EM and HMM

Leon Gu

CSD, CMU
The EM Algorithm

Suppose that we have observed some data $y = \{(y_1, y_2, \ldots, y_n)^T\}$, we want to fit a likelihood (or posterior) model by maximizing log-likelihood (or posterior)

$$\ell(\theta; y) = \log p(y \mid \theta).$$

Suppose that we don't know the explicit form of $p(y|\theta)$, instead we know there are some unobserved (hidden) variable $x$, and we can write down $p(y|\theta)$ as an integration of the joint probability of $y$ and $x$, so

$$\ell(\theta; y) = \log \sum_x p(y, x \mid \theta).$$

Directly maximizing $\ell(\theta; y)$ of this form is difficult because the log term “$\log \sum$” can not be further reduced. Instead of examining through all possible $x$ and maximizing their sum, we are going to use an iterative, greedy searching technique called Expectation-Maximization to maximize the log-likelihood.
Step One: Find a lower-bound of $\ell(\theta; y)$

First we introduce a density function $q(x)$ called “averaging distribution”. A lower-bound of the log-likelihood is given by,

$$
\ell(\theta; y) = \log p(y|\theta) \\
= \log \sum_x p(y, x|\theta) \\
= \log \sum_x q(x) \frac{p(y, x|\theta)}{q(x)} \\
\geq \sum_x q(x) \log \frac{p(x, y|\theta)}{q(x)} \\
= E_{q(x)} [\log p(y, x|\theta)] + \text{Entropy} [q(x)] \\
= L(q, \theta; y)
$$

The $\geq$ follows from Jensen’s inequality (log-concavity). More explicitly we can decouple $\ell(\theta; y)$ as the sum of three terms:

$$
\ell(\theta; y) = E_{q(x)} [\log p(y, x|\theta)] + KL[q(x) \parallel p(x|y, \theta)] + \text{Entropy} [q(x)]
$$

The expectation term $E_{q(x)} [\log p(y, x|\theta)]$ is called the-expected-complete-log-likelihood (or Q-function). The equation says that the sum of the Q-function and the entropy of averaging distribution provides a lower-bound of the log-likelihood.
Step Two: Maximize the bound over $\theta$ and $q(x)$ iteratively

Look at the bound $L(q, \theta; y)$. The equality is reached only at $q(x) = p(x|y, \theta)$, and the entropy term is independent of $\theta$. So we have

E-step: \[ q^t = \arg \max_q L(q, \theta^{t-1}; y) = p(x|y, \theta^{t-1}) \]

M-step: \[ \theta^t = \arg \max_\theta L(q^t, \theta; y) = \arg \max_\theta E_{q^t(x)} [\log p(y, x|\theta)] \]

or equivalently we have ,

One Step EM Update: \[ \theta^t = \arg \max_\theta E_{p(x|y, \theta^{t-1})} [\log p(y, x|\theta)] \] \tag{3} \]

If the complete-data-likelihood $\log p(y, x|\theta)$ is factorizable, optimizing the $Q$-function could be much easier than optimizing the log-likelihood.
Now we look at one example of EM which will provide more insights about the algorithm. Again, let $y$ denote the observed data and $x$ denote the hidden variable. Suppose that the joint probability $p(y, x|\theta)$ falls into exponential families, we can write it down as,

$$p(y, x|\theta) = \exp \{ \langle g(\theta), T(y, x) \rangle + d(\theta) + s(y, x) \}$$
If the MLE estimate of $\theta$ exists, then it must be some function of the sufficient statistics $T(y, x)$.

$$\theta_{MLE} = \arg\max_{\theta \in \Omega} \{ \langle g(\theta), T(y, x) \rangle + d(\theta) \}$$  \hspace{1cm} (4)$$

$$= f(T(y, x))$$  \hspace{1cm} (5)$$
According to its definition the Q-function $E_{q(x)}[\log p(y, x|\theta)]$ is,

$$Q(\theta', \theta) = E_{p(x|y, \theta')}[\log p(y, x|\theta)]$$

$$= E_{p(x|y, \theta')}[\langle g(\theta), T(y, x) \rangle + d(\theta) + s(y, x)]$$

$$= \langle g(\theta), E_{p(x|y, \theta')}[T(y, x)] \rangle + d(\theta) + \text{Constant}$$

Let $\overline{T(y, x)} = E_{p(x|y, \theta')}[T(y, x)]$, the EM updating is then given by the recursion

$$\theta''_{EM} = \arg\max_{\theta \in \Omega} Q(\theta', \theta)$$

$$= \arg\max_{\theta \in \Omega} \langle g(\theta), \overline{T(y, x)} \rangle + d(\theta)$$

$$= f(\overline{T(y, x)})$$

We conclude that when the complete data density is from exponential families, in the M step the EM estimate of the parameters take the exactly same form as the MLE estimate. The only difference is the sufficient statistics $T(y, x)$ are replaced by the expected sufficient statistics $\overline{T(y, x)}$. 
Suppose that we have observed a sequence of data \( \{y_1, y_2, \ldots, y_T\} \) (grey nodes), each of which is associated with a hidden state \( \{q_1, q_2, \ldots, q_T\} \).
Basic Settings

In Hidden Markov Model we make a few assumptions about the data:

1. *Discrete state space assumption*: the values of $q_t$ are discrete, $q_t \in \{S_1, \ldots, S_M\}$;

2. *Markov assumptions*:
   
   2.1 Given the state at time $t$, the state at time $t+1$ is independent to all previous states, that is, $q_{t+1} \perp q_i | q_t, \forall i < t$.
   
   2.2 Given the state at time $t$, the corresponding observation $y_t$ is independent to all other states, $y_t \perp q_i | q_t, \forall i \neq t$.

Then the behavior of a HMM is fully determined by three probabilities

1. the *transition probability* $p(q_{t+1} | q_t)$ - the probability of $q_{t+1}$ given its previous state $q_t$. Since the states are discrete, we can describe the transition probability by a $M \times M$ matrix which is called transition matrix. The $ij$-th element of the matrix denotes the probability of the state transiting from the $i$-th state to the $j$-th state.

2. the *emission probability* $p(y_t | q_t)$ - the probability of the observation $q_t$ given its hidden state $q_t$.

3. the *initial state distribution* $\pi(q_0)$. 
We are interested in the following problems:

1. (Inference) compute the probability of hidden states given observations, more specifically,
   1.1 the smoothing problem: compute $p(q_t | y_0 \sim y_T) \ (t < T)$;
   1.2 the filtering problem: compute $p(q_t | y_0, \sim y_t) \ (t = T)$
   1.3 the prediction problem: compute $p(q_t | y_0 \sim y_T) \ (t > T)$.
   1.4 find the most probable sequence of states $\{q_0 \sim q_t\}$ that maximizes $p(q_0 \sim q_t | y_0 \sim y_T)$

2. (Learning) decide the parameters of the models $p(q_{t+1} | q_t)$ and $\pi(q_0)$. 
The Forward-backward Algorithm (or $\alpha$-$\beta$ Algorithm)

Let us look at the the smoothing problem ($t < T$),

\[
p(q_t|y_0 \sim y_T) = \frac{p(q_t, y_0 \sim y_T)}{p(y_0 \sim y_T)}
\]

\[
p(q_t, y_0 \sim y_T) = p(y_0 \sim y_T|q_t) p(q_t)
\]

\[
= p(y_0 \sim y_T, q_t) p(y_{t+1} \sim y_T|q_t)
\]

\[
= \alpha(q_t) \beta(q_t)
\]

Note that we simplify notations by defining

\[
\alpha(q_t) = p(y_0 \sim y_t, q_t)
\]

\[
\beta(q_t) = p(y_{t+1} \sim y_T|q_t)
\]
Notice that both $\alpha(q_t)$ and $\beta(q_t)$ can be computed iteratively

$$
\alpha(q_t) = p(y_0 \sim y_t, q_t)
= \sum_{q_{t-1}} p(y_0 \sim y_t, q_t, q_{t-1})
= \sum_{q_{t-1}} p(y_0 \sim y_{t-1}, q_{t-1}) p(y_t, q_t | y_0 \sim y_{t-1}, q_{t-1})
= \sum_{q_{t-1}} p(y_0 \sim y_{t-1}, q_{t-1}) p(q_t | q_{t-1}) p(y_t | q_t)
= \sum_{q_{t-1}} \alpha(q_{t-1}) p(q_t | q_{t-1}) p(y_t | q_t)
$$
\[ \beta(q_t) = p(y_{t+1} \sim y_T | q_t) \]
\[ = \sum_{q_{t+1}} p(y_{t+1} \sim y_T, q_{t+1} | q_t) \]
\[ = \sum_{q_{t+1}} p(y_{t+1} \sim y_T | q_{t+1}, q_t) p(q_{t+1} | q_t) \]
\[ = \sum_{q_{t+1}} p(y_{t+2} \sim y_T | q_{t+1}) p(y_{t+1} | q_{t+1}) p(q_{t+1} | q_t) \]
\[ = \sum_{q_{t+1}} \beta(q_{t+1}) p(y_{t+1} | q_{t+1}) p(q_{t+1} | q_t) \]
Also notice that we can compute $\alpha(q_0)$ and $\beta(q_{T-1})$ by

$$\alpha(q_0) = p(y_0, q_0) = p(q_0)p(y_0|q_0)$$
$$\beta(q_{T-1}) = p(y_T|q_{T-1}) = \sum_{q_T} p(y_T|q_T)p(q_T|q_{T-1})$$

As a summary, the algorithm consists of two phases:

**forward phase:**

$$\alpha(q_t) = p(y_t|q_t) \sum_{q_{t-1}} p(q_t|q_{t-1}) \alpha(q_{t-1});$$

**backward phase:**

$$\beta(q_t) = \sum_{q_{t+1}} p(y_{t+1}|q_{t+1}) p(q_{t+1}|q_t) \beta(q_{t+1});$$

and the probability $p(q_t|y_0 \sim y_T)$ is given by

$$p(q_t|y_0 \sim y_T) = \frac{p(q_t, y_0 \sim y_T)}{p(y_0 \sim y_T)} \propto \alpha(q_t)\beta(q_t).$$
The $\gamma$ Algorithm

The backward step in the alpha-beta algorithm requests all the observations after the time $t$: $\{y_i|_{i=t+1,...,T}\}$. In practice we usually hope to throw the data away when we filter back. That motivates the $\gamma$-algorithm.

$$
\gamma(q_t) = p(q_t|y_0 \sim y_T) = \sum_{q_{t+1}} p(q_t, q_{t+1}|y_0 \sim y_T)
$$

$$
= \sum_{q_{t+1}} p(q_{t+1}|y_0 \sim y_T) p(q_t|q_{t+1}, y_0 \sim y_T)
$$

$$
= \sum_{q_{t+1}} \gamma(q_{t+1}) p(q_t|q_{t+1}, y_0 \sim y_t)
$$

$$
= \sum_{q_{t+1}} \gamma(q_{t+1}) \frac{p(q_t, q_{t+1}, y_0 \sim y_t)}{p(q_{t+1}, y_0 \sim y_t)}
$$

$$
= \sum_{q_{t+1}} \gamma(q_{t+1}) \frac{p(q_{t+1}|q_t) p(q_t, y_0 \sim y_t)}{p(q_{t+1}, y_0 \sim y_t)}
$$

$$
= \sum_{q_{t+1}} \gamma(q_{t+1}) \frac{p(q_{t+1}|q_t) \alpha(q_t)}{\sum_{q_t} p(q_{t+1}|q_t) \alpha(q_t)}
$$
The Max-Product Algorithm (or the *Viterbi algorithm*)

Now we look at the fourth inference problem: finding the most probable sequence of states \(\{q_0 \sim q_t\}\) that maximizes the posterior \(p(q_0 \sim q_t | y_0 \sim y_t)\). This problem can be solved by the so-called “max-product” algorithm.

\[
\max_{q_0 \sim q_t} p(q_0 \sim q_t | y_0 \sim y_t) = \max_{q_0 \sim q_t} p(q_0 \sim q_t, y_0 \sim y_t) = \max_{q_0 \sim q_t} \left\{ p(q_0) p(y_0 | q_0) \prod_{i=1}^{t} p(q_i | q_{i-1}) p(y_i | q_i) \right\}
\]

\[
= \max_{q_0 \sim q_t} \left\{ \max_{q_0 \sim q_{t-1}} \left\{ p(q_0) p(y_0 | q_0) \prod_{i=1}^{t} p(q_i | q_{i-1}) p(y_i | q_i) \right\} \right\}
\]

\[
= \max_{q_t} \left\{ p(y_t | q_t) \max_{q_0 \sim q_{t-1}} \left\{ p(q_0) p(y_0 | q_0) \prod_{i=1}^{t-1} p(q_i | q_{i-1}) p(y_i | q_i) p(q_t | q_{t-1}) \right\} \right\}
\]

\[
= \max_{q_t} \left\{ p(y_t | q_t) \max_{q_{t-1}} \left\{ p(y_{t-1} | q_{t-1}) p(q_t | q_{t-1}) \ldots \max_{q_0} \left\{ p(q_0) p(y_0 | q_0) p(q_1 | q_0) \right\} \right\} \right\}
\]
Now look at the inner optimization problems:

1. \( \max_{q_0} \{ p(q_0)p(y_0|q_0)p(q_1|q_0) \} \). For each possible value of \( q_1 \) (there are \( M \) of them), we find an optimal \( q_0 \) that maximizes \( p(q_0)p(y_0|q_0)p(q_1|q_0) \) and save the results;

2. \( \max_{q_1} \left\{ p(y_1|q_1)p(q_2|q_1) \max_{q_0} \{ p(q_0)p(y_0|q_0)p(q_1|q_0) \} \right\} \). For each possible value of \( q_2 \), we can find the optimal \( q_1 \) that maximizes \( p(y_1|q_1)p(q_2|q_1) \max_{q_0} \{ p(q_0)p(y_0|q_0)p(q_1|q_0) \} \). Notice that we don’t need to search for \( q_0 \), because we have already computed the optimal \( q_0 \) for each \( q_1 \).

3. Iterate until \( q_t \).

The computational cost of this algorithm is linear to \( t \).
Parameters Learning

Let us parameterize $q_t$ as a $M$-dimensional 0/1 vector, $q_t^i = 1$ indicates the state takes i-th value. The transition probability is defined by:

$$a(q_t, q_{t+1}) = \prod_{i,j=1}^{M} [a_{i,j}]q_t^i q_{t+1}^j$$

and the initial distribution is defined by:

$$\pi(q_0) = \prod_{i=1}^{M} [\pi_i]q_0^i$$

Similarly, we parameterize the observation $y_t$ as a $N$-dimensional vector. Assuming that $p(y_t|q_t)$ is multinomial, we have ($\eta$: observation matrix)

$$p(y_t|q_t, \eta) = \prod_{i,j=1}^{M,N} [\eta_{ij}]q_t^i y_t^j \quad \text{where} \quad \eta_{ij} = p(y_t^j = 1|q_t^i = 1, \eta)$$
The complete-data-log-likelihood is given by

$$
\log p (q, y) = \sum_{i=1}^{M} q_0^i \log \pi_i + \sum_{t=0}^{T} \sum_{i,j=1}^{M} q_t^i q_{t+1}^j \log a_{ij} + \sum_{t=0}^{T} \sum_{i,j=1}^{M,N} q_t^i y_t^j \log \eta_{ij}
$$

From the expression we see that the sufficient statistics for $\pi, a, \eta$ are:

$$
q_0^i; \quad m_{ij} = \sum_{t=0}^{T} q_t^i q_{t+1}^j; \quad n_{ij} = \sum_{t=0}^{T} q_t^i y_t^j
$$

And they are subjective to the constraints:

$$
\sum_{i=1}^{M} \pi_i = 1; \quad \sum_{j=1}^{M} a_{ij} = 1; \quad \sum_{j=1}^{N} \eta_{ij} = 1
$$
Applying Lagrange multiplier method, we obtain the MLE estimates of $\pi, a$ and $\eta$,

\[
\hat{\pi}_i = q^i_0; \\
\hat{a}_{ij} = \frac{m_{ij}}{\sum_{k=1}^{M} m_{ik}}; \\
\hat{\eta}_{ij} = \frac{n_{ij}}{\sum_{k=1}^{N} n_{ik}};
\]

We see the EM estimates just simply replaces the sufficient statistics $q^i_0, m_{ij}, n_{ij}$ by their expectation averaged over $p(q|y, \theta^{\text{old}})$. This is known as the Baum-Welch Algorithm.