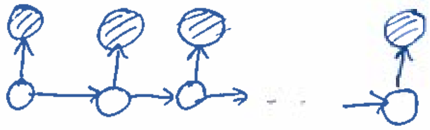
LECTURE - 14

11/12/2019

REVIEW Hidden Markov Models (HMMs)



Observations $o_t \in \{1, 2, \ldots, m\}$

States $s_t \in \{1, 2, \ldots, n\}$

Parameters : $a_{ij} = P(S_{t+1} = j \mid S_t = i)$ Transition matrix

$b_{ik} = P(O_t = k \mid S_t = i)$ Emission matrix

$\pi_i = P(S_1 = i)$ initial state distribution

Key Computations :

1) How to compute likelihood $P(o_1, o_2, \ldots, o_T)$?

2) How to decode $\underset{s_1, s_2, \ldots, s_T}{\arg\max} P(s_1, \ldots, s_T \mid o_1, o_2, \ldots, o_T)$?

3) How to estimate (learn) $\{\pi_i, a_{ij}, b_{ik}\}$ from data ?

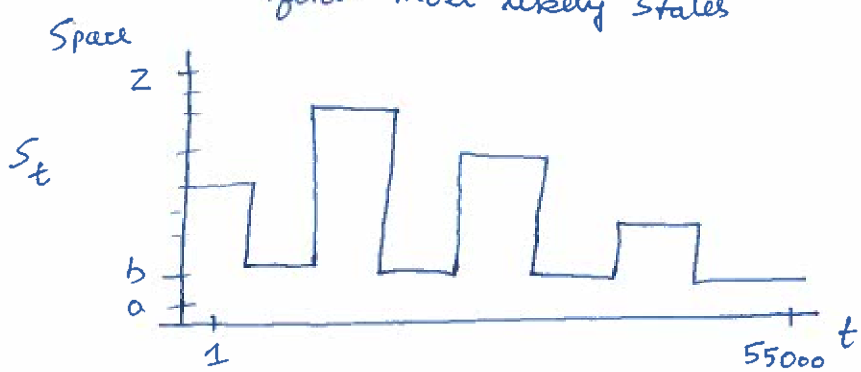
HW 7.1 Given $\{a_{ij}, b_{ik}, \pi_i\}$ with $n = 27$, $m = 2$

- observed time sequence

$$\{0101011110 \ldots \ldots 01\} \quad T = 55000$$

- decode $P(s_1, s_2, \ldots, s_T \mid o_1, \ldots, o_T)$

find most likely states



HW 7.4 : Belief Updating

How to compute $P(s_t \mid o_1, o_2, \ldots, o_t)$?

Important for real time monitoring.

(1) Computing likelihood $P(o_1, o_2, \dots, o_T)$

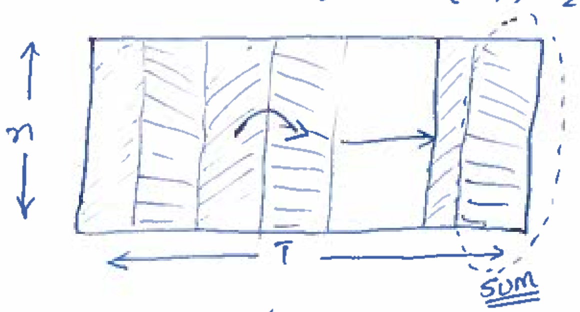
Efficient Recursion

$$P(o_1, o_2, \dots, o_{t+1}, S_{t+1}=j) = \sum_{i=1}^{n} P(o_1, o_2, \dots, o_{t+1}, S_t = i, S_{t+1}=j) \quad \text{marginalization}$$

$$= \sum_{i=1}^{n} P(o_1, o_2, \dots, o_t, S_t = i) \cdot P(S_{t+1}=j \mid o_1, o_2, \dots, o_t, S_t = i) \cdot P(o_{t+1} \mid o_1, \dots, o_t, S_t = i, S_{t+1}=j) \quad \text{Product rule \& C.I.}$$

$$= \sum_{i=1}^{n} \underbrace{P(o_1, o_2, \dots, o_t, S_t = i)}_{\text{recursive instance}} \cdot \underbrace{P(S_{t+1}=j \mid S_t = i) \cdot P(o_{t+1} \mid S_{t+1}=j)}_{\text{CPTs}}$$

Shorthand Notation : $\alpha_{it} = P(o_1, o_2, \dots, o_t, S_t = i)$



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

FORWARD ALGORITHM IN HMMs

Base Case : $\alpha_{i1} = P(o_1, S_1 = i) = P(S_1 = i) \cdot P(o_1 \mid S_1 = i)$ P.R

$\Rightarrow \alpha_{i1} = \pi_i\, b_i(o_1)$

*Likelihood $P(o_1, o_2, \dots, o_T) = \sum_{i=1}^{n} P(o_1, \dots, o_T, S_T = i)$ marginalization

$= \sum_{i=1}^{n} \alpha_{iT}$ (Sum of the last column of matrix)

* warning : for long sequences, watch out for underflow ...

(2) Computing most likely state sequence

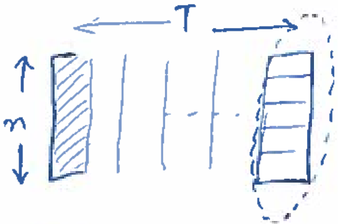
$$\{s_1^*, s_2^*, \dots, s_T^*\} = \arg\max_{s_1, s_2, \dots, s_T} P(s_1, s_2, \dots, s_T \mid o_1, o_2, \dots, o_T)$$

$$= \arg\max_{s_1, s_2, \dots, s_T} \left[\frac{P(s_1, s_2, \dots, s_T, o_1, o_2, \dots, o_T)}{P(o_1, o_2, \dots, o_T)} \right] \quad \text{Product rule}$$

$P(o_1, o_2, \dots, o_T) \longrightarrow$ Constant w.r.t $s_1, s_2, \dots, s_T$

$$= \arg\max_{s_1, s_2, \dots, s_T} \left[P(s_1, s_2, \dots, s_T, o_1, o_2, \dots, o_T) \right]$$

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



log-prob. of most likely "t-step path" of hidden States $S_1, S_2, \cdots, S_t$ that explains $O_1, O_2, \cdots, O_t$ and ends at $S_t = i$ at time t.

* Base Case : when $t = 1$

$$\ell^*_{i1} = \log P(O_1, S_1 = i) = \log\left[P(S_1 = i) P(O_1 | S_1 = i)\right] = \log \pi_i + \log b_i(O_1)$$

use this to fill first column of the matrix

* Recursive Step from time step t to $t+1$

$$\ell^*_{j,t+1} = \max_{S_1, S_2, \cdots, S_t} \log P(O_1, O_2, \cdots, O_{t+1}, S_1, S_2, \cdots, S_t, S_{t+1} = j)$$

$$= \max_{S_1, S_2, \cdots S_{t-1}} \underbrace{\max_{i}}_{\text{value of } S_t} \log P(O_1, O_2, \cdots, O_{t+1}, S_1, S_2, \cdots, S_t = i, S_{t+1} = j)$$

$$= \max_{S_1, \cdots, S_{t-1}} \max_{i} \log\Big[P(O_1, O_2, \cdots, O_t, S_1, \cdots, S_{t-1}, S_t = i) \cdot$$
$$P(S_{t+1} = j | \cancel{O_1, \cdots, O_t, S_1, \cdots, S_{t-1}}, S_t = i) \cdot$$
$$P(O_{t+1} | \cancel{O_1, \cdots, O_t, S_1, \cdots, S_{t-1}, S_t = i}, S_{t+1} = j)\Big]$$

Product rule

$$= \max_{S_1, \cdots, S_{t-1}} \max_{i} \Big\{ \log P(O_1, O_2, \cdots, O_t, S_1, S_2, \cdots, S_{t-1}, S_t = i)$$
$$+ \log P(S_{t+1} = j | S_t = i) + \log P(O_{t+1} | S_{t+1} = j) \Big\}$$

$$\ell^*_{j,t+1} = \max_{i} \Big[\underbrace{\max_{S_1, S_2, \cdots, S_{t-1}} \log P(O_1, \cdots, O_t, S_1, \cdots, S_{t-1}, S_t = i)}_{\text{recursive instance}} + \log \underbrace{P(S_{t+1} = j | S_t = i)}_{\text{CPT}} \Big]$$
$$+ \log \underbrace{P(O_{t+1} | S_{t+1} = j)}_{\text{CPT}}$$

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

(use this to recursively compute colums of matrix.)

- How to derive $\{s_1^*, s_2^*, \ldots, s_T^*\}$ from matrix $\{\ell^*_{it}\}$? $\begin{array}{l} i=1 \text{ to } n \\ t=1 \text{ to } T \end{array}$

* Record most likely state transitions:

$$\Phi_{t+1}(j) = \arg\max_i \left[\ell^*_{it} + \log a_{ij} \right]$$

- what is the most likely state at time 't' given '$s_{t+1} = j$' and we see $\{o_1, o_2, \ldots, o_{t+1}\}$?

* Compute $\{s_T^*, s_{T-1}^*, \ldots, s_1^*\}$ by backtracking:

$$s_T^* = \arg\max_i \left[\ell^*_{iT} \right]$$

for $t = T-1$ to 1,

$$\cancel{s_t^* = \Phi_{t+1}(s^*_{t+1})} \quad s_t^* = \Phi_{t+1}(s^*_{t+1})$$

- $\{s_1^*, s_2^*, \ldots, s_T^*\}$ called viterbi path

Viterbi algorithm: instance of dynamic programming.

(3) Learning in HMMs

Given: sequence of observations $\{o_1, o_2, \ldots, o_T\}$
(just one for simplicity)

Goal: Estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(o_1, o_2, \ldots, o_T)$

Assume cardinality 'n' of hidden states is fixed
$s_t \in \{1, 2, \ldots, n\}$

* CPTs to estimate :

$\pi_i = P(S_1 = i)$ root node

$a_{ij} = P(S_{t+1} = j \mid S_t = i)$ shared across time

$b_{ik} = P(O_t = k \mid S_t = i)$ shared across time

Recall from EM algorithm :

to update $P(X_i = x \mid Pa_i = \pi)$, need to compute $P(X_i = x, Pa_i = \pi \mid V)$

* E-step for HMMs :

Compute $P(S_1 = i \mid O_1, O_2, \cdots, O_T)$

$P(S_{t+1} = j, S_t = i \mid O_1, \cdots, O_T)$

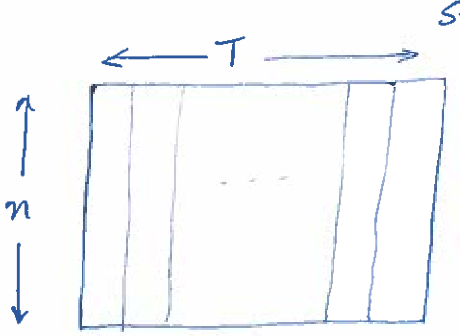
$P(O_t = k, S_t = i \mid O_1, \cdots, O_T) = I(O_t, k) P(S_t = i \mid O_1, \cdots, O_T)$

* How to compute these posteriors ?

Analogous to $\alpha_{it} = P(O_1, \cdots, O_t, S_t = i)$ up to and including time t

define $\beta_{it} = P(\underline{O_{t+1}}, O_{t+2}, \cdots, O_T \,\underline{\mid}\, S_t = i)$ beyond time t.

Start at '$t+1$' conditioning bar



* Recursion for β_{it} :

(1) base case : $t = T$

$\beta_{iT} = P(\_\_ \mid S_T = i)$? Set $\beta_{iT} = 1$ for all i

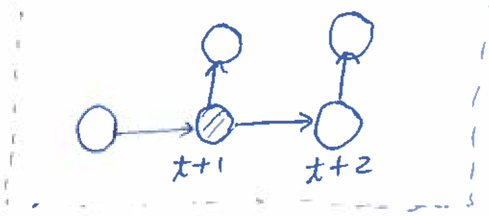
(2) Backwards step from time $t+1$ to t

$$\beta_{it} = P(O_{t+1}, \ldots, O_T \mid S_t = i)$$

$$= \sum_{j=1}^{n} P(O_{t+1}, \ldots, O_T, S_{t+1} = j \mid S_t = i) \quad \text{marginalization}$$

$$= \sum_{j=1}^{n} P(S_{t+1} = j \mid S_t = i)\, P(O_{t+1} \mid \cancel{S_t = i}, S_{t+1} = j) \cdot P(O_{t+2}, \ldots, O_T \mid \cancel{S_t = i}, S_{t+1} = j, \cancel{O_{t+1}})$$

Product Rule



$$= \sum_{j=1}^{n} P(S_{t+1} = j \mid S_t = i) \cdot P(O_{t+1} \mid S_{t+1} = j) \cdot P(O_{t+2}, \ldots, O_T \mid S_{t+1} = j) \quad \text{C.I}$$

$$\therefore \boxed{\beta_{it} = \sum_{j=1}^{n} a_{ij}\, b_j(O_{t+1}) \cdot \beta_{j,t+1}}$$