→ Learning CPTs from incomplete data

EM update:

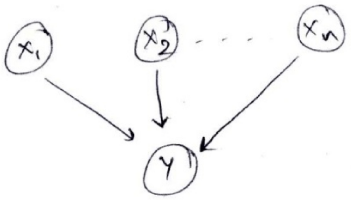
$$P(X_i = x \mid pa_i = \pi) \leftarrow \frac{\sum_t P(X_i = x, pa_i = \pi \mid v^t)}{\sum_t P(pa_i = \pi \mid v^t)} \quad \text{(for roots with parents)}$$

$$P(X_i = x) \leftarrow \frac{1}{T} \sum_t P(X_i = x \mid v^t) \quad \text{(root nodes)}$$

where v^t denotes visible nodes.

→ AND | NOISY-OR

$$P(Y = 1 \mid \vec{x}) = 1 - \prod_{i=1}^{n} (1 - p_i)^{x_i}$$



we learn from data $\{(\vec{x}_t, y_t)\}_{t=}^{T}$

EM update of this model is given by:

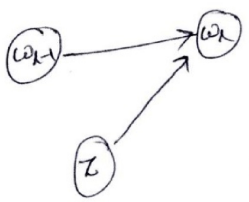
$$p_i \leftarrow \frac{1}{T_i} \sum_{t=1}^{T} \frac{y_t x_{it} p_i}{1 - \prod_{j=1}^{n} (1 - p_j)^{x_{jt}}}$$

where T_i is the number of examples in which $X_i = 1$

→ Linear interpolation (mixing) of n-gram models:

$$P_{in}(\omega_t \mid \omega_{t-1}) = \lambda P_1(\omega_t) + (1 - \lambda) P_2(\omega_t \mid \omega_{t-1})$$

To estimate λ, we rewrite this as a hidden variable model



$$P(\omega_t \mid \omega_{t-1}, z) = \begin{cases} P_1(\omega_t) & \text{if } z = 1 \\ P_2(\omega_t \mid \omega_{t-1}) & \text{if } z = 2 \end{cases}$$

$z = \{1, 2\}$

$P(z = 1) = \lambda$

$P(z = 2) = 1 - \lambda$

In this model,

$$P(\omega_t/\omega_{t-1}) = \sum_{z'} P(\omega_t, z'/\omega_{t-1})$$

$$= \sum_{z'} P(z'/\omega_{t-1})\, P(\omega_t/z', \omega_{t-1}) \quad \text{[Product Rule]}$$

$$= \sum_{z'} P(z')\, P(\omega_t/z', \omega_{t-1}) \quad \text{[Conditional independence]}$$

$$= \lambda P_1(\omega_t) + (1-\lambda) P_2(\omega_t/\omega_{t-1})$$

E-STEP

Compute posterior probability

* $P(z/\omega_t, \omega_{t-1}) = \dfrac{P(\omega_t/z, \omega_{t-1})\, P(z/\omega_{t-1})}{P(\omega_t/\omega_{t-1})}$ [Conditional Bayes' Rule]

* $P(z=1/\omega_t, \omega_{t-1}) = \dfrac{\lambda P_1(\omega_t/\omega_{t-1})}{\lambda P_1(\omega_t) + (1-\lambda) P_2(\omega_t/\omega_{t-1})}$

* $P(z=2/\omega_t, \omega_{t-1}) = 1 - P(z=1/\omega_t, \omega_{t-1})$

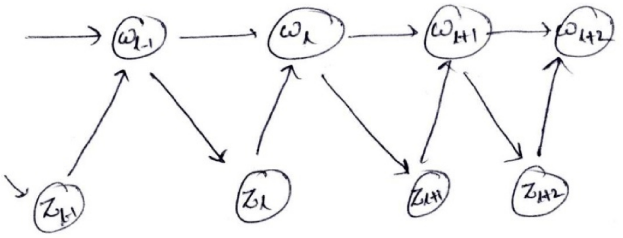
M-STEP

In this model

$$P(z=1) \leftarrow \frac{1}{L} \sum_{t=1}^{L} P(z=1/\omega_{t-1}, \omega_t)$$

λ

The above model over simplifies the belief network. In real world applications, smoothing parameter, λ would depend on the previous word.



$$P(z_\lambda = 1 / \omega_{\lambda-1}) = \lambda(\omega_{\lambda-1}) \qquad P(z_\lambda = 2 / \omega_{\lambda-1}) = 1 - \lambda(\omega_{\lambda-1})$$

$$P(\omega_\lambda / \omega_{\lambda-1}) = \lambda(\omega_{\lambda-1})\, P_1(\omega_\lambda) + 1 - \lambda(\omega_{\lambda-1})\, P_2(\omega_\lambda / \omega_{\lambda-1})$$

Using EM, we can estimate λ, which has the same size as the words in the vocabulary.

HIDDEN MARKOV MODELS

→ Random variables :

$S_t = \{1, 2, \ldots n\}$ state at time t

$O_t = \{1, 2, \ldots m\}$ observation at time t.

* Note that n, m have no relation ($n > m$, $n = m$, $n < m$ ← all possible)
* observations, O_t can be thought of as a noisy reflection of the hidden state S_t

→ Ex: puppy training.

$S = \{$"have-to-go", "doesn't-have-to-go", "went"$\}$

$O = \{$"wagging tail", "squeaking", "running in circles", "hiding in corners"$\}$

→ Ex: Speech recognition:

S_t = units of language : words, syllables, phonemes

O_t = acoustic measurement : energy in a window, no. of peaks in a window...

→ Ex: Robotics

S_t = location

O_t = sensor reading

MARKOV ASSUMPTIONS IN HMM:

finite Context:

$$P(S_t / S_1, S_2 \ldots S_{t-1}) = P(S_t / S_{t-1})$$

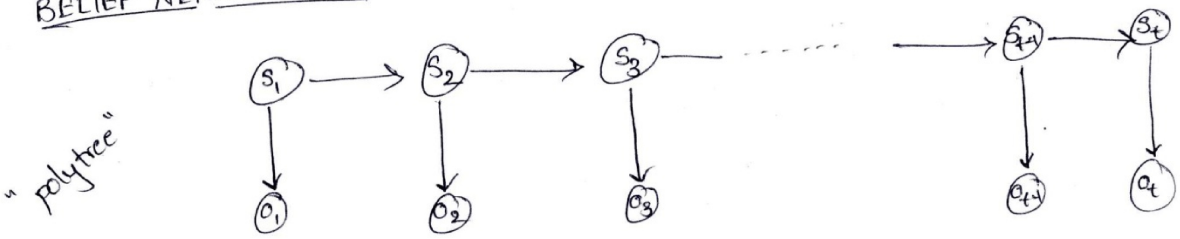
$$P(O_t / S_1, S_2 \ldots S_{t-1}, S_t \ldots S_T) = P(O_t / S_t)$$

shared CPTs

$$P(S_{t+1} = s' / S_t = s) = P(S_t = s' / S_{t-1} = s)$$

$$P(O_t = o / S_t = s) = P(O_{t+1} = o / S_{t+1} = s)$$

BELIEF NETWORK OF HMM



PARAMETERS OF HMM:

$\pi_i = \pi(s_1 = i)$ [Initial State Distribution]

$n \times n$ matrix $a_{ij} = P(s_{t+1} = j / s_t = i)$ [Transition matrix] $i = 1, 2, \ldots n$, $j = 1, 2, \ldots n$.

$n \times m$ matrix $b_{ik} = P(o_t = k / s_t = i)$ [Emission matrix] $i = 1, 2, \ldots n$, $k = 1, 2, \ldots m$

JOINT DISTRIBUTION OF HMM:

$$P(\vec{s}, \vec{o}) = P(s_1) \prod_{t=2}^{T} P(s_t / s_{t-1}) \prod_{t=1}^{T} P(o_t / s_t)$$

$s_1, s_2 \ldots s_T$ $\quad$ $o_1, o_2 \ldots o_T$

→ Ex: * Isolated word speech recognition

* Consider the problem of recognising the word "CAT"
* We have to build HMM that assigns high probability to "CAT" utterances
* " " " " low probability to other utterances
* " " " "

→ USES HMM with 5 states

State #	Sound
1	Initial silence
2	"C"
3	"A"
4	"T"
5	final silence

$$\pi_i = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$ ← always start with state 1

$$a_{ij} = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 & 0 \\ 0 & 0 & 0.99 & 0.01 & 0 \\ 0 & 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

→ this is an upper diagonal transition matrix

→ special case : left to right HMM

KEY QUESTIONS FOR HMMs

perform inference → given $\{\pi_i, a_{ij}, b_{ik}\}$ parameters

1) How to Compute likelihood $P(o_1, o_2 \cdots o_T)$

2) How to Compute most likely state sequence

$$(s_1^*, s_2^* \cdots s_T^*) = \underset{s_1, s_2 \cdots s_T}{\text{argmax}} \; P(s_1, s_2 \cdots s_T / o_1, o_2 \cdots o_T)$$

↳ n^T possible settings.

3) How to compute $P(s_t = i / o_1, o_2 \cdots o_t)$?

↳ this is can be thought of as updating belief in real time.

learning → given $\{o_1, o_2 \cdots o_T\}$ observations.

4) How to estimate parameters $\{\pi_i, a_{ij}, b_{ik}\}$ that maximises likelihood $P(o_1, o_2 \cdots o_T)$ } EM algorithm.

---

1) $P(o_1, o_2 \cdots o_T) = \sum_{s_1, s_2 \cdots s_T} P(s_1, s_2 \cdots s_T, o_1, o_2 \cdots o_T)$

(hidden, observed)

sum over n^T hidden state sequences

$$= \sum_{\vec{s}} P(s_1) \prod_{t=2}^{T} P(s_t / s_{t-1}) \prod_{t=1}^{T} P(o_t / s_t)$$