The EM algorithm for HMMs

Michael Collins

February 22, 2012

Maximum-Likelihood Estimation for Fully Observed Data (Recap from earlier)

- We have fully observed data, $x_{i,1} \dots x_{i,m}, s_{i,1} \dots s_{i,m}$ for $i = 1 \dots n$. The likelihood function is

$$L(\underline{\theta}) = \sum_{i=1}^{n} \log p(x_{i,1} \dots x_{i,m}, s_{i,1} \dots s_{i,m}; \underline{\theta})$$

- Maximum-likelihood estimates of transition probabilities are

$$t(s'|s) = \frac{\sum_{i=1}^{n} \text{count}(i, s \rightarrow s')}{\sum_{i=1}^{n} \sum_{s'} \text{count}(i, s \rightarrow s')}$$

- Maximum-likelihood estimates of emission probabilities are

$$e(x|s) = \frac{\sum_{i=1}^{n} \text{count}(i, s \rightsquigarrow x)}{\sum_{i=1}^{n} \sum_{x} \text{count}(i, s \rightsquigarrow x)}$$

Maximum-Likelihood Estimation for Partially Observed Data

- We have partially observed data, $x_{i,1} \ldots x_{i,m}$ for $i = 1 \ldots n$. Note we do *not* have state sequences. The likelihood function is

$$L(\underline{\theta}) = \sum_{i=1}^{n} \log \sum_{s_1 \ldots s_m} p(x_{i,1} \ldots x_{i,m}, s_1 \ldots s_m; \underline{\theta})$$

- We can maximize this function using EM... (the algorithm will converge to a local maximum of the likelihood function)

An Example

- Suppose we have an HMM with two states ($k = 2$) and 4 possible emissions (a, b, x, y) and our (partially observed) training data consists of the following counts of 4 different sequences (no other sequences are seen):

a	x	(100 times)
a	y	(100 times)
b	x	(100 times)
b	y	(100 times)

- What are the maximum-likelihood estimates for the HMM?

Forward and Backward Probabilities

- Define $\alpha[j, s]$ to be the sum of probabilities of all paths ending in state s at position j in the sequence, for $j = 1 \ldots m$ and $s \in \{1 \ldots k\}$. More formally:

$$\alpha[j, s] = \sum_{s_1 \ldots s_{j-1}} \left[t(s_1) e(x_1 | s_1) \left(\prod_{k=2}^{j-1} t(s_k | s_{k-1}) e(x_k | s_k) \right) t(s | s_{j-1}) e(x_j | s) \right]$$

- Define $\beta[j, s]$ for $s \in \{1 \ldots k\}$ and $j \in \{1 \ldots (m-1)\}$ to be the sum of probabilities of all paths starting with state s at position j and going to the end of the sequence. More formally:

$$\beta[j, s] = \sum_{s_{j+1} \ldots s_m} \left[t(s_{j+1} | s) e(x_{j+1} | s_{j+1}) \left(\prod_{k=j+2}^{m} t(s_k | s_{k-1}) e(x_k | s_k) \right) \right]$$

Recursive Definitions of the Forward Probabilities

- Initialization: for $s = 1 \ldots k$

$$\alpha[1, s] = t(s)e(x_1|s)$$

- For $j = 2 \ldots m$:

$$\alpha[j, s] = \sum_{s' \in \{1 \ldots k\}} (\alpha[j-1, s'] \times t(s|s') \times e(x_j|s))$$

Recursive Definitions of the Backward Probabilities

- Initialization: for $s = 1 \dots k$

$$\beta[m, s] = 1$$

- For $j = m - 1 \dots 1$:

$$\beta[j, s] = \sum_{s' \in \{1 \dots k\}} \left(\beta[j+1, s'] \times t(s'|s) \times e(x_{j+1}|s') \right)$$

The Forward-Backward Algorithm

- Given these definitions:

$$p(x_1 \ldots x_m, S_j = s; \underline{\theta})$$

$$= \sum_{s_1 \ldots s_m : s_j = s} p(x_1 \ldots x_m, s_1 \ldots s_m; \underline{\theta})$$

$$= \alpha[j, s] \times \beta[j, s]$$

- Note: we'll assume the special definition that $\beta[m, s] = 1$ for all s

The Forward-Backward Algorithm

- Given these definitions:

$$
\begin{aligned}
& p(x_1 \ldots x_m, S_j = s, S_{j+1} = s'; \underline{\theta}) \\
= & \sum_{s_1 \ldots s_m : s_j = s, s_{j+1} = s'} p(x_1 \ldots x_m, s_1 \ldots s_m; \underline{\theta}) \\
= & \alpha[j, s] \times t(s'|s) \times e(x_{j+1}|s') \times \beta[j+1, s']
\end{aligned}
$$

- Note: we'll assume the special definition that $\beta[m, s] = 1$ for all s

Things we can Compute Using Forward-Backward Probabilities

- The probability of any sequence:

$$
\begin{aligned}
p(x_1 \ldots x_m; \underline{\theta}) &= \sum_{s_1 \ldots s_m} p(x_1 \ldots x_m, s_1 \ldots s_m; \underline{\theta}) \\
&= \sum_{s} \alpha[m, s]
\end{aligned}
$$

- The probability of any state transition:

$$
\begin{aligned}
& p(x_1 \ldots x_m, S_j = s, S_{j+1} = s'; \underline{\theta}) \\
&= \sum_{s_1 \ldots s_m : s_j = s, s_{j+1} = s'} p(x_1 \ldots x_m, s_1 \ldots s_m; \underline{\theta}) \\
&= \alpha[j, s] \times t(s'|s) \times e(x_{j+1}|s') \times \beta[j+1, s']
\end{aligned}
$$

Things we can Compute Using Forward-Backward Probabilities (continued)

- The *conditional* probability of any state transition:

$$p(S_j = s, S_{j+1} = s'|x_1 \dots x_m; \underline{\theta})$$

$$= \frac{\alpha[j,s] \times t(s'|s) \times e(x_{j+1}|s') \times \beta[j+1,s']}{\sum_s \alpha[m,s]}$$

- The *conditional* probability of any state at any position:

$$p(S_j = s|x_1 \dots x_m; \underline{\theta}) = \frac{\alpha[j,s] \times \beta[j,s]}{\sum_s \alpha[m,s]}$$

Things we can Compute Using Forward-Backward Probabilities (continued)

- Define $\overline{\text{count}}(i, s \to s'; \underline{\theta})$ to be the expected number of times the transition $s \to s'$ is seen in the training example $x_{i,1}, x_{i,2}, \ldots, x_{i,m}$, for parameters $\underline{\theta}$. Then

$$\overline{\text{count}}(i, s \to s'; \underline{\theta}) = \sum_{j=1}^{m-1} p(S_j = s, S_{j+1} = s' | x_{i,1} \ldots x_{i,m}; \underline{\theta})$$

(We can compute $p(S_j = s, S_{j+1} = s' | x_{i,1} \ldots x_{i,m}; \underline{\theta})$ using the forward-backward probabilities, see previous slide)

Things we can Compute Using Forward-Backward Probabilities (continued)

- For completeness, a formal definition of $\overline{\text{count}}(i, s \to s'; \underline{\theta})$:

$$\begin{aligned} & \overline{\text{count}}(i, s \to s'; \underline{\theta}) \\ = & \sum_{s_1 \ldots s_m} p(s_1 \ldots s_m | x_{i,1} \ldots x_{i,m}; \underline{\theta}) \text{count}(s \to s', s_1 \ldots s_m) \end{aligned}$$

  where $\text{count}(s \to s', s_1 \ldots s_m)$ is the number of times the transition $s \to s'$ is seen in the sequence $s_1 \ldots s_m$

Things we can Compute Using Forward-Backward Probabilities (continued)

- Define $\overline{\text{count}}(i, s \rightsquigarrow z; \underline{\theta})$ to be the expected number of times the state s is paired with the emission z in the training sequence $x_{i,1}, x_{i,2}, \ldots, x_{i,m}$, for parameters $\underline{\theta}$. Then

$$\overline{\text{count}}(i, s \rightsquigarrow z; \underline{\theta}) = \sum_{j=1}^{m} p(S_j = s | x_{i,1} \ldots x_{i,m}; \underline{\theta}) [[x_{i,j} = z]]$$

(We can compute $p(S_j = s | x_{i,1} \ldots x_{i,m}; \underline{\theta})$ using the forward-backward probabilities, see previous slides)

The EM Algorithm for HMMs

- Initialization: set initial parameters $\underline{\theta}^0$ to some value

- For $t = 1 \ldots T$:

  - Use the forward-backward algorithm to compute all expected counts of the form

$$\overline{\mathsf{count}}(i, s \to s'; \underline{\theta}^{t-1}) \text{ or } \overline{\mathsf{count}}(i, s \rightsquigarrow z; \underline{\theta}^{t-1})$$

  - Update the parameters based on the expected counts:

$$t^t(s'|s) = \frac{\sum_{i=1}^n \overline{\mathsf{count}}(i, s \to s'; \underline{\theta}^{t-1})}{\sum_{i=1}^n \sum_{s'} \overline{\mathsf{count}}(i, s \to s'; \underline{\theta}^{t-1})}$$

$$e^t(x|s) = \frac{\sum_{i=1}^n \overline{\mathsf{count}}(i, s \rightsquigarrow x; \underline{\theta}^{t-1})}{\sum_{i=1}^n \sum_{x} \overline{\mathsf{count}}(i, s \rightsquigarrow x; \underline{\theta}^{t-1})}$$

The Initial State Probabilities

- For simplicity I've omitted the estimates for the initial state parameters $t(s)$, but these are simple to derive in a similar way to the transition and the emission parameters

- For completeness, the expected counts are:

$$\overline{\text{count}}(i, s; \underline{\theta}^{t-1}) = \frac{\alpha[1, s] \times \beta[1, s]}{\sum_s \alpha[m, s]}$$

  (the expected number of times state s is seen as the initial state)

- The parameter updates are then

$$t^t(s) = \frac{\sum_{i=1}^{n} \overline{\text{count}}(i, s; \underline{\theta}^{t-1})}{n}$$