Modelos Dinâmicos II: "Estimating Dynamic Models of Imperfect Competition" (Bajari, Benkard e Levin - BBL) e o Estimador em Dois Estágios de Hotz e Miller

June 12, 2020

Outline

Introduction

- Estimação de jogos dinâmicos.
- BBL = Rust + Berry
- Modelo ESTRUTURAL. Vários detalhes
- Método muito usado em outros papers

Modelo - definições

- N firmas, $i = 1, ..., N$, tomam decisões em $t = 1, ..., \infty$.
- $s_t \in S$, vetor de variáveis de estado
- dado s_t, firmas simultaneamente escolhem a ação em t.
- $a_{it} \in A_i$ e $\mathtt{a}_t = (a_{1t}, ..., a_{Nt}) \in A$
- Antes de escolher a_{it}, a firma i recebe um choque privado ν_{it}, retirado *iid* a partir da distribuicão $G_i(.|s_t)$.
- Vetor de choques: $\nu_t = (\nu_{1t}, ..., \nu_{Nt})$.

Modelo - função lucro

- Lucro da firma i: $\pi_i(\mathsf{a}_t, s_t, \nu_{it})$.
- Dado s_t, o lucro esperado da firma i é

$$E[\sum_{\tau=t}^{\infty} \beta^{\tau-t} \pi_i(\mathsf{a}_t, s_t, \nu_{it}) | s_t]$$

- Transição entre estados, s_{t+1}, é obtido de $P(s_{t+1}|\mathsf{a}_t, s_t)$

MPE

- Em um MPE, a ação de uma firma depende apenas do estado corrente (s) do seu choque privado, ν.
- Estratégia da firma i: $\sigma_i : S \times \nu_i \to A_i$
- $\sigma = (\sigma_1, ..., \sigma_n)$ é o vetor de estratégias de Markov.
- Se o comportamento é dado por estratégias markovianas, o lucro intertemporal pode ser escrito recursivamente:

$$V_i(s;\sigma) = E_\nu[\pi_i(\sigma(s,\nu), s, \nu_i)] + \beta \int V_i(s';\sigma) dP(s'|\sigma(s,\nu), s)|s]$$

Essa é a ex-ante value function.

MPE

Definição: σ é um MPE se, dado o vetor de estratégia dos oponentes, σ_{-i}, cada firma i prefere sua estratégia σ_i a qualquer outra alternativa σ_i'. Ou seja,

$$V_i(s;\sigma) \geq V_i(s;\sigma_i'), \quad \forall \sigma_i' \neq \sigma_i$$

O objetivo da estimação é obter $\theta, \Pi(\theta), G(\theta)$.

Estimação em 2 estágios

1. 1o.: Estima a função de política ótima, a função de transição entre estados e a função valor.
2. 2o. Estima os parâmetros estruturais.

1o. estágio - dados

- dados precisam vir de um único mercado, ou de um conjunto de mercados similares.
- Dados precisam ser gerados pelo mesmo MPE.
- Por que? Porque existe multiplicidade de equilíbrio.

Caso de modelo de escolha discreta (Rust)

- Choque privado da firma, ν_i, entra aditivamente na função lucro:

$$\pi_i(\mathtt{a}, s, \nu_i)$$

- Função valor para uma dada escolha:

$$v_i(a_i; s) = E_{\nu_{-i}}[\pi_i(a_i, \sigma_{-i}(s, \nu_{-i}), s)] + \beta \int V_i(s'; \sigma) dP(s' | \sigma_{-i}(s, \nu_{-i}), s)$$

Caso de modelo de escolha discreta (Rust)

- Firma escolhe a_i que satisfaz

$$v_i(a_i; s) + \nu_i(ai) \geq v_i(a_i'; s) + \nu_i(a_i'), \forall a_i' \in A_i$$

- Note que a firma precisa conhecer $v_i(a_i; s)$ para todas as ações e estados. Problema se houver muitos estados (curse of dimensionality)

Modelo de escolha discreta

- Firma escolhe a_i que satisfaz

$$v_i(a_i; s) + \nu_i(ai) \geq v_i(a_i'; s) + \nu_i(a_i'), \forall a_i' \in A_i$$

- Note que a firma precisa conhecer $v_i(a_i; s)$ para todas as ações e estados. Problema se houver muitos estados (curse of dimensionality)

Hotz & Miller

- obtém a função valor para uma escolha específica através da inversão da probabilidade de escolha condicional observada, em cada estado.

$$P(a_i|s) = P(v_i(a_i, s) + \nu_i(a_i) \geq v_i(a_i', s) + \nu_i(a_i')),$$

$$= P(\nu_i(a_i) - \nu_i(a_i')) \geq v_i(a_i', s) - v_i(a_i, s), \forall a_i' \in A_i$$

Hotz & Miller

- Exemplo: Se ν for distribuido como (advinha?) uma valor extremo tipo 1, e com duas opções (Rust) a integral de cálculo de probabilidade tem a tradicional forma do logit:

$$P(a_i = 1|s) = \frac{exp(v(1,s) - v(0,s))}{1 + exp(v(1,s) - v(0,s))}$$

É aqui que os dois métodos - nested fixed point e two step - divergem!

$$= exp(v(1,s) - v(0,s)) \frac{1}{1 + exp(v(1,s) - v(0,s))}$$

Hotz & Miller

- Logo,

$$P(a_i = 1|s) = exp(v(1,s) - v(0,s))P(a_i = 0|s)$$

$$\log(P(a_i = 1|s)) - \log(P(a_i = 0|s)) = v(1,s) - v(0,s) \tag{1}$$

- E daí? $P(a_i = 1|s)$ e $P(a_i = 0|s)$ são estimados a partir dos dados, como um probit da decisão de trocar o motor na quilometragem. Se conheço essas probabilidades, eu consigo saber o valor de $v(1,s) - v(0,s)$ para qualquer valor de s. Isso te fala o que fazer em cada valor de s.
- H & M te leva da CCP empírica para a optimal policy. Genial!

Value function

- Tendo a função de política ótima, e a probabilidade de transição, calculamos a value function por simulação.
- A função valor da firma i no estado s, assumindo que os jogadores jogam σ, é:

$$V_i(s;\sigma;\theta) = E[\sum_{t=0}^{\infty} \beta^t \pi_i(\sigma(s_t, \nu_t), s_t, \nu_{it};\theta)|s_0 = s;\theta]$$

Simulando a value function

Dado as estimativas do primeiro estágio da probabilidade de transição $\hat{P}(s_t + 1|s_t)$, e das CCP, $\hat{P}(a_t|s_t)$, a simulação de uma trajetória é da seguinte forma:

1. Começa em s_0, simula ν_{i0} usando $G_i(.|s_0, \theta)$, para cada firma i.
2. Calcula $a_{i0} = \sigma_i(s_0, \nu_{i0})$ e o lucro $\pi_i(\mathsf{a}_0, s_0, \nu_{i0}; \theta)$ para cada firma i.
3. Sorteia um novo estado s_1 usando $\hat{P}(s_1|\mathsf{a}_0, s_0)$.
4. Repita os passos 1-3 por T períodos, ou até que alguma firma atinja algum estado terminal e saia do mercado.
5. Para cada trajetória computar o somatório intertemporal dos lucros da firma. Simular várias trajetórias e tirar a média: $\hat{V}_i(s; \sigma; \theta)$

2o. estágio

- Usar a função de política, de transição entre os estados e a value function para obter os parâmetros estruturais.
- Equilíbrio

$$V_i(s;\sigma;\theta) \geq V_i(s;\sigma_i,\sigma_i';\theta), \forall i$$

- Define

$$g(\theta,\alpha) = V_i(s;\sigma;\theta) \geq V_i(s;\sigma_i,\sigma_i';\theta,\alpha), \forall i$$

- α é o desvio da política ótima (soma). 'one step deviation'.
- Escolha θ que maximiza $g(\theta,\alpha)$.

Rust (1987): troca ótima do motor

- Lucro:

$$\pi(a, s, \nu; \theta) = \begin{cases} -\mu s + \nu(0), & \text{se } a = 0 - R + \nu(1), \\ \text{se } a = 1 \end{cases}$$

- Simulação: $\mu = 1$, $R = 4$, $\beta = 0.9$, $s \in 1, 2, 3, 4, 5$, $\nu(0), \nu(1) \sim N(0, 1)$.
- Como simular os dados dos ônibus??

Rust (1987): troca ótima do motor

- Value functions escolha específica:

$$v(1, s) = -R + \beta V(1; \sigma)$$

$$v(0, s) = -\mu s + \beta V(0; \sigma)$$

Rust: primeiro estágio

1- Calcule a probabilidade de troca (CCP) para cada valor de s:

$$P(1|s=1) = \frac{\#[a=1|s=1]}{\#[s=1]}$$

$$\vdots$$

$$P(1|s=5) = \frac{\#[a=1|s=5]}{\#[s=5]}$$

Rust: primeiro estágio

2- Use HM para achar a política ótima:

$$\sigma(s,\nu) = 1 \iff v(1,s) + \nu(1) \geq v(0,s) + \nu(0)$$

$$P(1|s) = P(v(1,s) - v(0,s) \geq \nu(0) - \nu(1))$$

$$P(1|s) = \Phi(\frac{[v(1,s) - v(0,s)]}{\sqrt{2}})$$

$$\Phi^{-1}(P(1|s))\sqrt{2} = v(1,s) - v(0,s) \tag{2}$$

Essa á a política ótima $\sigma(s,\nu)$ em cada estado s.

Rust: value function

- Podemos reescrever o lucro como

$$\pi(a, s, \nu; \theta) = a[-R + \nu(1)] + (1 - a)[-\mu s + \nu(0)]$$

- A value function simulada:

$$V(s; \sigma; \theta) =$$

$$E[-\sum_{t=0}^{\infty} \beta^t \sigma(s_t, \nu_t) | s_0 = s].R$$

$$E[-\sum_{t=0}^{\infty} \beta^t (1 - \sigma(s_t, \nu_t)) s_t | s_0 = s].\mu$$

$$E[\sum_{t=0}^{\infty} \beta^t \nu(\sigma(s_t, \nu_t))) | s_0 = s]$$

Rust: segundo estágio

- Altera a política ótima computada em 2.
- Algoritmo procura R e μ que fazem $g_i(\theta, \alpha) > 0$ para todo i.
- ver tabela 1.

Linearização

- Apesar de não ter que iterar a value function, a simulação pode ficar proibitivamente demorada se tiver que ser repetida a cada etapa de procura pelo θ ótimo.
- Solução: lucro linear em θ: $\pi = [a, s, \nu]\theta$.
- Simula apenas uma vez as trajetórias, e então ache o θ.