EXPECTATION-MAXIMIZATION WITH LESS ARBITRARINESS

DAN PIPONI

1 INTRODUCTION

There are many introductions to the Expectation-Maximisation algorithm. Unfortunately every one I could find uses arbitrary seeming tricks that seem to be plucked out of a hat by magic. They can all be justified in retrospect, but I find it more useful to learn from reusable techniques that you can apply to further problems. Examples of tricks I've seen used are:

- Using Jensen's inequality. It's easy to find inequalities that apply in any situation. But there are often many ways to apply them. Why apply it to *this* way of writing this expression and not that one which is equal?
- Substituting $1 = A/A$ in the middle of an expression. Again, you can use $1 = A/A$ just about anywhere. Why choose this A at this time? Similarly I found derivations that insert a $B - B$ into an expression.
- Majorisation-Minimisation. This is a great technique, but involves choosing a function that majorises another. There are so many ways to do this, it's hard to imagine any general purpose method that tells you how to narrow down the choice.

My goal is to fill in the details of one key step in the derivation of the EM algorithm in a way that makes it inevitable rather than arbitrary. There's nothing original here, I'm merely expanding on a stackexchange answer http://stats.stackexchange.com/questions/44513/the-relationship-between-expectation-maximization-and-majorization-minimization/59470#59470.

2 GENERALITIES ABOUT EM

The EM algorithm seeks to construct a maximum likelihood estimator (MLE) with a twist: there are some variables in the system that we can't observe.

First assume no hidden variables. We assume there is a vector of parameters $\theta = (\theta_i)$ that defines some model. We make some observations $x = (x_j)$. We have a probability density $P(x|\theta)$ that depends on θ. The likelihood of θ given the observations x is $l(\theta|x) = P(x|\theta)$. The maximum likelihood estimator for θ is the choice of θ that maximises $l(\theta|x)$ for the x we have observed.

Now suppose there are also some variables $z = (z_k)$ that we didn't get to observe. We assume a density $P(x, z|\theta)$. We now have

$$P(x|\theta) = \sum_z P(x, z|\theta)$$

where we sum over all possible values of z. The MLE approach says we now need to maximise

$$l(\theta|x) = \sum_z P(x, z|\theta).$$

One of the things that is a challenge here is that the components of θ might be mixed up among the terms in the sum. If, instead, each term only referred to its own unique block of θ_i, then the maximisation would be easier as we could maximise each term independently of the others. Here's how we might move in that direction. Consider instead the log-likelihood

$$\log l(\theta|x) = \log \sum_z P(x, z|\theta).$$

Now imagine that by magic we could commute the logarithm with the sum. We'd need to maximise

$$\sum_z \log P(x, z|\theta).$$

One reason this would be to our advantage is that $P(x, z|\theta)$ often takes the form $\exp(f(x, z, \theta))$ where f is a simple function to optimise. In addition, f may break up as a sum of terms, each with its own block of θ_i's. Moving the logarithm inside the sum would give us something we could easily maximise term by term. What's more, the $P(x, z|\theta)$ for each z is often a standard probability distribution whose likelihood we already know how to maximise. But, of course, we can't just move that logarithm in.

3 MAXIMISATION BY PROXY

Sometimes a function is too hard to optimise directly. But if we have a guess for an optimum, we can replace our function with a proxy function that approximates it in the neighbourhood of our guess and optimise that instead. That will give us a new guess and we can continue from there. This is the basis of gradient descent. Suppose f is a differentiable function in a neighbourhood of x_0. Then around x_0 we have

$$f(x) \approx f(x_0) + f'(x_0) \cdot x.$$

We can try optimising $f(x_0) + f'(x_0) \cdot x$ with respect to x within a neighbourhood of x_0. If we pick a small circular neighbourhood then the optimal value will be in the direction of steepest descent. (Note that picking a circular neighbourhood is itself a somewhat arbitrary step, but that's another story.) For gradient descent we're choosing $f(x_0) + f'(x_0) \cdot x$ because it matches both the value and derivatives of f at x_0. We could go further and optimise a proxy that shares second derivatives too, and that leads to methods based on Newton-Raphson iteration.

We want our logarithm of a sum to be a sum of logarithms. But instead we'll settle for a proxy function that is a sum of logarithms. We'll make the derivatives of the proxy match those of the original function precisely so we're not making an arbitrary choice.

Write

$$\log l(\theta|x) = \log \sum_z P(x,z|\theta) \approx \sum_z \beta_z \log P(x,z|\theta) + \text{constant}.$$

The β_z are constants we'll determine. We want to match the derivatives on either side of the $\approx$ at $\theta = \theta_0$:

$$\frac{\partial \log l(\theta_0|x)}{\partial\theta_0} = \frac{1}{l(\theta_0|x)}\frac{\partial l(\theta_0|x)}{\partial\theta_0} = \sum_z \boxed{\frac{1}{l(\theta_0|x)}}\frac{\partial P(x,z|\theta_0)}{\partial\theta_0}.$$

On the other hand we have

$$\frac{\partial}{\partial\theta_0}\sum_z \beta_z \log P(x,z|\theta_0) = \sum_z \boxed{\beta_z \frac{1}{P(x,z|\theta_0)}}\frac{\partial P(x,z|\theta_0)}{\partial\theta_0}$$

To achieve equality we want to make the boxed subexpressions match. We choose

$$\beta_z = \frac{P(x,z|\theta_0)}{l(\theta_0|x)} = \frac{P(x,z|\theta_0)}{P(x|\theta_0)} = P(z|x,\theta_0).$$

Our desired proxy function is:

$$\sum_z P(z|x,\theta_0)\log P(x,z|\theta) + \text{const.} = E_{Z|x,\theta_0}(\log P(x,Z|\theta)) + \text{const.}.$$

So the procedure is to take a estimated θ_0 and obtain a new estimate by optimising this proxy function with respect to θ. This is the standard EM algorithm.

It turns out that this proxy has some other useful properties. For example, because of the concavity of the logarithm, the proxy is always smaller than the original likelihood. This means that when we optimise it we never optimise "too far" and that progress optimising the proxy is always progress optimising the original likelihood. But I don't need to say anything about this as it's all part of the standard literature.

4 AFTERWORD

As a side effect we have a general purpose optimisation algorithm that has nothing to do with statistics. If your goal is to compute

$$\arg\max_x \sum_i \exp(f_i(x))$$

you can iterate, at each step computing

$$\arg\max_x \sum_i \exp(f_i(x_0))f_i(x)$$

where x_0 is the previous iteration. If the f_i take a convenient form then this may turn out to be much easier.