DYNAMIC STRUCTURAL ECONOMETRICS
ECONOMETRIC SOCIETY SUMMER SCHOOL

—

Nested Pseudo Likelihood Estimation

Victor Aguirregabiria (University of Toronto)

Canberra, December 15, 2022

INTRODUCTION

- This lecture deals with the estimation of Dynamic Discrete Choice structural models using **Nested Pseudo Likelihood (NPL)** method.

- In previous lectures, you have seen two main estimation methods:
  - The NFXP algorithm for the computation of the MLE.
  - The Two-Step CCP or Hotz-Miller estimator.

- The NPL estimator was proposed by **Aguirregabiria & Mira, (ECMA, 2002)** in single-agent models with the purpose of dealing with potential issues in the application of NFXP and CCP methods.
  - NFXP: Computational cost of repeatedly solving the DP problem for every trial value of the structural parameters.
  - CCP: Statistical inefficiency of the two-step CCP method, asymptotically and in small samples.

INTRODUCTION [2/2]

- NPL method shares features with both NFXP and CCP.

- **Similarly as NFXP and in contrast to CCP:**

  a. **Full solution method**. Upon convergence, it provides estimates of structural parameters and CCPs that solve the DP problem.

  b. The algorithm does not require to be initialized with consistent estimates of CCPs (or of structural parameters).

  c. Upon convergence, it provides the ML estimator.

- **Similarly as CCP and in contrast to NFXP:**.

  d. An iteration of the NPL algorithm does not require solving the DP problem. It requires the same computations as the implementation of the (second step of) CCP method.

- Therefore, in single-agent models, the NPL method is **an alternative algorithm to NFXP to compute the MLE**.

OUTLINE

1. **SINGLE-AGENT MODEL**
   a. Model
   b. Policy Iteration Mapping
2. **NPL ESTIMATION IN SINGLE-AGENT MODELS**
   a. Estimation method
   b. Algorithm
   c. Computational & statistical properties
3. **DYNAMIC DISCRETE CHOICE GAMES**
   a. Model
   b. Policy Iteration Mapping
   c. Estimation with Multiple Equilibria
4. **NPL ESTIMATION IN DYNAMIC GAMES**
   a. Estimation method
   b. Algorithms

REFERENCES

- "Dynamic Discrete Choice Structural Models: A Survey," Victor Aguirregabiria and Pedro Mira. Journal of Econometrics, 156(1) (2010), 38-67.
- "Swapping the Nested Fixed Point Algorithm: A Class of Estimators for Discrete Markov Decision Models," Victor Aguirregabiria and Pedro Mira. Econometrica, 70 (2002), 1519-1543.
- "Sequential Estimation of Dynamic Discrete Games," Victor Aguirregabiria and Pedro Mira. Econometrica, 75(1) (2007), 1-53.
- "Dynamic Games in Empirical Industrial Organization," Victor Aguirregabiria, Allan Collard-Wexler, and Stephen P. Ryan. Handbook of Industrial Organization, Volume 4, Chapter 4, pp. 225-343. Elsevier (2021).
- "Imposing Equilibrium Restrictions in the Estimation of Dynamic Discrete Games," Victor Aguirregabiria, and Mathieu Marcoux. Quantitative Economics, 12(4), 1223-1271 (2021).

1. SINGLE-AGENT MODEL

GENERAL FEATURES: DECISION & STATES

- t represents **time**, and it is discrete: $t \in \{1, 2, ...\}$.
- **Econometric model** with a dependent variable y_t, explanatory variables $\mathbf{x}_t$, and unobservables to the researcher ε_t.
- $y_t =$ **agent's decision** at time t. It is discrete: $y_t \in \{0, 1, ..., J\}$
- The agent takes this action to **maximize her expected and discounted flow of utility** (infinite horizon) :

$$\mathbb{E}_t \left(\sum_{s=0}^{\infty} \beta^s \ \pi_{t+s} \right)$$

$\beta \in [0, 1)$ is the discount factor, and π_t is the utility at period t.

GENERAL FEATURES: UTILITY

- **Utility** depends on action y_t, and state variables $\mathbf{x}_t$ and ε_t:

$$\pi_t = \pi\left(y_t, \mathbf{x}_t, \boldsymbol{\theta}_\pi\right) + \varepsilon_t(y_t)$$

  and $\boldsymbol{\theta}_\pi$ is the vector of **structural parameters** in the utility function.

- State variables in $\mathbf{x}_t$ are **observable to us as researchers**.

- State variables in ε_t are **unobservable to us as researchers**.

- Because the model is **dynamic**, $\mathbf{x}_t$ should depend **on previous decisions**, say y_{t-1}.

GENERAL FEATURES: TRANSITIONS

- The model is completed with the specification of the **transition rules** or **transition probabilities** followed by the state variables $\mathbf{x}_t$ and ε_t.

- For ε_t, standard assumption is i.i.d. For concreteness, I assume that they are **i.i.d. Type 1 Extreme Value**.

- For $\mathbf{x}_t$, the standard assumption is that it has **discrete and finite support** and follows a **Controlled First Order Markov Process**:

$$Pr\left(\mathbf{x}_{t+1} = \mathbf{x}' \mid y_t = y,\ \mathbf{x}_t = \mathbf{x}\right) \ = \ f_x\left(\mathbf{x}' \mid y,\ \mathbf{x}\right)$$

- $\mathbf{F}_x(y)$ is the transition probability matrix of $\mathbf{x}_t$ when $y_t = y$.

GENERAL FEATURES: DYNAMIC PROGRAMMING

- The agent's decision is a **Dynamic Programming (DP)** problem.
- Let $V(\mathbf{x}_t, \varepsilon_t)$ be the value function. The **Bellman Equation** of this DP problem is:

$$V(\mathbf{x}_t, \varepsilon_t) = \max_{y_t} \left\{ \begin{array}{l} \pi(y_t, \mathbf{x}_t) + \varepsilon_t(y_t) + \\ \\ \beta \sum\limits_{x_{t+1}} \int V(\mathbf{x}_{t+1}, \varepsilon_{t+1}) \; f_\varepsilon(d\varepsilon_{t+1}) \; f_x(\mathbf{x}_{t+1} | y_t, \mathbf{x}_t) \end{array} \right\}$$

- The **Optimal Decision Rule**, $\alpha(\mathbf{x}_t, \varepsilon_t)$, is the **argmax** in y_t of the expression within brackets $\{\}$.

INTEGRATED BELLMAN EQUATION & CCPs

- The Integrated Value Function is: $V^{\sigma}(\mathbf{x}_t) \equiv \int V(\mathbf{x}_t, \varepsilon_t) \; f_{\varepsilon}(d\varepsilon_t)$

- The **Integrated Bellman Equation** for the Logit model is:

$$V^{\sigma}(\mathbf{x}_t) = \log\left(\sum_{y=0}^{J} \exp\left\{\pi(y, \mathbf{x}_t) + \beta \sum_{x_{t+1}} V^{\sigma}(\mathbf{x}_{t+1}) \; f_x(\mathbf{x}_{t+1} | y, \mathbf{x}_t)\right\}\right)$$

- **Conditional Choice Probability (CCP)** function for Logit model is:

$$P(y|\mathbf{x}_t) = \frac{\exp\left\{\pi(y, \mathbf{x}_t) + \beta \sum_{x_{t+1}} V^{\sigma}(\mathbf{x}_{t+1}) \; f_x(\mathbf{x}_{t+1} | y, \mathbf{x}_t)\right\}}{\sum_{j=0}^{J} \exp\left\{\pi(j, \mathbf{x}_t) + \beta \sum_{x_{t+1}} V^{\sigma}(\mathbf{x}_{t+1}) \; f_x(\mathbf{x}_{t+1} | j, \mathbf{x}_t)\right\}}$$

SOLUTION METHODS

- Two methods are the most commonly used for solving this DP.

1. **Fixed Point iterations in Integrated Bellman eq.**: $\mathbf{V}^{\sigma}_{n+1} = \Gamma_{\boldsymbol{\theta}}(\mathbf{V}^{\sigma}_{n})$.

    a. Integrated Bellman is a contraction: always converges to unique f.p.
    b. Converges slowly (linearly)
    c. Complexity (comput. cost) 1 iteration is linear in $|X|$.

2. **Policy (Newton-Kantorovich) iterations**: $\mathbf{P}_{n+1} = \Psi_{\boldsymbol{\theta}}(\mathbf{P}_{n})$.

    a. In this model, $\Psi_{\boldsymbol{\theta}}(.)$ is a contraction: always converges to unique f.p.
    b. Converges fast (quadratically).
    c. Complexity (comput. cost) 1 iteration is cubic in $|X|$.

    d. $\Psi_{\boldsymbol{\theta}}(.)$ plays a key role in **Nested Pseudo Likelihood estimation**.

POLICY (NEWTON-KANTOROVICH) MAPPING

- Let $\mathbf{P} \in [0,1]^{(J+1)|\mathcal{X}|}$ be vector of CCPs for every action-state $(y, \mathbf{x})$.

- $\Psi_{\boldsymbol{\theta}}(.)$ is a **fixed point mapping** in the space of $\mathbf{P}$:

$$\mathbf{P} = \Psi_{\boldsymbol{\theta}}(\mathbf{P})$$

- $\Psi_{\boldsymbol{\theta}}(.)$ is a **contraction**, and its unique fixed point is the vector of CCPs that solves the DP problem.

- $\Psi_{\boldsymbol{\theta}}(.)$ is the **composition of two mappings**:

$$\Psi_{\boldsymbol{\theta}}(\mathbf{P}) \equiv \Lambda_{\boldsymbol{\theta}}\left(\varphi_{\boldsymbol{\theta}}\left(\mathbf{P}\right)\right)$$

  - $\varphi_{\boldsymbol{\theta}}\left(.\right)$ is the **Policy Valuation mapping**.
  - $\Lambda_{\boldsymbol{\theta}}\left(.\right)$ is the **Policy Improvement mapping**

POLICY (NEWTON-KANTOROVICH) MAPPING [2/2]

- **Policy Valuation mapping** $\varphi_{\boldsymbol{\theta}}\left(\mathbf{P}\right)$ returns the vector of values $\mathbf{V}$ (one value for each state $\mathbf{x}$) if the agent behaves – now an in the future – according to the CCPs in $\mathbf{P}$.

$$\varphi_{\boldsymbol{\theta}}\left(\mathbf{P}\right) = \mathbf{V} =$$

$$\left[\mathbf{I} - \beta \sum_{y=0}^{J} \mathbf{P}(y) * \mathbf{F}_x(y)\right]^{-1} \left[\sum_{y=0}^{J} \mathbf{P}(y) * \left(\boldsymbol{\Pi}_{\boldsymbol{\theta}}(y) + \gamma - \ln \mathbf{P}(y)\right)\right]$$

- **Policy Improvement mapping** $\Lambda_{\boldsymbol{\theta}}\left(\mathbf{V}\right)$ returns the vector of CCPs $\mathbf{P}$ (one value for each action-state $(y, \mathbf{x})$) which are optimal if future values are given by vector $\mathbf{V}$.

$$\Lambda_{\boldsymbol{\theta}}\left(y, \mathbf{V}\right) = \mathbf{P}(y) = \frac{\exp\left\{\boldsymbol{\Pi}_{\boldsymbol{\theta}}(y) + \beta\ \mathbf{F}_x(y)\ \mathbf{V}\right\}}{\sum_{j=0}^{J} \exp\left\{\boldsymbol{\Pi}_{\boldsymbol{\theta}}(j) + \beta\ \mathbf{F}_x(j)\ \mathbf{V}\right\}}$$

POLICY ITERATION MAPPING with Linear-in-Parameters Utility

- Suppose that the utility function is:

$$\pi(y, \mathbf{x}_t) \;=\; \mathbf{z}(y, \mathbf{x}_t)'\boldsymbol{\theta}$$

where $\mathbf{z}(y, \mathbf{x}_t)$ is a vector of functions known by the researcher.

- Then, $\Psi_{\boldsymbol{\theta}}\left(y \mid \mathbf{x}_t, \mathbf{P}\right)$ has a simple structure:

$$\Psi_{\boldsymbol{\theta}}\left(y \mid \mathbf{x}_t, \mathbf{P}\right) \;=\; \frac{\exp\left\{\widetilde{\mathbf{z}}_t^{\mathbf{P}}(y)'\boldsymbol{\theta} + \widetilde{e}_t^{\mathbf{P}}(y)\right\}}{\sum_{j=0}^{J} \exp\left\{\widetilde{\mathbf{z}}_t^{\mathbf{P}}(j)'\boldsymbol{\theta} + \widetilde{e}_t^{\mathbf{P}}(j)\right\}},$$

with:

$$\widetilde{\mathbf{z}}_t^{\mathbf{P}}(y)' \equiv \mathbf{z}(y, \mathbf{x}_t)' + \beta\; F_x(y, \mathbf{x}_t)' \left[\mathbf{I} - \beta\; \mathbf{F}_x^{\mathbf{P}}\right]^{-1} \left[\sum_{j=0}^{J} \mathbf{P}(j) * \mathbf{Z}(j)\right]$$

$$\widetilde{e}_t^{\mathbf{P}}(y) \equiv \beta\; F_x(y, \mathbf{x}_t)' \left[\mathbf{I} - \beta\; \mathbf{F}_x^{\mathbf{P}}\right]^{-1} \left[\sum_{j=0}^{J} \mathbf{P}(j) * (\gamma - \ln \mathbf{P}(j)\right]$$

ZERO JACOBIAN OF POLICY VALUATION MAPPING

- **Aguirregabiria-Mira (2002) Proposition 1**. In this class of models, mapping $\Psi_{\boldsymbol{\theta}}(.)$ is a contraction.

- **Aguirregabiria-Mira (2002) Proposition 2**. Let $\mathbf{P}_{\boldsymbol{\theta}}$ be the fixed point of $\Psi_{\boldsymbol{\theta}}(.)$. Then, the valuation mapping have zero Jacobian matrix evaluated at $\mathbf{P}_{\boldsymbol{\theta}}$.

$$\frac{\partial \varphi_{\boldsymbol{\theta}}(\mathbf{P}_{\boldsymbol{\theta}})}{\partial \mathbf{P}'} = \mathbf{0}$$

- Intuition: $\mathbf{P}_{\boldsymbol{\theta}}$ is optimal, it maximizes the vector of values. Therefore, it maximizes the valuation operator $\varphi_{\boldsymbol{\theta}}(\mathbf{P})$.

- Since the PI operator $\Psi_{\boldsymbol{\theta}}$ is the composition of the valuation operator and the policy imporvement operator (which is bounded-valued):

$$\frac{\partial \Psi_{\boldsymbol{\theta}}(\mathbf{P}_{\boldsymbol{\theta}})}{\partial \mathbf{P}'} = \frac{\partial \Lambda_{\boldsymbol{\theta}}(\mathbf{V})}{\partial \mathbf{V}'} \frac{\partial \varphi_{\boldsymbol{\theta}}(\mathbf{P}_{\boldsymbol{\theta}})}{\partial \mathbf{P}'} = \frac{\partial \Lambda_{\boldsymbol{\theta}}(\mathbf{V})}{\partial \mathbf{V}'} \mathbf{0} = \mathbf{0}$$

IMPLICATION OF ZERO JACOBIAN FOR ESTIMATION

- In the estimation of the model – for instance, by MLE – we have the probabilities $\mathbf{P}_{\boldsymbol{\theta}}$ that solve the DP problem.

- Estimation algorithms (e.g., BHHH) use derivatives $\dfrac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}'}$. Given that $\mathbf{P}_{\boldsymbol{\theta}} = \Psi_{\boldsymbol{\theta}}(\mathbf{P}_{\boldsymbol{\theta}})$, and applying the derivative of an implicit function:

$$\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}'} = \left[\mathbf{I} - \beta \, \frac{\partial \Psi_{\boldsymbol{\theta}}}{\partial \mathbf{P}'} \right]^{-1} \frac{\partial \Psi_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}'}$$

- The zero Jacobian property implies that:

$$\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}'} = \left[\mathbf{I} - \beta \, \mathbf{0} \right]^{-1} \frac{\partial \Psi_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}'} = \frac{\partial \Psi_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}'}$$

- This result has important computational and statistical implications for the estimation of the model.

2. NPL ESTIMATION OF SINGLE-AGENT MODELS

PSEUDO LIKELIHOOD FUNCTION

- The researcher has panel data of N individuals over T periods of time.

$$Data \;=\; \{\; y_{it} \;,\; \mathbf{x}_{it} : i = 1, 2, ..., N \;;\; t = 1, 2, ..., T \;\}$$

- We are interested in estimation of parameters in utility function, $\boldsymbol{\theta}$.

- For any arbitrary value of $\boldsymbol{\theta}$ and of the vector of CCPs $\mathbf{P}$, the **Pseudo Likelihood Function** is defined as:

$$Q(\boldsymbol{\theta}, \mathbf{P}) = \sum_{i=1}^{N} \sum_{t=1}^{T} \ln \Psi_{\boldsymbol{\theta}} \left(y_{it} \mid \mathbf{x}_{it}, \mathbf{P} \right)$$

where $\Psi_{\boldsymbol{\theta}} \left(y \mid \mathbf{x}, \mathbf{P} \right)$ is element $(y, \mathbf{x})$ in vector $\Psi_{\boldsymbol{\theta}} \left(\mathbf{P} \right)$

PSEUDO LIKELIHOOD FUNCTION – Linear-in-Parameters Utility

- Suppose that the utility function is:

$$\pi(y, \mathbf{x}_{it}) \;=\; \mathbf{z}(y, \mathbf{x}_{it})'\boldsymbol{\theta}$$

where $\mathbf{z}(y, \mathbf{x}_{it})$ is a vector of functions known by the researcher.

- Then, $\Psi_{\boldsymbol{\theta}}\left(y \mid \mathbf{x}_{it}, \mathbf{P}\right)$ has a simple structure:

$$\Psi_{\boldsymbol{\theta}}\left(y \mid \mathbf{x}_{it}, \mathbf{P}\right) \;=\; \frac{\exp\left\{\widetilde{\mathbf{z}}_{it}^{\mathbf{P}}(y)'\boldsymbol{\theta} + \widetilde{e}_{it}^{\mathbf{P}}(y)\right\}}{\sum_{j=0}^{J}\exp\left\{\widetilde{\mathbf{z}}_{it}^{\mathbf{P}}(j)'\boldsymbol{\theta} + \widetilde{e}_{it}^{\mathbf{P}}(j)\right\}},$$

with:

$$\widetilde{\mathbf{z}}_{it}^{\mathbf{P}}(y)' \equiv \mathbf{z}(y, \mathbf{x}_{it})' + \beta\ F_x(y, \mathbf{x}_{it})'\left[\mathbf{I} - \beta\ \mathbf{F}_x^{\mathbf{P}}\right]^{-1}\left[\sum_{j=0}^{J}\mathbf{P}(j) * \mathbf{Z}(j)\right]$$

$$\widetilde{e}_{it}^{\mathbf{P}}(y) \equiv \beta\ F_x(y, \mathbf{x}_{it})'\left[\mathbf{I} - \beta\ \mathbf{F}_x^{\mathbf{P}}\right]^{-1}\left[\sum_{j=0}^{J}\mathbf{P}(j) * (\gamma - \ln \mathbf{P}(j)\right]$$

NPL ESTIMATOR

- NPL estimator is defined as a pair $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{P}})$ satisfying two conditions.
- **[NPL-1]** $\widehat{\boldsymbol{\theta}}$ maximizes the pseudo-likelihood given $\widehat{\mathbf{P}}$:

$$\widehat{\boldsymbol{\theta}} = \arg max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}, \widehat{\mathbf{P}})$$

- **[NPL-2]** $\widehat{\mathbf{P}}$ is a fixed point for $\Psi_{\widehat{\boldsymbol{\theta}}}$:

$$\widehat{\mathbf{P}} = \Psi_{\widehat{\boldsymbol{\theta}}}\left(\widehat{\mathbf{P}}\right)$$

- If there are multiple values $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{P}})$ satisfying conditions [NPL-1] and [NPL-2], the NPL estimator is the one with maximum value for Q.

RELATIONSHIP BETWEEN NPL ESTIMATOR & MLE

- For this class of models, the **Zero-Jacobian Property of PI mapping** Ψ_{θ} implies that **NPL and MLE are the same estimator**.

- **Aguirregabiria & Mira (2002) Proposition 3**.

    - $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{P}})$ satisfies conditions [NPL-1] and [NPL-2] if and only if $\widehat{\boldsymbol{\theta}}$ the likelihood equations from the "true" likelihood.

    - $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{P}})$ is the NPL solution with the maximum value of Q if and only if $\widehat{\boldsymbol{\theta}}$ is the solution to the likelihood equations with the maximum value for the "true" likelihood.

NPL ALGORITHM – "Swapping" the inner and outer alg. of NFXP

- AM (2002) propose the following **NPL Fixed Point Algorithm** to compute the NPL estimator.
- Start with an arbitrary vector of CCPs $\mathbf{P}_0$.
- At iteration $n \geq 1$, perform the following 4 tasks/steps.

Step 1 Calculate value $\widetilde{\mathbf{z}}_{it}^{\mathbf{P}_{n-1}}(y)$ and $\widetilde{e}_{it}^{\mathbf{P}_{n-1}}(y)$ for every y and obs. i, t.

Step 2 Obtain:

$$\boldsymbol{\theta}_n = \arg max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}, \mathbf{P}_{n-1})$$

Step 3 Update the vector of CCPs as:

$$\mathbf{P}_n = \Psi_{\boldsymbol{\theta}_n}(\mathbf{P}_{n-1})$$

Step 4 Check for convergence: $|| \; \mathbf{P}_n - \mathbf{P}_{n-1} \; || <$ small constant .

NPL ALGORITHM – COMPUTATIONAL COST

- In models with utility linear in $\boldsymbol{\theta}$, Step 2, $\boldsymbol{\theta}_n = \arg max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \mathbf{P}_{n-1})$ is very simple, as it is equivalent to estimating static logit model (likelihood is globally concave).
- The main cost comes from Step 3: $\mathbf{P}_n = \Psi_{\boldsymbol{\theta}_n}(\mathbf{P}_{n-1})$. This is equivalent to 1 Policy Iteration.
- In single-agent models, the number of NPL iterations to reach the NPL estimator is approximately equal to the number of Policy Iterations needed to solve once the DP problem,
- In contast, NFXP requires multiple Policy Iterations (up to convergence) for one single NFXP (outer) iteration.

RELATIONSHIP BETWEEN NPL & CCP 2-STEP ESTIMATOR

- Let $\widehat{\mathbf{P}}_0$ be a consistent nonparametric estimator of the true CCPs in the population.

- Given $\widehat{\mathbf{P}}_0$, apply one step of the NPL algorithm:

$$\widehat{\boldsymbol{\theta}}_1 \;=\; \arg max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}, \widehat{\mathbf{P}}_0)$$

- The estimator $\widehat{\boldsymbol{\theta}}_1$ is **Hotz-Miller 2-step CCP estimator**, version that does not exploit finite dependence property.

- The recursive application of NPL algorithm generates a sequence of **K-step CCP estimators**

MONTE CARLO EXPERIMENT

- Compare the computational cost of NFXP and NPL in the context of Rust (1987)'s bus engine replacement model.
- NFXP using Policy iterations in inner. Estimation of utility parameters (transitions in a first step).
- Experiment with different sample sizes (1,000, 5,000, 10,000), different state spaces (100, 200, 400, 800, ...), and different number of parameters in utility (1, 2, 3, 4).
- Ratio CPU-time NFXP/NPL is determined by ratio # PI NFXP / NPL. It is very stable over different sample sizes, and sizes of the state space, but it varies with number of parameters to estimate.
- **Ratio CPU time NFXP / NPL** is:
  - 6.9 with $K = 1$ parameters
  - 10.4 with $K = 2$ parameters
  - 13.2 with $K = 3$ parameters
  - 15.3 with $K = 4$ parameters

3. DYNAMIC DISCRETE GAMES

MODEL

- Time is discrete and indexed by t.
- The game is played by N firms that we index by i.
- Each player takes an action y_{it} to maximize the expected and discounted flow of payoffs:

$$\mathbb{E}_t \left(\sum_{s=0}^{\infty} \beta^s \; \pi_{i,t+s} \right)$$

- Payoff π_{it} depends on the player i's own action y_{it}, other players' actions, $\mathbf{y}_{-it} = \{y_{jt} : j \neq i\}$, and a vector of state variables $\mathbf{x}_t$.

$$\pi_{it} = \pi_i \left(y_{it}, \mathbf{y}_{-it}, \mathbf{x}_t \right)$$

Markov Perfect Equilibrium: Definition

- A key condition in this solution concept is that **players' strategies are functions of only payoff-relevant state variables**, $\boldsymbol{x}_t$.
- Let $\boldsymbol{\alpha} = \{\alpha_i(\mathbf{x}_t) : i = 1, 2, ..., N\}$ be a set of strategy functions.
- A MPE is an N-tuple of strategy functions α such that every player is maximizing its value given the strategies of the other players.
- For given strategies of the other players, the decision problem of a player is a single-agent dynamic programming (DP) problem.

Markov Perfect Equilibrium: Best Response DP

- Let $V_i^{\boldsymbol{\alpha}}(\mathbf{x}_t)$ be the value function of the DP problem that describes the best response of firm i to the strategies of the other firms in $\boldsymbol{\alpha}$.

- This value function is the unique solution to the Bellman equation:

$$V_i^{\boldsymbol{\alpha}}(\mathbf{x}_t) = \max_{y_{it}} \left\{ \pi_i^{\boldsymbol{\alpha}}(y_{it}, \mathbf{x}_t) + \beta \sum_{\mathbf{x}_{t+1}} V_i^{\boldsymbol{\alpha}}(\mathbf{x}_{t+1}) \ f_i^{\boldsymbol{\alpha}}(\mathbf{x}_{t+1}|y_{it}, \mathbf{x}_t) \right\}$$

- with:

$$\pi_i^{\boldsymbol{\alpha}}(y_{it}, \mathbf{x}_t) \quad = \quad \pi_i\left(y_{it}, \boldsymbol{\alpha}_{-i}(\mathbf{x}_t), \mathbf{x}_t\right)$$

- and:

$$f_i^{\boldsymbol{\alpha}}(\mathbf{x}_{t+1}|y_{it}, \mathbf{x}_t) \quad = \quad F_x(\mathbf{x}_{t+1}|y_{it}, \boldsymbol{\alpha}_{-i}(\mathbf{x}_t), \mathbf{x}_t)$$

Markov Perfect Equilibrium: Definition

- A Markov perfect equilibrium (MPE) is an N-tuple of strategy functions $\boldsymbol{\alpha}$ such that for any player i and for any $\mathbf{x}_t$, we have that:

$$\alpha_i(\mathbf{x}_t) = \arg\max_{y_{it}} \; v_i^{\boldsymbol{\alpha}}(y_{it}, \mathbf{x}_t)$$

with $v_i^{\boldsymbol{\alpha}}(y_{it}, \mathbf{x}_t)$ being the **Conditional-Choice Value Function:**

$$v_i^{\boldsymbol{\alpha}}(y_{it}, \mathbf{x}_t) \equiv \pi_i^{\boldsymbol{\alpha}}(y_{it}, \mathbf{x}_t) + \beta \sum_{\mathbf{x}_{t+1}} V_i^{\boldsymbol{\alpha}}(\mathbf{x}_{t+1}) \; f_i^{\boldsymbol{\alpha}}(\mathbf{x}_{t+1} | y_{it}, \mathbf{x}_t)$$

MPE AS FIXED POINT IN CCPs

- Let $\mathbf{P}_i \in [0,1]^{(J+1)|\mathcal{X}|}$ be vector of CCPs for player i.

- A MPE can be described as an N-tuple of CCP vectors, one for each player, such that, for every i:

$$\mathbf{P}_i = \Psi_{\theta,i}(\mathbf{P}_i, \mathbf{P}_{-i})$$

- $\Psi_{\boldsymbol{\theta},\boldsymbol{i}}(.)$ is a **Policy Iteration** mapping similar to the one defined for single-agent model.

- $\Psi_{\boldsymbol{\theta},\boldsymbol{i}}(.)$ is the **composition of two mappings**:

$$\Psi_{\boldsymbol{\theta},\boldsymbol{i}}(\mathbf{P}_i, \mathbf{P}_{-i}) \equiv \Lambda_{\boldsymbol{\theta},\boldsymbol{i}}\left(\varphi_{\boldsymbol{\theta},\boldsymbol{i}}\left(\mathbf{P}_i, \mathbf{P}_{-i}\right)\right)$$

  - $\varphi_{\boldsymbol{\theta},\boldsymbol{i}}(.)$ is the **Policy Valuation mapping**.
  - $\Lambda_{\boldsymbol{\theta},\boldsymbol{i}}(.)$ is the **Policy Improvement mapping**

MPE AS FIXED POINT IN CCPs [2/2]

- **Policy Valuation mapping** $\varphi_{\theta,i}\left(\mathbf{P}_i,\mathbf{P}_{-i}\right)$ returns the vector of values $\mathbf{V}_i$ for player i if all the players behave according to their CCPs in $\left(\mathbf{P}_i,\mathbf{P}_{-i}\right)$.

$$\varphi_{\theta}\left(\mathbf{P}_i,\mathbf{P}_{-i}\right)=\mathbf{V}_i=$$

$$\left[\mathbf{I}-\beta\sum_{y_i=0}^{J}\mathbf{P}_i(y)*\mathbf{F}_x^{\mathbf{P}_{-i}}(y_i)\right]^{-1}\left[\sum_{y_i=0}^{J}\mathbf{P}_i(y_i)*\left(\mathbf{\Pi}_{\theta}^{\mathbf{P}_{-i}}(y_i)+\gamma-\ln\mathbf{P}_i(y_i)\right)\right]$$

- **Policy Improvement mapping** $\Lambda_{\theta,i}\left(\mathbf{V}_i\right)$ returns the vector of CCPs $\mathbf{P}_i$ that is optimal for player i if future values are given by vector $\mathbf{V}_i$.

$$\Lambda_{\theta,i}\left(y_i,\mathbf{V}_i\right)=\mathbf{P}_i(y_i)=\frac{\exp\left\{\mathbf{\Pi}_{\theta}^{\mathbf{P}_{-i}}(y_i)+\beta\ \mathbf{F}_x^{\mathbf{P}_{-i}}(y_i)\ \mathbf{V_i}\right\}}{\sum_{j=0}^{J}\exp\left\{\mathbf{\Pi}_{\theta}^{\mathbf{P}_{-i}}(j)+\beta\ \mathbf{F}_x^{\mathbf{P}_{-i}}(j)\ \mathbf{V_i}\right\}}$$

SOLUTION METHOD: POLICY ITERATIONS

- Let $\mathbf{P}^0 \equiv \{\mathbf{P}_i^0 : \text{for any } i\}$ be arbitrary vector of CCPs.

- At iteration n, for any player i:

$$\mathbf{P}_i^n = \Psi_i\left(\mathbf{P}_i^{n-1}, \mathbf{P}_{-i}^{n-1}\right)$$

- We check for convergence:

$$\begin{cases} \text{if } \left\|\mathbf{P}^n - \mathbf{P}^{n-1}\right\| \leq \kappa & \text{then} \quad \mathbf{P}^n \text{ is a MPE} \\ \text{if } \left\|\mathbf{P}^n - \mathbf{P}^{n-1}\right\| > \kappa & \text{then} \quad \text{Proceed to iteration } n+1 \end{cases}$$

where κ is a small positive constant, e.g., $\kappa = 10^{-6}$.

- **Convergence is NOT guaranteed**. This is a serious limitation.

- Furthermore, in general, the game can have **multiple MPE**.

Some challenges in the computation of MPE

- A MPE is a vector $\mathbf{P}$ with dimension $N\,|\mathcal{Y}|^N\,|\mathcal{X}|$ such that:

$$\mathbf{P} = \Psi(\mathbf{P})$$

- By Brower's Theorem, there exists at least on MPE.

- **[1]** In general, $\Psi(.)$ is **NOT a contraction**. Fixed-point iterations do not guarantee convergence.

- **[2] Multiple equilibria.** In general, $\Psi(.)$ can have many MPE.

- **[3] Curse of dimensionality:** The dimension $|\mathcal{X}|$ increases exponentially with N.

  Example: Game of capital investment. A firm's capital stock can take 10 values. With $N = 6$ firms, we have that $|\mathcal{X}| > 1$ million.

4. FULL SOLUTION ESTIMATION METHODS FOR DYNAMIC GAMES

MLE-NFXP with equilibrium uniqueness

- Rust (1987) NFXP algorithm is a gradient method to obtain MLE.

- Originally proposed for single-agent models, it has been applied to the estimation of games with unique equilibrium for every $\boldsymbol{\theta}$.

- Let $\{P_i(a_i|\mathbf{x}, \boldsymbol{\theta}) : i \in \mathcal{I}\}$ be the equilibrium CCPs associated with $\boldsymbol{\theta}$. The **full log-likelihood function** is: $\ell(\boldsymbol{\theta}) = \sum_{m=1}^{M} \ell_m(\boldsymbol{\theta})$, where $\ell_m(\boldsymbol{\theta})$ is the contribution of market m:

$$\ell_m(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{t=1}^{T} \log P_i(a_{imt}|\mathbf{x}_{mt}, \boldsymbol{\theta}) + \log f_x(\mathbf{x}_{m,t+1}|\boldsymbol{a}_{mt}, \mathbf{x}_{mt}, \boldsymbol{\theta}_f)$$

MLE-NFXP with equilibrium uniqueness [2]

- NFXP combines BHHH iterations (**outer algorithm**) with equilibrium solution algorithm (**inner algorithm**) for each trial value $\boldsymbol{\theta}$.

- A BHHH iteration is:

$$\widehat{\boldsymbol{\theta}}_{k+1} = \widehat{\boldsymbol{\theta}}_k + \left(\sum_{m=1}^{M} \frac{\partial \ell_m(\widehat{\boldsymbol{\theta}}_k)}{\partial \boldsymbol{\theta}} \frac{\partial \ell_m(\widehat{\boldsymbol{\theta}}_k)}{\partial \boldsymbol{\theta}'} \right)^{-1} \left(\sum_{m=1}^{M} \frac{\partial \ell_m(\widehat{\boldsymbol{\theta}}_k)}{\partial \boldsymbol{\theta}} \right)$$

- The score vector $\partial \ell_m(\widehat{\boldsymbol{\theta}}_k) / \partial \boldsymbol{\theta}$ depends on $\partial \log P_i(a_{imt} | \mathbf{x}_{mt}, \widehat{\boldsymbol{\theta}}_k) / \partial \boldsymbol{\theta}$. To obtain these derivatives, the inner algorithm of NFXP solves for the equilibrium CCPs given $\widehat{\boldsymbol{\theta}}_k$.

MLE-NFXP with multiple equilibria

- With Multiple Equilibria, $\ell_m(\boldsymbol{\theta})$ is not a function but a correspondence.

- To define the MLE in a model with multiple equilibria, it is convenient to define an *extended* or **Pseudo Likelihood function**.

- For arbitrary values of $\boldsymbol{\theta}$ and firms' CCPs $\mathbf{P}$, define:

$$Q(\boldsymbol{\theta}, \mathbf{P}) = \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T} \log \Psi_i(a_{imt} \mid \mathbf{x}_{mt}, \boldsymbol{\theta}, \mathbf{P})$$

where Ψ_i is the *best response probability function*.

MLE-NFXP with multiple equilibria [2]

- A modified version of NFXP can be applied to obtain the MLE in games with multiple equilibria.

- The MLE is the pair $(\widehat{\boldsymbol{\theta}}_{MLE}, \widehat{\mathbf{P}}_{MLE})$ that maximizes the Q subject to the constraint that CCPs are equilibrium strategies associated:

$$(\widehat{\boldsymbol{\theta}}_{MLE}, \widehat{\mathbf{P}}_{MLE}, \widehat{\boldsymbol{\lambda}}_{MLE}) = \arg \max_{(\boldsymbol{\theta}, \mathbf{P}, \boldsymbol{\lambda})} \; Q(\boldsymbol{\theta}, \mathbf{P}) + \boldsymbol{\lambda}' \left[\mathbf{P} - \Psi(\boldsymbol{\theta}, \mathbf{P}) \right]$$

- The F.O.C. are the Lagrangian equations:

$$\left\{ \begin{array}{rcl} \widehat{\mathbf{P}}_{MLE} - \Psi(\widehat{\boldsymbol{\theta}}_{MLE}, \widehat{\mathbf{P}}_{MLE}) & = & \mathbf{0} \\ \nabla_{\boldsymbol{\theta}} Q(\widehat{\boldsymbol{\theta}}_{MLE}, \widehat{\mathbf{P}}_{MLE}) - \widehat{\boldsymbol{\lambda}}'_{MLE} \; \nabla_{\boldsymbol{\theta}} \Psi(\widehat{\boldsymbol{\theta}}_{MLE}, \widehat{\mathbf{P}}_{MLE}) & = & \mathbf{0} \\ \nabla_{\mathbf{P}} Q(\widehat{\boldsymbol{\theta}}_{MLE}, \widehat{\mathbf{P}}_{MLE}) - \widehat{\boldsymbol{\lambda}}'_{MLE} \; \nabla_{\mathbf{P}} \Psi(\widehat{\boldsymbol{\theta}}_{MLE}, \widehat{\mathbf{P}}_{MLE}) & = & \mathbf{0} \end{array} \right.$$

MLE-NFXP with multiple equilibria [3]

- A Newton method can be used to obtain a root of this system of Lagrangian equations.
- A key computational problem is the very high dimensionality of this system of equations.
- The most costly part of this algorithm is the calculation of the Jacobian matrix $\nabla_{\mathbf{P}} \Psi(\widehat{\theta}, \widehat{\mathbf{P}})$. In dynamic games, in general, this is not a sparse matrix, and can contain billions or trillions of elements.
- The evaluation of the best response mapping $\Psi(\theta, \mathbf{P})$ for a new value of $\mathbf{P}$ requires solving for a valuation operator and solving a system of equations with the same dimension as $\mathbf{P}$.

Nested Pseudo Likelihood (NPL)

- Imposes equilibrium restrictions but does NOT require:

  - Repeatedly solving for MPE for each trial value of $\boldsymbol{\theta}$ (as NFXP)
  - Computing $\nabla_{\mathbf{P}} \Psi(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{P}})$ (as NFXP and MPEC)

- A NPL $(\widehat{\boldsymbol{\theta}}_{NPL}, \widehat{\mathbf{P}}_{NPL})$, that satisfy two conditions:

  (1) given $\widehat{\mathbf{P}}_{NPL}$, $\widehat{\boldsymbol{\theta}}_{NPL} = \arg\max_{\boldsymbol{\theta}} \ Q(\boldsymbol{\theta}, \widehat{\mathbf{P}}_{NPL})$;

  (2) given $\widehat{\boldsymbol{\theta}}_{NPL}$, $\widehat{\mathbf{P}}_{NPL} = \Psi(\widehat{\boldsymbol{\theta}}_{NPL}, \widehat{\mathbf{P}}_{NPL})$.

- The NPL estimator is consistent and asymptotically normal under the same regularity conditions as the MLE. For dynamic games, the NPL estimator has larger asymptotic variance than the MLE.

Differences ML and NPL estimators

NPL root	ML root
1. $\hat{\mathbf{P}} - \mathbf{\Psi}\left(\hat{\theta}, \hat{\mathbf{P}}\right) = \mathbf{0}$	1. $\hat{\mathbf{P}} = \mathbf{\Psi}\left(\hat{\theta}, \hat{\mathbf{P}}\right)$
2. $\nabla_{\boldsymbol{\theta}} \hat{Q}\left(\hat{\theta}, \hat{\mathbf{P}}\right) = \mathbf{0}$	2. $\nabla_{\boldsymbol{\theta}} \hat{Q}\left(\hat{\theta}, \hat{\mathbf{P}}\right) - \lambda' \nabla_{\boldsymbol{\theta}} \mathbf{\Psi}\left(\hat{\theta}, \hat{\mathbf{P}}\right) = \mathbf{0}$
	3. $\nabla_{\mathbf{P}} \hat{Q}\left(\hat{\theta}, \hat{\mathbf{P}}\right) - \lambda' \nabla_{\mathbf{P}} \mathbf{\Psi}\left(\hat{\theta}, \hat{\mathbf{P}}\right) = \mathbf{0}$

- In **single-agent** DDC models, the two sets of conditions are equivalent [because the zero Jacobian property].
- In **dynamic games**, the two sets of conditions are not equivalent.

Algorithms to Implement the NPL Estimator

NPL FIXED POINT ALGORITHM

- An algorithm to compute the NPL is the **NPL fixed point algorithm**.
- Starting with an initial $\widehat{\mathbf{P}}_0$, at iteration $k \geq 1$:

  (Step 1) given $\widehat{\mathbf{P}}_{k-1}$, $\widehat{\boldsymbol{\theta}}_k = \arg\max_{\boldsymbol{\theta}} \ Q(\boldsymbol{\theta}, \widehat{\mathbf{P}}_{k-1})$;

  (Step 2) given $\widehat{\boldsymbol{\theta}}_k$, $\widehat{\mathbf{P}}_k = \Psi(\widehat{\boldsymbol{\theta}}_k, \widehat{\mathbf{P}}_{k-1})$.

- Step 1 is very simple in most applications, as it is equivalent to obtaining the MLE in a static single-agent discrete choice model.
- Step 2 is equivalent to solving once a system of linear equations with the same dimension as $\mathbf{P}$.
- A limitation of this fixed point algorithm is that **convergence is not guaranteed**. An alternative algorithm that has been used to compute NPL is a **Spectral Residual algorithm**.

SPECTRAL ALGORITHM: Motivation

- NPL estimator is a solution to a stochastic system of nonlinear equations

- A possible approach: **Newton method**
  - Positive: Local convergence guaranteed despite fixed point instability
  - Negative: Computing and inversion of high-dimensional Jacobians

- **Spectral algorithm**: Derivative free non-monotone spectral residual methods
  - Initially proposed by Barzilai and Borwein (1988).
  - Applicable to high-dimensional settings
  - La Cruz, Martinez, and Raydan (2006) propose non-monotone line search as a globalization strategy.

Spectral Algorithm

- Some definitions:

$$\hat{\boldsymbol{\phi}}\left(\mathbf{P}\right) = \mathbf{P} - \boldsymbol{\Psi}\left(\widehat{\theta}(\mathbf{P}), \mathbf{P}\right)$$

$$\Delta\hat{\mathbf{P}}_k \equiv \hat{\mathbf{P}}_k - \hat{\mathbf{P}}_{k-1}$$

$$\Delta\hat{\boldsymbol{\phi}}\left(\hat{\mathbf{P}}_k\right) \equiv \hat{\boldsymbol{\phi}}\left(\hat{\mathbf{P}}_k\right) - \hat{\boldsymbol{\phi}}\left(\hat{\mathbf{P}}_{k-1}\right)$$

- Spectral algorithm iteration: $\hat{\mathbf{P}}_{k+1} = \hat{\mathbf{P}}_k - \alpha_k \; \hat{\boldsymbol{\phi}}\left(\hat{\mathbf{P}}_k\right)$ with

$$\alpha_k = \frac{\Delta\hat{\mathbf{P}}'_k \; \Delta\hat{\mathbf{P}}_k}{\Delta\hat{\mathbf{P}}'_k \; \Delta\hat{\boldsymbol{\phi}}\left(\hat{\mathbf{P}}_k\right)}$$

- The spectral step-length α_k is a scalar!