Chapter 11

EM algorithms

In this set of notes, we discuss the EM (Expectation-Maximization) algorithm for density estimation.

11.1 EM for mixture of Gaussians

Suppose that we are given a training set $\{x^{(1)}, \ldots, x^{(n)}\}$ as usual. Since we are in the unsupervised learning setting, these points do not come with any labels.

We wish to model the data by specifying a joint distribution $p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$. Here, $z^{(i)} \sim \text{Multinomial}(\phi)$ (where $\phi_j \geq 0$, $\sum_{j=1}^{k} \phi_j = 1$, and the parameter ϕ_j gives $p(z^{(i)} = j)$), and $x^{(i)}|z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j)$. We let k denote the number of values that the $z^{(i)}$'s can take on. Thus, our model posits that each $x^{(i)}$ was generated by randomly choosing $z^{(i)}$ from $\{1, \ldots, k\}$, and then $x^{(i)}$ was drawn from one of k Gaussians depending on $z^{(i)}$. This is called the **mixture of Gaussians** model. Also, note that the $z^{(i)}$'s are **latent** random variables, meaning that they're hidden/unobserved. This is what will make our estimation problem difficult.

The parameters of our model are thus ϕ, μ and Σ. To estimate them, we can write down the likelihood of our data:

$$\begin{aligned} \ell(\phi, \mu, \Sigma) &= \sum_{i=1}^{n} \log p(x^{(i)}; \phi, \mu, \Sigma) \\ &= \sum_{i=1}^{n} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}|z^{(i)}; \mu, \Sigma) p(z^{(i)}; \phi). \end{aligned}$$

However, if we set to zero the derivatives of this formula with respect to the parameters and try to solve, we'll find that it is not possible to find the maximum likelihood estimates of the parameters in closed form. (Try this yourself at home.)

The random variables $z^{(i)}$ indicate which of the k Gaussians each $x^{(i)}$ had come from. Note that if we knew what the $z^{(i)}$'s were, the maximum likelihood problem would have been easy. Specifically, we could then write down the likelihood as

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{n} \log p(x^{(i)}|z^{(i)}; \mu, \Sigma) + \log p(z^{(i)}; \phi).$$

Maximizing this with respect to ϕ, μ and Σ gives the parameters:

$$\begin{aligned}
\phi_j &= \frac{1}{n}\sum_{i=1}^{n} 1\{z^{(i)} = j\}, \\
\mu_j &= \frac{\sum_{i=1}^{n} 1\{z^{(i)} = j\}x^{(i)}}{\sum_{i=1}^{n} 1\{z^{(i)} = j\}}, \\
\Sigma_j &= \frac{\sum_{i=1}^{n} 1\{z^{(i)} = j\}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{n} 1\{z^{(i)} = j\}}.
\end{aligned}$$

Indeed, we see that if the $z^{(i)}$'s were known, then maximum likelihood estimation becomes nearly identical to what we had when estimating the parameters of the Gaussian discriminant analysis model, except that here the $z^{(i)}$'s playing the role of the class labels.[1]

However, in our density estimation problem, the $z^{(i)}$'s are *not* known. What can we do?

The EM algorithm is an iterative algorithm that has two main steps. Applied to our problem, in the E-step, it tries to "guess" the values of the $z^{(i)}$'s. In the M-step, it updates the parameters of our model based on our guesses. Since in the M-step we are pretending that the guesses in the first part were correct, the maximization becomes easy. Here's the algorithm:

Repeat until convergence: {

(E-step) For each i, j, set

$$w_j^{(i)} := p(z^{(i)} = j|x^{(i)}; \phi, \mu, \Sigma)$$

[1] There are other minor differences in the formulas here from what we'd obtained in PS1 with Gaussian discriminant analysis, first because we've generalized the $z^{(i)}$'s to be multinomial rather than Bernoulli, and second because here we are using a different Σ_j for each Gaussian.

(M-step) Update the parameters:

$$\begin{aligned}
\phi_j &:= \frac{1}{n}\sum_{i=1}^{n} w_j^{(i)}, \\
\mu_j &:= \frac{\sum_{i=1}^n w_j^{(i)} x^{(i)}}{\sum_{i=1}^n w_j^{(i)}}, \\
\Sigma_j &:= \frac{\sum_{i=1}^n w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^n w_j^{(i)}}
\end{aligned}$$

}

In the E-step, we calculate the posterior probability of our parameters the $z^{(i)}$'s, given the $x^{(i)}$ and using the current setting of our parameters. I.e., using Bayes rule, we obtain:

$$p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma) = \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{\sum_{l=1}^k p(x^{(i)} | z^{(i)} = l; \mu, \Sigma) p(z^{(i)} = l; \phi)}$$

Here, $p(x^{(i)} | z^{(i)} = j; \mu, \Sigma)$ is given by evaluating the density of a Gaussian with mean μ_j and covariance Σ_j at $x^{(i)}$; $p(z^{(i)} = j; \phi)$ is given by ϕ_j, and so on. The values $w_j^{(i)}$ calculated in the E-step represent our "soft" guesses[2] for the values of $z^{(i)}$.

Also, you should contrast the updates in the M-step with the formulas we had when the $z^{(i)}$'s were known exactly. They are identical, except that instead of the indicator functions "$1\{z^{(i)} = j\}$" indicating from which Gaussian each datapoint had come, we now instead have the $w_j^{(i)}$'s.

The EM-algorithm is also reminiscent of the K-means clustering algorithm, except that instead of the "hard" cluster assignments $c(i)$, we instead have the "soft" assignments $w_j^{(i)}$. Similar to K-means, it is also susceptible to local optima, so reinitializing at several different initial parameters may be a good idea.

It's clear that the EM algorithm has a very natural interpretation of repeatedly trying to guess the unknown $z^{(i)}$'s; but how did it come about, and can we make any guarantees about it, such as regarding its convergence? In the next set of notes, we will describe a more general view of EM, one

[2]The term "soft" refers to our guesses being probabilities and taking values in $[0, 1]$; in contrast, a "hard" guess is one that represents a single best guess (such as taking values in $\{0, 1\}$ or $\{1, \ldots, k\}$).

that will allow us to easily apply it to other estimation problems in which there are also latent variables, and which will allow us to give a convergence guarantee.

11.2 Jensen's inequality

We begin our discussion with a very useful result called **Jensen's inequality**

Let f be a function whose domain is the set of real numbers. Recall that f is a convex function if $f''(x) \geq 0$ (for all $x \in \mathbb{R}$). In the case of f taking vector-valued inputs, this is generalized to the condition that its hessian H is positive semi-definite ($H \geq 0$). If $f''(x) > 0$ for all x, then we say f is **strictly** convex (in the vector-valued case, the corresponding statement is that H must be positive definite, written $H > 0$). Jensen's inequality can then be stated as follows:

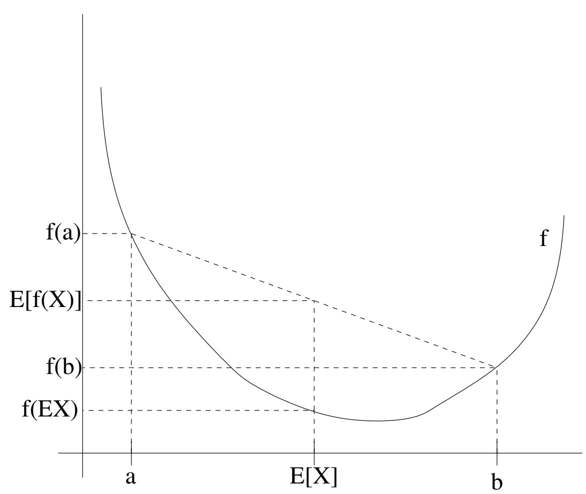
Theorem. Let f be a convex function, and let X be a random variable. Then:

$$\mathrm{E}[f(X)] \geq f(\mathrm{E}X).$$

Moreover, if f is strictly convex, then $\mathrm{E}[f(X)] = f(\mathrm{E}X)$ holds true if and only if $X = \mathrm{E}[X]$ with probability 1 (i.e., if X is a constant).

Recall our convention of occasionally dropping the parentheses when writing expectations, so in the theorem above, $f(\mathrm{E}X) = f(\mathrm{E}[X])$.

For an interpretation of the theorem, consider the figure below.



Here, f is a convex function shown by the solid line. Also, X is a random variable that has a 0.5 chance of taking the value a, and a 0.5 chance of taking the value b (indicated on the x-axis). Thus, the expected value of X is given by the midpoint between a and b.

We also see the values $f(a)$, $f(b)$ and $f(\mathrm{E}[X])$ indicated on the y-axis. Moreover, the value $\mathrm{E}[f(X)]$ is now the midpoint on the y-axis between $f(a)$ and $f(b)$. From our example, we see that because f is convex, it must be the case that $\mathrm{E}[f(X)] \geq f(\mathrm{E}X)$.

Incidentally, quite a lot of people have trouble remembering which way the inequality goes, and remembering a picture like this is a good way to quickly figure out the answer.

Remark. Recall that f is [strictly] concave if and only if $-f$ is [strictly] convex (i.e., $f''(x) \leq 0$ or $H \leq 0$). Jensen's inequality also holds for concave functions f, but with the direction of all the inequalities reversed ($\mathrm{E}[f(X)] \leq f(\mathrm{E}X)$, etc.).

11.3 General EM algorithms

Suppose we have an estimation problem in which we have a training set $\{x^{(1)}, \ldots, x^{(n)}\}$ consisting of n independent examples. We have a latent variable model $p(x, z; \theta)$ with z being the latent variable (which for simplicity is assumed to take finite number of values). The density for x can be obtained by marginalized over the latent variable z:

$$p(x; \theta) = \sum_z p(x, z; \theta) \tag{11.1}$$

We wish to fit the parameters θ by maximizing the log-likelihood of the data, defined by

$$\ell(\theta) = \sum_{i=1}^{n} \log p(x^{(i)}; \theta) \tag{11.2}$$

We can rewrite the objective in terms of the joint density $p(x, z; \theta)$ by

$$\begin{aligned} \ell(\theta) &= \sum_{i=1}^{n} \log p(x^{(i)}; \theta) && (11.3) \\ &= \sum_{i=1}^{n} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta). && (11.4) \end{aligned}$$

But, explicitly finding the maximum likelihood estimates of the parameters θ may be hard since it will result in difficult non-convex optimization problems.[3] Here, the $z^{(i)}$'s are the latent random variables; and it is often the case that if the $z^{(i)}$'s were observed, then maximum likelihood estimation would be easy.

In such a setting, the EM algorithm gives an efficient method for maximum likelihood estimation. Maximizing $\ell(\theta)$ explicitly might be difficult, and our strategy will be to instead repeatedly construct a lower-bound on ℓ (E-step), and then optimize that lower-bound (M-step).[4]

It turns out that the summation $\sum_{i=1}^{n}$ is not essential here, and towards a simpler exposition of the EM algorithm, we will first consider optimizing the the likelihood $\log p(x)$ for **a single example** x. After we derive the algorithm for optimizing $\log p(x)$, we will convert it to an algorithm that works for n examples by adding back the sum to each of the relevant equations. Thus, now we aim to optimize $\log p(x;\theta)$ which can be rewritten as

$$\log p(x;\theta) = \log \sum_z p(x,z;\theta) \tag{11.5}$$

Let Q be a distribution over the possible values of z. That is, $\sum_z Q(z) = 1$, $Q(z) \geq 0$).

Consider the following:[5]

$$\begin{aligned}
\log p(x;\theta) &= \log \sum_z p(x,z;\theta) \\
&= \log \sum_z Q(z) \frac{p(x,z;\theta)}{Q(z)} && (11.6) \\
&\geq \sum_z Q(z) \log \frac{p(x,z;\theta)}{Q(z)} && (11.7)
\end{aligned}$$

The last step of this derivation used Jensen's inequality. Specifically, $f(x) = \log x$ is a concave function, since $f''(x) = -1/x^2 < 0$ over its domain

[3]It's mostly an empirical observation that the optimization problem is difficult to optimize.

[4]Empirically, the E-step and M-step can often be computed more efficiently than optimizing the function $\ell(\cdot)$ directly. However, it doesn't necessarily mean that alternating the two steps can always converge to the global optimum of $\ell(\cdot)$. Even for mixture of Gaussians, the EM algorithm can either converge to a global optimum or get stuck, depending on the properties of the training data. Empirically, for real-world data, often EM can converge to a solution with relatively high likelihood (if not the optimum), and the theory behind it is still largely not understood.

[5]If z were continuous, then Q would be a density, and the summations over z in our discussion are replaced with integrals over z.

$x \in \mathbb{R}^+$. Also, the term

$$\sum_z Q(z) \left[\frac{p(x,z;\theta)}{Q(z)} \right]$$

in the summation is just an expectation of the quantity $[p(x,z;\theta)/Q(z)]$ with respect to z drawn according to the distribution given by Q.[6] By Jensen's inequality, we have

$$f\left(\mathrm{E}_{z\sim Q}\left[\frac{p(x,z;\theta)}{Q(z)} \right] \right) \geq \mathrm{E}_{z\sim Q}\left[f\left(\frac{p(x,z;\theta)}{Q(z)} \right) \right],$$

where the "$z \sim Q$" subscripts above indicate that the expectations are with respect to z drawn from Q. This allowed us to go from Equation (11.6) to Equation (11.7).

Now, for **any** distribution Q, the formula (11.7) gives a lower-bound on $\log p(x;\theta)$. There are many possible choices for the Q's. Which should we choose? Well, if we have some current guess θ of the parameters, it seems natural to try to make the lower-bound tight at that value of θ. I.e., we will make the inequality above hold with equality at our particular value of θ.

To make the bound tight for a particular value of θ, we need for the step involving Jensen's inequality in our derivation above to hold with equality. For this to be true, we know it is sufficient that the expectation be taken over a "constant"-valued random variable. I.e., we require that

$$\frac{p(x,z;\theta)}{Q(z)} = c$$

for some constant c that does not depend on z. This is easily accomplished by choosing

$$Q(z) \propto p(x,z;\theta).$$

Actually, since we know $\sum_z Q(z) = 1$ (because it is a distribution), this further tells us that

$$\begin{aligned} Q(z) &= \frac{p(x,z;\theta)}{\sum_z p(x,z;\theta)} \\ &= \frac{p(x,z;\theta)}{p(x;\theta)} \\ &= p(z|x;\theta) \end{aligned} \tag{11.8}$$

[6] We note that the notion $\frac{p(x,z;\theta)}{Q(z)}$ only makes sense if $Q(z) \neq 0$ whenever $p(x,z;\theta) \neq 0$. Here we implicitly assume that we only consider those Q with such a property.

Thus, we simply set the Q's to be the posterior distribution of the z's given x and the setting of the parameters θ.

Indeed, we can directly verify that when $Q(z) = p(z|x;\theta)$, then equation (11.7) is an equality because

$$\begin{aligned}
\sum_z Q(z) \log \frac{p(x,z;\theta)}{Q(z)} &= \sum_z p(z|x;\theta) \log \frac{p(x,z;\theta)}{p(z|x;\theta)} \\
&= \sum_z p(z|x;\theta) \log \frac{p(z|x;\theta)p(x;\theta)}{p(z|x;\theta)} \\
&= \sum_z p(z|x;\theta) \log p(x;\theta) \\
&= \log p(x;\theta) \sum_z p(z|x;\theta) \\
&= \log p(x;\theta) \qquad\qquad (\text{because } \textstyle\sum_z p(z|x;\theta) = 1)
\end{aligned}$$

For convenience, we call the expression in Equation (11.7) the **evidence lower bound** (ELBO) and we denote it by

$$\mathrm{ELBO}(x;Q,\theta) = \sum_z Q(z) \log \frac{p(x,z;\theta)}{Q(z)} \tag{11.9}$$

With this equation, we can re-write equation (11.7) as

$$\forall Q,\theta,x, \quad \log p(x;\theta) \geq \mathrm{ELBO}(x;Q,\theta) \tag{11.10}$$

Intuitively, the EM algorithm alternatively updates Q and θ by a) setting $Q(z) = p(z|x;\theta)$ following Equation (11.8) so that $\mathrm{ELBO}(x;Q,\theta) = \log p(x;\theta)$ for x and the current θ, and b) maximizing $\mathrm{ELBO}(x;Q,\theta)$ w.r.t θ while fixing the choice of Q.

Recall that all the discussion above was under the assumption that we aim to optimize the log-likelihood $\log p(x;\theta)$ for a single example x. It turns out that with multiple training examples, the basic idea is the same and we only needs to take a sum over examples at relevant places. Next, we will build the evidence lower bound for multiple training examples and make the EM algorithm formal.

Recall we have a training set $\{x^{(1)},\ldots,x^{(n)}\}$. Note that the optimal choice of Q is $p(z|x;\theta)$, and it depends on the particular example x. Therefore here we will introduce n distributions $Q_1,\ldots,Q_n$, one for each example $x^{(i)}$. For each example $x^{(i)}$, we can build the evidence lower bound

$$\log p(x^{(i)};\theta) \geq \mathrm{ELBO}(x^{(i)};Q_i,\theta) = \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}$$

Taking sum over all the examples, we obtain a lower bound for the log-likelihood

$$\ell(\theta) \geq \sum_i \text{ELBO}(x^{(i)}; Q_i, \theta) \tag{11.11}$$
$$= \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

For *any* set of distributions $Q_1, \ldots, Q_n$, the formula (11.11) gives a lower-bound on $\ell(\theta)$, and analogous to the argument around equation (11.8), the Q_i that attains equality satisfies

$$Q_i(z^{(i)}) \quad = \quad p(z^{(i)}|x^{(i)}; \theta)$$

Thus, we simply set the Q_i's to be the posterior distribution of the $z^{(i)}$'s given $x^{(i)}$ with the current setting of the parameters θ.

Now, for this choice of the Q_i's, Equation (11.11) gives a lower-bound on the loglikelihood ℓ that we're trying to maximize. This is the E-step. In the M-step of the algorithm, we then maximize our formula in Equation (11.11) with respect to the parameters to obtain a new setting of the θ's. Repeatedly carrying out these two steps gives us the EM algorithm, which is as follows:

Repeat until convergence {

(E-step) For each i, set

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)}; \theta).$$

(M-step) Set

$$\theta := \arg\max_\theta \sum_{i=1}^n \text{ELBO}(x^{(i)}; Q_i, \theta)$$
$$= \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}. \tag{11.12}$$

}

How do we know if this algorithm will converge? Well, suppose $\theta^{(t)}$ and $\theta^{(t+1)}$ are the parameters from two successive iterations of EM. We will now prove that $\ell(\theta^{(t)}) \leq \ell(\theta^{(t+1)})$, which shows EM always monotonically improves the log-likelihood. The key to showing this result lies in our choice of the Q_i's. Specifically, on the iteration of EM in which the parameters had started out as $\theta^{(t)}$, we would have chosen $Q_i^{(t)}(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta^{(t)})$. We saw earlier that this choice ensures that Jensen's inequality, as applied to get Equation (11.11), holds with equality, and hence

$$\ell(\theta^{(t)}) = \sum_{i=1}^{n} \text{ELBO}(x^{(i)}; Q_i^{(t)}, \theta^{(t)}) \tag{11.13}$$

The parameters $\theta^{(t+1)}$ are then obtained by maximizing the right hand side of the equation above. Thus,

$$\begin{aligned}
\ell(\theta^{(t+1)}) &\geq \sum_{i=1}^{n} \text{ELBO}(x^{(i)}; Q_i^{(t)}, \theta^{(t+1)}) \\
&\qquad\qquad \text{(because ineqaulity (11.11) holds for all } Q \text{ and } \theta) \\
&\geq \sum_{i=1}^{n} \text{ELBO}(x^{(i)}; Q_i^{(t)}, \theta^{(t)}) \qquad \text{(see reason below)} \\
&= \ell(\theta^{(t)}) \qquad \text{(by equation (11.13))}
\end{aligned}$$

where the last inequality follows from that $\theta^{(t+1)}$ is chosen explicitly to be

$$\arg\max_{\theta} \quad \sum_{i=1}^{n} \text{ELBO}(x^{(i)}; Q_i^{(t)}, \theta)$$

Hence, EM causes the likelihood to converge monotonically. In our description of the EM algorithm, we said we'd run it until convergence. Given the result that we just showed, one reasonable convergence test would be to check if the increase in $\ell(\theta)$ between successive iterations is smaller than some tolerance parameter, and to declare convergence if EM is improving $\ell(\theta)$ too slowly.

Remark. If we define (by overloading ELBO$(\cdot)$)

$$\text{ELBO}(Q, \theta) = \sum_{i=1}^{n} \text{ELBO}(x^{(i)}; Q_i, \theta) = \sum_{i} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \tag{11.14}$$

then we know $\ell(\theta) \geq \text{ELBO}(Q, \theta)$ from our previous derivation. The EM can also be viewed an alternating maximization algorithm on ELBO(Q, θ), in which the E-step maximizes it with respect to Q (check this yourself), and the M-step maximizes it with respect to θ.