BST 227: Expectation maximization notes

Contents

1 The EM (Expectation Maximization) algorithm

This is covered in Chapter 9 of the PRML book, if you would like more in depth reading; below is a brief summary of the EM algorithm.

The EM algorithm is used to perform maximum likelihood estimation of a set of model parameters $\boldsymbol{\theta}$, when latent variables are present. In the following sections, let's assume X represents input (observed) data, and C represents a set of latent (unobserved) variables that we do not observe as real data, but make describing the joint probability of X easier. In the case of our running example of enhancer sequences in which we think there are TFBS, C might indicate the location of the individual TFBS in the enhancer sequences.

Normally, to maximize likelihood when some unobserved variables C are present, we would just marginalize over C (assume C are discrete variables below):

$$p(X|\boldsymbol{\theta}) = \sum_{C} p(X, C|\boldsymbol{\theta}) \tag{1}$$

However, as we found out with the simple sequence model (with no C), we often have to maximize log likelihood, to convert products of probabilities to sum of log probabilities (which makes derivatives easier). Therefore:

$$\ln p(X|\boldsymbol{\theta}) = \ln \left[\sum_{C} p(X, C|\boldsymbol{\theta}) \right] \tag{2}$$

And herein lies the problem: with latent variables C, our expression for $\ln p(X|\boldsymbol{\theta})$ includes a summation inside the $\ln[\cdot]$, which makes taking the derivative with respect to $\boldsymbol{\theta}$ difficult.

The EM algorithm was developed to maximize $p(X|\boldsymbol{\theta})$ in the presence of latent variables C. A brief sketch of how the EM algorithm is derived is shown below, and holds true for any distribution $q(C)$.

$$
\begin{aligned}
\ln p(X|\boldsymbol{\theta}) &= \ln p(X|\boldsymbol{\theta}) && (3)\\
&= \sum_C q(C) \ln p(X|\boldsymbol{\theta}) && (4)\\
&= \sum_C q(C) \ln \frac{p(X, C|\boldsymbol{\theta})}{p(C|X, \boldsymbol{\theta})} && (5)\\
&= \sum_C q(C) \ln p(X, C|\boldsymbol{\theta}) - \sum_C q(C) \ln p(C|X, \boldsymbol{\theta}) && (6)\\
&= \sum_C q(C) \ln \frac{p(X, C|\boldsymbol{\theta})}{q(C)} + \sum_C q(C) \ln \frac{q(C)}{p(C|X, \boldsymbol{\theta})} && (7)\\
&= \text{ELBO}(q(C), \boldsymbol{\theta}) + \text{KL}(q(C)||p(C|X, \boldsymbol{\theta})) + K && (8)\\
& && (9)
\end{aligned}
$$

Where

$$
\text{ELBO}(q(C), \boldsymbol{\theta}) = \boldsymbol{E}_q[\ln p(X, C|\boldsymbol{\theta})] - \boldsymbol{E}_q[\ln q(C)] \quad (10)
$$
$$
(11)
$$

Here, $\text{ELBO}(q(C), \boldsymbol{\theta})$ is called the "evidence lower bound", and $\text{KL}(\cdot)$ is the KL divergence function. Because the KL divergence function is non-negative, then $\text{ELBO}(q(C), \boldsymbol{\theta})$ is a lower bound on the log likelihood function $\ln p(X|\boldsymbol{\theta})$. Note that the ELBO function depends on the choice of distribution $q(C)$ and a specific set of parameter values $\boldsymbol{\theta}$; changing either $q()$ or $\boldsymbol{\theta}$ will change the ELBO function value.

The key idea of the EM algorithm is that the ELBO function is easy to optimize because it primarily involves the term $\ln p(X, C|\boldsymbol{\theta})$, which does not require marginalization over C. The problem with optimizing ELBO is that it is only a lower bound on the log likelihood function (with the difference between the two being the KL divergence term), and in general we cannot guarantee that increasing the ELBO function (by optimizing $\boldsymbol{\theta}$) will increase the log likelihood function.

However, through careful choice of the distribution $q(C)$, we can ensure optimizing ELBO will lead to optimization of the log likelihood function.

More specifically, note that when $q(C) = p(C|X, \boldsymbol{\theta})$, then the KL term goes to zero. In other words, the KL divergence between a probability distribution and itself is 0. When this happens, we have $\ln p(X|\boldsymbol{\theta}) = \text{ELBO}(q(C), \boldsymbol{\theta})$, and increasing $\text{ELBO}(q(C), \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ will also increase $\ln p(X|\boldsymbol{\theta})$.

A very basic implementation of the EM algorithm might look like the following:

1. initialize the parameters randomly to $\boldsymbol{\theta}^{(0)}$
2. set $L^{(-1)} = -\infty$
3. set $L^{(0)} = 0$
4. set convergence criterion $L_t = 1e - 4$ (or some other number)
5. set $t = 1$
6. while $L^{(t-1)} - L^{(t-2)} > L_t$:
   (a) for the current iteration t, compute posteriors based on the previous best estimate of the parameters $\boldsymbol{\theta}^{(t-1)}$ by computing $q_t(C) = p(C|X, \boldsymbol{\theta}^{(t-1)})$. (Note the posterior of C depends on a specific set of parameter values $\boldsymbol{\theta}$)
   (b) find a new set of optimal parameters $\boldsymbol{\theta}^{(t)}$ by maximizing $\text{ELBO}(q_t(\cdot), \boldsymbol{\theta}^{(t)})$ with respect to $\boldsymbol{\theta}^{(t)}$.
   (c) $t = t + 1$

Note that the EM algorithm is iterative: by computing a set of posteriors at time t (based on the previous best guess of the model parameters, $\boldsymbol{\theta}^{(t-1)}$), we are setting the log likelihood equal to the ELBO term.

Finally, also note that when we compute ELBO in practice, the entropy term $-\boldsymbol{E}_q[\ln q(C)]$ critically does not depend on $\boldsymbol{\theta}$, so when we take the derivative with respect to $\boldsymbol{\theta}$, the entropy term disappears. Thus, in the M step derivation, we can effectively ignore this term. For calculation of the ELBO term for monitoring convergence, the entropy of all standard distributions can be found on e.g. Wikipedia.

2 Basic latent variable model derivation

Some notation:

- indices i index over input sequences (enhancers).
- indices j index over positions of individual input sequences. For ease of notation, all input sequences are assumed to be of length L, though they need not be.
- indices k index over bases (A, C, G, T). E.g. $k \in 1, ..., 4$.
- indices l index over models ($l = 0$ indicates TFBS, $l = 1$ indicate background).
- $X_{i,j,k}$: an indicator variable in which $X_{i,j,k} = 1$ if base j of sequence i is equal to base k, otherwise $X_{i,j,k} = 0$.
- $C_{i,j,l}$: an indicator variable in which $C_{i,j,l} = 1$ if base j of sequence i was drawn from model l, otherwise $C_{i,j,l} = 0$. The model l can either be the foreground (TFBS) model $l = 0$, or the background (non-TFBS) model $l = 1$.
- $\boldsymbol{X}$: the set of all $X_{i,j,k}$, $\forall i, j, k$.
- $\boldsymbol{C}$: the set of all $C_{i,j,l}$, $\forall i, j, l$.

The generative sequence model we discussed is:

- $P(C_{i,j,l} = 1) = \lambda_l$, where either $l = 0$ (TFBS) or $l = 1$ (background). Note $0 \leq \lambda_l \leq 1$, and $\lambda_0 + \lambda_1 = 1$.
- $P(X_{i,j,k} = 1 | C_{i,j,l} = 1) = \psi_k^{(l)}$, where $\sum_k \psi_k^{(l)} = 1$.
- We will denote the set of all model parameters $\boldsymbol{\theta} = \{\lambda_l, \psi_k^{(l)}\}$

The complete log likelihood of the data can be then defined as:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{C}) &= \log P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta}) && (12)\\
&= \log \prod_i \prod_j P(X_{i,j}, C_{i,j}|\boldsymbol{\theta}) && (13)\\
&= \log \prod_i \prod_j \left(\prod_k \prod_l \left[P(C_{i,j,l} = 1|\boldsymbol{\theta}) P(X_{i,j,k} = 1|C_{i,j,l} = 1, \boldsymbol{\theta}) \right]^{X_{i,j,k} C_{i,j,l}} \right) && (14)\\
&= \log \prod_i \prod_j \left(\prod_k \prod_l \left[\lambda_l \psi_k^{(l)} \right]^{X_{i,j,k} C_{i,j,l}} \right) && (15)\\
&= \sum_i \sum_j \sum_k \sum_l X_{i,j,k} C_{i,j,l} \log \left[\lambda_l \psi_k^{(l)} \right] && (16)\\
&= \sum_i \sum_j \sum_k \sum_l X_{i,j,k} C_{i,j,l} \log \lambda_l + \sum_i \sum_j \sum_k \sum_l X_{i,j,k} C_{i,j,l} \log \psi_k^{(l)} && (17)
\end{aligned}$$

2.1 Expectation step (E-step)

To perform the E-step in the EM algorithm, we discussed in class how we must calculate the posteriors over the latent variables $\boldsymbol{C}$, e.g., $q(\boldsymbol{C}) = P(\boldsymbol{C}|\boldsymbol{X}, \boldsymbol{\theta})$. Note from Bayes theorem, where $K_1 = P(\boldsymbol{X}|\boldsymbol{\theta})$, we have:

$$
\begin{aligned}
P(\boldsymbol{C}|\boldsymbol{X}, \boldsymbol{\theta}) &= K_1^{-1} P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta}) && (18)\\
&\propto P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta}) && (19)\\
&= \prod_i \prod_j P(X_{i,j}, C_{i,j}|\boldsymbol{\theta}) && (20)\\
&= \prod_i \prod_j P(C_{i,j}|X_{i,j}, \boldsymbol{\theta}) P(X_{i,j}|\boldsymbol{\theta}) && (21)\\
&\propto \prod_i \prod_j P(C_{i,j}|X_{i,j}, \boldsymbol{\theta}) && (22)
\end{aligned}
$$

From the above, we see that the posterior over all latent variables, $P(\boldsymbol{C}|\boldsymbol{X}, \boldsymbol{\theta})$, is proportional to the product of the individual posteriors, $P(C_{i,j}|X_{i,j}, \boldsymbol{\theta})$, up to some normalization constant. To compute the individual posteriors, let $K_2 = P(X_{i,j}|\boldsymbol{\theta})$ and:

$$
\begin{aligned}
P(C_{i,j}|X_{i,j}, \boldsymbol{\theta}) &= K_2^{-1} P(C_{i,j}, X_{i,j}|\boldsymbol{\theta}) && (23)\\
&= K_2^{-1} \prod_k \prod_l [P(C_{i,j,l} = 1|\boldsymbol{\theta}) P(X_{i,j,k} = 1|C_{i,j,l} = 1, \boldsymbol{\theta})]^{X_{i,j,k} C_{i,j,l}} && (24)\\
&= K_2^{-1} \prod_k \prod_l \left(\lambda_l \psi_k^{(l)}\right)^{X_{i,j,k} C_{i,j,l}} && (25)
\end{aligned}
$$

Consider now $K_2 = P(X_{i,j}|\boldsymbol{\theta})$. We know:

$$
\begin{aligned}
K_2 &= P(X_{i,j}|\boldsymbol{\theta}) && (26)\\
&= \sum_{l'=0}^{1} P(X_{i,j}, C_{i,j,l} = l'|\boldsymbol{\theta}) && (27)\\
&= \sum_{l'=0}^{1} \prod_k \left(\lambda_{l'} \psi_k^{(l')}\right)^{X_{i,j,k}} && (28)\\
&= \prod_k \left(\lambda_0 \psi_k^{(0)}\right)^{X_{i,j,k}} + \prod_k \left(\lambda_1 \psi_k^{(1)}\right)^{X_{i,j,k}} && (29)
\end{aligned}
$$

Therefore, we finally get:

$$
P(C_{i,j}|X_{i,j}, \boldsymbol{\theta}) = \frac{\prod_k \prod_l \left(\lambda_l \psi_k^{(l)}\right)^{X_{i,j,k} C_{i,j,l}}}{\prod_k \left(\lambda_0 \psi_k^{(0)}\right)^{X_{i,j,k}} + \prod_k \left(\lambda_1 \psi_k^{(1)}\right)^{X_{i,j,k}}} \quad (30)
$$

2.2 Maximization step (M-step)

We discussed in class how:

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}) &= \log P(\boldsymbol{X}|\boldsymbol{\theta}) && (31)\\
&= \boldsymbol{E}_q[\log P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta})] + \mathrm{KL}(q(C)||P(\boldsymbol{C}|\boldsymbol{X}, \boldsymbol{\theta})) - \boldsymbol{E}_q[\log q(C)] && (32)
\end{aligned}
$$

We discussed how when we set $q(\boldsymbol{C}) = P(\boldsymbol{C}|\boldsymbol{X}, \boldsymbol{\theta})$, then the term $\text{KL}(q(C)||P(\boldsymbol{C}|\boldsymbol{X}, \boldsymbol{\theta})) = 0$. Furthermore, the entropy term $\boldsymbol{E}_q[\log q(C)]$ does not depend on $\boldsymbol{\theta}$. Therefore, when $q(\boldsymbol{C}) = P(\boldsymbol{C}|\boldsymbol{X}, \boldsymbol{\theta})$, we have $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}) = \boldsymbol{E}_q[\log P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta})]$. We can therefore maximize $\boldsymbol{E}_q[\log P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta})]$ as a surrogate for maximizing $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X})$.

$$\boldsymbol{E}_q[\log P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta})] = \boldsymbol{E}_q\left[\sum_i \sum_j \sum_k \sum_l X_{i,j,k} C_{i,j,l} \log \lambda_l + \sum_i \sum_j \sum_k \sum_l X_{i,j,k} C_{i,j,l} \log \psi_k^{(l)}\right] \tag{33}$$

$$= \sum_i \sum_j \sum_k \sum_l \boldsymbol{E}_q\left[X_{i,j,k} C_{i,j,l} \log \lambda_l\right] + \sum_i \sum_j \sum_k \sum_l \boldsymbol{E}_q\left[X_{i,j,k} C_{i,j,l} \log \psi_k^{(l)}\right] \tag{34}$$

$$= \sum_i \sum_j \sum_k \sum_l X_{i,j,k} \boldsymbol{E}_q\left[C_{i,j,l}\right] \log \lambda_l + \sum_i \sum_j \sum_k \sum_l X_{i,j,k} \boldsymbol{E}_q\left[C_{i,j,l}\right] \log \psi_k^{(l)} \tag{35}$$

Upon closer inspection, we see that

$$\boldsymbol{E}_q\left[C_{i,j,l}\right] = 0 \times q(C_{i,j,l} = 0) + 1 \times q(C_{i,j,l} = 1) \tag{36}$$

$$= q(C_{i,j,l} = 1) \tag{37}$$

$$= P(C_{i,j} = l|X_{i,j}, \boldsymbol{\theta}) \tag{38}$$

In the M-step, we therefore just need to take the derivative of $\boldsymbol{E}_q[\log P(\boldsymbol{X}, \boldsymbol{C}|\boldsymbol{\theta})]$ with respect to the model parameters in $\boldsymbol{\theta}$, and solve. Note, for parameters like $\psi_k^{(l)}$ where the parameter vector must sum to 1 (and be non-negative), you must use Lagrange multipliers to solve. See Appendix E of the Pattern Recognition in Machine Learning (PRML) book to see how to use them: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf .

As an example, consider the MLE of λ_l. To use the Lagrange multipliers to enforce $\lambda_0 + \lambda_1 = 1$, we define:

$$h(\lambda_0, \lambda_1) = \text{ELBO}(q_t(C), \boldsymbol{\theta}) + \phi \cdot g(\lambda_0, \lambda_1) \tag{39}$$

Where ϕ is our Lagrange multiplier, and $g(\lambda_0, \lambda_1) = 1 - \lambda_0 - \lambda_1$.

Now we have:

$$\frac{\partial h}{\partial \lambda_l} = \sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q\left[C_{i,j,l}\right] \frac{1}{\lambda_l} - \phi \tag{40}$$

Setting $\frac{\partial h}{\partial \lambda_l} = 0$ and solving for λ_l gives us:

$$\lambda_l = \frac{1}{\phi} \sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q\left[C_{i,j,l}\right] \tag{41}$$

Note when we take the gradient of h with respect to the Lagrange multliplier ϕ, we get the constraint equation back:

$$\frac{\partial h}{\partial \phi} = g(\lambda_0, \lambda_1) \tag{42}$$

Setting $\frac{\partial h}{\partial \phi} = 0$ and substituting in our expression for $\lambda_l = \frac{1}{\phi} \sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l}]$ gives us:

$$\phi = \sum_{l'} \sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l'}] \tag{43}$$

Substituting back into our expression for $\lambda_l = \frac{1}{\phi} \sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l}]$ gives us:

$$\hat{\lambda}_l = \frac{\sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l}]}{\sum_{l'} \sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l'}]} \tag{44}$$

$$= \frac{\sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l}]}{\sum_i \sum_j \sum_k X_{i,j,k} \left[\sum_{l'} \boldsymbol{E}_q [C_{i,j,l'}]\right]} \tag{45}$$

$$= \frac{\sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l}]}{\sum_i \sum_j \sum_k X_{i,j,k}} \tag{46}$$

$$= \frac{\sum_i \sum_j \sum_k X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l}]}{N \times L} \tag{47}$$

Where N is the number of input sequences, and L is their (common) length. Note this is intuitive: λ_l is the prior probability that model l is selected to generate an individual base of a sequence, and its MLE is equal to the estimated number of bases in the training data whose posterior points to model l (numerator), divided by the total number of bases in the training data (denominator). One can see the only difference between the numerator and denominator is the weight $\boldsymbol{E}_q [C_{i,j,l}]$, which is the posterior probability of each base being generated from model l.

Similarly, we can use the Lagrange multipliers to derive the update for $\psi_k^{(l)}$, to enforce that $\sum_k \psi_k^{(l)} = 1$. We therefore would get:

$$\hat{\psi}_k^{(l)} = \frac{\sum_i \sum_j X_{i,j,k} \boldsymbol{E}_q [C_{i,j,l}]}{\sum_i \sum_j \sum_{k'} X_{i,j,k'} \boldsymbol{E}_q [C_{i,j,l}]} \tag{48}$$