A, B, \pi\}$ and an observation sequence $O = o_1 o_2 o_3 \ldots o_T$

<u>output</u>: the most likely sequence of states $Q = q_1 q_2 \cdots q_T$ given the observation sequence O

The Viterbi Algorithm

- $\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P[q_1, \ldots, q_{t-1}, q_t = s_i \mid o_1, o_2, \ldots o_t]$ = the most probable sequence of states $q_1, \ldots, q_{t-1}$?

① <u>initialization</u>:

$\delta_1(i) = \pi_i b_j(o_1)$ $1 \le i \le N$

$\psi_1(i) = 0$ $1 \le i \le N$ #backtracking

② <u>recurrence</u>:

$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}]\, b_j(o_t)$ $1 \le j \le N$, $2 \le t \le T$

$\psi_t(j) = \operatorname{argmax}_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}]$ $1 \le j \le N$, $2 \le t \le T$

③ <u>Termination</u>:

$p^* = \max_{1 \le i \le N} [\delta_T(i)]$

$q_T^* = \operatorname{argmax}_{1 \le i \le N} [\delta_T(i)]$

q^* : state index

④ <u>Backtracking</u>:

$q_t^* = \psi_{t+1}(q_{t+1}^*)$ $t = T-1, T-2, \ldots, 2, 1$

$Q_{opt} = q_1^* q_2^* \ldots q_T^*$

also $O(N^2 T)$