Announcements

• Oct 22: Midterm review
• Oct 24: Midterm (in-class)
• Nov 5: Project details + PyTorch tutorial
• Nov 11: Project proposal due
  • Start forming teams now! (2-3 members)
    • Can use Piazza
• Nov 18: Assignment 4 due date changed
In general, we can use all observations and all previous states:

\[ \hat{S} = \arg \max_S P(S \mid O) = \arg \max_S \prod_i P(s_i \mid o_n, o_{n-1}, \ldots, o_1, s_{i-1}, \ldots, s_1) \]

\[ P(s_i \mid s_{i-1}, \ldots, s_1, O) \propto \exp(w \cdot f(s_i, s_{i-1}, \ldots, s_1, O)) \]
Features in an MEMM

Figure 8.13 An MEMM for part-of-speech tagging showing the ability to condition on more features.

\[
\begin{align*}
&\langle t_i, w_{i-2} \rangle, \langle t_i, w_{i-1} \rangle, \langle t_i, w_i \rangle, \langle t_i, w_{i+1} \rangle, \langle t_i, w_{i+2} \rangle \\
&\langle t_i, t_{i-1} \rangle, \langle t_i, t_{i-2}, t_{i-1} \rangle, \\
&\langle t_i, t_{i-1}, w_i \rangle, \langle t_i, w_{i-1}, w_i \rangle, \langle t_i, w_i, w_{i+1} \rangle,
\end{align*}
\]

Feature templates

Features

\[
\begin{align*}
t_i &= \text{VB and } w_{i-2} = \text{Janet} \\
t_i &= \text{VB and } w_{i-1} = \text{will} \\
t_i &= \text{VB and } w_i = \text{back} \\
t_i &= \text{VB and } w_{i+1} = \text{the} \\
t_i &= \text{VB and } w_{i+2} = \text{bill} \\
t_i &= \text{VB and } t_{i-1} = \text{MD} \\
t_i &= \text{VB and } t_{i-1} = \text{MD and } t_{i-2} = \text{NNP} \\
t_i &= \text{VB and } w_i = \text{back and } w_{i+1} = \text{the}
\end{align*}
\]
MEMM: Learning

- Gradient descent: similar to logistic regression!

\[ P(s_i | s_1, \ldots, s_{i-1}, O) \propto \exp(w \cdot f(s_1, \ldots, s_i, O)) \]

- Given: pairs of \((S, O)\) where each \(S = \langle s_1, s_2, \ldots, s_n \rangle\)

Loss for one sequence, \(L = - \sum_i \log P(s_i | s_1, \ldots, s_{i-1}, O)\)

- Compute gradients with respect to weights \(w\) and update
EM: Some intuition

• Let’s say I have 3 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:

  • 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 the results of the coin 0 toss, or whether coin 1 or coin 2 was tossed, but I tell you how many heads/tails are seen after each trial

• You see the following sequence:
  $\langle H, H, H \rangle, \langle T, T, T \rangle, \langle H, H, H \rangle, \langle T, T, T \rangle, \langle H, H, H \rangle$

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

- Data points \(x_1, x_2, \ldots, x_n\) from (finite or countable) set \(\mathcal{X}\)
- Parameter vector \(\theta\)
- Parameter space \(\Omega\)
- We have a distribution \(P(x \mid \theta)\) for any \(\theta \in \Omega\), such that
  \[
  \sum_{x \in \mathcal{X}} P(x \mid \theta) = 1 \quad \text{and} \quad P(x \mid \theta) \geq 0 \quad \forall x
  \]
- Assume data points are drawn independently and identically distributed from a distribution \(P(x \mid \theta^*)\) for some \(\theta^* \in \Omega\)
Log Likelihood

- Data points $x_1, x_2, \ldots, x_n$ from (finite or countable) set $\mathcal{X}$
- Parameter vector $\theta$ and a parameter space $\Omega$
- Probability distribution $P(x \mid \theta)$ for any $\theta \in \Omega$

\[
\text{Likelihood}(\theta) = P(x_1, x_2, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} P(x_i \mid \theta)
\]

- Log-likelihood, $L(\theta) = \sum_{i=1}^{n} \log P(x_i \mid \theta)$
Example 1: 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.
  - HTHTHHHHTTT
- Parameter vector $\theta$ is a single parameter, i.e. probability of coin coming up heads
- Parameter space $\Omega = [0, 1]$
- Distribution $P(x \mid \theta) = \begin{cases} \theta & \text{if } x = H \\ 1 - \theta & \text{if } x = T \end{cases}$
Example 2: Markov chains

- $\mathcal{X}$ is the set of all possible state (e.g. tag) sequences generated by the underlying generative process. Our sample is $n$ sequences $X_1, \ldots, X_n$ such that each $X_i \in \mathcal{X}$, consists of a sequence of states.

- $\theta_T$ is the vector of all transition $(s_i \rightarrow s_j)$ parameters. Without loss of generality, assume a dummy start state $\phi$ and initial transition $\phi \rightarrow s_1$ (how many parameters?)

- Let $T(\alpha) \subset T$ be all the transitions of the form $\alpha \rightarrow \beta$

- Parameter space $\Omega$ is the set of $\theta \in [0,1]^{(|S+1||S|}$ where $S$ is set of all states (tags), such that:

$$\text{for all } \alpha \in S, \sum_{t \in T(\alpha)} \theta_t = 1$$
Example 2: Markov chains

• $\theta_T$ is the vector of all transition parameters

• We have:

$$P(X \mid \theta) = \prod_{t \in T} \theta_t^{\text{Count}(X,t)}$$

where $\text{Count}(X, t)$ is the number of times transition $t$ is seen in sequence $X$

$$\implies \log P(X \mid \theta) = \sum_{t \in T} \text{Count}(X, t) \log \theta_t$$
MLE for Markov chains

- We have
  \[ \log P(X | \theta) = \sum_{t \in T} \text{Count}(X, t) \log \theta_t \]

  where \( \text{Count}(X, t) \) is the number of times transition \( t \) is seen in sequence \( X \)

- And,
  \[ L(\theta) = \sum_i \log P(X_i | \theta) = \sum_i \sum_{t \in T} \text{Count}(X_i, t) \log \theta_t \]
MLE for Markov chains

\[ L(\theta) = \sum_i \log P(X_i | \theta) = \sum_i \sum_{t \in T} \text{Count}(X_i, t) \log \theta_t \]

Solve \( \theta_{MLE} = \arg \max_{\theta \in \Omega} L(\theta) \)

\[ \implies \text{find } \theta \text{ s. t. } \frac{\partial L(\theta)}{\partial \theta} = 0 \text{ with appropriate probability constraints} \]

This gives: \( \theta_t = \frac{\sum_i \text{Count}(X_i, t)}{\sum_i \sum_{t' \in T(\alpha)} \text{Count}(X_i, t')} \)

where \( t \) is of the form \( \alpha \rightarrow \beta \) for some \( \beta \)
Models with hidden variables

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

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

\[
L(\theta) = \sum_i \log P(x_i, y_i | \theta)
\]

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

\[
L(\theta) = \sum_i \log P(x_i | \theta)
\]

\[
= \sum_i \log \sum_{y \in \mathcal{Y}} P(x_i, y | \theta)
\]

Unsupervised Learning
Expectation Maximization

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

\[
L(\theta) = \sum_i \log \sum_{y \in \mathcal{Y}} P(x_i, y | \theta)
\]

• The EM (Expectation Maximization) algorithm is a method for finding

\[
\theta_{MLE} = \arg \max_\theta L(\theta) = \arg \max_\theta \sum_i \log \sum_{y \in \mathcal{Y}} P(x_i, y | \theta)
\]
The three coins example

• In the three coins example,
  \[ \mathcal{Y} = \{H, T\} \] (possible outcomes of coin 0)
  \[ \mathcal{X} = \{HHH, TTT, HTT, THH, HHT, TTH, HTH, THT\} \]
  \[ \theta = \{\lambda, p_1, p_2\} \]

• and \( P(x, y | \theta) = P(y | \theta) \ P(x | y, \theta) \)
  where
  \[ P(y | \theta) = \begin{cases} 
    \lambda & \text{if } y = H \\
    1 - \lambda & \text{if } y = T 
  \end{cases} \]
  and
  \[ P(x | y, \theta) = \begin{cases} 
    p_1^y (1 - p_1)^{1-y} & \text{if } y = H \\
    p_2^y (1 - p_2)^{1-y} & \text{if } y = T 
  \end{cases} \]
The three coins example

- Calculating various probabilities:

  \[ P(x = THT, y = H \mid \theta) = \lambda p_1(1 - p_1)^2 \]
  \[ P(x = THT, y = T \mid \theta) = (1 - \lambda)p_2(1 - p_2)^2 \]

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

  \[ P(y = H \mid x = THT, \theta) = \frac{P(x = THT, y = H \mid \theta)}{P(x = THT \mid \theta)} \]
  \[ = \frac{\lambda p_1(1 - p_1)^2}{\lambda p_1(1 - p_1)^2 + (1 - \lambda)p_2(1 - p_2)^2} \]
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:

\[
\lambda = \frac{3}{5}
\]
\[
p_1 = \frac{9}{9}
\]
\[
p_2 = \frac{0}{6}
\]
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 MLE 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 the current parameters are \( \lambda, p_1, p_2 \)

\[
P(y = H | x = \langle HHH \rangle) = \frac{P(\langle HHH \rangle, H)}{P(\langle HHH \rangle, H) + P(\langle HHH \rangle, T)}
\]

\[
= \frac{\lambda p_1^3}{\lambda p_1^3 + (1 - \lambda) p_2^3}
\]

\[
P(y = H | x = \langle TTT \rangle) = \frac{P(\langle TTT \rangle, H)}{P(\langle TTT \rangle, H) + P(\langle TTT \rangle, T)}
\]

\[
= \frac{\lambda (1 - p_1)^3}{\lambda (1 - p_1)^3 + (1 - \lambda)(1 - p_2)^3}
\]
The three coins example

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

\[
P(y = H | x = \langle HHH \rangle) = \frac{P(\langle HHH \rangle, H)}{P(\langle HHH \rangle, H) + P(\langle HHH \rangle, T)}
\]

\[
= \frac{\lambda p_1^3}{\lambda p_1^3 + (1 - \lambda)p_2^3}
\]

\[
P(y = H | x = \langle TTT \rangle) = \frac{P(\langle HHH \rangle, H)}{P(\langle TTT \rangle, H) + P(\langle TTT \rangle, T)}
\]

\[
= \frac{\lambda(1 - p_1)^3}{\lambda(1 - p_1)^3 + (1 - \lambda)(1 - p_2)^3}
\]

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

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

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

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

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

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

\[
p_2 = \frac{3 \times 3 \times 0.9492 + 0 \times 2 \times 0.3033}{3 \times 3 \times 0.9492 + 3 \times 2 \times 0.3033} = 0.8244
\]
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
  
  $\lambda = 0.3092, \ p_1 = 0.0987, \ p_2 = 0.8244$
EM iterations

<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>0.0680</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 $x = \{\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle\}$. The solution that EM reaches is intuitively correct: the coin tosser has two coins, one which always shows heads, and another which always shows tails, and is picking between them with equal probability ($\lambda = 0.5$).

Posterior probabilities $\tilde{p}_i$ show that we are certain that coin 1 (tail-biased) generate $x_2$ and $x_4$, whereas coin 2 generated $x_1$ and $x_3$. 
EM iterations

<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>

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 coin 0 has a probability 0.4 of selecting the tail-biased coin.
EM iterations

<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>

Coin example for $x = \{ \langle HHT \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle \}$.

EM selects a tails-only coin, and a coin which is heavily heads-biased ($p_2 = 0.8284$). It’s certain that $x_1$ and $x_3$ were generated by coin 2 since they contain heads. $x_2$ and $x_4$ could have been generated by either coin but coin 1 (tail-biased) is far more likely.
Initialization matters

Coin example for $x = \{\langle HHH \rangle, \langle TTT \rangle, \langle HHH \rangle, \langle TTT \rangle \}$. 

In this case, EM is stuck at a **saddle point**.
If we initialize $p_1$ and $p_2$ even a small amount away from the saddle point $p_1 = p_2$, EM diverges and eventually reaches the global maximum

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\lambda$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$\hat{p}_1$</th>
<th>$\hat{p}_2$</th>
<th>$\hat{p}_3$</th>
<th>$\hat{p}_4$</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>

Coin example for $x = \{ \langle HHH\rangle, \langle TTT\rangle, \langle HHH\rangle, \langle TTT\rangle \}$. 
The EM algorithm

• $\theta^t$ is the parameter vector at the $t^{th}$ iteration

• Choose $\theta^0$ at random (or using smart heuristics)

• Iterative procedure defined as:

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

where

$$Q(\theta, \theta^{t-1}) = \sum_i \sum_{y \in Y} P(y | x_i, \theta^{t-1}) \log P(x_i, y | \theta)$$
The EM algorithm

- $\theta^t$ is the parameter vector at the $t^{th}$ iteration

- Choose $\theta^0$ at random (or using smart heuristics)

- (E step): Compute expected counts

$$\overline{\text{Count}}(r) = \sum_{i=1}^{n} \sum_{y} P(y | x_i, \theta^{t-1}) \text{Count}(x_i, y, r)$$

for every parameter $\theta_r$

- e.g.

$$\overline{\text{Count}}(DT \rightarrow NN) = \sum_{i} \sum_{y} P(S | O_i, \theta^{t-1}) \text{Count}(O_i, S, \theta_{DT \rightarrow NN})$$
The EM algorithm

• \( \theta^t \) is the parameter vector at the \( t^{th} \) iteration

• Choose \( \theta^0 \) at random (or using smart heuristics)

• (E step): Compute expected counts

\[
\bar{\text{Count}}(r) = \sum_{i=1}^{n} \sum_{y} P(y | x_i, \theta^{t-1}) \, \text{Count}(x_i, y, r)
\]

for every parameter \( \theta_r \)

• (M step): Re-estimate parameters using expected counts to maximize likelihood

\[
e.g. \, \theta_{DT \rightarrow NN} = \frac{\bar{\text{Count}}(DT \rightarrow NN)}{\sum_{\beta} \bar{\text{Count}}(DT \rightarrow \beta)}
\]
The EM algorithm

- Iterative procedure defined as \( \theta^t = \arg \max \theta \ Q(\theta, \theta^{t-1}) \) where

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

- Key points:
  - Intuition: Fill in hidden variables \( y \) according to \( P(y | x_i, \theta) \)
  - EM is guaranteed to converge to a local maximum, or saddle-point, of the likelihood function
  
  In general, if \( \arg \max \theta \sum_i \log P(x_i, y_i | \theta) \) has a simple analytic solution, then

\[
\arg \max \theta \sum_i \sum_y P(y | x_i, \theta) \log P(x_i, y | \theta)
\] also has a simple solution.
Example: EM for HMM

- We observe only word sequences $X_1, X_2, \ldots, X_n$ (no tags $Y$)

- $\theta$ is the vector of all transition parameters (include initial state distribution as a special case, $\emptyset \rightarrow s$

- $\phi$ is the vector of all emission parameters

- Initialize parameters $\theta^0$ and $\phi^0$
Example: EM for HMM

- Initialize parameters $\theta^0$ and $\phi^0$

- **(E-Step)**

$$\overline{\text{Count}}(\theta_k) = \sum_{i=1}^{n} \sum_{Y} P(Y | X_i, \theta^{t-1}, \phi^{t-1}) \ \text{Count}(X_i, Y, \theta_k)$$

$$= \sum_{i=1}^{n} \sum_{Y} P(Y | X_i, \theta^{t-1}, \phi^{t-1}) \ \text{Count}(Y, \theta_k)$$

$$\overline{\text{Count}}(\phi_k) = \sum_{i=1}^{n} \sum_{Y} P(Y | X_i, \theta^{t-1}, \phi^{t-1}) \ \text{Count}(X_i, Y, \phi_k)$$
Example: EM for HMM

- Initialize parameters $\theta^0$ and $\phi^0$

- (M-Step)

$$\theta_k^t = \frac{\text{Count}(\theta_k)}{\sum_{\theta' \in M(\theta_k)} \text{Count}(\theta')}$$

where $M(\theta_k)$ is the set of all transitions $(a \rightarrow b, \text{ all } b)$ that share the same previous state as the $k^{th}$ transition $(a \rightarrow c$ for some $c$).

$$\phi_k^t = \frac{\text{Count}(\phi_k)}{\sum_{\phi' \in M'(\phi_k)} \text{Count}(\phi')}$$

where $M'(\phi_k)$ is the set of all emissions $(a \rightarrow x, \text{ all } x)$ that share the same hidden state as the $k^{th}$ emission $(a \rightarrow x', \text{ for some } x')$. 
Efficient EM?

- **(E-Step)**

\[
\text{Count}(\theta_k) = \sum_{i=1}^{n} \sum_{Y} P(Y \mid X_i, \theta^{t-1}, \phi^{t-1}) \text{ Count}(Y, \theta_k)
\]

\[
\text{Count}(\phi_k) = \sum_{i=1}^{n} \sum_{Y} P(Y \mid X_i, \theta^{t-1}, \phi^{t-1}) \text{ Count}(X_i, Y, \phi_k)
\]

*Cannot enumerate all possible Y!*
Efficient EM?

- (E-Step)

\[ \overline{\text{Count}}(\theta_{NN \rightarrow VBD}) = \sum_{i=1}^{n} \sum_{Y} P(Y | X_i, \theta^{t-1}, \phi^{t-1}) \text{ Count}(Y, \theta_k) \]

\[ = \sum_{i} \sum_{j=1}^{m} P(y_j = NN, y_{j+1} = VBD | X_i, \theta^{t-1}, \phi^{t-1}) \]

where \( m \) is the length of the sequence \( X_i \)
Efficient EM?

\[
\overline{\text{Count}}(\theta_{NN\rightarrow VBD}) = \sum_{i=1}^{n} \sum_{Y} P(Y | X_i, \theta^{t-1}, \phi^{t-1}) \overline{\text{Count}}(Y, \theta_k) \\
= \sum_{i} \sum_{j=1}^{m} P(y_j = NN, y_{j+1} = VBD | X_i, \theta^{t-1}, \phi^{t-1})
\]

where \( m \) is the length of the sequence \( X_i \).

Similarly, \( \overline{\text{Count}}(\phi_{NN\rightarrow \text{cat}}) = \sum_{i} \sum_{j:X_{ij} = \text{cat}} P(y_j = NN | X_i, \theta^{t-1}, \phi^{t-1}) \)
Forward-backward algorithm

• Define:
  \[ \alpha_s(j) = P(x_1, \ldots, x_{j-1}, y_j = s \mid \theta, \phi) \] (forward probability)
  \[ \beta_s(j) = P(x_j, \ldots, x_m \mid y_j = s, \theta, \phi) \] (backward probability)

• Observation likelihood,
  \[ Z = P(x_1, x_2, \ldots, x_m \mid \theta, \phi) = \sum_s \alpha_s(j)\beta_s(j) \text{ for any } j \in 1,\ldots,m \]

• \[ P(y_j = s \mid X, \theta, \phi) = \frac{\alpha_s(j)\beta_s(j)}{Z} \]

• \[ P(y_j = s, y_{j+1} = s' \mid X, \theta, \phi) = \frac{\alpha_s(j) \theta_{s \rightarrow s'} \phi_{s \rightarrow x_j} \beta_{s'}(j+1)}{Z} \]
Forward-backward algorithm

\[ \alpha_{NN}(2) \]

\[ \beta_{VBD}(3) \]

\[ P(y_j = s \mid X, \theta, \phi) = \frac{\alpha_s(j)\beta_s(j)}{Z} \]

\[ P(y_j = s, y_{j+1} = s' \mid X, \theta, \phi) = \frac{\alpha_s(j) \theta_{s \rightarrow s'} \phi_{s \rightarrow x_j} \beta_{s'} (j + 1)}{Z} \]
Forward-backward algorithm

- \( P(y_j = s \mid X, \theta, \phi) = \frac{\alpha_s(j)\beta_s(j)}{Z} \)

- \( P(y_j = s, y_{j+1} = s' \mid X, \theta, \phi) = \frac{\alpha_s(j) \theta_{s \rightarrow s'} \phi_{s \rightarrow x_j} \beta_{s'}(j + 1)}{Z} \)

- Given these, we can now estimate:

  \[
  \text{Count}(\theta_{s \rightarrow s'}) = \sum_i \sum_{j=1}^m P(y_j = s, y_{j+1} = s' \mid X_i, \theta, \phi)
  \]

  \[
  \text{Count}(\phi_{s \rightarrow o}) = \sum_i \sum_{j:X_{ij} = o} P(y_j = s \mid X_i, \theta, \phi)
  \]
Dynamic programming

\[
\alpha_s(j) = P(y_j = s, x_1, \ldots, x_{j-1})
\]

\[
= \sum_{s'} P(y_{j-1} = s', x_1, \ldots, x_{j-2}) P(x_{j-1} | y_{j-1} = s') P(y_j = s | y_{j-1} = s')
\]

\[
= \sum_{s'} \alpha_{s'} (j - 1) \phi_{s'\rightarrow x_{j-1}} \theta_{s'\rightarrow s}
\]

\[
\beta_s(j) = P(y_j = s, x_1, \ldots, x_j)
\]

\[
= \sum_{s'} \beta_{s'} (j + 1) \phi_{s\rightarrow x_{j+1}} \theta_{s\rightarrow s'}
\]
Dynamic programming

\[ \alpha_s(j) = P(y_j = s, x_1, \ldots, x_{j-1}) \]

\[ = \sum_{s'} P(y_{j-1} = s', x_1, \ldots, x_{j-2}) P(x_{j-1} | y_{j-1} = s') P(y_j = s | y_{j-1} = s') \]

\[ = \sum_{s'} \alpha_{s'} (j - 1) \phi_{s' \rightarrow x_{j-1}} \theta_{s' \rightarrow s} \]

- Similarly,

\[ \beta_s(j) = \phi_{s \rightarrow x_j} \sum_{s'} \beta_{s'} (j + 1) \theta_{s \rightarrow s'} \]

- Runtime: \( O(|S|^2 \cdot m) \)