Mixed Logit

Chris Conlon

Spring 2023

NYU Stern: MSQE Applied Econometrics

Reading

Today's reading is Chapters 6 from:

Ken Train's Discrete Choice Methods with Simulation

Mixed Logit

We relax the IIA property by mixing over various logits with random effects:

$$
\begin{aligned}
u_{ijt} &= \beta\, x_j + \mu_{ij} + \varepsilon_{ij} \\
s_{ij} &= \int \frac{\beta\, exp[x_j + \mu_{ij}]}{1 + \sum_k \exp[\beta\, x_k + \mu_{ik}]} f(\boldsymbol{\mu_i}|\theta)
\end{aligned}
$$

- Each individual draws a vector $\boldsymbol{\mu_i}$ of μ_{ij} (separately from $\boldsymbol{\varepsilon}$).
- Conditional on $\boldsymbol{\mu_i}$ each person follows an IIA logit model.
- However we integrate (or mix) over many such individuals giving us a mixed logit or heirarchical model (if you are a statistician)
- In practice these are not that different from linear random effects models you have learned about previously.
- It helps to think about fixing $\boldsymbol{\mu_i}$ first and then integrate out over $\boldsymbol{\varepsilon_i}$

Mixed/ Random Coefficients Logit

As an alternative, we could have specified an error components structure on ε_i.

$$U_{ij} = \beta x_{ij} + \underbrace{\nu_i z_{ij} + \varepsilon_{ij}}_{\tilde{\varepsilon}_{ij}}$$

- The key is that ν_i is unobserved and mean zero. But that x_{ij}, z_{ij} are observed per usual and ε_{ij} is IID Type I EV.
- This allows for a heteroskedastic structure on ε_i, but only one which we can project down onto the space of z.

An alternative is to allow for individuals to have random variation in β_i:

$$U_{ij} = \beta_i x_{ij} + \varepsilon_{ij}$$

Which is the random coefficients formulation (these are the same model).

Mixed/ Random Coefficients Logit

- Kinds of heterogeneity
  - We can allow for there to be two types of β_i in the population (high-type, low-type). latent class model.
  - We can allow β_i to follow an independent normal distribution for each component of x_{ij} such as $\beta_i = \overline{\beta} + \nu_i\sigma$.
  - We can allow for correlated normal draws using the Cholesky root of the covariance matrix.
  - Can allow for non-normal distributions too (lognormal, exponential). Why is normal so easy?
- The structure is extremely flexible but at a cost.
- We generally must perform the integration numerically.
- High-dimensional numerical integration is difficult. In fact, integration in dimension 8 or higher makes me very nervous.
- We need to be parsimonious in how many variables have unobservable heterogeneity.
- Again observed heterogeneity does not make life difficult so the more of that the better!

Mixed Logit

How does it work?

- Well we are mixing over individuals who conditional on β_i or μ_i follow logit substitution patterns, however they may differ wildly in their s_{ij} and hence their substitution patterns.
- For example if we are buying cameras: I may care a lot about price, you may care a lot about megapixels, and someone else may care mostly about zoom.
- The basic idea is that we need to explain the heteroskedasticity of $Cov(\varepsilon_i, \varepsilon_j)$ what random coefficients do is let us use a basis from our X's.
- If our X's are able to span the space effectively, then an RC logit model can approximate any arbitrary RUM (McFadden and Train 2002).
- Of course if you have 1000 products and two random coefficients, you are asking for a lot.

Mixed/ Random Coefficients Logit

Suppose there is only one random coefficient, and the others are fixed:

- $f(\beta_i \theta) \sim N(\overline{\beta}, \sigma)$.
- We can re-write this as the integral over a transformed standard normal density

$$s_{ij}(\theta) = \int \frac{e^{V_{ij}(\nu_\iota, \theta)}}{\sum_k e^{V_{ik}(\nu_\iota, \theta)}} f(\nu_\iota) \partial \nu \approx \sum_{\iota=1}^{I} w_\iota \cdot \frac{e^{V_{ij}(\nu_\iota, \theta)}}{\sum_k e^{V_{ik}(\nu_\iota, \theta)}}$$

Numerical Integration: Monte Carlo

How do we choose (w_ι, ν_ι)

$$s_{ij}(\theta) \approx \sum_{\iota=1}^{I} w_\iota \cdot \frac{e^{V_{ij}(\nu_\iota,\theta)}}{\sum_k e^{V_{ik}(\nu_\iota,\theta)}}$$

Monte Carlo Integration: Independent Normal Case

- Draw ν_i from the standard normal distribution.
- Set $w_\iota = \frac{1}{I}$ (the number of draws).
- Now we can rewrite $\beta_\iota = \overline{\beta} + \nu_\iota \sigma$
- For each β_ι calculate $s_{ij}(\beta_\iota)$.
- $\frac{1}{I} \sum_{\iota=1}^{I} s_{ij}(\beta_\iota) = \widehat{s_{ij}}$

Numerical Integration: Multivariate Normal

Suppose instead we want to integrate out over a multivariate normal so that $\nu_i \sim \mathcal{N}(0, \Sigma)$.

- Work with the Cholesky Root of $LL' = \Sigma$
- If ν_i is a k-dimensional standard normal then $L\,\nu_i \sim \mathcal{N}(0, \Sigma)$.
- The vector $\beta_\iota = \overline{\beta} + L\,\nu_\iota$
- Otherwise the same as before, except now we are looking for lower triangle L instead of σ.

Numerical Integration: Gaussian Quadrature

- Quadrature rules give us a set of (w_ι, ν_ι) to approximate

$$s_{ij}(\theta) \approx \sum_{\iota=1}^{I} w_\iota \cdot \frac{e^{V_{ij}(\nu_\iota,\theta)}}{\sum_k e^{V_{ik}(\nu_\iota,\theta)}}$$

  to a high degree of accuracy with a small number of nodes.
- These points are chosen to approximate the function to a polynomial order and then integrate the polynomial exactly.
- ie: approximate with 13th order polynomial.
- For smooth, bounded, and continuously differentiable functions, these work really well!
- We will use Gauss Hermite rules on the homework.

Quadrature in higher dimensions

- Quadrature is great in low dimensions – but scales badly in high dimensions.
- If we need N_a points to accurately approximate the integral in $d = 1$ then we need N_a^d points in dimension d (using the tensor product of quadrature rules).
- There is some research on quadrature rules that nest and also how to carefully eliminate points so that the number doesn't grow so quickly.
- Try `http://sparse-grids.de`

Estimation

How do we actually estimate these models?

- In practice we should be able to do MLE.

$$\max_{\theta} \sum_{i=1}^{N} y_{ij} \log s_{ij}(\theta)$$

- When we are doing IIA logit, this problem is globally convex and is easy to estimate using Newton's Method.
- When doing nested logit or random coefficients logit, it generally is non-convex which can make life difficult.
- The tough part is generally working out what $\frac{\partial \log s_{ij}}{\partial \theta}$ is, especially when we need to simulate to obtain s_{ij}.
- It turns out that MSLE actually has consistent problems for fixed S. Why?
- Alternative? MSM/MoM type estimators

Mixed Logit: Estimation

- Just like before, we do MLE
- One wrinkle–how do we compute the integral?

$$s_{ij} = \int \frac{\exp[x_j\beta_i]}{1+\sum_k \exp[x_k\beta_i]} f(\beta_i|\theta) \approx \sum_{s=1}^{ns} w_s \frac{\exp[x_j(\overline{\beta}+\Sigma\nu_{is})]}{1+\sum_k \exp[x_k(\overline{\beta}+\Sigma\nu_{is})]}$$

- Option 1: Monte Carlo integration. Draw $NS = 1000$ or so samples of ν_i from the standard normal and set $w_i = \frac{1}{NS}$.
- Option 2: Quadrature. Choose ν_i and w_i according to a Gaussian quadrature rule. Like `quad` in MATLAB or `mvquad` in R or `scipy.integrate.quadrature` in `Python`.
- Personally I get nervous about integrals in dimension greater than 5. People routinely have 20 or more though.

Mixed Logit: Hints

How bad is the simulation error?

- Depends how small your shares are.
- Since you care about $\log s_{jt}$ when shares are small, tiny errors can be enormous.
- Often it is pretty bad.
- I recommend sticking with quadrature at a high level of precision.
- `sparse-grids.de` provide efficient high dimensional quadrature rules.

A Semiparametric Estimator

Even More Flexibility (Fox, Kim, Ryan, Bajari)

Suppose we wanted to nonparametrically estimate $f(\beta_i|\theta)$ instead of assuming that it is normal or log-normal.

$$s_{ij} = \int \frac{\exp[x_j\beta_i]}{1+\sum_k \exp[x_k\beta_i]} f(\beta_i|\theta)$$

- Choose a distribution $g(\beta_i)$ that is more spread out that $f(\beta_i|\theta)$
- Draw several β_s from that distribution (maybe 500-1000).
- Compute $\hat{s}_{ij}(\beta_s)$ for each draw of β_s and each j.
- Holding $\hat{s}_{ij}(\beta_s)$ fixed, look for w_s that solve

$$\min_w \left(s_j - \sum_{s=1}^{ns} w_s \hat{s}_{ij}(\beta_s) \right)^2 \quad \text{s.t.} \sum_{s=1}^{ns} w_s = 1, \quad w_s \geq 0 \quad \forall s$$

Even More Flexibility (Fox, Kim, Ryan, Bajari)

- Like other semi-/non- parametric estimators, when it works it is both general and very easy.
- We are solving a least squares problem with constraints: positive coefficients, coefficients sum to 1.
- It tends to produce sparse models with only a small number of β_s getting positive weights.
- This is way easier than solving a random coefficients logit model with all but the simplest distributions.
- There is a bias-variance tradeoff in choosing $g(\beta_i)$.
- Incorporating parameters that are not random coefficients loses some of the simplicity.
- I have no idea how to do this with large numbers of fixed effects.

Thanks