Multinomial Discrete Choice: Beyond Logit

Chris Conlon

April 17, 2020

Applied Econometrics II

Multinomial Logit: IIA

The multinomial logit is frequently criticized for producing unrealistic substitution patterns

- Suppose we got rid of a product k then $s_j^{(1)} = s_j^{(0)} \frac{1}{1-s_k}$.
- Substitution is just proportional to your pre-existing shares s_j
- No concept of "closeness" of competition!

Can we do better?

Multinomial Probit?

- The probit has $\varepsilon_i \sim N(0, \Sigma)$.
- If Σ is unrestricted, then this can produce relatively flexible substitution patterns.
- Flexible is relative: still have normal tails, only pairwise correlations, etc.
- It might be that ρ_{12} is large if $1, 2$ are similar products.
- Much more flexible than Logit

Downside

- Σ has potentially J^2 parameters (that is a lot)!
- Maybe $J * (J - 1)/2$ under symmetry. (still a lot).
- Each time we want to compute $s_j(\theta)$ we have to simulate an integral of dimension J.
- I wouldn't do this for $J \geq 5$.

Relaxing IIA

Let's make ε_{ij} more flexible than IID. Hopefully still have our integrals work out.

$$u_{ij} = x_{ij}\beta + \varepsilon_{ij}$$

- One approach is to allow for a block structure on ε_{ij} (and consequently on the elasticities).
- We assign products into groups g and add a group specific error term

$$u_{ij} = x_{ij}\beta + \eta_g + \varepsilon_{ij}$$

- The trick putting a distribution on $\eta_g + \varepsilon_{ij}$ so that the integrals still work out.
- Do not try this at home: it turns out the required distribution is known as GEV and the resulting model is known as the nested logit.

Nested Logit

A traditional (and simple) relaxation of the IIA property is the Nested Logit. This model is often presented as two sequential decisions.

- First consumers choose a category (following an IIA logit).
- Within a category consumers make a second decision (following the IIA logit).
- This leads to a situation where while choices within the same nest follow the IIA property (do not depend on attributes of other alternatives) choices among different nests do not!

Alternative Interpretation

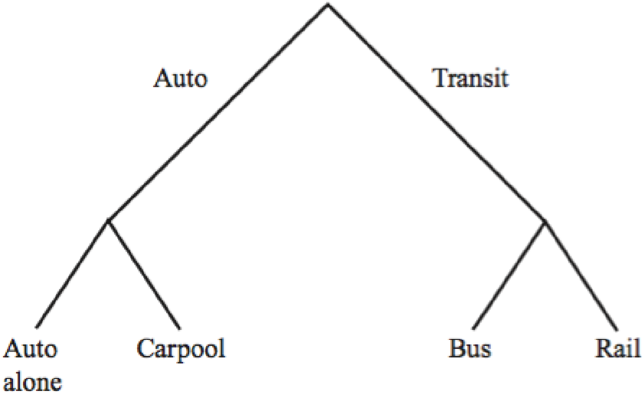

Figure 4.1. Tree diagram for mode choice.

Nested Logit

Utility looks basically the same as before:

$$U_{ij} = V_{ij} + \underbrace{\eta_{ig} + \widetilde{\varepsilon_{ij}}}_{\varepsilon_{ij}(\lambda_g)}$$

- We add a new term that depends on the group g but not the product j and think about it as varying unobservably over individuals i just like ε_{ij}.
- Now $\varepsilon_i \sim F(\varepsilon)$ where $F(\varepsilon) = \exp[-\sum_{g=G}^{G} \left(\sum_{j \in J_g} \exp[-\varepsilon_{ij}/\lambda_g]\right)^{\lambda_g}$. This is no longer Type I EV but GEV.
- The key is the addition of the λ_g parameters which govern (roughly) the within group correlation.
- This distribution is a bit cooked up to get a closed form result, but for $\lambda_g \in [0, 1]$ for all g it is consistent with random utility maximization.

Nested Logit

The nested logit choice probabilities are:

$$s_{ij} = \frac{e^{V_{ij}/\lambda_g} \left(\sum_{k \in J_g} e^{V_{ik}/\lambda_g}\right)^{\lambda_g - 1}}{\sum_{h=1}^{G} \left(\sum_{k \in J_h} e^{V_{ik}/\lambda_h}\right)^{\lambda_h}}$$

Within the same group g we have IIA and proportional substitution

$$\frac{s_{ij}}{s_{ik}} = \frac{e^{V_{ij}/\lambda_g}}{e^{V_{ik}/\lambda_g}}$$

But for different groups we do not:

$$s_{ij} = \frac{e^{V_{ij}/\lambda_g} \left(\sum_{k \in J_g} e^{V_{ik}/\lambda_g}\right)^{\lambda_g - 1}}{e^{V_{ik}/\lambda_h} \left(\sum_{k \in J_h} e^{V_{ik}/\lambda_h}\right)^{\lambda_h - 1}}$$

Nested Logit

We can take the probabilities and re-write them slightly with the substitution that $\log\left(\sum_{k\in J_g} e^{V_{ik}}\right) \equiv IV_{ig}$:

$$\begin{aligned} s_{ij} &= \frac{e^{V_{ij}/\lambda_g}}{\left(\sum_{k\in J_g} e^{V_{ik}/\lambda_g}\right)} \cdot \frac{\left(\sum_{k\in J_g} e^{V_{ik}/\lambda_g}\right)^{\lambda_g}}{\sum_{h=1}^{G}\left(\sum_{k\in J_h} e^{V_{ik}/\lambda_h}\right)^{\lambda_h}} \\ &= \underbrace{\frac{e^{V_{ij}/\lambda_g}}{\left(\sum_{k\in J_g} e^{V_{ik}/\lambda_g}\right)}}_{s_{ij|g}} \cdot \underbrace{\frac{e^{\lambda_g IV_{ig}}}{\sum_{h=1}^{G} e^{\lambda_h IV_{ih}}}}_{s_{ig}} \end{aligned}$$

This is the decomposition into two logits that leads to the "sequential logit" story.

Nested Logit : Notes

- $\lambda_g = 1$ is the simple logit case (IIA)
- $\lambda_g \to 0$ implies that all consumers stay within the nest.
- $\lambda < 0$ or $\lambda > 1$ can happen and usually means something is wrong. These models are not generally consistent with RUM. (If you report one in your paper I will reject it).
- λ is often interpreted as a correlation parameter and this is almost true but not exactly!
- There are other extensions: overlapping nests, or three level nested logit.
- In general the hard part is understanding what the appropriate nesting structure is ex ante. Maybe for some problems this is obvious but for many not.

Nested Logit

In practice we end up with the following:

$$s_{ij} = s_{ij|g}(\theta) s_{ig}(\theta)$$

- Because the nested logit can be written as the within group share $s_{ij|g}$ and the share of the group s_{ig} we often explain this model as sequential choice
- First you pick a category, then you pick a product within a category.
- This is a sometimes helpful (sometimes unhelpful) way to think about this.
- We can also think about this imposing a block structure on the covariance matrix of ε_i
- You need to assign products to categories before you estimate and you can't make mistakes!

Nested Logit

How does it actually look?

$$
\begin{aligned}
IV_{ig}(\theta) &= \log\left(\sum_{k\in G}\exp[x_k\beta/(1-\lambda_g)]\right) = E_\varepsilon[\max_{j\in G} u_{ij}] \\
s_{ij|g}(\theta) &= \frac{\exp[x_j\beta/(1-\lambda_g)]}{\sum_{k\in G}\exp[x_k\beta/(1-\lambda_g)]} \\
s_{ig}(\theta) &= \frac{\exp[IV_{ig}]^{1-\lambda_g}}{\sum_h \exp[IV_{ih}]^{1-\lambda_h}}
\end{aligned}
$$

Nested Logit

How does it actually look?

$$\log\left(\frac{s_{ij|g}(\theta)}{s_{ik|g}(\theta)}\right) = (x_j - x_k)\cdot\frac{\beta}{1-\lambda_g}$$

- We are back to having the IIA property but now within the group G.
- We also have IIA across groups g, h
- λ_g and α govern the elasticities, which also have a block structure.
- Sometimes people refer to this as the product of two logits
- In the old days people used to estimate by fitting sequential IIA logit models – this is consistent but inefficient – you shouldn't do this today!
- Estimation happens via MLE. This can be tricky because the model is non-convex. It helps to substitute $\tilde{\beta} = \beta/(1-\lambda_g)$

Parametric Identification

Look at derivatives:

$$\begin{aligned}
\frac{\partial\, s_{j|g}}{\partial X_j} &= \beta_1 s_{j|g}(1-s_{j|g}) \\
\frac{\partial\, s_g}{\partial X} &= (1-\lambda)\beta_1 s_g(1-s_g) \\
\frac{\partial\, s_g}{\partial J} &= \frac{1-\lambda}{J} s_g(1-s_g)
\end{aligned}$$

- We get β by changing x_j within group
- We get nesting parameter λ by varying X
- We don't have any parameters left to explain changing number of products J.

Nested Logit

There are more potential generalizations though they are less frequently used:

- You can have multiple levels of nesting: first I select a size car (compact, mid-sized, full-sized) then I select a manufacturer, finally a car.
- You can have potentially overlapping nests: Yogurt brands are one nest, Yogurt flavors are a second nest. This way strawberry competes with strawberry and/or Dannon substitutes for Dannon.

Mixed Logit

We relax the IIA property by mixing over various logits:

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

- Each individual draws a vector μ_i of μ_{ij} (separately from ε).
- Conditional on μ_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 μ_i first and then integrating out over ε_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

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

$$s_{ij}(\theta) = \int \frac{e^{V_{ij}(\nu_i,\theta)}}{\sum_k e^{V_{ik}(\nu_i,\theta)}} f(\nu_i) \partial \nu$$

- Monte Carlo Integration: Independent Normal Case
  - Draw ν_i from the standard normal distribution.
  - Now we can rewrite $\beta_i = \overline{\beta} + \nu_i \sigma$
  - For each β_i calculate $s_{ij}(\beta_i)$.
  - $\frac{1}{S} \sum_{s=1}^{S} s_{ij} = \widehat{s_j^s}$
- Gaussian Quadrature
  - Or we can draw a non-random set of points ν_i and corresponding weights w_i and approximate the integral to a high level of polynomial accuracy.

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 `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 (next time).

Mixed Logit: Estimation

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

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

- 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.
- 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.

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.

Convexity and Computation

Convexity

An optimization problem is convex if

$$\min_{x} f(\mathbf{x}) \quad s.t. \quad h(\mathbf{x}) \leq 0 \quad A\mathbf{x} = 0$$

- $f(\mathbf{x}), h(\mathbf{x})$ are convex (PSD second derivative matrix)
- Equality Constraint is affine

Some helpful identities about convexity

- Compositions and sums of convex functions are convex.
- Norms $||$ are convex, $\max$ is convex, $\log$ is convex
- $\log(\sum_{i=1}^{n} \exp(x_i))$ is convex.
- Fixed Points can introduce non-convexities.
- Globally convex problems have a unique optimum

Properties of Convex Optimization

- If a program is globally convex then it has a unique minimizer that will be found by convex optimizers.
- If a program is not globally convex, but is convex over a region of the parameter space, then most convex optimization routines find any local minima in the convex hull
- Convex optimization routines are unlikely to find local minima (including the global minimum) if they do not begin in the same convex hull as the optimum (starting values matter!).
- Most good commercial routines are clever about dealing with multiple starting values and handling problems that are well approximated by convex functions.
- Good Routines use information about sparseness of Hessian – this generally determines speed.

Nested Logit Model

FIML Nested Logit Model is Non-Convex

$$\min_{\theta} \sum_j q_j \ln P_j(\theta) \quad \text{s.t.} \quad P_j(\theta) = \frac{e^{x_j \beta/\lambda} (\sum_{k \in g_l} e^{x_j \beta/\lambda})^{\lambda - 1}}{\sum_{\forall l'} (\sum_{k \in g_l'} e^{x_j \beta/\lambda})^{\lambda}}$$

This is a pain to show but the problem is with the cross term $\frac{\partial^2 P_j}{\partial \beta \partial \lambda}$ because $\exp[x_j \beta/\lambda]$ is not convex.

A Simple Substitution Saves the Day: let $\gamma = \beta/\lambda$

$$\min_{\theta} \sum_j q_j \ln P_j(\theta) \quad \text{s.t.} \quad P_j(\theta) = \frac{e^{x_j \gamma} (\sum_{k \in g_l} e^{x_j \gamma})^{\lambda - 1}}{\sum_{\forall l'} (\sum_{k \in g_l'} e^{x_j \gamma})^{\lambda}}$$

This is much better behaved and easier to optimize.

Nested Logit Model

	Original[1]	Substitution[2]	No Derivatives[3]
Parameters	49	49	49
Nonlinear λ	5	5	5
Likelihood	2.279448	2.279448	2.27972
Iterations	197	146	352
Time	59.0 s	10.7 s	192s

Discuss Nelder-Meade

Computing Derivatives

A key aspect of any optimization problem is going to be computing the derivatives (first and second) of the model. There are some different approaches

- Numerical: Often inaccurate and error prone (why?)
- Pencil and Paper: this tends to be mistake prone – but often actually the fastest
- Automatic (AMPL): Software brute forces through a chain rule calculation at every step (limited language).
- Symbolic (Maple/Mathematica): software "knows" derivatives of certain objects and can do its own simplification. (limited language).