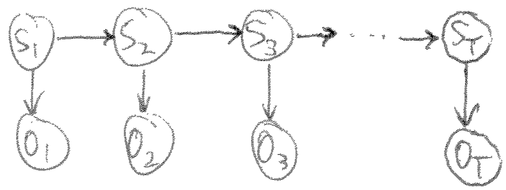
Review

- Hidden Markov Models (HMMs)



states $S_t \in \{1, 2, \ldots, n\}$
observations $O_t \in \{1, 2, \ldots, m\}$

- Joint distribution

$$P(\vec{S}, \vec{O}) = P(S_1) \prod_{t=2}^{T} P(S_t | S_{t-1}) \prod_{t=1}^{T} P(O_t | S_t)$$

- Parameters

$$\pi_i = P(S_1 = i)$$
$$a_{ij} = P(S_{t+1} = j | S_t = i)$$
$$b_{ik} = P(O_t = k | S_t = i)$$

Q1: How to compute likelihood $P(O_1, O_2, \ldots, O_T)$?

$$P(O_1, O_2, \ldots, O_T) = \sum_{\vec{S}} P(S_1, \ldots, S_T, O_1, \ldots, O_T)$$

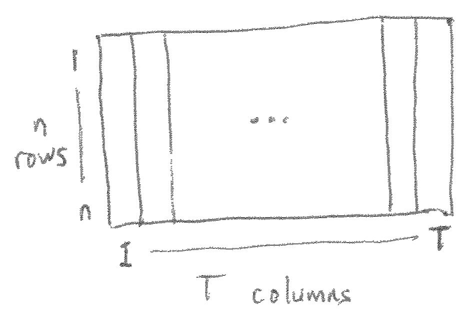
sum over n^T hidden state sequences!

- Efficient recursion

$$P(O_1, \ldots, O_{t+1}, S_{t+1} = j) = \sum_{i=1}^{n} \underbrace{P(O_1, \ldots, O_t, S_t = i)}_{\text{recursive instance}} \underbrace{P(S_{t+1} = j | S_t = i) P(O_{t+1} | S_{t+1} = j)}_{\text{CPTs}}$$

$$\alpha_{j,t+1} = \sum_{i=1}^{n} \alpha_{it} \, a_{ij} \, b_j(O_{t+1})$$

α : $n \times T$ matrix



- Initial condition

$$\alpha_{i1} = \pi_i \, b_i(O_1)$$

- Likelihood computation

$$P(O_1, \ldots, O_T) = \sum_{i=1}^{n} P(O_1, \ldots, O_T, S_T = i)$$
$$= \sum_{i=1}^{n} \alpha_{iT}$$

sum of last column of α matrix!

- "Forward algorithm" scales as $O(n^2 T)$

Q2: How to compute most likely state sequence?

$$S^* = \{S_1^*, S_2^*, \ldots, S_T^*\}$$

$$= \underset{S}{\operatorname{argmax}}\ P(S_1, S_2, \ldots, S_T \mid O_1, O_2, \ldots, O_T)$$

$$= \underset{S}{\operatorname{argmax}} \left[\frac{P(S_1, \ldots, S_T, O_1, \ldots, O_T)}{P(O_1, \ldots, O_T)} \right] \quad \text{product rule}$$

$$= \underset{S}{\operatorname{argmax}}\ P(S_1, \ldots, S_T, O_1, \ldots, O_T) \quad \text{b/c denominator is constant w.r.t. } S$$

How to compute $S^* = \underset{S}{\operatorname{argmax}} \log P(S_1, \ldots, S_T, O_1, \ldots, O_T)$?

monotonic function

Define: $$\ell_{it}^* = \max_{\{S_1, S_2, \ldots, S_{t-1}\}} \log P(S_1, S_2, \ldots, S_{t-1}, S_t = i, O_1, O_2, \ldots, O_t)$$

= log-probability of most likely t-step state sequence that ends in state i at time t, and that accounts for first t observations

Recursion

(1) Base case ($t=1$)

$$\ell_{i1}^* = \log P(S_1 = i, O_1) = \log\left[P(S_1 = i) P(O_1 \mid S_1 = i)\right] \quad \text{product rule}$$

$$= \log\left[\pi_i\, b_i(O_1)\right]$$

$$= \log \pi_i + \log b_i(O_1)$$

(2) From time t to time $t+1$:

$$l^*_{j,t+1} = \max_{S_1,\ldots,S_t} \log P(S_1,\ldots,S_t,S_{t+1}=j,O_1,O_2,\ldots,O_t,O_{t+1})$$

$$= \max_{S_1,\ldots,S_{t-1}} \max_i \log P(S_1,\ldots,S_t=i,S_{t+1}=j,O_1,\ldots,O_t,O_{t+1})$$

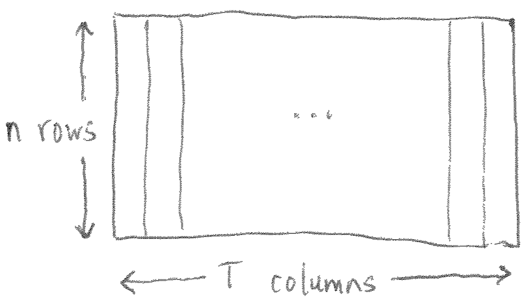
$$= \max_{S_1,\ldots,S_{t-1}} \max_i \log \Big[P(S_1,\ldots,S_{t-1},S_t=i,O_1,\ldots,O_t)$$
$$P(S_{t+1}=j \mid \cancel{S_1,\ldots,}S_t=i,\cancel{O_1,\ldots,O_t})$$
$$P(O_{t+1} \mid \cancel{S_1,\ldots,S_t=i},S_{t+1}=j,\cancel{O_1,\ldots,O_t})\Big]$$

product rule + CI

$$= \max_{S_1,\ldots,S_{t-1}} \max_i \log\Big[P(S_1,\ldots,S_t=i,O_1,\ldots,O_t)\,P(S_{t+1}=j \mid S_t=i)\,P(O_{t+1} \mid S_{t+1}=j)\Big]$$

$$= \max_i \Big[\max_{S_1,\ldots,S_{t-1}} \log P(S_1,\ldots,S_t=i,O_1,\ldots,O_t) + \log P(S_{t+1}=j \mid S_t=i)$$
$$+ \log P(O_{t+1} \mid S_{t+1}=j) \Big]$$

$$\boxed{l^*_{j,t+1} = \max_i \left[l^*_{it} + \log a_{ij} \right] + \log b_j(O_{t+1})}$$



fill in l^*_{it} column-by-column

- How to derive S^* from ℓ^*?

Record most likely transitions

$$\Phi_{t+1}(j) = \underset{i}{\operatorname{argmax}} \left[\ell_{it}^* + \log a_{ij} \right]$$

= most likely state at time t given we land in state j at time $t+1$ (with observations $O_1, O_2, \ldots, O_{t+1}$)

- Compute S^* by back-tracking:

$$S_T^* = \underset{i}{\operatorname{argmax}} \left[\ell_{iT}^* \right]$$

$$\left. \begin{array}{l} \text{for } t = T-1 \text{ to } 1: \\ \quad S_t^* = \Phi_{t+1}(S_{t+1}^*) \end{array} \right\} \text{"backward pass"}$$

S^* is known as "Viterbi" path (state sequence)

Viterbi algorithm is instance of dynamic programming

Q3: Belief updating

How to update belief in real time based on incoming evidence?

Define $q_{it} = P(S_t = i \mid O_1, O_2, \ldots, O_t)$ conditional prob. that $S_t = i$ based on evidence only up to time t

How to update from $q_{j,t-1}$ to q_{it}?

$$q_{it} = \frac{P(S_t = i, O_t \mid O_1, O_2, \ldots, O_{t-1})}{P(O_t \mid O_1, O_2, \ldots, O_{t-1})}$$ product rule

Numerator:

$$P(S_t = i, O_t \mid O_1, \ldots, O_{t-1}) = P(S_t = i \mid O_1, \ldots, O_{t-1}) P(O_t \mid S_t = i, \cancel{O_1, \ldots, O_{t-1}})$$ product rule

$$= P(S_t = i \mid O_1, \ldots, O_{t-1}) P(O_t \mid S_t = i)$$ cond. ind.

$$= \left[\sum_{j=1}^{n} P(S_t = i, S_{t-1} = j \mid O_1, \ldots, O_{t-1})\right] P(O_t \mid S_t = i)$$ marginalization

$$= \left[\sum_{j=1}^{n} P(S_{t-1} = j \mid O_1, O_2, \ldots, O_{t-1}) P(S_t = i \mid S_{t-1} = j, O_1, \ldots, O_{t-1})\right] P(O_t \mid S_t = i)$$ product rule

$$= \sum_j q_{j,t-1}\, a_{ji}\, b_i(O_t)$$

numerator!

Denominator:

$$P(O_t \mid O_1, O_2, \ldots, O_{t-1}) = \sum_{i=1}^{n} P(O_t, S_t = i \mid O_1, O_2, \ldots, O_{t-1})$$

Ratio (recursion):

$$q_{it} = \frac{\sum_j q_{j,t-1}\, a_{ji}\, b_i(O_t)}{\sum_{i',j'} q_{j',t-1}\, a_{j'i'}\, b_{i'}(O_t)}$$

for $i = 1 \ldots n$

$O(n^2)$ update from time $t-1$ to time t

Base case: $q_{i1} = ?$ Exercise!