Deriving the mean-field approximation

Jonathan Parkinson

A helpful note: This derivation will become a little complicated and it's easy to lose sight of the forest for the trees, so it helps to keep in mind where we're trying to go. We have an intractable posterior that we want to approximate with a tractable posterior, using a "fully-factored" approximation where each variable is independent of every other. Ultimately, we'll show we can do so using an iterative technique where we evaluate the following expression:

$$log(q_n(\theta_n)) = E_{k \neq n}[log(p(\theta, X))] - constant$$

for each parameter n in our model in turn. Essentially we average across all other parameters aside from the one of interest and then cycle through the parameters, doing this for each of them in turn, until the fit converges – hence the name "mean-field". Clearly the underlying assumption – that all parameters are independent of all others – is a strong assumption. As a result, the posterior density estimate generated by variational inference is not usually very good compared to MCMC. Nonetheless, mean-field is much faster than MCMC and gives a very good estimate of the MAP values for our parameters (like EM) together with a crude estimate of uncertainty (which EM can't provide).

For mixture models in particular, variational inference has some interesting benefits, because a mixture model fit using variational inference can "kill off" unneeded components so that the number of components can act as an upper bound (this is in stark contrast to EM, which will try to use all the components you give it).

Having explained what we're trying to get and why...let's see how we get there.

As always, begin with Bayes' Rule. We'll start out general by talking about a dataset X to which we fit a model with a set of parameters θ.

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}$$

The denominator is intractable for our mixture model. We can of course just $argmax_\theta p(X|\theta)$, and that's what the finite mixture model in studenttmixture does – use the EM algorithm to maximize the likelihood. But what if we want the posterior?

We could use MCMC sampling, but MCMC is computationally very expensive for a large dataset. Another alternative is to approximate the intractable $p(\theta|X)$ using a simpler distribution $q(\theta)$ that is as close as we can make it to $p(X|\theta)$ while keeping it tractable. It turns out this approach has some hidden and surprising benefits. One way to make $q(\theta)$ as much like $p(\theta|X)$ as possible is to minimize the Kullback-Leibler or KL divergence between the two, which is:

$$D_{KL}(P||Q) = \int p(\theta|X)log(\frac{p(\theta|X)}{q(\theta)})d\theta$$

$$D_{KL}(Q||P) = \int q(\theta)log(\frac{q(\theta)}{p(\theta|X)})d\theta$$

Notice the divergence is not symmetric! $D_{KL}(P||Q)$ does not equal $D_{KL}(Q||P)$.

Which one to use? If we use $D_{KL}(P||Q)$, any time $p(\theta|X)$ is nonzero and $q(\theta)$ is close to zero, the divergence will blow up to infinity. If we fit our model by minimizing $D_{KL}(P||Q)$, $q(\theta)$ will be forced to spread out, becoming very thin and broad, and cover all the places where $p(\theta|X)$ is nonzero. Not good, because our mixture model has a multimodal posterior. (If you permute the class labels on the components of a fitted mixture model you get the same result, so the posterior distribution for a mixture model will be multimodal.)

$D_{KL}(Q||P)$ by contrast works out nicely. The only requirement to avoid a weird result that blows up to infinity is that $q(\theta)$ be zero everywhere that $p(\theta|X)$ is zero. Minimizing the divergence for $D_{KL}(Q||P)$ will therefore result in an approximate $q(\theta)$ model that has "zoomed in" on one of the modes of the posterior for our true model $p(\theta|X)$ and probably does a reasonably nice job approximating our true model around that mode. We'll get a use-
ful, locally valid approximation! So let's look at how to minimize $D_{KL}(Q||P)$.

Recall from basic probability that $p(\theta|X) = \frac{p(X,\theta)}{p(X)}$, therefore:

$$D_{KL}(Q||P) = \int q(\theta)log\left(\frac{q(\theta)}{p(\theta|X)}\right)d\theta =$$

$$\int q(\theta)log\left(\frac{q(\theta)p(X)}{p(\theta,X)}\right)d\theta =$$

$$\int q(\theta)log\left(\frac{q(\theta)}{p(\theta,X)}\right)d\theta + \int q(\theta)log(p(X))d\theta$$

$$\int q(\theta)log(p(X))d\theta =$$

$$log(p(X))\int q(\theta)d\theta =$$

$$log(p(X))$$

therefore

$$D_{KL}(Q||P) - \int q(\theta)log\left(\frac{q(\theta)}{p(\theta,X)}\right) = log(p(X))$$

Because log(p(X)) is constant (as long as we don't switch out the dataset or model), if we maximize $-\int q(\theta)log\left(\frac{q(\theta)}{p(\theta,X)}\right)$, we minimize $D_{KL}(Q||P)$.So, "all" we have to do is maximize $-\int q(\theta) log\left(\frac{q(\theta)}{p(\theta,X)}\right)$. It turns out it's really convenient if our Q approximation is fully factored, i.e. if every parameter of our approximate model is assumed to be completely independent of every other parameter. This type of variational approximation is called a mean-field approximation.

For notational purposes, we'll say that our approximate distribution $q(\theta)$ is given by:

$$q(\theta) = \prod_{j}^{M} q_j(\theta_j)$$

where we have M parameters for the model, so each parameter j is independent of all others.

Plugging $q(\theta)$ into the term we need to maximize, we get:

$$-\int \prod_{j}^{M} q_j(\theta_j) log\left(\frac{\prod_j^M q_j(\theta_j)}{p(\theta, X)}\right) d\theta =$$

$$\int log(p(\theta, X)) \prod_{j}^{M} q_j(\theta_j) - \sum_{j}^{M} log(q_j(\theta_j)) \prod_{j}^{M} q_j(\theta_j) d\theta$$

To simplify this,
let's distinguish between a θ_n of immediate interest and all the other $\theta_{k\neq n}$, i.e.:

$$\int_{\theta_n} q_n(\theta_n) \int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) \left(log(p(\theta, X)) - \sum_{j}^{M} log(q_j(\theta_j)) \right) d\theta =$$

$$\int_{\theta_n} q_n(\theta_n) \int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) log(p(\theta, X)) d\theta - \int_{\theta_n} q_n(\theta_n) \int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) \sum_{j}^{M} log(q_j(\theta_j)) d\theta$$

This is starting to look a little ugly, and at this point it may be hard to keep track of where we're headed, but don't worry, it will become clear once we reshuffle this a little. To clean this up, let's introduce some additional notation and say that

$$E_{k\neq n}[log(p(\theta,X))] = \int_{k\neq n} \prod_{k\neq n}^{M} q_k(\theta_k)(log(p(\theta,X))d\theta_{k\neq n}$$

In other words, this is the expectation across all variables EXCEPT n. Let's rearrange this a little by pulling the term that involves q_n out of the sum.

$$\int_{\theta_n} q_n(\theta_n) E_{k\neq n}[log(p(\theta,X))]d\theta_n -$$

$$\int_{\theta_n} q_n(\theta_n) log(q_n(\theta_n)) \int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) d\theta -$$

$$\int_{\theta_n} q_n(\theta_n) \int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) \sum_{k\neq n}^{M} log(q_k(\theta_k))d\theta$$

Probability distributions integrate to 1, so the last expression simplifies to:

$$\int_{\theta_n} q_n(\theta_n) E_{k\neq n}[log(p(\theta,X))]d\theta_n - \int_{\theta_n} q_n(\theta_n) log(q_n(\theta_n))d\theta_n -$$

$$\int_{\theta_n} q_n(\theta_n) \int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) \sum_{k\neq n}^{M} log(q_k(\theta_k))d\theta =$$

$$\int_{\theta_n} q_n(\theta_n) \left(E_{k\neq n}[log(p(\theta,X))] - log(q_n(\theta_n))\right) d\theta_n -$$

$$\int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) \sum_{k\neq n}^{M} log(q_k(\theta_k))d\theta$$

Now we're ready to maximize this! But first, we have to enforce a constraint: all $q_j(\theta_j)$ must each integrate to 1. The most obvious way to enforce this constraint is the Lagrange multiplier technique, which here obviously yields:

$$\int_{\theta_n} q_n(\theta_n) \left(E_{k\neq n}[log(p(\theta,X))] - log(q_n(\theta_n))\right) d\theta_n -$$

$$\int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k) \sum_{k\neq n}^{M} log(q_k(\theta_k))d\theta - \sum_{j}^{M} \lambda_j \left(1 - \int q_j(\theta_j)\right)$$

Now we can take the derivative. However, we are taking the derivative with respect to a function, because the expression above is a functional (a function of a function, $q_j(\theta_j)$. The techniques required to do so are found in variational calculus – hence, variational approximations. To save space I won't derive this here, but it can be shown using the Euler-Lagrange equation that:

$$\frac{\delta F}{\delta f(x)} = \frac{\delta L}{\delta f} - \frac{d}{dx}\frac{\delta L}{\delta f'}$$

when:

$$F[f(x)] = \int_a^b L(x, f(x), f'(x))dx$$

So, we just need to plug our last expression into this formula, taking derivatives with respect to each q_n. The derivative of q_n does not appear in our expression, so the second term in Euler-Lagrange, $\frac{d}{dx}\frac{\delta L}{\delta f'}$, goes to zero. To recap: We want to take the derivative of this:

$$\int_{\theta_n} q_n(\theta_n)\left(E_{k\neq n}[log(p(\theta, X))] - log(q_n(\theta_n))\right)d\theta_n -$$

$$\int_{\theta_{k\neq n}} \prod_{k\neq n}^{M} q_k(\theta_k)\sum_{k\neq n}^{M} log(q_k(\theta_k))d\theta - \sum_{j}^{M}\lambda_j\left(1 - \int q_j(\theta_j)\right)$$

with respect to q_n using Euler-Lagrange, which in this case means we need to take the functional derivative of all the terms under the integral signs with respect to q_n. BUT...the second term in that expression doesn't involve q_n at all! Nice, right? So now we have:

$$\frac{\partial}{\partial q_n}\left(q_n(\theta_n)\left(E_{k\neq n}[log(p(\theta, X))] - log(q_n(\theta_n))\right) - \lambda_n q_n(\theta_n)\right) =$$

$$E_{k\neq n}[log(p(\theta, X))] - log(q_n(\theta_n)) - constant$$

Setting this equal to zero obtains:

$$0 = E_{k\neq n}[log(p(\theta, X))] - log(q_n(\theta_n)) - constant$$

$$log(q_n(\theta_n)) = E_{k\neq n}[log(p(\theta, X))] - constant$$

This last one is the key mean-field equation you need to know to derive update equations. The constant is a normalization constant, and if

$e^{E_{k \neq n}[log(p(\theta, X))}$ follows the form of some basic distribution (e.g. a Gaussian), we can often figure out what it is from that. Consequently, mean-field approximations boil down to the following strategy: update each $q_n(\theta_n)$ by taking the expectation across all other parameters, and cycle over your parameters in this way until the algorithm converges. Clearly this is not always as straightforward as it sounds – choosing good starting parameters, for example, can sometimes be a problem, and deriving all of the update equations can be a nuisance – but compared to MCMC, it can be a cheap way to approximate your posterior distribution.

For the derivation of the Student's t-mixture update equations using the mean field formula, see the next section of the docs.