Lecture 17: Simulation-Based Estimation I

ResEcon 703: Topics in Advanced Econometrics

Matt Woerman
University of Massachusetts Amherst

Agenda

Last time

- Mixed Logit Model Example in R

Today

- Simulated Choice Probabilities
- Simulation-Based Estimators

Upcoming

- Reading for next time
  - Optional: Handel (2013)
- Problem sets
  - Problem Set 4 is posted, due November 21

Mixed Logit Model

Mixed logit choice probability

$$P_{ni} = \int \frac{e^{\beta' x_{ni}}}{\sum_{j=1}^{J} e^{\beta' x_{nj}}} f(\beta \mid \theta) d\beta$$

- This integral does not have a closed-form expression
- We cannot estimate a mixed logit model using MLE, as we did for logit and nested logit
- Instead, we approximate choice probabilities through simulation and estimate the maximum simulated likelihood estimator (MSLE)

How do we simulate mixed logit choice probabilities?

What is maximum simulated likelihood estimation?

Simulated Choice Probabilities

Simulated Choice Probabilities

The mixed logit choice probability is

$$P_{ni} = \int L_{ni}(\beta) f(\beta \mid \theta) d\beta$$

where $L_{ni}(\beta)$ is the logit choice probability at a given set of coefficients, β,

$$L_{ni}(\beta) = \frac{e^{\beta' x_{ni}}}{\sum_{j=1}^{J} e^{\beta' x_{nj}}}$$

To simulate this choice probability for a given set of parameters, θ,

1. Draw a set of coefficients, β^r, from the density $f(\beta \mid \theta)$
2. Calculate the conditional probability, $L_{ni}(\beta^r)$, for each alternative
3. Repeat steps 1 and 2 for a total of R draws from $f(\beta \mid \theta)$
4. Average over these R draws to get $\check{P}_{ni}$ for each alternative

$$\check{P}_{ni} = \frac{1}{R} \sum_{r=1}^{R} L_{ni}(\beta^r)$$

Some Simulation Technicalities

For a given set of coefficient density parameters, θ, perform the steps on the previous slide for every alternative for each decision maker

- For a given decision maker, use the same set of β^r draws to simulate the choice probability for every alternative
  - That is, for each draw of β^r, calculate the full set of J choice probabilities (one for each alternative), and then average each choice probability over the same R draws of β^r
- Use a different set of β^r draws for each decision maker in order to maintain independence over decision makers
  - That is, we need N different sets of R draws of β^r

Simulated Choice Probabilities and Numerical Optimization

We use these simulated choice probabilities within a numerical optimization algorithm to find the set of parameters, $\hat{\theta}$, that are our estimators

- We want to use the same set of "draws" for a given decision maker throughout the numerical optimization algorithm
- If we use different "draws" for each iteration of the algorithm, we introduce additional noise that impedes convergence

In order to avoid this noise

1. Draw many $(K \times N \times R)$ random variables from a standard normal distribution before starting the optimization algorithm
2. Transform this same set of standard normal random variables in each iteration of the optimization algorithm to represent $f(\beta \mid \theta)$ for the set of parameters, θ, of each iteration

Transforming a Standard Normal Random Variable

We can transform a standard normal random variable (or a vector of standard normals) into many other distributions

1. Draw K standard normal random variables, $\eta \sim N(0,1)$, where K is the number of random parameters
2. Transform these standard normals into the desired distributions
    - Normal: $\varepsilon = b + s\eta$ gives $\varepsilon \sim N(b, s^2)$
    - Log-normal: $\varepsilon = e^{b+s\eta}$ gives $\ln \varepsilon \sim N(b, s^2)$
    - Multivariate normal: $\varepsilon = b + L\eta$ gives $\varepsilon \sim N(b, \Omega)$ where ε, b, and η are each a vector of length equal to the number of multivariate normal random variables, Ω is the variance-covariance matrix of these variables, and L is the Choleski factor of Ω
    - Comparable transformations exist for most distributions

See Chapter 9 in the Train textbook for more on drawing and transforming random variables

Simulated-Based Estimators

Simulated-Based Estimation

Simulation-based estimators are roughly equivalent to their traditional analogs

- We replace terms that are difficult or impossible to calculate with their simulated counterparts
  - Example: Mixed logit choice probabilities include an integral and do not have a closed-form expression, so we replace them with simulated choice probabilities
- Simulation can potentially introduce bias or noise into the estimation, which we need to consider when using simulation-based estimators

Maximum Simulated Likelihood Estimation

Maximum simulated likelihood estimation (MSLE), or simulated maximum likelihood estimation (SMLE), is the simulation analog of maximum likelihood estimation (MLE)

MSLE is the estimator that maximizes the simulated log-likelihood

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \ln \check{L}(\theta \mid y)$$

where $\ln \check{L}(\theta \mid y)$ is the log of the simulated likelihood

$$\ln \check{L}(\theta \mid y) = \sum_{n=1}^{N} \ln \check{f}(y_n \mid \theta)$$

and $\check{f}(y_n \mid \theta)$ is a simulated density function

MSLE with Discrete Choice

For discrete choice applications, the log of simulated likelihood is a function of simulated choice probabilities

$$\ln \check{L}(\theta \mid y) = \sum_{n=1}^{N} \sum_{j=1}^{J} y_{nj} \ln \check{P}_{nj}(\theta)$$

so the MSLE estimator is

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{n=1}^{N} \sum_{j=1}^{J} y_{nj} \ln \check{P}_{nj}(\theta)$$

which yields the first-order condition

$$0 = \frac{\partial \ln \check{P}_n(\hat{\theta})}{\partial \theta}$$

where $\check{P}_n(\theta)$ is the simulated choice probability of the chosen alternative

Method of Simulated Moments

Method of simulated moments (MSM), or simulated method of moments (SMM), is the simulation analog of generalized method of moments (GMM)

MSM is the estimator that "solves" simulated moments

$$\frac{1}{N}\sum_{n=1}^{N}\check{m}(w_n,\hat{\theta})=0$$

where $\check{m}(w_n,\hat{\theta})$ are the simulated empirical analogs of population moments that satisfy

$$E[m(w_n,\theta^*)]=0$$

With more moments than parameters, we cannot actually solve the first equation, so we minimize a weighted function of the moments

MSM with Discrete Choice

For discrete choice applications, the population moments result from the model residuals being uncorrelated with a set of exogenous instruments

$$E\left[\sum_{j=1}^{J}[y_{nj} - P_{nj}(\theta^*)]z_{nj}\right] = 0$$

so the MSM estimator "solves" the simulated empirical analogs

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{J}[y_{nj} - \check{P}_{nj}(\hat{\theta})]z_{nj} = 0$$

Steps for Simulation-Based Estimation

1. Draw $K \times N \times R$ standard normal random variables
   - K random coefficients for each of
   - N different decision makers for each of
   - R different simulation draws
2. Find the set of parameters that maximizes or minimizes the objective function of a simulation-based estimator
   1. Start with some set of parameters, θ^0
   2. Simulate choice probabilities for this set of parameters, $\check{P}_{ni}(\theta^t)$
      1. Transform each set of K standard normals using θ^t to get a set of β_n^r
      2. Calculate the choice probabilities for each individual and draw, $L_{ni}(\beta_n^r)$
      3. Average over all R simulation draws to get $\check{P}_{ni}(\theta^t)$
   3. Use these simulated choice probabilities to calculate simulated log-likelihood, simulated moments, etc.
   4. Step to a better set of parameters, θ^{t+1}
   5. Repeat steps 2 and 3 until the algorithm converges to a set of parameters that is your simulation-based estimator

Traditional Estimators

Traditional estimators are the set of parameters that solve a specific function

$$g(\hat{\theta}) = \frac{1}{N}\sum_{n=1}^{N} g_n(\hat{\theta}) = 0$$

Maximum likelihood estimation of a discrete choice model

$$g_n(\theta) = \frac{\partial \ln P_n(\theta)}{\partial \theta}$$

Generalized method of moments estimation of a discrete choice model

$$g_n(\theta) = \sum_{j=1}^{J} (y_{nj} - P_{nj}) z_{nj}$$

When the some assumptions are met, these estimators yield consistent estimates of the true set of parameters, θ^*

Simulation-Based Estimators

Simulation-based estimators are the set of parameters that solve a specific function

$$\check{g}(\hat{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \check{g}_n(\hat{\theta}) = 0$$

where $\check{g}_n(\theta)$ is the simulation-based analog of $g_n(\theta)$

We can compare this function for the simulation-based and traditional estimators by writing

$$\begin{aligned}
\check{g}(\theta) &= \check{g}(\theta) + \{g(\theta) - g(\theta)\} + \{E_r[\check{g}(\theta)] - E_r[\check{g}(\theta)]\} \\
&= g(\theta) + \{E_r[\check{g}(\theta)] - g(\theta)\} + \{\check{g}(\theta) - E_r[\check{g}(\theta)]\}
\end{aligned}$$

A simulation-based estimator can differ from a traditional estimator for two reasons

- Simulation bias: $E_r[\check{g}(\theta)] - g(\theta)$
- Simulation noise: $\check{g}(\theta) - E_r[\check{g}(\theta)]$

Simulation Bias and Noise

How do we reduce the simulation bias and noise in a simulation-based estimator?

- Increase the sample size, N
- Increase the number of simulation draws, R

With sufficient sample size and simulation draws, a simulation-based estimator is:

- Consistent
- Asymptotically normal
- Sometimes equivalent (or converging) to its traditional analog

The specifics depend on the estimator in question

- See Chapter 10 in the Train textbook for details

Announcements

Reading for next time

- Optional: Handel (2013)

Office hours

- Reminder: 2:00–3:00 on Tuesdays in 218 Stockbridge

Upcoming

- Problem Set 4 is posted, due November 21