

TA Session 3  
Dynamic Discrete Choice: Full Solution  
Microeconometrics II with Joan Lull  
IDEA, Fall 2024

---

TA: Conghan Zheng

November 25, 2024

# A Renewal Problem<sup>1</sup>

- 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:  $(x, \varepsilon)$ 
  - $x_t$ : the accumulated milage since last replacement.
  - *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  $x = 0$ .
  - If the manager keeps the engine, the bus mileage advances to  $x_{t+1} = x_t + 1$ ; if the manager replaces the engine,  $x_t = 0$  and  $x_{t+1} = 1$ .
  - 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).

---

<sup>1</sup>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  $90 \times 90$  Markov transition matrix.
- Do we then need to estimate  $90 \times 90 = 8,100$  parameters?

---

<sup>2</sup>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  $90 \times 90$  discrete transition probability matrix for the uncontrolled Markov process (i.e. assuming replacement never occurs) is given by:

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

- Whereas  $F_{x_t, x_{t+1}}^1$  has only one element 1: if the replacement action is taken, then  $x_{t+1} = 0$ .
- Then, for the entire transition distribution, you need to estimate three numbers:  $\varphi_0$ ,  $\varphi_1$ , and  $\varphi_2$ .

# Estimation

## Transitions

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

```
% Get transition probs by summing all 3 possible transitions and dividing by
% total amount of observations
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
end
```

# Optimization

- The manager maximizes the expected discounted sum of payoffs:

$$\mathbb{E} \left\{ \sum_{t=1}^{\infty} \beta^{t-1} [d_{t0} (\theta_0 x_t + \varepsilon_{t0}) + d_{t1} (\theta_1 + \varepsilon_{t1})] \right\}$$

- This is a stationary infinite horizon problem, age and time have no role.
- For each period, let  $EV(x)$  denote
  - the ex-ante value function at the beginning of the period
  - which is, the discounted sum of current and future payoffs just before  $\varepsilon$  of that period is realized and before the decision of that period is made
- Given conditional independence, the choice-specific conditional value function is

$$v_j(x) = \begin{cases} \beta EV(1), & j = 1 \\ \theta_0 x + \beta EV(x+1), & j = 0 \end{cases}$$

which has the renewal property:  $v_1(x+1) = v_1(1)$ .

- By Bellman's principle, optimal behavior implies:

$$d_1^O(x, \varepsilon) = \mathbb{1} \{ \varepsilon_0 - \varepsilon_1 \leq v_1(x) - v_0(x) \} = 1 - d_0^O(x, \varepsilon)$$

# Conditional Choice Probabilities (CCP)

- Given stationarity and T1EV, the conditional choice probability of replacing the engine given  $x$  can be described by a conditional logit:

$$\begin{aligned}
 p_1(x) &\equiv \int_{\varepsilon} d_1^o(x, \varepsilon) f(\varepsilon) d\varepsilon \\
 &= \int_{\varepsilon} \mathbb{1}\{\epsilon_{t0} - \epsilon_{t1} \leq v_1(x) - v_0(x)\} f(\varepsilon \mid x_t) d\varepsilon \\
 &= \frac{e^{v_1(x)}}{e^{v_1(x)} + e^{v_0(x)}} = \frac{1}{1 + e^{v_0(x) - v_1(x)}}
 \end{aligned}$$

$p_j$  is found by integrating over  $(\varepsilon_1, \dots, \varepsilon_J)$ . Now, fortunately, we have  $J = 2$ .

- An ML estimator could be formed from the CCP.



# Conditional Choice Probabilities (CCP)

## Full Solution Approach

- Nested Fixed Point Algorithm (Rust, 1987):

empirical states and actions  $\xrightarrow[\text{Inner Loop}]{} \text{value function} \rightarrow \text{predicted CCP} \xrightarrow[\text{Outer Loop}]{\text{MLE}} \theta$

# Estimation

## Value Function

- For example, part of loop in `valuefunction.m`<sup>3</sup> in MATLAB:

```
% eq.(10) of Joan's notes
exp1 = exp(-theta(1)+beta*ev0(1)); % exp(v1), 1*1
exp0 = exp(-c+beta*ev0); % exp(v0), 90*1

% update the value function (v0)
% eq.(20) of Joan's notes
ev0_1 = trans_mat*log(exp1 + exp0); % 90*1

% update the norm for convergence
rule = norm(ev0_1-ev0);

% go to next iteration
ev0 = ev0_1;
count = count+1;
```

---

<sup>3</sup>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:

```
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.