Short note on EM

Brendan O'Connor (http://brenocon.com)

October 18, 2012

1 EM

The EM algorithm confused me for years. I don't like how the Bishop textbook presents it. Here's another attempt at it. (There are many more EM tutorials on the web.)

EM is a particular optimization approach to solving a maximum likelihood problem. Normally for maximum likelihood, we have data X and parameters Θ, and want to solve the MLE problem $\max_\Theta P(X;\Theta)$.

EM is designed for settings where we also have a set of hidden/latent variables Z, for which if you knew them, parameter estimation would be easy. The two examples we've seen so far are (1) missing data in a Bernoulli Bayes Net (in homework 3), and (2) cluster memberships (in a mixture-of-gaussians). So you would want to use EM in a setting where you have three sets of things,

- X - the data
- Z - the hidden/latent variables
- Θ - the parameters

And it is the case that

- $P(X;\Theta)$ is nasty to optimize for Θ
- $P(X,Z;\Theta)$ is easy to optimize for Θ (or perhaps $P(Z)P(X|Z)$ is easy to optimize)
- $P(Z|X;\Theta)$ is easy to compute

And we care about solving, again,

$$\max_\Theta \log P(X;\Theta) = \log \sum_Z P(X,Z;\Theta)$$

$P(X;\Theta)$ is sometimes called the "incomplete data likelihood" since it doesn't include the latent Z variables, and in fact, you want to integrate them out.

It turns out that EM iteratively finds better estimates of Θ that improve the likelihood, while letting you stick with the simpler $P(X,Z)$ optimization steps. The algorithm not only maintains a current solution for Θ, but also keeps soft probabilistic decisions for all the Z variables. Call this q, which will be updated each iteration. The algorithm is, after initializing Θ to something,

- **E-Step**: Set $q(Z) := P(Z|X;\Theta)$
- **M-Step**: Set $\Theta := \arg\max_\Theta \sum_Z q(Z) \log P(X,Z;\Theta)$

The M-Step is to maximize an *expected* log-likelihood, as a proxy for the true incomplete log-likelihood. Once you do that, your beliefs about what the Z states might be should change, so you go back to the E-Step and recalculate them as the posteriors. The $q(Z)$ values you might call "pseudo-posteriors" or soft beliefs.

EM isn't really an algorithm, but more of a meta-algorithm or a template. You have to work out an instantiation for any particular problem. This can sometimes be tricky. Consider two examples we've seen so far:

For the missing data example in Homework 3, there's one hidden Z variable, and several Bernoulli Θ parameters. The E-Step is pretty straightforward from Bayes Rule and applying some Markov blankets. It turns out, if you work out the M-Step, that the weighted MLE is to do weighted counting: if you have $q(Z) = 0.8$, then add in a fractional count of 0.8 for the appropriate multinomial.

For mixture of Gaussians, there are many z_i variables, one for each data point: the cluster membership variables. We make q an $(N \times K)$-sized matrix to represent a posterior over them. Θ describes the means and variances of the Gaussian clusters. If you take the template above and work it out, you end up with

- **E-Step**: For all i, k: Set $q(z_i = k) := P(z_i = k|x_i; \Theta)$
- **M-Step**: Set $\Theta := \arg\max_\Theta \sum_i \sum_k q(z_i = k) \log [p(x_i|z_i = k; \Theta)\ p(z_i = k; \Theta)]$

Where $P(z_k = k|x_i; \Theta)$ comes from the relative likelihood of the data point under the different Gaussian clusters, and where $p(x_i|z_i = k; \Theta)$ is just the Gaussian density function for the k'th Gaussian cluster (and $p(z_i = k)$ is the mixture prior, which might be a parameter to learn as well). It turns out, if you work out the M-Step above, you come up with the usual sum and sum-of-squares equations for computing mean and variance, except you weight the datapoints by the strength of q-beliefs for which one they're in.

2 Why would EM be good to do?

You can show, via some calculus tricks, that the general form of the EM algorithm is coordinate ascent for a thing related to the weighted log-likelihood.

$$\max_{\Theta,q} \sum_Z q(Z) \log P(X, Z; \Theta) - q(Z) \log q(Z)$$

The M-Step is when you fix q and maximize Θ. The E-Step is when you fix Θ and maximize q; it turns out that the solution is to set them to Z's current posteriors. The above quantity is actually a lower bound on the true incomplete likelihood, so EM is guaranteed to at least improve likelihood (though there are no guarantees how good a solution it will find; for example, EM is notorious for being initialization-dependent).

3 More things to consider

(1) In most all the cases we'll consider in this course, the Z variables (I sometimes call them "E-Step variables") are discrete, and the Θ parameters are continuous. This is not required, but it seems to be a common case for when people use EM. The version shown above is for discrete Z's.

(2) EM is not Bayesian. It's really a maximum likelihood algorithm, albeit with a side effect of getting these pseudo-posterior q's, which may be useful. (They are not Bayesian posteriors since they're based on a point estimate of Θ.) Philosophically EM is non-Bayesian because it thinks of parameters as different from hidden variables. A true Bayesian would say, Θ and Z are all just unknowns, and we should just compute posteriors over all of them. So Bayesian estimation is an alternative to EM.

(3) Another alternative to EM is to make *hard* decisions for the Z's,

$$\max_{\Theta,Z} P(X, Z; \Theta)$$

This is sometimes called "hard EM," and is in fact usually easier to implement. Note that it's like EM where the q beliefs are spiked, putting all beliefs for one z into one particular category. Sometimes hard EM doesn't work as well as standard EM because you can't account for multiple possibilities when you just don't have good knowledge which Z is true. But sometimes it's OK. In the world of latent variable algorithms, no one has a simple good answer for all situations.

(4) EM is very widespread. Speech recognition, machine translation, state-space tracking (for vehicles, robots, missiles...) all use lots of EM.

(5) More references: The Dempster et al 1977 paper that (debatably) invented EM is beautiful and still good to read. Another important paper is Neal and Hinton 1998. I also like the derivation in Andrew Ng's CS229 notes. There are many other resources out there to read too.