TA Session 3
Dynamic Discrete Choice: Full Solution

Microeconometrics II with Joan Llull

IDEA, Fall 2024

TA: Conghan Zheng

November 25, 2024

A Renewal Problem[1]

- The choice: $j \in \{0, 1\}$
  - The maintenance manager decides whether to replace the existing bus engine $(d_{t1} = 1)$ in period t or keep it for at least one more period $(d_{t0} = 1)$.
  - The tradeoff: engine replacement is costly, but the maintenance costs also increase with use.
- The state vector: (x_t, ε_t), revealed to the individual at the beginning of t
  - x_t: the accumulated milage since last replacement.
  - If the manager keeps the engine, the bus mileage x_t advances; if the manager replaces the engine, $x_t = 0$.
  - *Renewal*: once the bus engine is replaced it is assumed to be "as good as new" so the state of the system regenerates to the state $x_t = 0$.
  - Buses may also be differentiated by a fixed characteristics included in x.
  - The unobservable states (choice-specific shocks) $\varepsilon_t = \{\varepsilon_{t0}, \varepsilon_{t1}\}$ are i.i.d. bivariate Type I extreme value distributed (T1EV).

[1] In this slides, I'm not using the same notation as in Joan's lecture notes or as in the problem set statement, to avoid giving you the exact solution in this tutorial. In your own solution, please follow the problem set notation.

Dynamics

- To free us from many computational challenges, we assume *conditional independence*: $F(x_{t+1}, \varepsilon_{t+1} | d_t, x_t, \varepsilon_t) = F_x(x_{t+1} | d_t, x_t) F_\varepsilon(\varepsilon_{t+1})$.
- The loglikelihood function can be factorized as:

$$\ln \mathcal{L} = \sum_i \sum_t^{T-1} \sum_j d_{itj} \ln \underbrace{F_{tj}(x_{i,t+1} | x_{it}; \theta)}_{\text{transition}} + \sum_i \sum_t^{T} \sum_j d_{itj} \ln \underbrace{p_{tj}(x_t; \theta)}_{\text{choice prob.}}$$

- The only essential difference between a estimating a static discrete choice model using MLE and a estimating a dynamic model satisfying conditional independence using MLE is that parameterizations of v based on u do not have a closed form, but must be computed numerically.
- Estimating $F_{tj}(x_{t+1} | x_t)$: It's identified for each (t, j) from the transitions, so there is no conceptual reason for parameterizing this distribution.

State Space

The state space should be defined compatible to the transition matrix

- Theoretically, for n states in space, you have a $n \times n$ transition matrix for each (j, t), while you can average this matrix over t. And our case becomes even simpler because we only have two actions.
- A natural way to discretize the state space in the bus engine replacement problem is to divide mileage into equally sized ranges.
- In the Rust's example, there are 90 bins of length 5,000, with a upper bound of 450,000[2] (then this upper bound will be the absorbing state).
- The transition probability with this discretization becomes a 90 × 90 Markov transition matrix.
- Do we then need to estimate $90 \times 90 = 8{,}100$ parameters?

[2] In the data set actually no bus ever got more than 400,000 miles.

State Transitions

- Consider a trinomial distribution on the set $\{0, 1, 2\}$, corresponding to the mileage ranges $[0, 5000]$, $[5000, 10000]$, and $[10000, \infty]$, respectively.
- And we define

$$\varphi_j = \Pr(x_{t+1} = x_t + j \mid x_t, d_t = 0), \quad j \in \{0, 1, 2\}$$

Then the 90 × 90 discrete transition probability matrix for the uncontrolled Markov process (i.e. assuming replacement never occurs) is given by:

$$F^0_{x_t, x_{t+1}} = \begin{bmatrix} \varphi_0 & \varphi_1 & \varphi_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & \varphi_0 & \varphi_1 & \varphi_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \varphi_0 & \varphi_1 & \varphi_2 \\ 0 & 0 & 0 & 0 & \dots & 0 & \varphi_0 & 1 - \varphi_0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

- Whereas $F^1_{x_t, x_{t+1}}$ has only one element 1: if the replacement action is taken, then $x_t = 0$.
- Then, for the entire transition distribution, you need to estimate three numbers: φ_0, φ_1, and φ_2.

Estimation

Transitions

- A consistent estimator of $F^0_{x_t,x_{t+1}}$ can be obtained from the proportion of observations in the $(t, j = 0, x_t)$ cell transitioning to x_t, $x_t + 1$, and $x_t + 2$.
- For example, part of the code in MATLAB:

```
% Among all obs that did not undergo replacement in the last period (TTT-
    repl)
% Get transition probs by summing all 3 possible transitions and
% Dividing by the total no of obs
phi0 = sum(sum(d0))/(TTT-repl);
phi1 = sum(sum(d1))/(TTT-repl);
phi2 = sum(sum(d2))/(TTT-repl);
phi = [phi0 phi1 phi2];

% Create the uncontrolled transition matrix
F0 = zeros(90,90);
for i = 1:90
    if i < 89
        F0(i,i:i+2) = phi;
    elseif i == 89
        F0(i,i:i+1) = [phi0 1-phi0];
    else
        F0(i,i) = 1;
    end
end
```

Value Function

- The manager chooses $\{d_{jt}\}_{j,t}$ to maximize the expected discounted sum of payoffs in the first period:

$$\mathbb{E}\left\{\sum_{t=1}^{T}\sum_{j=1}^{J}\beta^{t-1}d_{jt}\cdot\left[u_{jt}(x_t)+\varepsilon_{jt}\right]\middle|x_1\right\}$$

- Formally the subjective discount factor β is redundant if the current period payoff u is subscripted by t, we typically include a geometric discount factor so that the infinite sums of utility are bounded and the optimization is well posed.
- For a stationary infinite horizon problem, age and time have no role.
- Denote by $d_t^O(x_t,\varepsilon_t)$ the optimal decision rule at t which has J elements: $d_{jt}^O(x_t,\varepsilon_t), j=1,\ldots,J$.
- The value function in period t, conditional on behaving according to the optimal decision rule:

$$V_t(x_t)\equiv\mathbb{E}\left\{\sum_{\tau=t}^{T}\sum_{j=1}^{J}\beta^{\tau-t}d_{j\tau}^O\cdot\left[u_{j\tau}(x_\tau)+\varepsilon_{j\tau}\right]\middle|x_\tau\right\}$$

Optimization

- Applying Bellman (1958) principle, when the state space is finite and Euclidean, we have

$$
\begin{aligned}
V_t(x_t) &= \mathbb{E}\left\{ \sum_j d^O_{jt} \left[u_{jt}(x_t) + \varepsilon_{jt} + \beta \int V_{t+1}(x) \mathrm{d}F_{jt}(x, \varepsilon | x_t, \varepsilon_t) \right] \middle| x_t \right\} \\
\left(\substack{\text{conditional} \\ \text{independence}}\right) &= \sum_j \int_\varepsilon d^O_{jt} \left\{ u_{jt}(x_t) + \varepsilon_{jt} + \beta \int_x V_{t+1}(x) \mathrm{d}F_{jt}(x | x_t) \right\} \mathrm{d}F(\varepsilon)
\end{aligned}
$$

- The choice-specific conditional value function:

$$
v_{jt}(x_t) = \underbrace{u_{jt}(x_t)}_{\substack{\text{flow payoff of} \\ \text{action } j \text{ at } t \\ = \mathbb{E}[u_{jt}(x_t) + \varepsilon_t | x_t]}} + \beta \underbrace{\int V_{t+1}(x) \mathrm{d}F_{jt}(x | x_t)}_{\substack{\text{The expected future utility} \\ \text{of behaving optimally from} \\ \text{period } t+1 \text{ and on}}}
$$

and the optimal decision rule:

$$
\begin{aligned}
d^O_{jt}(x_t, \varepsilon_t) &= \mathbb{1}\left\{ v_{jt}(x_t, \varepsilon_t) \geq v_{kt}(x_t, \varepsilon_t), \forall k \right\} \\
&= \Pi_{k=1}^{J} \mathbb{1}\left\{ v_{jt}(x_t, \varepsilon_t) \geq v_{kt}(x_t, \varepsilon_t) \right\}
\end{aligned}
$$

Conditional Choice Probabilities (CCP)

Given stationarity and the multivariate T1EV errors, the conditional choice probability (CCP) of replacing the engine given x can be described by a **multinomial logit** (McFadden, 1974).

- The CCP is found by integrating the optimal decision rule d_j^o over $(\varepsilon_1, \ldots, \varepsilon_J)$ (Actually, one of the errors needs to be normalized, and you only need to take $J-1$ integrals.):

  $$p_j(x) \equiv P(d=j|x) = \int_\varepsilon d_j^o(x,\varepsilon) f(\varepsilon)\, d\varepsilon$$
  $$= \int_\varepsilon \Pi_{k=1}^J \mathbb{1}\left\{v_j(x,\varepsilon) \geq v_k(x,\varepsilon)\right\} f(\varepsilon \mid x)\, d\varepsilon = \frac{e^{v_j(x)}}{\sum_l e^{v_l(x)}}$$

  where the denominator is the exponential of the **inclusive value** (you saw this in the nested logit):

  $$V_t = \ln \sum_j e^{v_{jt}(x)}$$

  The Euler constant is omitted as it doesn't affect the maximization.

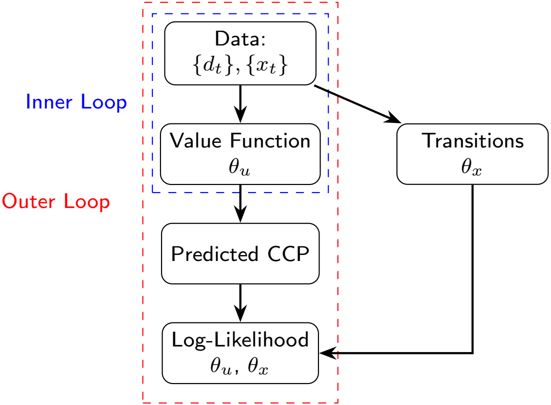
Conditional Choice Probabilities (CCP)

- Specifically, the CCP of replacing the engine in Rust (1987):

$$
\begin{aligned}
p_1(x) &= \int_{\varepsilon} \mathbb{1}\{\varepsilon_0 - \varepsilon_1 \leq v_1(x) - v_0(x)\} f(\varepsilon \mid x)\, d\varepsilon \\
&= \frac{e^{v_1(x)}}{e^{v_1(x)} + e^{v_0(x)}} = \frac{1}{1 + e^{v_0(x) - v_1(x)}}
\end{aligned}
$$

Full Solution Approach

- Nested Fixed Point Algorithm (Rust, 1987):



- θ_u: parameters of the utility function
- θ_x: parameters of the transition matrix.
- "Full Solution": the way we get CCP is by solving the model for every parameter evaluation

Solving the Problem

Finite horizon (not our case)

- To compute the optimum for T finite, we first solve **a static problem in the last period** T, to obtain $d_T^O(x_T, \varepsilon_T)$ for all possible (x_T, ε_T) in the state space.
- Then by **backwards induction**, in period t, j is chosen to maximize the choice-specific conditional value function $v_{jt}(x_t)$.

Solving the Problem

Infinite horizon or large T

- In the stationary infinite horizon case, we assume $u_{jt}(x) \equiv u_j(x)$.
- Consequently, the expected utility in each period is bounded and the **contraction mapping** theorem applies, proving $d_t^O(x) \to d^O(x)$ for large T.
- And V is given by the unique **fixed point** to the contraction mapping $V_{t+1} = V_t \to V = \Gamma(V)$ for large T:

$$\begin{aligned} V_t(x_t) &= \ln \sum_j e^{v_j(x_t)} = \ln \sum_j e^{u_j(x_t) + \beta \int V_{t+1}(x_{t+1}) \mathrm{d}F_j(x_{t+1}|x_t)} \\ \Rightarrow \quad v_j(x_t) &= u_j(x_t) + \beta \int V_{t+1}(x_{t+1}) \mathrm{d}F_j(x_{t+1}|x_t) \\ &= u_j(x_t) + \beta \underbrace{\int \ln \sum_j e^{v_j(x_{t+1})} \mathrm{d}F_j(x_{t+1}|x_t)}_{\equiv EV(x_t)} \end{aligned}$$

Solving the Problem

- Specifically, in Rust (1987):

$$\begin{aligned} v_j(x_t) &= u_j(x_t) + \beta EV(x_t) \\ &= u_j(x_t) + \beta \int \ln \left[e^{u_0(x)+EV(x)} + e^{u_1(0)+EV(0)} \right] \mathrm{d}F_j(x|x_t) \end{aligned}$$

as $d_t = 1 \Rightarrow x_t = 0, x_{t+1} = (0,1,2) \cdot (\varphi_0, \varphi_1, \varphi_2)'$. And this is why in my code on the next slide, $EV(0)$ in calculating the log-sum is the first element of the EV vector (`ev`).

Estimation

Value Function

- For example, the valuefunction.m[3] in MATLAB:

```
function [ev,count]=valuefunction(beta, theta, trans_mat)
    % ... (some description)
    tol = 1e-6; % tolerance level for the iterations
    ev = zeros(90,1); % initialization, EV
    rule=1; % convergence norm

    while rule>tol
        x = [1:1:90]'; % state vector, 90*1
        c = 0.001*theta(2)*x; % maintainance cost, 90*1
        exp1 = exp(-theta(1)+beta*ev(1)); % exp(v1), 1*1
        exp0 = exp(-c+beta*ev); % exp(v0), 90*1

        % update the value function (v0)
        ev_new = trans_mat*log(exp1 + exp0); % 90*1

        % update the norm
        rule = norm(ev_new-ev);

        % go to next iteration
        ev = ev_new;
    end
end
```

[3] You can put this function in a separate .m file and call it in the main script, or you can just put it at the end of the main file.

Estimation

Log Likelihood

- For example, part of `log_lik.m` in MATLAB:

```matlab
for i=1:size(EExt,2) % loop over all i

    st = state(:,i); % state, one column for each i (across t)
    it = EEit(:,i); % action, one column for each i (across t)

    for t=1:size(EExt,1) % loop over all t
        if ~isnan(st(t))

        % choice prob.
        % payoff_diff = v1t - v0t
        p1t = 1 - 1/(1+exp(payoff_diff(st(t))));

        % log-likelihood contribution
        f(t,i) = it(t)*log(p1t) + (1-it(t))*log(1-p1t);
        end
    end

end
```

References

- Rust, J. (1987). Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. Econometrica: Journal of the Econometric Society, 999-1033.
- Rust, J. (2000). Nested fixed point algorithm documentation manual. Unpublished Manuscript (6), 1-43.