FAQ August 2011

The key features that slow down the GMM analysis is the combination of mixture modeling, 3 growth factors which leads to 3 dimensions of integration given the categorical outcomes, a large sample size (n=2635), and using more than one latent class variable (one latent and one tx-Knownclass). Here are some suggestions for an analysis strategy, breaking it down into several steps.

As a preliminary step, it is useful to do an LCA – so no growth factors – to gauge the functional form of the development over time. That can be done without covariates.

As a first growth modeling step, it is useful to do an LCGA (latent class growth analysis a la Nagin) as a baseline model. This means no numerical integration. This can be done efficiently with a reduced set of covariates. In the attached Step1 run I use only tx as a covariate so that I have only one latent class variable. I use Starts = 800 200 to make sure to replicate the best LL. This takes only 15 seconds on my 4-processor computer. It gives BIC=9620.

As a second step, I make the intercept random to give one dimension of integration. Again, tx is a covariate so that I have only one latent class variable. I use Starts = 800 200 to make sure to replicate the best LL (it won't necessarily be perfect replication due to the numerical integration). Note that I say Processors = 4(Starts). This takes 3 minutes, 51 seconds, but if I omit Starts in parenthesis the parallelization is less efficient and takes 7 minutes, 23 seconds. This run gives BIC = 9488, so much better than LCGA. Note that I request SVALUES in the Output command for use in the next step.

As a third step, I use the SVALUES from the second step as starting values in a run with Starts=0, with the idea that the solution is not far from that because with mixtures the linear and quadratic slopes seldom have large variances. To be more confident you get the best solution, you can also try Starts not set at zero, perhaps with STSCALE=1 to reduce the magnitude of the perturbation. In this run, I change the SVALUES setup so that the linear slope is random so that I have 2 dimensions of integration. This run takes 6 seconds and gives BIC = 9504.

As a fourth step, I also make the quadratic slope random and get 3 dimensions of integration. This Starts=0 run takes 8 minutes and 17 seconds and gives BIC = 9509.

From this I conclude that the best BIC is for the random intercept only model. I can then fine-tune this model by letting the variances vary across the classes. I can also re-introduce the tx-Knownclass variable and make the model more flexible. And I can add the other covariates to the model, using the starting values from the second step run in a relatively fast Starts=0 run.