Hierarchies and chains

Ben Klemens

9 July 2013

I've shown you how we can transform base models to produce new models, but in relatively simple ways. It gets much more interesting when we have multiple models to combine.

The mixture This transformation takes several models that all have the same data space, and produces a linear combination of the models. I think this is a realistic model, in two senses.

First, there really are multiple explanations for many interesting situations. In a social science context, we might see some pattern among actions people take—more people using some product, or moving in to some location, or buying a stock. A claim that everybody who moves in to a neighborhood does so for the same reason or based on the same factor is *prima facie* false. One approach would be to brainstorm the motivations and write down a model like (moves in) $= \beta_0 + \beta_1$(school quality) $+\beta_2$(proximity to good pick-up bars) $+\beta_3$(motivation measure 3) $+\beta_4$(motivation measure 4) $+\epsilon$. This model (herein, the flat model) posits that every person who moves in has all of the motivations at once, in some fixed proportion. However, the people who are looking for schools for their kids are typically a different set of people from the ones looking for a good time at 2AM.

We could express that the overall measure is the sum of diverse subgroups via a mixture model: the total move-in rate is λ(parent move-ins) + $(1 - \lambda)$(barfly move-ins). The move-in rates for the subpopulations are their own models, which might be another linear combination of elements, like the regression form above. With a sufficiently detailed model and good data, we should be able to pick up that there's one set of βs that put high weight on schools and another that puts high weight on nightlife.

The second sense in which this is realistic is that there is unobservable information. The flat model assumes that everybody is the same type, which means that we don't have to estimate the type for each person moving in. The mixture model doesn't assume away types, which means that now we have the problem of determining which observation comes from which model. For some models, this is easier than for others (maybe we're comfortable placing people based on whether they currently have kids, for example).

My understanding of the standard in the mixture estimation literature is to consider the unobserved type of each observation to be a parameter to be estimated. The estimation is then a two-step, EM-type process:

- Start with an arbitrary initial guess for the parameters of the models.
- Repeat the following until convergence:
  - Assign each observation to the most likely model
  - Re-estimate the parameters of the models using the observations currently in the model.

There are a lot of details in how all this happens, and the reader is welcome to improve the estimation algorithm in `apop_mixture.estimate`, which is currently just OK.

But to bring this section to a conclusion, being able to mix models at least lets us consider the possibility of more descriptive models. The flat model is, to put it flatly, false, and we can make it at least a little less false by combining several submodels. But then we have the engineering problem of estimating the more complex model using information (group membership) that we may not have.

Bayesian updating This form maps from $\mathbb{M}\times\mathbb{M} \to \mathbb{M}$. We have a likelihood model, which is a category regarding the spaces $\mathbb{D}_L$ and $\mathbb{P}_L$, and has a likelihood function $L : \mathbb{D}_L \times \mathbb{P}_L \to \mathbb{R}^+$. If we were certain about the parameters for the model, then that'd be the whole story. If we are uncertain about the parameters, however, we could model them as well, using a prior model, which has its own data and parameter spaces, $\mathbb{D}_P$ and $\mathbb{P}_P$. Because a draw from the prior will (at least conceivably) be used as a parameter for the likelihood model, we need to have $\mathbb{D}_P = \mathbb{P}_L$. Our aggregate model, then, has the same parameter space as the prior model and the same data space as the likelihood model, and the overall likelihood is the integral of the likelihood's likelihood over all possible parameters (weighted by their likelihood as given by the prior model):

$$L'(\mathbf{d}, \mathbf{p}) = \int_{\mathbf{p}} L_{\text{like}}(\mathbf{d}, \rho) dL_{\text{prior}}(\mathbf{d}, \rho)$$

The textbook usage of this form is when the prior model is given. In theory it is derived from prior data, but in practice it is usually selected to facilitate closed-form calculations.

But like the mixture model, the form provides a means of expressing a richer model, for the sake of reducing the number of arbitrary assumptions. Instead of taking the parameters of the prior as given, we can estimate the parameters of the larger model as we would any other model.

Today's code sample almost gets there. It is another round-trip, making draws from one model and using that data set for estimation of another model. It is pretending to be out of a functional textbook, and is a single expression: an estimate of the parameters of a Normal distribution using draws from a posterior distribution generated from a prior of a truncated Normal, a likelihood of a Poisson, and data consisting of 10,000 draws from a mixture of three Poisson distributions, with λ taking on three different values.

[In what I worry will be a recurring theme, those of you playing along at home should download a recent copy of Apophenia to make this example work better.]

```
#include <apop.h>

apop_model truncated_model; //these are from last episode.
apop_model *truncate_model(apop_model *in, double cutoff_in);

int main(){
    apop_model_print (
        apop_estimate(
            apop_update(
                apop_model_draws(
                    apop_model_mixture(
                        apop_model_set_parameters(apop_poisson, 2.8),
                        apop_model_set_parameters(apop_poisson, 2.0),
                        apop_model_set_parameters(apop_poisson, 1.3)
                    ),
                    1e4
                ),
                truncate_model(
                    apop_model_set_parameters(apop_normal, 2, 1),
                    0
                ),
                &apop_poisson
            )->data,
            apop_normal
        )
    , NULL);
}
```

Hierarchical models Bayesian updating joined models by requiring that $\mathbb{D}_P = \mathbb{P}_L$. Conversely, we could also the parameters from the initial model be used as data for the subsequent. For example, we might have z ZIP codes, each with its own data set. We estimate the parameters of the above model for each, and thus wind up with a set of parameters for each neighborhood: $\vec{\beta}^i = \beta_0^i, \ldots, \beta_n^i$ (assuming away any notational problems that a mixture model would generate). Then the sequence of parameters $\vec{\beta}^1, \ldots, \vec{\beta}^z$ is a data set that can itself be modeled, maximum-likelihood estimated, and so on.

Next time, I'll talk about some computational epistemology. The last few posts established a system based on some well-defined models, and then transformations of those models that allow us to describe how base models are transformed and combined to produce complex models. So what are the rules about what can be a base model? Can any old likelihood be the basis of a model?