Structural Econometrics in Labor and IO
Week 5 - Single Agent Dynamic Discrete Choice in IO

Hannes Ullrich

DIW Berlin and University of Copenhagen

DIW Office 5.2.020, email: hullrich@diw.de

2 June 2022, DIW

Organization

- Next class on June 9, 14:15-17:15
  - Single agent dynamic discrete choice in Labor
  - Read Keane and Wolpin (1997): "The Career Decisions of Young Men"
  - Peter will hand out problem set next week.
- Exam
  - One IO problem on any issue discussed in Weeks 2 to 5 and problem sets 1, 2, and 4
  - Date: Thursday, July 14, 14:15-17:15
  - Place: DIW

Plan for today

- Recap: Week 4
- Discuss code for Rust problem set
- Two-step estimation: Hotz and Miller (ReStud 1993)

Recap: Week 5

- Discrete choice framework with forward-looking agents
- Model primitives: $u(x_t, i, \theta), p(x_{t+1}|x_t, i, \theta), \beta$
- Estimation: introduce unobserved state variable ϵ as structural error term
- Dynamic models involve solution to an optimization problem: dynamic programming
- Under additive separability, conditional independence, and i.i.d. EV type 1 distribution assumptions, obtain dynamic logit choice probabilities:

$$P(i_t|x_t, \theta) = \frac{exp\{V^i(x_t, \theta)\}}{\sum_{i \in \{0,1\}} exp\{V^i(x_t, \theta)\}},$$

Recap: Week 5

- Reduced form vs. structural error specification
- Conditional independence assumption
- EV type I distribution assumption for unobserved state variable ϵ
- (Non)identification

Error term ϵ

- ϵ is modeled as a state variable observed by agents but unobserved by the econometrician: $u(s_t, i, \theta) = u(x_t, i, \theta) + \epsilon_t(i)$, so that $i_t = \delta(x_t, \epsilon_t(i))$
- Alternative: agent's optimization error, random shock to decisions: $i_t = \delta(x_t) + \epsilon_t$
- This alternative would be internally inconsistent
  - Optimizing agents expect such errors, adapt optimal policy $\delta(x_t)$ accordingly
  - If irrational or non-maximizing behavior, deviations from $\delta(x_t)$ not random

Conditional independence assumption

- Assumption CI has two powerful implications:
  - We can write $EV_\theta(x_t, i_t)$ instead of $EV_\theta(x_t, \epsilon_t, i_t)$,
  - We can consider a Bellman equation for $EV_\theta(x_t, i_t)$, which is computationally simpler than the Bellman equation for $V_\theta(x_t, \epsilon_t)$.
- Excludes persistent unobserved heterogeneity, serially correlated ϵ
- Breaks down if a) agents know more about their futures than the econometrician and b) this knowledge influences decision-making

Conditional independence assumption

- In Rust example
  - Unobserved quality of the engine, inexperienced/abusive drivers, systematic dispatch into 'rougher' routes
  - impacts maintenance cost and optimal replacement period
- Dynamic selection problem: In data, (unobservedly) cheaper engines used longer than others
- $\rightarrow$ CI omits part of dynamic nature of decision
- Blevins (JAE 2016), Norets (Ecta 2009), and Reich (2014/OR 2018) find CI assumption violated in Rust data, discuss solutions with serially correlated ϵ. Early attempts by Pakes (1986).

Extreme value type I distribution assumption for ϵ

- Dramatic gains in estimation time due to closed-form solutions for choice probabilities *and* value functions
- In static models, with no random coefficients, EV type I assumption leads to IIA property
- No IIA in dynamic logit because choice probabilities depend on differences in value functions, which include information about all choice alternatives
- Larsen, Oswald, Reich, and Wunderli (EL 2012) show that reasonable modeling assumption in Rust (1987)

Nonparametric (non)identification

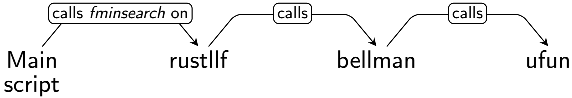
- Hotz and Miller (1993), Magnac and Thesmar (2002), Bajari, Chernozhukov, Hong, and Nekipelov (2009)
  - Logit example: can invert $P(i_t|x_t) = \frac{exp\{V^i(x_t)\}}{\sum_{i \in \{\mathbf{0},\mathbf{1}\}} exp\{V^i(x_t)\}}$,
  - so that $\ln p(i_t = 1|x_t) - \ln p(i_t = 0|x_t) = V^1(x_t) - V^0(x_t)$
  - Only difference in choice-specific values identified
- To identify $u(x, d)$, need to
  - assume $u(x, d_0) = z \;\; \forall \; x$, for reference choice d_0 (Rust: replacement)
  - fix discount factor β
  - know distribution of ϵ
- Remark: Even though the utility function is under-identified, Kalouptsidi, Scott, and Souza-Rodrigues (2016) show that (some) counterfactuals of DDC models are identified and when they are (not).

Nonparametric (non)identification

- Possible exclusion restriction to identify β: State variable that
  - can be excluded from u_t and
  - affects choice-specific value function through payoff-relevant states' transition probabilities.
- In practice: calibrate β to some interest rate r, $\beta = 1/(1+r)$. Typically, $[0.90, 0.99]$, which corresponds to r between 10% and 1%

Problem set 3: Rust's engine replacement problem

Code structure



Two-step estimators

Two-step estimators

- Nested fixed point algorithm is computationally intensive: solve DP problem for each trial value of structural parameters
- Hotz and Miller (1993), Arcidiacono and Miller (2011): Avoid computing DP problem
- HM: Data contain information about agents' expectations about the future – transition *and* choice probabilities
- HM inversion: one-to-one mapping between normalized value functions and conditional choice probabilities (CCP)
- “CCP estimators"

Two-step estimators

Applicable in two broad classes of problems:

- decisions that exhibit finite dependence
  - terminal choice
  - choice making previous period irrelevant (e.g. Rust's engine replacement, durable goods demand)
- stationary settings

Two-step estimators: Intuition

- Recall the conditional (or choice-specific) value function

$$v_t(x_t, d_t) = u(x_t, d_t) + \beta \int \bar{V}_{t+1}(x_{t+1}) f(x_{t+1}|x_t, d_t) dx_{t+1}$$

  where $\bar{V}_{t+1}$ is the ex ante value function, before ϵ_{t+1} is realized.

- Probability of observing d_t conditional on x_t, $p_t(d_t|x_t)$ by integrating ϵ out from decision rule:

$$p_t(d_t|x_t) = \int \mathbb{1}\left(d_t = \arg\max_{d_t \in D_t} \{v_t(x_t, d_t) + \epsilon_t(d_t)\} \right) g(\epsilon_t) d\epsilon_t$$

Two-step estimators: Intuition

- Recall, with ϵ_t following a EV type 1 distribution, conditional choice probability:

$$p_t(d_t|x_t) = \frac{exp(v_t(x_t, d_t))}{\sum_{d_t' \in D} exp(v_t(x_t, d_t'))}$$

and ex-ante value function

$$\bar{V}_t(x_t) = \ln\left(\sum_{d_t' \in D} exp(v_t(x_t, d_t'))\right) + \gamma$$

Two-step estimators: Intuition

- Trick: rewrite $\bar{V}_t(x_t)$ wrt conditional value function for arbitrary d_t^*:

$$\begin{aligned}
\bar{V}_t(x_t) &= \ln\left\{ exp[v_t(x_t, d_t^*)] \left[\frac{\sum_{d_t' \in D} exp[v_t(x_t, d_t')]}{exp[v_t(x_t, d_t^*)]} \right] \right\} + \gamma \\
&= \ln\left[\frac{\sum_{d_t' \in D} exp[v_t(x_t, d_t')]}{exp[v_t(x_t, d_t^*)]} \right] + v_t(x_t, d_t^*) + \gamma \\
&= -\ln[p(d_t^* | x_t)] + v_t(x_t, d_t^*) + \gamma
\end{aligned}$$

Two-step estimators: Intuition

- Plugging into conditional value function:

$$
\begin{aligned}
v_t(x_t, d_t) = u(x_t, d_t) + \beta\gamma + \\
\beta \int \left\{ v_{t+1}(x_{t+1}, d^*_{t+1}) - \ln[p_{t+1}(d^*_{t+1}|x_{t+1})] \right\} f(x_{t+1}|x_t, d_t) dx_{t+1}
\end{aligned}
$$

- Crucial how we can define the normalizing choice d^*_t
- With terminal choice, $v_{t+1}(x_{t+1}, d^*_{t+1})$ reduces to static flow utility term $u(x_{t+1}, d^*_{t+1})$ with parametric form, or normalized to zero

Two-step estimator for stationary settings

- Same setting as Rust: AS, CI, finite state space, stationarity
- Ex ante / integrated value function again, written recursively:

$$\bar{V}(x,\theta) = \int \max_{d \in D} \left\{ u(x,d,\theta) + \epsilon(d) + \beta \sum_{x'} \bar{V}(x',\theta) Pr(x'|x,d) \right\} q(\epsilon|x) d\epsilon$$

Two-step estimator for stationary settings

- With CCP, integrated value function can be written as

$$\bar{V}(x,\theta) = \sum_{d \in D} Pr(d|x) \left(u(x,d,\theta) + E[\epsilon(d)|x,d] + \beta \sum_{x'} \bar{V}(x',\theta) Pr(x'|x,d) \right)$$

  where $E[\epsilon(d)|x,d]$ the expectation of $\epsilon(d)$ conditional on optimal choice d

- HM show that $E[\epsilon(d)|x,d]$ can be written as functions of estimable choice probabilities P

$$E[\epsilon(d)|x,d] = e(d,P)$$

- For ϵ EV type I, the function $e(d,P)$ has a closed form

Two-step estimator for stationary settings

- Using matrix notation, write $\bar{V}$ as

$$\bar{V} = \sum_{d \in D} P(d). * [u(d) + e(d, P) + \beta F(d)\bar{V}]$$

  where
  - $\bar{V}, P(d), u(d), e(d, P)$ are $M \times 1$ vectors with M states,
  - P a $M(J-1) \times 1$ vector of CCPs, and
  - $F(d)$ the $M \times M$ matrix of conditional transition probabilities $Pr(x'|x, d)$.

- System of fixed point equations can be solved for value functions

$$\bar{V} = \left(I_M - \beta F^U(P)\right)^{-1} \left(\sum_{d \in D} P(d). * [u(d) + e(d, P)]\right)$$

  where $F^U(P) = \sum_{d \in D} P(d). * F(d)$ the $M \times M$ matrix of unconditional transition probabilities induced by P

Two-step estimator: Rust problem set

- $d_t \in \{0,1\}$, $(\epsilon_{0t}, \epsilon_{1t}) \sim$ iid type 1 extreme value, so that $e(d,P)$:

$$E[\epsilon_{1t}|x_t, d_t = 1] = \gamma - \ln(Pr(d_t = 1|x_t)),$$
$$E[\epsilon_{0t}|x_t, d_t = 0] = \gamma - \ln(1 - Pr(d_t = 1|x_t)).$$

  where γ is Euler's constant.

- $\hat{T}^d = \hat{T}^0, \hat{T}^1$: cond. transition probability matrices ($M \times M$, estimated λ)
- $\hat{P} = Pr(d_t = 1|x_t)$: vector of state-dependent CCPs ($M \times 1$, estimated)
- $\hat{F}^U(P) = (1-\hat{P})\hat{T}^0 + \hat{P}\hat{T}^1$

- Then, the solution of dynamic programming problem is

$$\bar{V} = \left(I - \beta \hat{F}^U(P)\right)^{-1} \times$$
$$\left((1-\hat{P})\left[u(\cdot,0,\theta) + 0.5772 - \ln(1-\hat{P})\right] + \hat{P}\left[u(\cdot,1,\theta) + 0.5772 - \ln \hat{P}\right]\right)$$

Two-step estimator: Rust problem set

- Model predictions in logit case

$$\tilde{P}r(d_t = 1|x_t, \theta) = \frac{\exp\left[u(x_t, 1, \theta) + \beta \hat{T}^1 \bar{V}(x_t, \theta)\right]}{\exp\left[u(x_t, 0, \theta) + \beta \hat{T}^0 \bar{V}(x_t, \theta)\right] + \exp\left[u(x_t, 1, \theta) + \beta \hat{T}^1 \bar{V}(x_t, \theta)\right]}$$

- Then, pseudo log-likelihood function to maximize:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \sum_{t=0}^{T} \left[i_{it} * \ln \tilde{P}r(d_{it} = 1|x_{it}, \theta) + (1 - i_{it}) * \ln(1 - \tilde{P}r(d_{it} = 1|x_{it}, \theta))\right]$$

Two-step estimator: Rust problem set

- Model predictions in logit case

$$\tilde{P}r(d_t = 1|x_t, \theta) = \frac{\exp\left[u(x_t, 1, \theta) + \beta \hat{T}^1 \bar{V}(x_t, \theta)\right]}{\exp\left[u(x_t, 0, \theta) + \beta \hat{T}^0 \bar{V}(x_t, \theta)\right] + \exp\left[u(x_t, 1, \theta) + \beta \hat{T}^1 \bar{V}(x_t, \theta)\right]}$$

- Then, pseudo log-likelihood function to maximize:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \sum_{t=0}^{T} \left[i_{it} * \ln \tilde{P}r(d_{it} = 1|x_{it}, \theta) + (1 - i_{it}) * \ln(1 - \tilde{P}r(d_{it} = 1|x_{it}, \theta))\right]$$

Switch to code

Two-step estimators

- HM: significant computational gain, but efficiency loss
- Aguirregabiria and Mira (Ecta 2002) nest as special cases HM and Rust's NFXP algorithm
- Reasonable for stable markets