Problem Set: Single Agent Dynamic Discrete-Choice Models
Spring 2009

Due date: April 29th

In this problem set, we explore computation and estimation of single-agent dynamic discrete-choice models, with an emphasis on the Harold Zurcher model.[1]

The four exercises build progressively on each other.

1. Compute the Harold Zurcher model
   - Use the parameter estimates of θ from the top of table X in Rust's (1987) paper. (Use parameters for the case where $\beta = 0.9999$ and for bus groups 1,2,3.)
   - Compute $EV(x, i; \theta)$ using the value iteration procedure, described in the Rust paper and lecture notes.
   - Graph $EV(x, i; \theta)$ separately for $i = 0, 1$.
2. Simulate the Harold Zurcher model
   - Assume there are $N = 50$ homogenous buses, and you observe each for $T = 52$ weeks. HZ makes a replacement decision every week.
   - Initial values: take $x_{n0} = 0, i_{n0} = 0$ for all buses n.
   - For each week $t = 1, \ldots, 52$, simulate the utility shocks $\epsilon_{0nt}, \epsilon_{1nt}$, the mileage x_{nt} and the replacement decision i_{nt}:
     - Draw $\epsilon_{0nt}, \epsilon_{1nt}$, independently from type 1 extreme value distribution, with CDF $F(\epsilon) = \exp[-\exp[-(\epsilon - 0.577)]]$.[2]
     - Draw mileage x_{nt} from transition $G(x|x_{n,t-1}, i_{n,t-1})$, which is the multinomial given in the Rust paper.

[1] This homework was developed by Matt Shum.

[2] To simulate form a desired CDF $F(x)$, draw uniform random variables $u \sim U[0, 1]$, and transform $x = F^{-1}(u)$.

- Compute the replacement decision

$$i_{n,t} \equiv \text{argmax}_{i=0,1}\big(u(x_{nt}, i; \theta) + \epsilon_{int} + \beta EV(x_{nt}, i; \theta)\big) \tag{1}$$

where you use $EV(x_{nt}, i; \theta)$ as computed in problem 1 above.

- After sequences of (x, i) are simulated for all buses, provide summary statistics of your simulated data.

3. Estimate the model using Rust's MLE/nested-fixed-point algorithm using your simulated data.

4. Estimate the model using the indirect Hotz-Miller method using your simulated data.