EM for HMMs
a.k.a. The Baum-Welch Algorithm

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

April 11th, 2007
The general learning problem with missing data

Marginal likelihood – $x$ is observed, $z$ is missing:

\[
\ell(\theta : D) = \log \prod_{j=1}^{m} P(x_j | \theta) \\
= \sum_{j=1}^{m} \log P(x_j | \theta) \\
= \sum_{j=1}^{m} \log \sum_{z} P(x_j, z | \theta)
\]

\text{sum over hidden vars (marginalize out)}
EM is coordinate ascent

\[ \ell(\theta : \mathcal{D}) \geq F(\theta, Q) = \sum_{j=1}^{m} \sum_{z} Q(z | x_j) \log \frac{P(z, x_j | \theta)}{Q(z | x_j)} \]

- **M-step**: Fix Q, maximize F over \( \theta \) (a lower bound on \( \ell(\theta : \mathcal{D}) \)):
  \[ \ell(\theta : \mathcal{D}) \geq F(\theta, Q^{(t)}) = \sum_{j=1}^{m} \sum_{z} Q^{(t)}(z | x_j) \log P(z, x_j | \theta) + m.H(Q^{(t)}) \]

- **E-step**: Fix \( \theta \), maximize F over Q:
  \[ \ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) = \ell(\theta^{(t)} : \mathcal{D}) - m \sum_{j=1}^{m} KL\left( Q(z | x_j) || P(z | x_j, \theta^{(t)}) \right) \]

  “Realigns” F with likelihood:
  \[ F(\theta^{(t)}, Q^{(t+1)}) = \ell(\theta^{(t)} : \mathcal{D}) \]
What you should know about EM

- K-means for clustering:
  - algorithm
  - converges because it’s coordinate ascent

- EM for mixture of Gaussians:
  - How to “learn” maximum likelihood parameters (locally max. like.) in the case of unlabeled data

- Be happy with this kind of probabilistic analysis

- Remember, E.M. can get stuck in local minima, and empirically it DOES

- EM is coordinate ascent

- General case for EM
Learning HMMs from fully observable data is easy

Learn 3 distributions:

\[ P(X_1) \]

\[ P(O_i \mid X_i) \]

\[ P(X_i \mid X_{i-1}) \]
Learning HMMs from fully observable data is easy

\[ X_1 = \{a, ..., z\} \rightarrow X_2 = \{a, ..., z\} \rightarrow X_3 = \{a, ..., z\} \rightarrow X_4 = \{a, ..., z\} \rightarrow X_5 = \{a, ..., z\} \]

Learn 3 distributions:

\[ P(X_1) = \frac{\text{count (first letter was } a)}{\text{dataset size}} \]

\[ P(O_i \mid X_i) = \frac{\text{count (pixel } i \text{ was white, } X_i = a)}{\text{anything in }} \]

What if \( O \) is observed, but \( X \) is hidden
Log likelihood for HMMs when $X$ is hidden

Marginal likelihood – $O$ is observed, $X$ is missing

- For simplicity of notation, training data consists of only one sequence:

$$
\ell(\theta : D) = \log P(o \mid \theta) \\
= \log \sum_x P(x, o \mid \theta)
$$

- If there were $m$ sequences:

$$
\ell(\theta : D) = \sum_{j=1}^{m} \log \sum_x P(x, o^{(j)} \mid \theta)
$$
Computing Log likelihood for HMMs when $X$ is hidden

\[ \ell(\theta : D) = \log P(o \mid \theta) = \log \sum_{x} P(x, o \mid \theta) \]
Computing Log likelihood for HMMs when $X$ is hidden – variable elimination

- Can compute efficiently with variable elimination:

$$
\ell(\theta : D) = \log P(o | \theta) = \log \sum_x P(x, o | \theta)
$$
EM for HMMs when $X$ is hidden

- E-step: Use inference (forwards-backwards algorithm)

- M-step: Recompute parameters with weighted data

$X_1 = \{a, \ldots, z\} \rightarrow X_2 = \{a, \ldots, z\} \rightarrow X_3 = \{a, \ldots, z\} \rightarrow X_4 = \{a, \ldots, z\} \rightarrow X_5 = \{a, \ldots, z\}$

$O_1 = \text{[symbol]} \quad O_2 = \text{[symbol]} \quad O_3 = \text{[symbol]} \quad O_4 = \text{[symbol]} \quad O_5 = \text{[symbol]}$
E-step

- E-step computes probability of hidden vars $x$ given $o$

$$Q^{(t+1)}(x \mid o) = P(x \mid o, \theta^{(t)})$$

- Will correspond to inference
  - use forward-backward algorithm!
The M-step

- **Maximization step:**

\[
\theta^{(t+1)} \leftarrow \arg \max_\theta \sum_x Q^{(t+1)}(x | o) \log P(x, o | \theta)
\]

- Use expected counts instead of counts:
  - If learning requires \( \text{Count}(x, o) \)
  - Use \( \mathbb{E}_{Q^{(t+1)}}[\text{Count}(x, o)] \)
Decomposition of likelihood revisited

\[ P(X_1) \]
\[ P(O_i \mid X_i) \]
\[ P(X_i \mid X_{i-1}) \]

- Likelihood optimization decomposes:

\[
\max_\theta \sum_x Q(x \mid o) \log P(x, o \mid \theta) =
\]
\[
\max_\theta \sum_x Q(x \mid o) \log P(x_1 \mid \theta_{X_1})P(o_1 \mid x_1, \theta_{O \mid X}) \prod_{t=2}^{n} P(x_t \mid x_{t-1}, \theta_{X_t \mid X_{t-1}})P(o_t \mid x_t, \theta_{O \mid X})
\]
Starting state probability $P(X_1)$

- Using expected counts
  
  $P(X_1 = a) = \theta_{X_1=a}$

$$\max_{\theta_{X_1}} \sum_x Q(x \mid o) \log P(x_1 \mid \theta_{X_1})$$

$$\theta_{X_1=a} = \frac{\sum_{j=1}^m Q(X_1 = a \mid o^{(j)})}{m}$$
Transition probability $P(X_t|X_{t-1})$

- Using expected counts

$$P(X_t=a|X_{t-1}=b) = \theta_{X_t=a|X_{t-1}=b}$$

$$\max_{\theta_{X_t|X_{t-1}}} \sum_x Q(x | o) \log \prod_{t=2}^{n} P(x_t | x_{t-1}, \theta_{X_t|X_{t-1}})$$

$$\theta_{X_t=a|X_{t-1}=b} = \frac{\sum_{j=1}^{m} \sum_{t=2}^{n} Q(X_t = a, X_{t-1} = b | o^{(j)})}{\sum_{j=1}^{m} \sum_{t=2}^{n} \sum_{i=1}^{k} Q(X_t = i, X_{t-1} = b | o^{(j)})}$$
Observation probability $P(O_t | X_t)$

- Using expected counts
  
  $P(O_t = a | X_t = b) = \theta_{O_t = a | X_t = b}$

$$\max_{\theta_{O|X}} \sum_x Q(x | o) \log \prod_{t=1}^n P(o_t | x_t, \theta_{O|X})$$

$$\theta_{O_t = a | X_t = b} = \frac{\sum_{j=1}^m \sum_{t=1}^n \delta(o_t^{(j)} = a) Q(X_t = b | o^{(j)})}{\sum_{j=1}^m \sum_{t=1}^n Q(X_t = b | o^{(j)})}$$
E-step revisited

\[ Q^{(t+1)}(x \mid o) = P(x \mid o, \theta^{(t)}) \]

- E-step computes probability of hidden vars \( x \) given \( o \)
- Must compute:
  - \( Q(x_t=a \mid o) \) – marginal probability of each position
  - \( Q(x_{t+1}=a, x_t=b \mid o) \) – joint distribution between pairs of positions
The forwards-backwards algorithm

**Initialization:** $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$

For $i = 2$ to $n$

- Generate a forwards factor by eliminating $X_{i-1}$
  
  $\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$

**Initialization:** $\beta_n(X_n) = 1$

For $i = n-1$ to $1$

- Generate a backwards factor by eliminating $X_{i+1}$
  
  $\beta_i(X_i) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid X_i)\beta_{i+1}(x_{i+1})$

8 i, probability is:

$$P(X_i \mid o_1..n) \propto \alpha_i(X_i)\beta_i(X_i)$$
E-step revisited

\[ Q^{(t+1)}(x | o) = P(x | o, \theta^{(t)}) \]

- E-step computes probability of hidden vars \( x \) given \( o \)

- Must compute:
  - \( Q(x_t = a | o) \) – marginal probability of each position
    - Just forwards-backwards!
  - \( Q(x_{t+1} = a, x_t = b | o) \) – joint distribution between pairs of positions
What can you do with EM for HMMs?

1. Clustering sequences

Independent clustering:

Sequence clustering:
What can you do with EM for HMMs? 2

- Exploiting unlabeled data

- Labeling data is hard work! Save (graduate student) time by using both labeled and unlabeled data

- Labeled data:
  - \(<X=\text{"brace"}, O=\) >

- Unlabeled data:
  - \(<X=???????, O=\) >
Exploiting unlabeled data in clustering

- A few data points are labeled
  - \(<x,o>\)

- Most points are unlabeled
  - \(<?,o>\)

- In the E-step of EM:
  - If i’th point is unlabeled:
    - compute \(Q(X|o_i)\) as usual
  - If i’th point is labeled:
    - set \(Q(X=x|o_i)=1\) and \(Q(X\neq x|o_i)=0\)

- M-step as usual
Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol $D$ indicates an arbitrary digit.

<table>
<thead>
<tr>
<th>Iteration 0</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>intelligence</td>
<td>$DD$</td>
<td>$D$</td>
</tr>
<tr>
<td>$DD$</td>
<td>lecture</td>
<td>$D$</td>
</tr>
<tr>
<td>artificial</td>
<td>cc</td>
<td>$DD$</td>
</tr>
<tr>
<td>understanding</td>
<td>$D*D$</td>
<td>$DD:DD$</td>
</tr>
<tr>
<td>$DDw$</td>
<td>handout</td>
<td>due</td>
</tr>
<tr>
<td>dist</td>
<td>due</td>
<td>$D*$</td>
</tr>
<tr>
<td>identical</td>
<td>problem</td>
<td>homework</td>
</tr>
<tr>
<td>rus</td>
<td>set</td>
<td>assignment</td>
</tr>
<tr>
<td>arrange</td>
<td>tay</td>
<td>handout</td>
</tr>
<tr>
<td>games</td>
<td>$DDAm$</td>
<td>set</td>
</tr>
<tr>
<td>dartmouth</td>
<td>yurttas</td>
<td>hw</td>
</tr>
<tr>
<td>natural</td>
<td>homework</td>
<td>exam</td>
</tr>
<tr>
<td>cognitive</td>
<td>kfoury</td>
<td>problem</td>
</tr>
<tr>
<td>logic</td>
<td>sec</td>
<td>$DDAm$</td>
</tr>
<tr>
<td>proving</td>
<td>postscript</td>
<td>postscript</td>
</tr>
<tr>
<td>prolog</td>
<td>solution</td>
<td>solution</td>
</tr>
<tr>
<td>knowledge</td>
<td>exam</td>
<td>quiz</td>
</tr>
<tr>
<td>human</td>
<td>solution</td>
<td>chapter</td>
</tr>
<tr>
<td>representation</td>
<td>assaf</td>
<td>ascii</td>
</tr>
</tbody>
</table>

Using one labeled example per class
20 Newsgroups data – advantage of adding unlabeled data
20 Newsgroups data – Effect of additional unlabeled data
Exploiting unlabeled data in HMMs

- A few data points are labeled
  - `<x,o>`

- Most points are unlabeled
  - `<?,o>`

- In the E-step of EM:
  - If i’th point is unlabeled:
    - compute $Q(X|o_i)$ as usual
  - If i’th point is labeled:
    - set $Q(X=x|o_i)=1$ and $Q(X\neq x|o_i)=0$

- M-step as usual
  - Speed up by remembering counts for labeled data
What you need to know

- Baum-Welch = EM for HMMs
- E-step:
  - Inference using forwards-backwards
- M-step:
  - Use weighted counts
- Exploiting unlabeled data:
  - Some unlabeled data can help classification
  - Small change to EM algorithm
    - In E-step, only use inference for unlabeled data
Acknowledgements

- Experiments combining labeled and unlabeled data provided by Tom Mitchell