Review

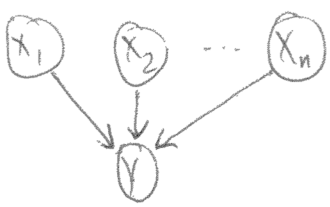
- Learning CPTs from incomplete data

EM update $$P(X_i = x \mid pa_i = \pi) \leftarrow \frac{\sum_t P(X_i = x, pa_i = \pi \mid v^{(t)})}{\sum_t P(pa_i = \pi \mid v^{(t)})}$$ for nodes w/ parents

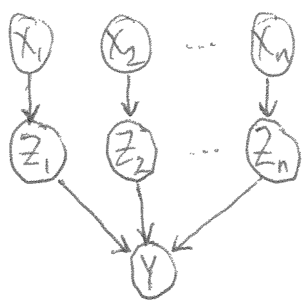
$$P(X_i = x) \leftarrow \frac{1}{T} \sum_t P(X_i = x \mid v^{(t)})$$ for root nodes

where $v^{(t)}$ denotes visible nodes

- Ex: Noisy-OR



Hidden variable model



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

Same as noisy-OR!

How to estimate p_i from data $\{(\vec{x}_t, y_t)\}_{t=1}^{T}$?

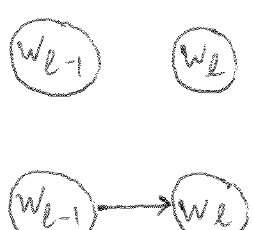
EM update : $$p_i = P(Z_i = 1 \mid X_i = 1) \leftarrow \frac{1}{T_i} \sum_{t=1}^{T} \frac{y_t x_{it} p_i}{1 - \prod_{j=1}^{n} (1-p_j)^{x_{jt}}}$$

from posterior $P(Z_i = 1 \mid \vec{X}, Y)$, computed in terms of CPTs

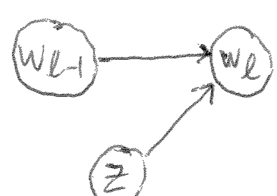
- Ex: Mixture of n-gram models

unigram

bigram



Hidden variable model



$$P(w_\ell \mid w_{\ell-1}) = \lambda P_1(w_\ell) + (1-\lambda) P_2(w_\ell \mid w_\ell$$
$$= P_m(w_\ell \mid w_{\ell-1})$$

same as mixture model!

How to estimate λ from data $\{(w_{\ell-1}, w_\ell)\}_{\ell=1}^{L}$?

EM update : $$\lambda = P(Z=1) \longleftarrow \frac{1}{L} \sum_{\ell=1}^{L} P(Z=1 \mid w_{\ell-1}, w_\ell)$$

posterior computed in terms of CPTs

Hidden Markov Models (HMMs)

- Random variables

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

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

  "noisy" reflection of hidden state S_t

- Ex: puppy training

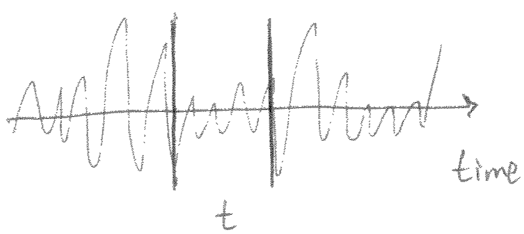
  $S = \{$"have-to-go", "don't-have-to-go", "went"$\}$

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

- Ex: speech recognition

  S = units of language: words, syllables, phonemes

  O = acoustic measurements



- Ex: robotics

  S_t = location

  O_t = sensor readings

- Belief Network of HMM

$$
\begin{tikzcd}
S_1 \arrow[r] \arrow[d] & S_2 \arrow[r] \arrow[d] & S_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & S_{T-1} \arrow[r] \arrow[d] & S_T \arrow[d] \\
O_1 & O_2 & O_3 & & O_{T-1} & O_T
\end{tikzcd}
$$

Polytree?

YES!

- Markov assumptions

  - finite context

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

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

  - shared CPTs

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

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

- Joint distribution

$$P(\underbrace{\vec{S}}_{(S_1, S_2, \ldots, S_T)}, \underbrace{\vec{O}}_{(O_1, O_2, \ldots, O_T)}) = \underbrace{P(S_1)}_{\text{initial state}} \cdot \left\{ \prod_{t=2}^{T} P(S_t \mid S_{t-1}) \right\} \cdot \left\{ \prod_{t=1}^{T} P(O_t \mid S_t) \right\}$$

- Parameters (CPTs)

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

$a_{ij} = P(S_{t+1} = j \mid S_t = i)$ transition matrix $(n \times n)$

$b_{ik} = P(O_t = k \mid S_t = i)$ emission matrix $(n \times m)$

For clarity: $b_{ik} = b_i(k)$ alternate notation

Ex: isolated word speech recognizer

recognize word "CAT"

build HMM that assigns high probability to "CAT" utterances
low probability to other utterances

use HMM with 5 states

state #	sound
1	initial silence
2	"C"
3	"A"
4	"T"
5	final silence

$$\pi = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

must start in state 1

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

upper diagonal transition matrix

special case: left to right HMM

Key Questions for HMMs

Inference: given $\{\pi_i, a_{ij}, b_{ik}\}$ parameters

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

2) How to compute most likely state sequence?

$$(s_1^*, s_2^*, \ldots, s_T^*) = \underset{s_1, s_2, \ldots, s_T}{\text{argmax}} \; P(\underbrace{S_1, S_2, \ldots, S_T}\,|\, O_1, O_2, \ldots, O_T)$$

sequence of states that maximize posterior probability

3) How to compute $P(S_t = i \,|\, O_1, O_2, \ldots, O_t)$?

updating belief in real-time

Learning: given $\{o_1, o_2, \ldots, o_T\}$ observations

4) How to estimate parameters $\{\pi_i, a_{ij}, b_{ik}\}$ that maximize likelihood $P(o_1, o_2, \ldots, o_T)$?

EM algorithm!

---

1) Computing likelihood

$$P(O_1, O_2, \ldots, O_T) = \sum_{\vec{S}} P(\overbrace{S_1, S_2, \ldots, S_T}^{\text{hidden}}, \overbrace{O_1, O_2, \ldots, O_T}^{\text{observed}})$$

Sum over n^T hidden state sequences

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

• Efficient recursion

$$P(O_1, O_2, \ldots, O_t, O_{t+1}, S_{t+1} = j) = \sum_{i=1}^{n} P(O_1, \ldots, O_t, O_{t+1}, S_{t+1} = j, S_t = i)$$ marginalization

$$= \sum_{i=1}^{n} P(O_1, \ldots, O_t, S_t = i) P(S_{t+1} = j, O_{t+1} | S_t = i, O_1, \ldots, O_t)$$ product rule

$$= \sum_{i=1}^{n} P(O_1, \ldots, O_t, S_t = i) P(S_{t+1} = j, O_{t+1} | S_t = i)$$ conditional independence

$$= \sum_{i=1}^{n} P(O_1, \ldots, O_t, S_t = i) P(S_{t+1} = j | S_t = i) P(O_{t+1} | S_{t+1} = j, S_t = i)$$ product rule

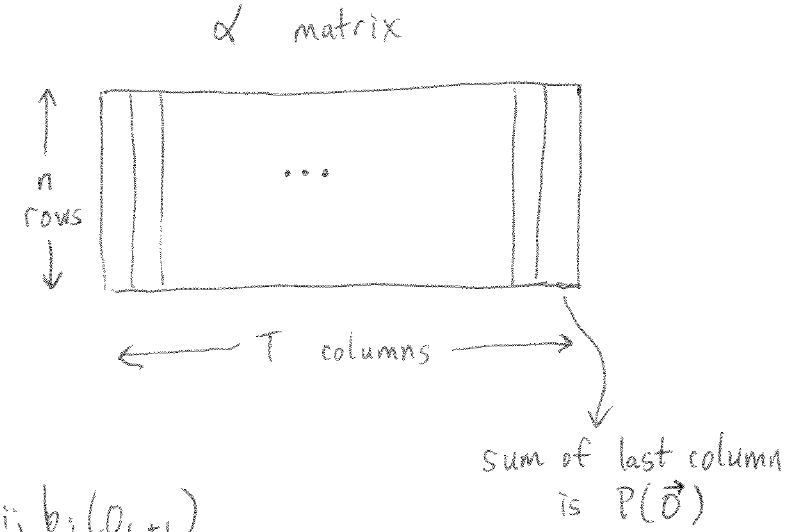
$$= \sum_{i=1}^{n} \underbrace{P(O_1, \ldots, O_t, S_t = i)}_{\text{recursive instance}} \underbrace{P(S_{t+1} = j | S_t = i) P(O_{t+1} | S_{t+1} = j)}_{\text{CPTs}}$$ conditional independence

- Shorthand notation

$$\alpha_{it} = P(O_1, O_2, \ldots, O_t, S_t = i)$$

$i = 1, \ldots, n$ # hidden states

$t = 1, \ldots, T$ sequence length



- Forward algorithm

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

- Initial condition (1st column of α)

$$\alpha_{i1} = P(O_1, S_1 = i) = P(S_1 = i)P(O_1 | S_1 = i) = \pi_i b_i(O_1) \quad \text{for } i = 1, \ldots, n$$

- Back to likelihood computation

$$P(O_1, O_2, \ldots, O_T) = \sum_{i=1}^{n} P(O_1, O_2, \ldots, O_T, S_T = i) \qquad \text{marginalization}$$

$$= \sum_{i=1}^{n} \alpha_{iT}$$

- Scales as $O(n^2 T)$

linear, not exponential, in sequence length

quadratic in # of states

- Warning: naive calculations will underflow for long sequences

because $P(O_1, O_2, \ldots, O_T) \ll 1$