Learning CPTs from incomplete data

EM update: 
\[ P(X_i = x | p_a = \pi) \leftarrow \frac{\sum_{t} P(X_i = x, p_a = \pi | v(t))}{\sum_{t} P(p_a = \pi | v(t))} \]
for nodes w/ parents

\[ P(X_i = x) \leftarrow \frac{1}{T} \sum_{t} P(X_i = x | v(t)) \]
for root nodes

where \( v(t) \) denotes visible nodes

Ex: Noisy-OR

Hidden variable model

\[ P(Y = 1 | \vec{x}) = 1 - \prod_{i=1}^{n} (1 - p_i)^{x_i} \]
Same as noisy-OR!

How to estimate \( p_i \) from data \( \{(x_t, y_t)\}_t^T \)?

EM update: 
\[ p_i = P(Z_i = 1 | x_i = 1) \leftarrow \frac{1}{T_i} \sum_{t=1}^{T} \frac{y_t x_i + p_i}{1 - \prod_{j=1}^{n} (1 - p_j)^{x_j}} \]
from posterior 
computed in terms of CPTs

Ex: Mixture of n-gram models

Hidden variable model

\[ P(w_l | w_{l-1}) = \lambda P_i(w_l) + (1 - \lambda) P_m(w_l | w_{l-1}) \]
Same as mixture model!

How to estimate \( \lambda \) from data \( \{(w_{l-1}, w_l)\}_l^L \)?

EM update: 
\[ \lambda = P(Z = 1) \leftarrow \frac{1}{L} \sum_{l=1}^{L} P(Z = 1 | w_{l-1}, w_l) \]
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

Polytree? YES!
• Markov assumptions

- Finite context

\[ P(S_t | S_1, S_2, \ldots, S_{t-1}) = P(S_t | S_{t-1}) \]
\[ P(O_t | S_1, S_2, \ldots, S_{t-1}, S_t, S_{t+1}, \ldots, S_T) = P(O_t | S_t) \]

- Shared CPTs

\[ P(S_{t+1} = s' | S_t = s) = P(S_t = s' | S_{t-1} = s) \]
\[ P(O_t = o | S_t = s) = P(O_{t+1} = o | S_{t+1} = s) \]

• Joint distribution

\[ P(S', O) = P(S_1) \cdot \prod_{t=2}^{T} P(S_t | S_{t-1}) \cdot \prod_{t=1}^{T} P(O_t | S_t) \]

\((S_1, S_2, \ldots, S_T)\quad (O_1, O_2, \ldots, O_T)\quad \text{initial state}\)

• Parameters (CPTs)

\[ \Pi_i = P(S_1 = i) \quad \text{initial state distribution} \]
\[ a_{ij} = P(S_{t+1} = j | S_t = i) \quad \text{transition matrix} \ (n \times n) \]
\[ b_{ik} = P(O_t = k | S_t = i) \quad \text{emission matrix} \ (n \times m) \]

For clarity: \( b_{ik} = b_i(k) \quad \text{alternate notation} \)
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

<table>
<thead>
<tr>
<th>state #</th>
<th>sound</th>
<th>( \pi = \begin{bmatrix} 1 \ 0 \ 0 \ 0 \ 0 \end{bmatrix} ) must start in state 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>initial silence</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>&quot;C&quot;</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&quot;A&quot;</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&quot;T&quot;</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>final silence</td>
<td></td>
</tr>
</tbody>
</table>

\[ a_{ij} = \begin{bmatrix} 0.9 & 0.1 & \emptyset \\ 0.9 & 0.1 & \emptyset \\ 0.99 & 0.01 & \emptyset \\ \emptyset & 0.4 & 0.6 \\ \emptyset & \emptyset & 1 \end{bmatrix} \]  
upper diagonal transition matrix

special case: left to right HMM

Key Questions for HMMs

Inference: given \( \{ \pi, a_{ij}, b_{ik} \} \) parameters

1) How to compute likelihood \( P(O_1, O_2, \ldots, O_T) \)?
2) How to compute most likely state sequence?

\( (s_1^*, s_2^*, \ldots, s_T^*) = \arg\max_{s_1, s_2, \ldots, s_T} P(s_1, s_2, \ldots, s_T | O_1, O_2, \ldots, O_T) \)

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**

\[
P(O_1, O_2, \ldots, O_T) = \sum_{\pi, a, b} P(S_1, S_2, \ldots, S_T, O_1, O_2, \ldots, O_T)
\]

\[
\sum_{\pi, a, b} = \sum_{\pi} \sum_{a} \sum_{b} P(S_1) \prod_{t=2}^{T} P(S_t | S_{t-1}) \prod_{t=1}^{T} P(O_t | S_t)
\]

- **Efficient recursion**

\[
P(O_1, O_2, \ldots, O_T, O_{t+1}, S_{t+1} = j) = \sum_{i=1}^{n} P(O_1, \ldots, O_t, O_{t+1}, S_{t+1} = j, S_t = i) \ 	ext{ marginalsization}
\]

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

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

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

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

- **Recursive instance**

- **CPTs**
• Shorthand notation

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

\[ \alpha \text{ matrix} \]

\[ \begin{array}{c}
\uparrow \\
\downarrow \\
\downarrow \\
\text{n rows} \\
\text{T columns} \\
\uparrow \\
\downarrow \\
\text{sum of last column} \\
is \quad P(\hat{O})
\end{array} \]

\[ t = 1, \ldots, T \quad \text{sequence length} \]

\[ i = 1, \ldots, n \quad \text{# hidden states} \]

• Forward algorithm

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

• Initial condition (1st column of \( \alpha \))

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

• Back to likelihood computation

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

\[ = \sum_{i=1}^{n} \alpha_{iT} \]

• Scales as \( O(n^2 T) \)

linear, not exponential, in sequence length
quadritic in # of states

• Warning: naive calculations will underflow for long sequences
because \( P(O_1, O_2, \ldots, O_T) \ll 1 \)