6.864: Lecture 5 (September 22nd, 2005)

The EM Algorithm
Overview

• The EM algorithm in general form

• The EM algorithm for hidden markov models (brute force)

• The EM algorithm for hidden markov models (dynamic programming)
An Experiment/Some Intuition

• I have three coins in my pocket,
  
  Coin 0 has probability $\lambda$ of heads;
  Coin 1 has probability $p_1$ of heads;
  Coin 2 has probability $p_2$ of heads

• For each trial I do the following:
  
  First I toss Coin 0
  If Coin 0 turns up heads, I toss coin 1 three times
  If Coin 0 turns up tails, I toss coin 2 three times

  I don’t tell you whether Coin 0 came up heads or tails,
  or whether Coin 1 or 2 was tossed three times,
  but I do tell you how many heads/tails are seen at each trial

• you see the following sequence:
  
  $\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle$

  What would you estimate as the values for $\lambda$, $p_1$ and $p_2$?
Maximum Likelihood Estimation

- We have data points $x_1, x_2, \ldots, x_n$ drawn from some (finite or countable) set $\mathcal{X}$.

- We have a parameter vector $\Theta$.

- We have a parameter space $\Omega$.

- We have a distribution $P(x \mid \Theta)$ for any $\Theta \in \Omega$, such that
  \[ \sum_{x \in \mathcal{X}} P(x \mid \Theta) = 1 \text{ and } P(x \mid \Theta) \geq 0 \text{ for all } x \]

- We assume that our data points $x_1, x_2, \ldots, x_n$ are drawn at random (independently, identically distributed) from a distribution $P(x \mid \Theta^*)$ for some $\Theta^* \in \Omega$. 
Log-Likelihood

- We have data points $x_1, x_2, \ldots x_n$ drawn from some (finite or countable) set $\mathcal{X}$

- We have a parameter vector $\Theta$, and a parameter space $\Omega$

- We have a distribution $P(x \mid \Theta)$ for any $\Theta \in \Omega$

- The likelihood is

$$Likelihood(\Theta) = P(x_1, x_2, \ldots x_n \mid \Theta) = \prod_{i=1}^{n} P(x_i \mid \Theta)$$

- The log-likelihood is

$$L(\Theta) = \log Likelihood(\Theta) = \sum_{i=1}^{n} \log P(x_i \mid \Theta)$$
A First Example: Coin Tossing

- $\mathcal{X} = \{H, T\}$. Our data points $x_1, x_2, \ldots x_n$ are a sequence of heads and tails, e.g.

  HHTTHHNHTTHH

- Parameter vector $\Theta$ is a single parameter, i.e., the probability of coin coming up heads

- Parameter space $\Omega = [0, 1]$

- Distribution $P(x \mid \Theta)$ is defined as

  $$P(x \mid \Theta) = \begin{cases} 
  \Theta & \text{If } x = H \\
  1 - \Theta & \text{If } x = T 
  \end{cases}$$
Maximum Likelihood Estimation

• Given a sample $x_1, x_2, \ldots x_n$, choose

$$\Theta_{ML} = \arg\max_{\Theta \in \Omega} L(\Theta) = \arg\max_{\Theta \in \Omega} \sum_i \log P(x_i | \Theta)$$

• For example, take the coin example:
  say $x_1 \ldots x_n$ has $\text{Count}(H)$ heads, and $(n - \text{Count}(H))$ tails

$$\Rightarrow \quad L(\Theta) = \log \left( \Theta^{\text{Count}(H)} \times (1 - \Theta)^{n - \text{Count}(H)} \right)$$

$$= \text{Count}(H) \log \Theta + (n - \text{Count}(H)) \log(1 - \Theta)$$

• We now have

$$\Theta_{ML} = \frac{\text{Count}(H)}{n}$$
A Second Example: Probabilistic Context-Free Grammars

- $\mathcal{X}$ is the set of all parse trees generated by the underlying context-free grammar. Our sample is $n$ trees $T_1 \ldots T_n$ such that each $T_i \in \mathcal{X}$.

- $R$ is the set of rules in the context free grammar
  $N$ is the set of non-terminals in the grammar

- $\Theta_r$ for $r \in R$ is the parameter for rule $r$

- Let $R(\alpha) \subset R$ be the rules of the form $\alpha \rightarrow \beta$ for some $\beta$

- The parameter space $\Omega$ is the set of $\Theta \in [0, 1]^{|R|}$ such that

  $\text{for all } \alpha \in N \sum_{r \in R(\alpha)} \Theta_r = 1$
We have

\[ P(T \mid \Theta) = \prod_{r \in R} \Theta_r^{\text{Count}(T, r)} \]

where \( \text{Count}(T, r) \) is the number of times rule \( r \) is seen in the tree \( T \)

\[ \Rightarrow \quad \log P(T \mid \Theta) = \sum_{r \in R} \text{Count}(T, r) \log \Theta_r \]
Maximum Likelihood Estimation for PCFGs

- We have
  \[
  \log P(T \mid \Theta) = \sum_{r \in R} \text{Count}(T, r) \log \Theta_r
  \]
  where \( \text{Count}(T, r) \) is the number of times rule \( r \) is seen in the tree \( T \)

- And,
  \[
  L(\Theta) = \sum_i \log P(T_i \mid \Theta) = \sum_i \sum_{r \in R} \text{Count}(T_i, r) \log \Theta_r
  \]

- Solving \( \Theta_{ML} = \arg\max_{\Theta \in \Omega} L(\Theta) \) gives
  \[
  \Theta_r = \frac{\sum_i \text{Count}(T_i, r)}{\sum_i \sum_{s \in R(\alpha)} \text{Count}(T_i, s)}
  \]
  where \( r \) is of the form \( \alpha \to \beta \) for some \( \beta \)
Multinomial Distributions

- $\mathcal{X}$ is a finite set, e.g., $\mathcal{X} = \{\text{dog, cat, the, saw}\}$

- Our sample $x_1, x_2, \ldots x_n$ is drawn from $\mathcal{X}$
  e.g., $x_1, x_2, x_3 = \text{dog, the, saw}$

- The parameter $\Theta$ is a vector in $\mathbb{R}^m$ where $m = |\mathcal{X}|$
  e.g., $\Theta_1 = P(\text{dog}), \Theta_2 = P(\text{cat}), \Theta_3 = P(\text{the}), \Theta_4 = P(\text{saw})$

- The parameter space is
  \[ \Omega = \{\Theta : \sum_{i=1}^{m} \Theta_i = 1 \text{ and } \forall i, \Theta_i \geq 0\} \]

- If our sample is $x_1, x_2, x_3 = \text{dog, the, saw}$, then
  \[ L(\Theta) = \log P(x_1, x_2, x_3 = \text{dog, the, saw}) = \log \Theta_1 + \log \Theta_3 + \log \Theta_4 \]
Models with Hidden Variables

• Now say we have two sets $\mathcal{X}$ and $\mathcal{Y}$, and a joint distribution $P(x, y \mid \Theta)$

• If we had fully observed data, $(x_i, y_i)$ pairs, then

$$L(\Theta) = \sum_i \log P(x_i, y_i \mid \Theta)$$

• If we have partially observed data, $x_i$ examples, then

$$L(\Theta) = \sum_i \log P(x_i \mid \Theta)$$

$$= \sum_i \log \sum_{y \in \mathcal{Y}} P(x_i, y \mid \Theta)$$
The **EM (Expectation Maximization) algorithm** is a method for finding

$$\Theta_{ML} = \arg\max_{\Theta} \sum_i \log \sum_{y \in Y} P(x_i, y | \Theta)$$
The Three Coins Example

• e.g., in the three coins example:
  \[ \mathcal{Y} = \{H, T\} \]
  \[ \mathcal{X} = \{HHH, TTT, HTT, THH, HHT, TTH, HTH, THT\} \]
  \[ \Theta = \{\lambda, p_1, p_2\} \]

• and
  \[ P(x, y \mid \Theta) = P(y \mid \Theta)P(x \mid y, \Theta) \]
  where
  \[ P(y \mid \Theta) = \begin{cases} 
  \lambda & \text{If } y = H \\
  1 - \lambda & \text{If } y = T
  \end{cases} \]
  and
  \[ P(x \mid y, \Theta) = \begin{cases} 
  p_1^h(1 - p_1)^t & \text{If } y = H \\
  p_2^h(1 - p_2)^t & \text{If } y = T
  \end{cases} \]
  where \( h = \text{number of heads in } x, \ t = \text{number of tails in } x \)
The Three Coins Example

- Various probabilities can be calculated, for example:

\[
P(x = \text{THT}, y = \text{H} \mid \Theta) = \lambda p_1 (1 - p_1)^2
\]
The Three Coins Example

- Various probabilities can be calculated, for example:

\[
P(x = \text{THT}, y = \text{H} \mid \Theta) = \lambda p_1 (1 - p_1)^2
\]

\[
P(x = \text{THT}, y = \text{T} \mid \Theta) = (1 - \lambda)p_2 (1 - p_2)^2
\]
The Three Coins Example

- Various probabilities can be calculated, for example:

\[ P(x = THT, y = H \mid \Theta) = \lambda p_1 (1 - p_1)^2 \]

\[ P(x = THT, y = T \mid \Theta) = (1 - \lambda) p_2 (1 - p_2)^2 \]

\[ P(x = THT \mid \Theta) = P(x = THT, y = H \mid \Theta) + P(x = THT, y = T \mid \Theta) \]
\[ = \lambda p_1 (1 - p_1)^2 + (1 - \lambda) p_2 (1 - p_2)^2 \]
The Three Coins Example

• Various probabilities can be calculated, for example:

\[ P(x = \text{THT}, y = \text{H} \mid \Theta) = \lambda p_1 (1 - p_1)^2 \]

\[ P(x = \text{THT}, y = \text{T} \mid \Theta) = (1 - \lambda) p_2 (1 - p_2)^2 \]

\[ P(x = \text{THT} \mid \Theta) = P(x = \text{THT}, y = \text{H} \mid \Theta) + P(x = \text{THT}, y = \text{T} \mid \Theta) = \lambda p_1 (1 - p_1)^2 + (1 - \lambda) p_2 (1 - p_2)^2 \]

\[ P(y = \text{H} \mid x = \text{THT}, \Theta) = \frac{P(x = \text{THT}, y = \text{H} \mid \Theta)}{P(x = \text{THT} \mid \Theta)} = \frac{\lambda p_1 (1 - p_1)^2}{\lambda p_1 (1 - p_1)^2 + (1 - \lambda) p_2 (1 - p_2)^2} \]
The Three Coins Example

- Fully observed data might look like:

  \((\langle HHH \rangle, H), (\langle TTT \rangle, T), (\langle HHH \rangle, H), (\langle TTT \rangle, T), (\langle HHH \rangle, H)\)

- In this case maximum likelihood estimates are:

  \[ \lambda = \frac{3}{5} \]

  \[ p_1 = \frac{\lambda}{\lambda} = \frac{9}{9} \]

  \[ p_2 = \frac{\lambda}{1 - \lambda} = \frac{0}{6} \]
The Three Coins Example

• Partially observed data might look like:

\[ \langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle \]

• How do we find the maximum likelihood parameters?
The Three Coins Example

- Partially observed data might look like:

  \[ \langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle \]

- If current parameters are \( \lambda, p_1, p_2 \)

  \[
P(y = H \mid x = \langle HHH \rangle) = \frac{P(\langle HHH \rangle, H)}{P(\langle HHH \rangle, H) + P(\langle HHH \rangle, T)} = \frac{\lambda p_1^3}{\lambda p_1^3 + (1 - \lambda)p_2^3}
  \]

  \[
P(y = H \mid x = \langle TTT \rangle) = \frac{P(\langle TTT \rangle, H)}{P(\langle TTT \rangle, H) + P(\langle TTT \rangle, T)} = \frac{\lambda(1 - p_1)^3}{\lambda(1 - p_1)^3 + (1 - \lambda)(1 - p_2)^3}
  \]
The Three Coins Example

- If current parameters are $\lambda, p_1, p_2$

\[
P(y = H \mid x = \langle HHH \rangle) = \frac{\lambda p_1^3}{\lambda p_1^3 + (1 - \lambda)p_2^3}
\]

\[
P(y = H \mid x = \langle TTT \rangle) = \frac{\lambda(1 - p_1)^3}{\lambda(1 - p_1)^3 + (1 - \lambda)(1 - p_2)^3}
\]

- If $\lambda = 0.3, p_1 = 0.3, p_2 = 0.6$:

\[
P(y = H \mid x = \langle HHH \rangle) = 0.0508
\]

\[
P(y = H \mid x = \langle TTT \rangle) = 0.6967
\]
The Three Coins Example

- After filling in hidden variables for each example, partially observed data might look like:

\[
\begin{align*}
(\langle HHH \rangle, H) &\quad P(y = H \mid HHH) = 0.0508 \\
(\langle HHH \rangle, T) &\quad P(y = T \mid HHH) = 0.9492 \\
(\langle TTT \rangle, H) &\quad P(y = H \mid TTT) = 0.6967 \\
(\langle TTT \rangle, T) &\quad P(y = T \mid TTT) = 0.3033 \\
(\langle HHH \rangle, H) &\quad P(y = H \mid HHH) = 0.0508 \\
(\langle HHH \rangle, T) &\quad P(y = T \mid HHH) = 0.9492 \\
(\langle TTT \rangle, H) &\quad P(y = H \mid TTT) = 0.6967 \\
(\langle TTT \rangle, T) &\quad P(y = T \mid TTT) = 0.3033 \\
(\langle HHH \rangle, H) &\quad P(y = H \mid HHH) = 0.0508 \\
(\langle HHH \rangle, T) &\quad P(y = T \mid HHH) = 0.9492
\end{align*}
\]
The Three Coins Example

• New Estimates:

\[
\begin{align*}
(\langle HHH \rangle, H) & \quad P(y = H \mid HHH) = 0.0508 \\
(\langle HHH \rangle, T) & \quad P(y = T \mid HHH) = 0.9492 \\
(\langle TTT \rangle, H) & \quad P(y = H \mid TTT) = 0.6967 \\
(\langle TTT \rangle, T) & \quad P(y = T \mid TTT) = 0.3033
\end{align*}
\]

\[\lambda = \frac{3 \times 0.0508 + 2 \times 0.6967}{5} = 0.3092\]

\[p_1 = \frac{3 \times 3 \times 0.0508 + 0 \times 2 \times 0.6967}{3 \times 3 \times 0.0508 + 3 \times 2 \times 0.6967} = 0.0987\]

\[p_2 = \frac{3 \times 3 \times 0.9492 + 0 \times 2 \times 0.3033}{3 \times 3 \times 0.9492 + 3 \times 2 \times 0.3033} = 0.8244\]
The Three Coins Example: Summary

- Begin with parameters $\lambda = 0.3, p_1 = 0.3, p_2 = 0.6$

- Fill in hidden variables, using

  \[ P(y = H \mid x = \langle HHH \rangle) = 0.0508 \]

  \[ P(y = H \mid x = \langle TTT \rangle) = 0.6967 \]

- Re-estimate parameters to be $\lambda = 0.3092, p_1 = 0.0987, p_2 = 0.8244$
<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\lambda$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$\tilde{p}_1$</th>
<th>$\tilde{p}_2$</th>
<th>$\tilde{p}_3$</th>
<th>$\tilde{p}_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.6000</td>
<td>0.0508</td>
<td>0.6967</td>
<td>0.0508</td>
<td>0.6967</td>
</tr>
<tr>
<td>1</td>
<td>0.3738</td>
<td>0.6800</td>
<td>0.7578</td>
<td>0.0004</td>
<td>0.9714</td>
<td>0.0004</td>
<td>0.9714</td>
</tr>
<tr>
<td>2</td>
<td>0.4859</td>
<td>0.0004</td>
<td>0.9722</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>0.5000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

The coin example for $y = \{\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle\}$. The solution that EM reaches is intuitively correct: the coin-tosser has two coins, one which always shows up heads, the other which always shows tails, and is picking between them with equal probability ($\lambda = 0.5$). The posterior probabilities $\tilde{p}_i$ show that we are certain that coin 1 (tail-biased) generated $y_2$ and $y_4$, whereas coin 2 generated $y_1$ and $y_3$. 
<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\lambda$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$\tilde{p}_1$</th>
<th>$\tilde{p}_2$</th>
<th>$\tilde{p}_3$</th>
<th>$\tilde{p}_4$</th>
<th>$\tilde{p}_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.6000</td>
<td>0.0508</td>
<td>0.6967</td>
<td>0.0508</td>
<td>0.6967</td>
<td>0.0508</td>
</tr>
<tr>
<td>1</td>
<td>0.3092</td>
<td>0.0987</td>
<td>0.8244</td>
<td>0.0008</td>
<td>0.9837</td>
<td>0.0008</td>
<td>0.9837</td>
<td>0.0008</td>
</tr>
<tr>
<td>2</td>
<td>0.3940</td>
<td>0.0012</td>
<td>0.9893</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>0.4000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

The coin example for $\{\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle\}$. $\lambda$ is now 0.4, indicating that the coin-tosser has probability 0.4 of selecting the tail-biased coin.
<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\lambda$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$\tilde{p}_1$</th>
<th>$\tilde{p}_2$</th>
<th>$\tilde{p}_3$</th>
<th>$\tilde{p}_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.6000</td>
<td>0.1579</td>
<td>0.6967</td>
<td>0.0508</td>
<td>0.6967</td>
</tr>
<tr>
<td>1</td>
<td>0.4005</td>
<td>0.0974</td>
<td>0.6300</td>
<td>0.0375</td>
<td>0.9065</td>
<td>0.0025</td>
<td>0.9065</td>
</tr>
<tr>
<td>2</td>
<td>0.4632</td>
<td>0.0148</td>
<td>0.7635</td>
<td>0.0014</td>
<td>0.9842</td>
<td>0.0000</td>
<td>0.9842</td>
</tr>
<tr>
<td>3</td>
<td>0.4924</td>
<td>0.0005</td>
<td>0.8205</td>
<td>0.0000</td>
<td>0.9941</td>
<td>0.0000</td>
<td>0.9941</td>
</tr>
<tr>
<td>4</td>
<td>0.4970</td>
<td>0.0000</td>
<td>0.8284</td>
<td>0.0000</td>
<td>0.9949</td>
<td>0.0000</td>
<td>0.9949</td>
</tr>
</tbody>
</table>

The coin example for $y = \{\langle HHT \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle\}$. EM selects a tails-only coin, and a coin which is heavily heads-biased ($p_2 = 0.8284$). It’s certain that $y_1$ and $y_3$ were generated by coin 2, as they contain heads. $y_2$ and $y_4$ could have been generated by either coin, but coin 1 is far more likely.
<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\lambda$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$\tilde{p}_1$</th>
<th>$\tilde{p}_2$</th>
<th>$\tilde{p}_3$</th>
<th>$\tilde{p}_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3000</td>
<td>0.7000</td>
<td>0.7000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
<tr>
<td>1</td>
<td>0.3000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
<tr>
<td>2</td>
<td>0.3000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
<tr>
<td>3</td>
<td>0.3000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
<tr>
<td>4</td>
<td>0.3000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
<tr>
<td>5</td>
<td>0.3000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
<tr>
<td>6</td>
<td>0.3000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
</tbody>
</table>

The coin example for $y = \{\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle\}$, with $p_1$ and $p_2$ initialised to the same value. EM is stuck at a saddle point.
<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\lambda$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$\tilde{p}_1$</th>
<th>$\tilde{p}_2$</th>
<th>$\tilde{p}_3$</th>
<th>$\tilde{p}_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3000</td>
<td>0.7001</td>
<td>0.7000</td>
<td>0.3001</td>
<td>0.2998</td>
<td>0.3001</td>
<td>0.2998</td>
</tr>
<tr>
<td>1</td>
<td>0.2999</td>
<td>0.5003</td>
<td>0.4999</td>
<td>0.3004</td>
<td>0.2995</td>
<td>0.3004</td>
<td>0.2995</td>
</tr>
<tr>
<td>2</td>
<td>0.2999</td>
<td>0.5008</td>
<td>0.4997</td>
<td>0.3013</td>
<td>0.2986</td>
<td>0.3013</td>
<td>0.2986</td>
</tr>
<tr>
<td>3</td>
<td>0.2999</td>
<td>0.5023</td>
<td>0.4990</td>
<td>0.3040</td>
<td>0.2959</td>
<td>0.3040</td>
<td>0.2959</td>
</tr>
<tr>
<td>4</td>
<td>0.3000</td>
<td>0.5068</td>
<td>0.4971</td>
<td>0.3122</td>
<td>0.2879</td>
<td>0.3122</td>
<td>0.2879</td>
</tr>
<tr>
<td>5</td>
<td>0.3000</td>
<td>0.5202</td>
<td>0.4913</td>
<td>0.3373</td>
<td>0.2645</td>
<td>0.3373</td>
<td>0.2645</td>
</tr>
<tr>
<td>6</td>
<td>0.3009</td>
<td>0.5605</td>
<td>0.4740</td>
<td>0.4157</td>
<td>0.2007</td>
<td>0.4157</td>
<td>0.2007</td>
</tr>
<tr>
<td>7</td>
<td>0.3082</td>
<td>0.6744</td>
<td>0.4223</td>
<td>0.6447</td>
<td>0.0739</td>
<td>0.6447</td>
<td>0.0739</td>
</tr>
<tr>
<td>8</td>
<td>0.3593</td>
<td>0.8972</td>
<td>0.2773</td>
<td>0.9500</td>
<td>0.0016</td>
<td>0.9500</td>
<td>0.0016</td>
</tr>
<tr>
<td>9</td>
<td>0.4758</td>
<td>0.9983</td>
<td>0.0477</td>
<td>0.9999</td>
<td>0.0000</td>
<td>0.9999</td>
<td>0.0000</td>
</tr>
<tr>
<td>10</td>
<td>0.4999</td>
<td>1.0000</td>
<td>0.0001</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>11</td>
<td>0.5000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

The coin example for $\mathbf{y} = \{\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle\}$. If we initialise $p_1$ and $p_2$ to be a small amount away from the saddle point $p_1 = p_2$, the algorithm diverges from the saddle point and eventually reaches the global maximum.
The coin example for $y = \{\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle \}$. If we initialise $p_1$ and $p_2$ to be a small amount away from the saddle point $p_1 = p_2$, the algorithm diverges from the saddle point and eventually reaches the global maximum.
The EM Algorithm

- $\Theta^t$ is the parameter vector at $t$’th iteration
- Choose $\Theta^0$ (at random, or using various heuristics)
- Iterative procedure is defined as

$$\Theta^t = \arg\max_{\Theta} Q(\Theta, \Theta^{t-1})$$

where

$$Q(\Theta, \Theta^{t-1}) = \sum_i \sum_{y \in \mathcal{Y}} P(y \mid x_i, \Theta^{t-1}) \log P(x_i, y \mid \Theta)$$
The EM Algorithm

- Iterative procedure is defined as \( \Theta^t = \arg\max_{\Theta} Q(\Theta, \Theta^{t-1}) \), where

\[
Q(\Theta, \Theta^{t-1}) = \sum_i \sum_{y \in \mathcal{Y}} P(y \mid x_i, \Theta^{t-1}) \log P(x_i, y \mid \Theta)
\]

- Key points:
  - Intuition: fill in hidden variables \( y \) according to \( P(y \mid x_i, \Theta) \)
  - EM is guaranteed to converge to a local maximum, or saddle-point, of the likelihood function
  - In general, if
    \[
    \arg\max_{\Theta} \sum_i \log P(x_i, y_i \mid \Theta)
    \]
    has a simple (analytic) solution, then
    \[
    \arg\max_{\Theta} \sum_i \sum_y P(y \mid x_i, \Theta) \log P(x_i, y \mid \Theta)
    \]
    also has a simple (analytic) solution.
Overview

- The EM algorithm in general form
- The EM algorithm for hidden Markov models (brute force)
- The EM algorithm for hidden Markov models (dynamic programming)
The Structure of Hidden Markov Models

• Have $N$ states, states $1 \ldots N$

• Without loss of generality, take $N$ to be the final or stop state

• Have an alphabet $K$. For example $K = \{a, b\}$

• Parameter $\pi_i$ for $i = 1 \ldots N$ is probability of starting in state $i$

• Parameter $a_{i,j}$ for $i = 1 \ldots (N - 1)$, and $j = 1 \ldots N$ is probability of state $j$ following state $i$

• Parameter $b_i(o)$ for $i = 1 \ldots (N - 1)$, and $o \in K$ is probability of state $i$ emitting symbol $o$
An Example

• Take $N = 3$ states. States are $\{1, 2, 3\}$. Final state is state 3.

• Alphabet $K = \{the, dog\}$.

• Distribution over initial state is $\pi_1 = 1.0$, $\pi_2 = 0$, $\pi_3 = 0$.

• Parameters $a_{i,j}$ are

<table>
<thead>
<tr>
<th></th>
<th>j=1</th>
<th>j=2</th>
<th>j=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=1</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>i=2</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

• Parameters $b_i(o)$ are

<table>
<thead>
<tr>
<th></th>
<th>o=the</th>
<th>o=dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=1</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>i=2</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>
A Generative Process

• Pick the start state $s_1$ to be state $i$ for $i = 1 \ldots N$ with probability $\pi_i$.

• Set $t = 1$

• Repeat while current state $s_t$ is not the stop state ($N$):
  - Emit a symbol $o_t \in K$ with probability $b_{s_t}(o_t)$
  - Pick the next state $s_{t+1}$ as state $j$ with probability $a_{s_t,j}$.
  - $t = t + 1$
Probabilities Over Sequences

- An **output sequence** is a sequence of observations \( o_1 \ldots o_T \) where each \( o_i \in K \)
  
e.g. the dog the dog dog the

- A **state sequence** is a sequence of states \( s_1 \ldots s_T \) where each \( s_i \in \{1 \ldots N\} \)
  
e.g. 1 2 1 2 2 1

- HMM defines a probability for each state/output sequence pair

  e.g. the/1 dog/2 the/1 dog/2 the/2 dog/1 has probability

\[
\pi_1 \ b_1(\text{the}) \ a_{1,2} \ b_2(\text{dog}) \ a_{2,1} \ b_1(\text{the}) \ a_{1,2} \ b_2(\text{dog}) \ a_{2,2} \ b_2(\text{the}) \ a_{2,1} \ b_1(\text{dog})a_{1,3}
\]

Formally:

\[
P(s_1 \ldots s_T, o_1 \ldots o_T) = \pi_{s_1} \times \left( \prod_{i=2}^{T} P(s_i \mid s_{i-1}) \right) \times \left( \prod_{i=1}^{T} P(o_i \mid s_i) \right) \times P(N \mid s_T)
\]
A Hidden Variable Problem

- We have an HMM with $N = 3$, $K = \{e, f, g, h\}$

- We see the following output sequences in training data

  
  
  e  g
  e  h
  f  h
  f  g

- How would you choose the parameter values for $\pi_i$, $a_{i,j}$, and $b_i(o)$?
Another Hidden Variable Problem

- We have an HMM with $N = 3$, $K = \{e, f, g, h\}$

- We see the following output sequences in training data:
  
  - $e \ g \ h$
  - $e \ h$
  - $f \ h \ g$
  - $f \ g \ g$
  - $e \ h$

- How would you choose the parameter values for $\pi_i$, $a_{i,j}$, and $b_i(o)$?
A Reminder: Models with Hidden Variables

• Now say we have two sets \( \mathcal{X} \) and \( \mathcal{Y} \), and a joint distribution \( P(x, y \mid \Theta) \)

• If we had **fully observed data**, \((x_i, y_i)\) pairs, then

\[
L(\Theta) = \sum_i \log P(x_i, y_i \mid \Theta)
\]

• If we have **partially observed data**, \(x_i\) examples, then

\[
L(\Theta) = \sum_i \log P(x_i \mid \Theta)
= \sum_i \log \sum_{y \in \mathcal{Y}} P(x_i, y \mid \Theta)
\]
Hidden Markov Models as a Hidden Variable Problem

- We have two sets $\mathcal{X}$ and $\mathcal{Y}$, and a joint distribution $P(x, y \mid \Theta)$

- In Hidden Markov Models:
  each $x \in \mathcal{X}$ is an output sequence $o_1 \ldots o_T$
  each $y \in \mathcal{Y}$ is a state sequence $s_1 \ldots s_T$
Maximum Likelihood Estimates

• We have an HMM with \( N = 3, K = \{e, f, g, h\} \)

We see the following **paired sequences** in training data:

\[
\begin{align*}
e/1 & \quad g/2 \\
e/1 & \quad h/2 \\
f/1 & \quad h/2 \\
f/1 & \quad g/2
\end{align*}
\]

• Maximum likelihood estimates:

\[
\begin{align*}
\pi_1 &= 1.0, & \pi_2 &= 0.0, & \pi_3 &= 0.0 \\

\begin{array}{c|ccc}
\text{for parameters } a_{i,j}: & j=1 & j=2 & j=3 \\
i=1 & 0 & 1 & 0 \\
i=2 & 0 & 0 & 1
\end{array}
\]

\[
\begin{array}{c|cccc}
\text{for parameters } b_i(o): & o=e & o=f & o=g & o=h \\
i=1 & 0.5 & 0.5 & 0 & 0 \\
i=2 & 0 & 0 & 0.5 & 0.5
\end{array}
\]
The Likelihood Function for HMMs: Fully Observed Data

- Say \((x, y) = \{o_1 \ldots o_T, s_1 \ldots s_T\}\), and

\[
\begin{align*}
  f(i, j, x, y) &= \text{Number of times state } j \text{ follows state } i \text{ in } (x, y) \\
  f(i, x, y) &= \text{Number of times state } i \text{ is the initial state in } (x, y) \text{ (1 or 0)} \\
  f(i, o, x, y) &= \text{Number of times state } i \text{ is paired with observation } o
\end{align*}
\]

- Then

\[
P(x, y) = \prod_{i \in \{1 \ldots N-1\}} \pi_i \prod_{i \in \{1 \ldots N-1\}} \prod_{j \in \{1 \ldots N\}} a_{i,j} \prod_{i \in \{1 \ldots N-1\}, \ o \in K} b_i(o) f(i, o, x, y)
\]
The Likelihood Function for HMMs: Fully Observed Data

- If we have training examples \((x_l, y_l)\) for \(l = 1 \ldots m\),

\[
L(\Theta) = \sum_{l=1}^{m} \log P(x_l, y_l)
\]

\[
= \sum_{l=1}^{m} \left( \sum_{i \in \{1 \ldots N-1\}} f(i, x_l, y_l) \log \pi_i + \sum_{i \in \{1 \ldots N-1\}, \ j \in \{1 \ldots N\}} f(i, j, x_l, y_l) \log a_{i,j} + \sum_{i \in \{1 \ldots N-1\}, \ o \in K} f(i, o, x_l, y_l) \log b_i(o) \right)
\]
• Maximizing this function gives maximum-likelihood estimates:

\[
\pi_i = \frac{\sum_{l} f(i, x_l, y_l)}{\sum_{l} \sum_{k} f(k, x_l, y_l)}
\]

\[
a_{i,j} = \frac{\sum_{l} f(i, j, x_l, y_l)}{\sum_{l} \sum_{k} f(i, k, x_l, y_l)}
\]

\[
b_i(o) = \frac{\sum_{l} f(i, o, x_l, y_l)}{\sum_{l} \sum_{o' \in K} f(i, o', x_l, y_l)}
\]
The Likelihood Function for HMMs: Partially Observed Data

- If we have training examples \((x_l)\) for \(l = 1 \ldots m\),

\[
L(\Theta) = \sum_{l=1}^{m} \log \sum_{y} P(x_l, y)
\]

\[
Q(\Theta, \Theta^{t-1}) = \sum_{l=1}^{m} \sum_{y} P(y \mid x_l, \Theta^{t-1}) \log P(x_l, y \mid \Theta)
\]
\[Q(\Theta, \Theta^{t-1}) = \sum_{l=1}^{m} \sum_{y} P(y \mid x_l, \Theta^{t-1}) \left( \sum_{i \in \{1 \ldots N-1\}} f(i, x_l, y) \log \pi_i + \right. \]

\[
\sum_{i \in \{1 \ldots N-1\}, \quad j \in \{1 \ldots N\}} \left. f(i, j, x_l, y) \log a_{i,j} + \sum_{i \in \{1 \ldots N-1\}, \quad o \in K} f(i, o, x_l, y) \log b_i(o) \right) \]

\[= \sum_{l=1}^{m} \left( \sum_{i \in \{1 \ldots N-1\}} g(i, x_l) \log \pi_i + \sum_{i \in \{1 \ldots N-1\}, \quad j \in \{1 \ldots N\}} g(i, j, x_l) \log a_{i,j} + \sum_{i \in \{1 \ldots N-1\}, \quad o \in K} g(i, o, x_l) \log b_i(o) \right) \]

where each \(g\) is an expected count:

\[g(i, x_l) = \sum_{y} P(y \mid x_l, \Theta^{t-1}) f(i, x_l, y)\]

\[g(i, j, x_l) = \sum_{y} P(y \mid x_l, \Theta^{t-1}) f(i, j, x_l, y)\]

\[g(i, o, x_l) = \sum_{y} P(y \mid x_l, \Theta^{t-1}) f(i, o, x_l, y)\]
• Maximizing this function gives EM updates:

\[ \pi_i = \frac{\sum_l g(i, x_l)}{\sum_l \sum_k g(k, x_l)} \quad a_{i,j} = \frac{\sum_l g(i, j, x_l)}{\sum_l \sum_k g(i, k, x_l)} \quad b_i(o) = \frac{\sum_l g(i, o, x_l)}{\sum_l \sum_{o' \in K} g(i, o', x_l)} \]

• Compare this to maximum likelihood estimates in fully observed case:

\[ \pi_i = \frac{\sum_l f(i, x_l, y_l)}{\sum_l \sum_k f(k, x_l, y_l)} \quad a_{i,j} = \frac{\sum_l f(i, j, x_l, y_l)}{\sum_l \sum_k f(i, k, x_l, y_l)} \quad b_i(o) = \frac{\sum_l f(i, o, x_l, y_l)}{\sum_l \sum_{o' \in K} f(i, o', x_l, y_l)} \]
A Hidden Variable Problem

- We have an HMM with $N = 3$, $K = \{e, f, g, h\}$

- We see the following output sequences in training data:

  - e g
  - e h
  - f h
  - f g

- How would you choose the parameter values for $\pi_i$, $a_{i,j}$, and $b_i(o)$?
• Four possible state sequences for the first example:

  e/1  g/1  
  e/1  g/2  
  e/2  g/1  
  e/2  g/2
• Four possible state sequences for the first example:

- e/1  g/1
- e/1  g/2
- e/2  g/1
- e/2  g/2

• Each state sequence has a different probability:

- e/1  g/1  $\pi_1 a_{1,1} a_{1,3} b_1(e) b_1(g)$
- e/1  g/2  $\pi_1 a_{1,2} a_{2,3} b_1(e) b_2(g)$
- e/2  g/1  $\pi_2 a_{2,1} a_{1,3} b_2(e) b_1(g)$
- e/2  g/2  $\pi_2 a_{2,2} a_{2,3} b_2(e) b_2(g)$
A Hidden Variable Problem

- Say we have initial parameter values:

\[ \pi_1 = 0.35, \quad \pi_2 = 0.3, \quad \pi_3 = 0.35 \]

<table>
<thead>
<tr>
<th>(a_{i,j})</th>
<th>j=1</th>
<th>j=2</th>
<th>j=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>i=2</td>
<td>0.3</td>
<td>0.2</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b_i(o))</th>
<th>o=e</th>
<th>o=f</th>
<th>o=g</th>
<th>o=h</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=1</td>
<td>0.2</td>
<td>0.25</td>
<td>0.3</td>
<td>0.25</td>
</tr>
<tr>
<td>i=2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
</tr>
</tbody>
</table>

- Each state sequence has a different probability:

\[
\pi_1 a_{1,1} a_{1,3} b_1(e) b_1(g) = 0.0021 \\
\pi_1 a_{1,2} a_{2,3} b_1(e) b_2(g) = 0.00315 \\
\pi_2 a_{2,1} a_{1,3} b_2(e) b_1(g) = 0.00135 \\
\pi_2 a_{2,2} a_{2,3} b_2(e) b_2(g) = 0.0009
\]
A Hidden Variable Problem

• Each state sequence has a different probability:

\[
\begin{align*}
\text{e/1 g/1} & \quad \pi_1 a_{1,1} a_{1,3} b_1(e) b_1(g) = 0.0021 \\
\text{e/1 g/2} & \quad \pi_1 a_{1,2} a_{2,3} b_1(e) b_2(g) = 0.00315 \\
\text{e/2 g/1} & \quad \pi_2 a_{2,1} a_{1,3} b_2(e) b_1(g) = 0.00135 \\
\text{e/2 g/2} & \quad \pi_2 a_{2,2} a_{2,3} b_2(e) b_2(g) = 0.0009
\end{align*}
\]

• Each state sequence has a different conditional probability, e.g.:

\[
P(1 1 \mid e \ g, \Theta) = \frac{0.0021}{0.0021 + 0.00315 + 0.00135 + 0.0009} = 0.28
\]

\[
\begin{align*}
\text{e/1 g/1} & \quad P(1 1 \mid e \ g, \Theta) = 0.28 \\
\text{e/1 g/2} & \quad P(1 2 \mid e \ g, \Theta) = 0.42 \\
\text{e/2 g/1} & \quad P(2 1 \mid e \ g, \Theta) = 0.18 \\
\text{e/2 g/2} & \quad P(2 2 \mid e \ g, \Theta) = 0.12
\end{align*}
\]
fill in hidden values for (e g), (e h), (f h), (f g)

\begin{align*}
\text{e/1} & \quad \text{g/1} & P(1 \ 1 \mid e \ g, \Theta) &= 0.28 \\
\text{e/1} & \quad \text{g/2} & P(1 \ 2 \mid e \ g, \Theta) &= 0.42 \\
\text{e/2} & \quad \text{g/1} & P(2 \ 1 \mid e \ g, \Theta) &= 0.18 \\
\text{e/2} & \quad \text{g/2} & P(2 \ 2 \mid e \ g, \Theta) &= 0.12 \\
\text{e/1} & \quad \text{h/1} & P(1 \ 1 \mid e \ h, \Theta) &= 0.211 \\
\text{e/1} & \quad \text{h/2} & P(1 \ 2 \mid e \ h, \Theta) &= 0.508 \\
\text{e/2} & \quad \text{h/1} & P(2 \ 1 \mid e \ h, \Theta) &= 0.136 \\
\text{e/2} & \quad \text{h/2} & P(2 \ 2 \mid e \ h, \Theta) &= 0.145 \\
\text{f/1} & \quad \text{h/1} & P(1 \ 1 \mid f \ h, \Theta) &= 0.181 \\
\text{f/1} & \quad \text{h/2} & P(1 \ 2 \mid f \ h, \Theta) &= 0.434 \\
\text{f/2} & \quad \text{h/1} & P(2 \ 1 \mid f \ h, \Theta) &= 0.186 \\
\text{f/2} & \quad \text{h/2} & P(2 \ 2 \mid f \ h, \Theta) &= 0.198 \\
\text{f/1} & \quad \text{g/1} & P(1 \ 1 \mid f \ g, \Theta) &= 0.237 \\
\text{f/1} & \quad \text{g/2} & P(1 \ 2 \mid f \ g, \Theta) &= 0.356 \\
\text{f/2} & \quad \text{g/1} & P(2 \ 1 \mid f \ g, \Theta) &= 0.244 \\
\text{f/2} & \quad \text{g/2} & P(2 \ 2 \mid f \ g, \Theta) &= 0.162
\end{align*}
Calculate the expected counts:

\[ \sum_{l} g(1, x_l) = 0.28 + 0.42 + 0.211 + 0.508 + 0.181 + 0.434 + 0.237 + 0.356 = 2.628 \]

\[ \sum_{l} g(2, x_l) = 1.372 \]

\[ \sum_{l} g(3, x_l) = 0.0 \]

\[ \sum_{l} g(1, 1, x_l) = 0.28 + 0.211 + 0.181 + 0.237 = 0.910 \]

\[ \sum_{l} g(1, 2, x_l) = 1.72 \]

\[ \sum_{l} g(2, 1, x_l) = 0.746 \]

\[ \sum_{l} g(2, 2, x_l) = 0.626 \]

\[ \sum_{l} g(1, 3, x_l) = 1.656 \]

\[ \sum_{l} g(2, 3, x_l) = 2.344 \]
Calculate the expected counts:

\[
\sum_{l} g(1, e, x_l) = 0.28 + 0.42 + 0.211 + 0.508 = 1.4
\]

\[
\sum_{l} g(1, f, x_l) = 1.209
\]

\[
\sum_{l} g(1, g, x_l) = 0.941
\]

\[
\sum_{l} g(1, h, x_l) = 0.827
\]

\[
\sum_{l} g(2, e, x_l) = 0.6
\]

\[
\sum_{l} g(2, f, x_l) = 0.385
\]

\[
\sum_{l} g(2, g, x_l) = 1.465
\]

\[
\sum_{l} g(2, h, x_l) = 1.173
\]
Calculate the new estimates:

\[
\pi_1 = \frac{\sum_l g(1, x_l)}{\sum_l g(1, x_l) + \sum_l g(2, x_l) + \sum_l g(3, x_l)} = \frac{2.628}{2.628 + 1.372 + 0} = 0.657
\]

\[\pi_2 = 0.343 \quad \pi_3 = 0\]

\[
a_{1,1} = \frac{\sum_l g(1, 1, x_l)}{\sum_l g(1, 1, x_l) + \sum_l g(1, 2, x_l) + \sum_l g(1, 3, x_l)} = \frac{0.91}{0.91 + 1.72 + 1.656} = 0.212
\]

<table>
<thead>
<tr>
<th>(a_{i,j})</th>
<th>(j=1)</th>
<th>(j=2)</th>
<th>(j=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i=1)</td>
<td>0.212</td>
<td>0.401</td>
<td>0.387</td>
</tr>
<tr>
<td>(i=2)</td>
<td>0.201</td>
<td>0.169</td>
<td>0.631</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b_i(o))</th>
<th>(o=e)</th>
<th>(o=f)</th>
<th>(o=g)</th>
<th>(o=h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i=1)</td>
<td>0.320</td>
<td>0.276</td>
<td>0.215</td>
<td>0.189</td>
</tr>
<tr>
<td>(i=2)</td>
<td>0.166</td>
<td>0.106</td>
<td>0.404</td>
<td>0.324</td>
</tr>
</tbody>
</table>
Iterate this 3 times:

\[
\pi_1 = 0.9986, \quad \pi_2 = 0.00138 \quad \pi_3 = 0
\]

<table>
<thead>
<tr>
<th>(a_{i,j})</th>
<th>j=1</th>
<th>j=2</th>
<th>j=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=1</td>
<td>0.0054</td>
<td>0.9896</td>
<td>0.00543</td>
</tr>
<tr>
<td>i=2</td>
<td>0.0</td>
<td>0.0013627</td>
<td>0.9986</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b_i(o))</th>
<th>o=e</th>
<th>o=f</th>
<th>o=g</th>
<th>o=h</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=1</td>
<td>0.497</td>
<td>0.497</td>
<td>0.00258</td>
<td>0.00272</td>
</tr>
<tr>
<td>i=2</td>
<td>0.001</td>
<td>0.000189</td>
<td>0.4996</td>
<td>0.4992</td>
</tr>
</tbody>
</table>
Overview

- The EM algorithm in general form

- The EM algorithm for hidden markov models (brute force)

- The EM algorithm for hidden markov models (dynamic programming)
The Forward-Backward or Baum-Welch Algorithm

- Aim is to (efficiently!) calculate the expected counts:

\[ g(i, x_l) = \sum_y P(y \mid x_l, \Theta^{t-1}) f(i, x_l, y) \]

\[ g(i, j, x_l) = \sum_y P(y \mid x_l, \Theta^{t-1}) f(i, j, x_l, y) \]

\[ g(i, o, x_l) = \sum_y P(y \mid x_l, \Theta^{t-1}) f(i, o, x_l, y) \]
The Forward-Backward or Baum-Welch Algorithm

- Suppose we could calculate the following quantities, given an input sequence \( o_1 \ldots o_T \):

\[
\alpha_i(t) = P(o_1 \ldots o_{t-1}, s_t = i \mid \Theta) \quad \text{forward probabilities}
\]

\[
\beta_i(t) = P(o_t \ldots o_T \mid s_t = i, \Theta) \quad \text{backward probabilities}
\]

- The probability of being in state \( i \) at time \( t \), is

\[
p_t(i) = P(s_t = i \mid o_1 \ldots o_T, \Theta)
\]

\[
= \frac{P(s_t = i, o_1 \ldots o_T \mid \Theta)}{P(o_1 \ldots o_T \mid \Theta)}
\]

\[
= \frac{\alpha_t(i)\beta_t(i)}{P(o_1 \ldots o_T \mid \Theta)}
\]

also,

\[
P(o_1 \ldots o_T \mid \Theta) = \sum_i \alpha_t(i)\beta_t(i) \text{ for any } t
\]
Expected Initial Counts

- As before,

\[ g(i, o_1 \ldots o_T) = \text{expected number of times state } i \text{ is state 1} \]

- We can calculate this as

\[ g(i, o_1 \ldots o_T) = p_1(i) \]
Expected Emission Counts

* As before,

\[ g(i, o, o_1 \ldots o_T) = \text{expected number of times state } i \text{ emits the symbol } o \]

* We can calculate this as

\[
g(i, o, o_1 \ldots o_T) = \sum_{t: o_t = o} p_t(i)
\]
The Forward-Backward or Baum-Welch Algorithm

- Suppose we could calculate the following quantities, given an input sequence \( o_1 \ldots o_T \):

\[
\alpha_i(t) = P(o_1 \ldots o_{t-1}, s_t = i \mid \Theta) \quad \text{forward probabilities}
\]

\[
\beta_i(t) = P(o_t \ldots o_T \mid s_t = i, \Theta) \quad \text{backward probabilities}
\]

- The probability of being in state \( i \) at time \( t \), and in state \( j \) at time \( t + 1 \), is

\[
p_{t}(i, j) = P(s_t = i, s_{t+1} = j \mid o_1 \ldots o_T, \Theta)
\]

\[
= \frac{P(s_t = i, s_{t+1} = j, o_1 \ldots o_T \mid \Theta)}{P(o_1 \ldots o_T \mid \Theta)}
\]

\[
= \frac{\alpha_t(i) a_{i,j} b_i(o_t) \beta_{t+1}(j)}{P(o_1 \ldots o_T \mid \Theta)}
\]

also,

\[
P(o_1 \ldots o_T \mid \Theta) = \sum_i \alpha_t(i) \beta_t(i) \text{ for any } t
\]
Expected Transition Counts

- As before,

\[ g(i, j, o_1 \ldots o_T) = \text{expected number of times state } j \text{ follows state } i \]

- We can calculate this as

\[ g(i, j, o_1 \ldots o_T) = \sum_{t} p_t(i, j) \]
Recursive Definitions for Forward Probabilities

- Given an input sequence $o_1 \ldots o_T$:
  \[
  \alpha_i(t) = P(o_1 \ldots o_{t-1}, s_t = i \mid \Theta)
  \]  
  forward probabilities

- Base case:
  \[
  \alpha_i(1) = \pi_i \quad \text{for all } i
  \]

- Recursive case:
  \[
  \alpha_j(t+1) = \sum_i \alpha_i(t) a_{i,j} b_i(o_t) \quad \text{for all } j = 1 \ldots N \text{ and } t = 2 \ldots T
  \]
Recursive Definitions for Backward Probabilities

- Given an input sequence $o_1 \ldots o_T$:

$$\beta_i(t) = P(o_t \ldots o_T \mid s_t = i, \Theta)$$

backward probabilities

- Base case:

$$\beta_i(T + 1) = 1 \quad \text{for } i = N$$

$$\beta_i(T + 1) = 0 \quad \text{for } i \neq N$$

- Recursive case:

$$\beta_i(t) = \sum_j a_{i,j} b_i(o_t) \beta_j(t+1) \quad \text{for all } j = 1 \ldots N \text{ and } t = 1 \ldots T$$
Overview

• The EM algorithm in general form
  (more about the 3 coin example)

• The EM algorithm for hidden markov models (brute force)

• The EM algorithm for hidden markov models (dynamic programming)

• Briefly: The EM algorithm for PCFGs
EM for Probabilistic Context-Free Grammars

- A PCFG defines a distribution $P(S, T \mid \Theta)$ over tree/sentence pairs $(S, T)$

- If we had tree/sentence pairs (fully observed data) then

  $$L(\Theta) = \sum_i \log P(S_i, T_i \mid \Theta)$$

- Say we have sentences only, $S_1 \ldots S_n$
  $\Rightarrow$ trees are hidden variables

  $$L(\Theta) = \sum_i \log \sum_T P(S_i, T \mid \Theta)$$
EM for Probabilistic Context-Free Grammars

- Say we have sentences only, $S_1 \ldots S_n$
  $\Rightarrow$ trees are hidden variables

$$L(\Theta) = \sum_i \log \sum_T P(S_i, T \mid \Theta)$$

- EM algorithm is then $\Theta^t = \arg\max_{\Theta} Q(\Theta, \Theta^{t-1})$, where

$$Q(\Theta, \Theta^{t-1}) = \sum_i \sum_T P(T \mid S_i, \Theta^{t-1}) \log P(S_i, T \mid \Theta)$$
• Remember:

\[
\log P(S_i, T \mid \Theta) = \sum_{r \in R} \text{Count}(S_i, T, r) \log \Theta_r
\]

where \( \text{Count}(S, T, r) \) is the number of times rule \( r \) is seen in the sentence/tree pair \((S, T)\)

\[
\Rightarrow Q(\Theta, \Theta^{t-1}) = \sum_i \sum_T P(T \mid S_i, \Theta^{t-1}) \log P(S_i, T \mid \Theta)
\]

\[
= \sum_i \sum_T P(T \mid S_i, \Theta^{t-1}) \sum_r \text{Count}(S_i, T, r) \log \Theta_r
\]

\[
= \sum_i \sum_{r \in R} \text{Count}(S_i, r) \log \Theta_r
\]

where \( \text{Count}(S_i, r) = \sum_T P(T \mid S_i, \Theta^{t-1}) \text{Count}(S_i, T, r) \)

the expected counts
• Solving $\Theta_{ML} = \arg\max_{\Theta \in \Omega} L(\Theta)$ gives

$$\Theta_{\alpha \rightarrow \beta} = \frac{\sum_i Count(S_i, \alpha \rightarrow \beta)}{\sum_i \sum_{s \in R(\alpha)} Count(S_i, s)}$$

• There are efficient algorithms for calculating

$$Count(S_i, r) = \sum_T P(T | S_i, \Theta^{t-1}) Count(S_i, T, r)$$

for a PCFG. See (Baker 1979), called “The Inside Outside Algorithm”. See also Manning and Schuetze section 11.3.4.