

Estimating **STATIC** discrete-choice games of incomplete information and Multiplicity of equilibria

Bertel Schjerning, University of Copenhagen

2025 Econometric Society Summer School in Dynamic Structural Econometrics

University of Hong Kong
Dec 8th - Dec 13th , 2025

Estimating Games of Incomplete Information

 Pakes, Ostrovsky and Berry, 2007

Various 2-Step CCP estimators: pseudo-ML, MoM, $\min \chi^2$

 Aguirregabiria and Mira, 2007

Recursive 2-Step CCP estimator, Nested Pseudo-Likelihood (NPL)

 Pesendorfer and Schmidt-Dengler, 2008

2-Step least squares

 Kasahara and Shimotsu, 2012

Modified (dampened) NPL

 Su, 2013 and Egesdal, Lai and Su, 2015

Constrained optimization (MPEC)

 Blevins and Dearing, 2025

Efficient Pseudo-Likelihood (EPL)

Example: Static Entry Game

- ▶ Two firms: a and b
- ▶ Actions: each firm has two possible actions:

$$d_a = \begin{cases} 1, & \text{if firm } \underline{a} \text{ choose to enter the market} \\ 0, & \text{if firm } \underline{a} \text{ choose not to enter the market} \end{cases} \quad (1)$$

$$d_b = \begin{cases} 1, & \text{if firm } \underline{b} \text{ choose to enter the market} \\ 0, & \text{if firm } \underline{b} \text{ choose not to enter the market} \end{cases} \quad (2)$$

Example: Static Entry Game of Incomplete Information

Utility: Ex-post payoff to firms

$$u_a(d_a, d_b, x_a, \epsilon_a) = \begin{cases} [\alpha + d_b * (\beta - \alpha)]x_a + \epsilon_{a1}, & \text{if } d_a = 1 \\ 0 + \epsilon_{a0}, & \text{if } d_a = 0 \end{cases}$$
$$u_b(d_a, d_b, x_b, \epsilon_b) = \begin{cases} [\alpha + d_a * (\beta - \alpha)]x_b + \epsilon_{b1}, & \text{if } d_b = 1 \\ 0 + \epsilon_{b0}, & \text{if } d_b = 0 \end{cases}$$

- ▶ (α, β) : structural parameters to be estimated
- ▶ (x_a, x_b) : firms' observed types; **common knowledge**
- ▶ $\epsilon_a = (\epsilon_{a0}, \epsilon_{a1}), \epsilon_b = (\epsilon_{b0}, \epsilon_{b1})$: firms' unobserved types, **private information**
- ▶ (ϵ_a, ϵ_b) are observed only by each firm, but not by their opponent firm nor by the econometrician

Example: Static Entry Game of Incomplete Information

- ▶ Assume the error terms (ϵ_a, ϵ_b) have EV Type I distribution
- ▶ A Bayesian Nash equilibrium (p_a, p_b) satisfies

$$\begin{aligned} p_a &= \frac{\exp[p_b \beta x_a + (1 - p_b) \alpha x_a]}{1 + \exp[p_b \beta x_a + (1 - p_b) \alpha x_a]} \\ &= \frac{1}{1 + \exp[-\alpha x_a + p_b x_a (\alpha - \beta)]} \\ &\equiv \Psi_a(p_b, x_a; \alpha, \beta) \end{aligned}$$

$$\begin{aligned} p_b &= \frac{1}{1 + \exp[-\alpha x_b + p_a x_b (\alpha - \beta)]} \\ &\equiv \Psi_b(p_a, x_b; \alpha, \beta) \end{aligned}$$

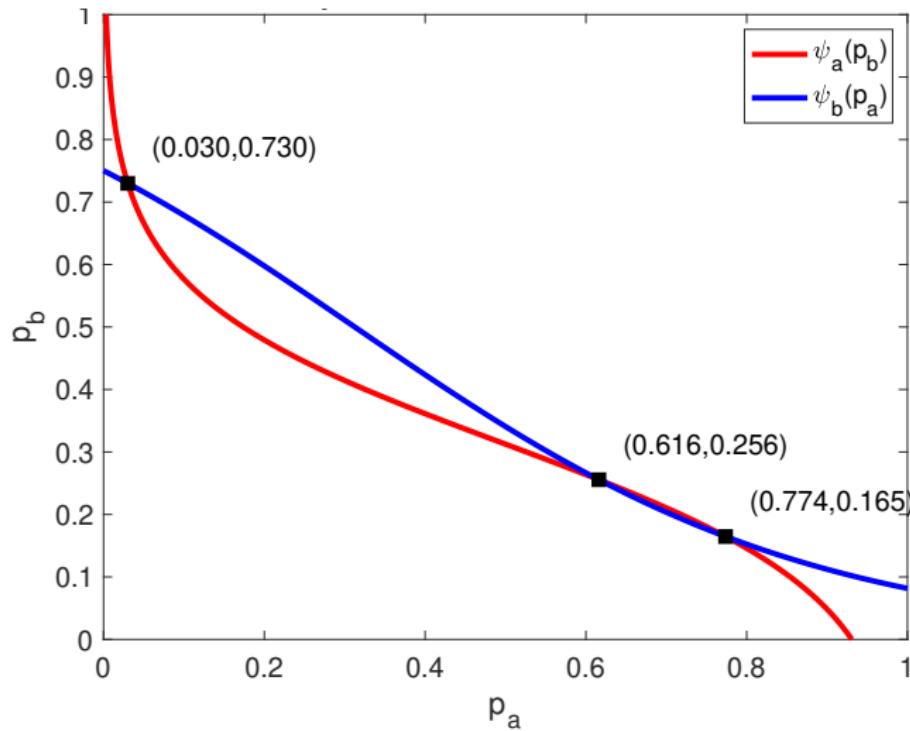
Static Game Example: Parameters

We consider a very contestable game throughout

- ▶ Monopoly profits: $\alpha * x_j = 5 * x_j$
- ▶ Duopoly profits: $\beta * x_j = -11 * x_j$
- ▶ Firm types: $(x_a, x_b) = (0.52, 0.22)$

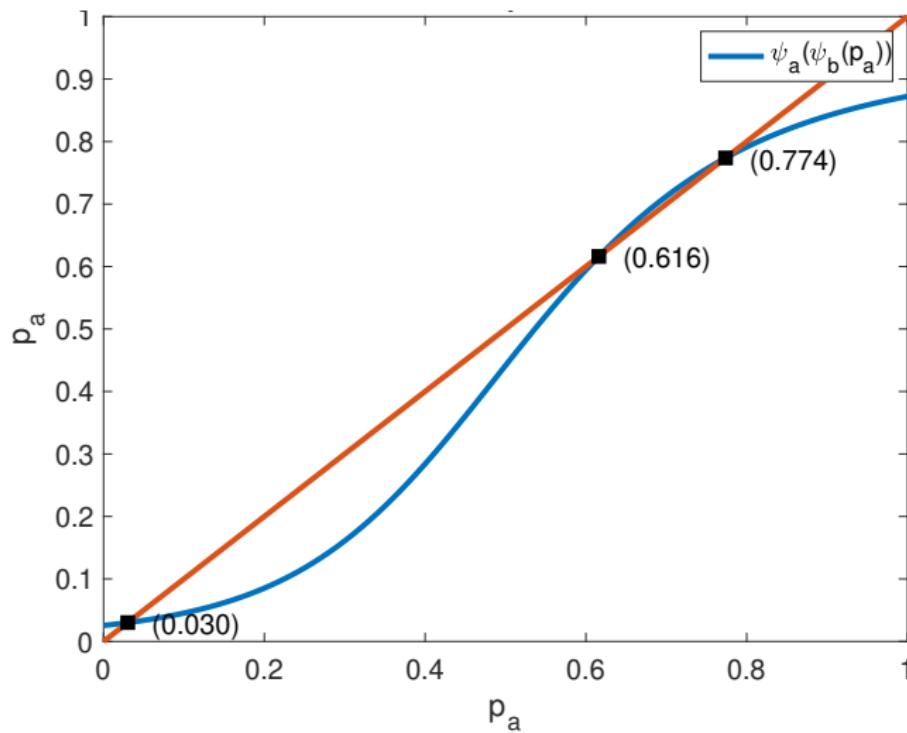
Static Game Example: Three Bayesian Nash Equilibria

Figure: Equilibria at intersections of best response functions



Static Game Example: Solving for Equilibria

Figure: Fixed points on second order best response function



Static Game Example: Solving for Equilibria

Solution method: Combination of successive approximations and bisection algorithm

Succesive approximations (SA)

- ▶ Converge to nearest stable equilibrium.
- ▶ Start SA at $p_a = 0$ and $p_a = 1$.
- ▶ Unique equilibrium ($K=1$): SA will converge to it from anywhere.
- ▶ Three equilibria ($K=3$): Two will be stable, and one will be unstable.
- ▶ More equilibria ($K>3$): Not in this model.

Bisection method

- ▶ Use this to find the unstable equilibrium (if $K=3$).
- ▶ The bisection method that repeatedly bisects an interval and then selects a subinterval in which the fixed point (or root) must lie.
- ▶ The two stable equilibria, defines the initial interval to search over.
- ▶ The bisection method is a very simple and robust method, but it is also relatively slow.

Static Game Example: Data Generation and Identification

- ▶ Data Generating Process (DGP): the data are generated by a single equilibrium
- ▶ The two players use the **same** equilibrium to play 1000 times
- ▶ Data: $X = \{(d_a^i, d_b^i)_{i=1}^{1000}, (x_a, x_b) = (0.52, 0.22)\}$
- ▶ Given data X , we want to recover structural parameters α and β

Static Game Example: Maximum Likelihood Estimation

- Maximize the likelihood function

$$\begin{aligned} \max_{\alpha, \beta} \quad & \log \mathcal{L}(p_a(\alpha, \beta); X) \\ = \quad & \sum_{i=1}^N (d_a^i * \log(p_a(\alpha, \beta)) + (1 - d_a^i) \log(1 - p_a(\alpha, \beta))) \\ + \quad & \sum_{i=1}^N (d_b^i * \log(p_b(\alpha, \beta)) + (1 - d_b^i) \log(1 - p_b(\alpha, \beta))) \end{aligned}$$

- $p_a(\alpha, \beta)$ and $p_b(\alpha, \beta)$ are the solutions of the Bayesian-Nash Equilibrium equations

$$\begin{aligned} p_a &= \frac{1}{1 + \exp[-\alpha x_a + p_b x_a (\alpha - \beta)]} \equiv \Psi_a(p_b, x_a; \alpha, \beta) \\ p_b &= \frac{1}{1 + \exp[-\alpha x_b + p_a x_b (\alpha - \beta)]} \equiv \Psi_b(p_a, x_b; \alpha, \beta) \end{aligned}$$

Static Game Example: MLE via NFXP

- ▶ Outer Loop
 - ▶ Choose (α, β) to maximize the likelihood function

$$\log \mathcal{L}(p_a(\alpha, \beta), p_b(\alpha, \beta); X)$$

- ▶ Inner loop:
 - ▶ For a given (α, β) , solve the BNE equations for **ALL** equilibria:
 $(p_a^k(\alpha, \beta), p_b^k(\alpha, \beta))$, $k = 1, \dots, K$
 - ▶ Choose the equilibrium that gives the highest likelihood value:

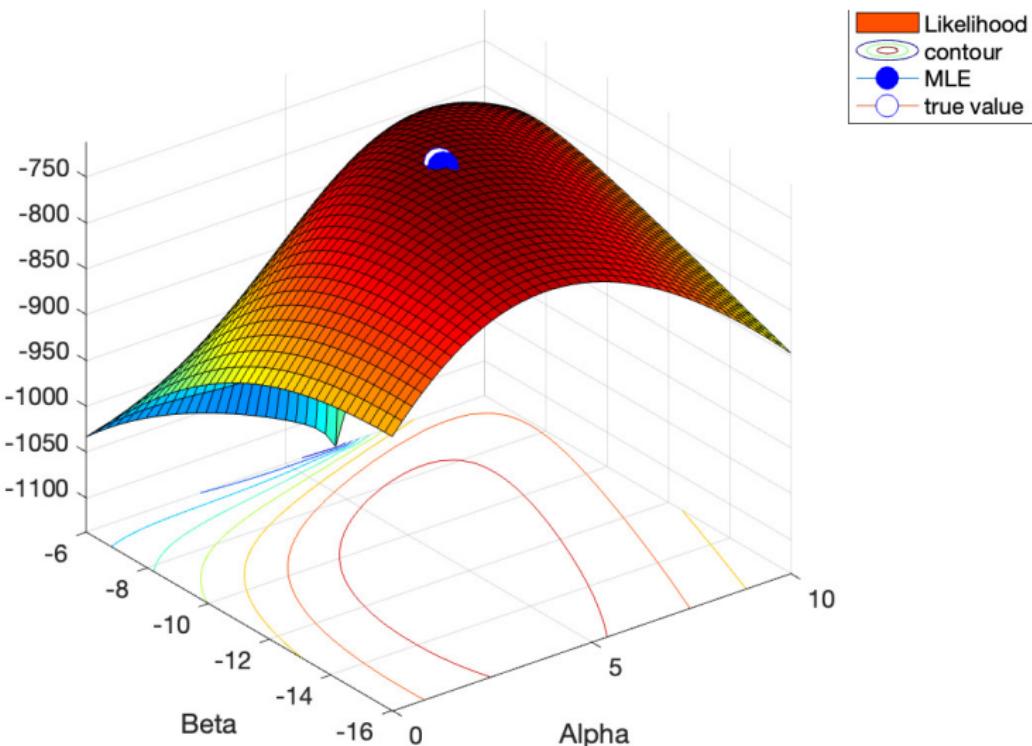
$$k^* = \arg \max_{k=1, \dots, K} \log \mathcal{L}(p_a^k(\alpha, \beta), p_b^k(\alpha, \beta); X)$$

such that

$$(p_a(\alpha, \beta), p_b(\alpha, \beta)) = (p_a^{k^*}(\alpha, \beta), p_b^{k^*}(\alpha, \beta))$$

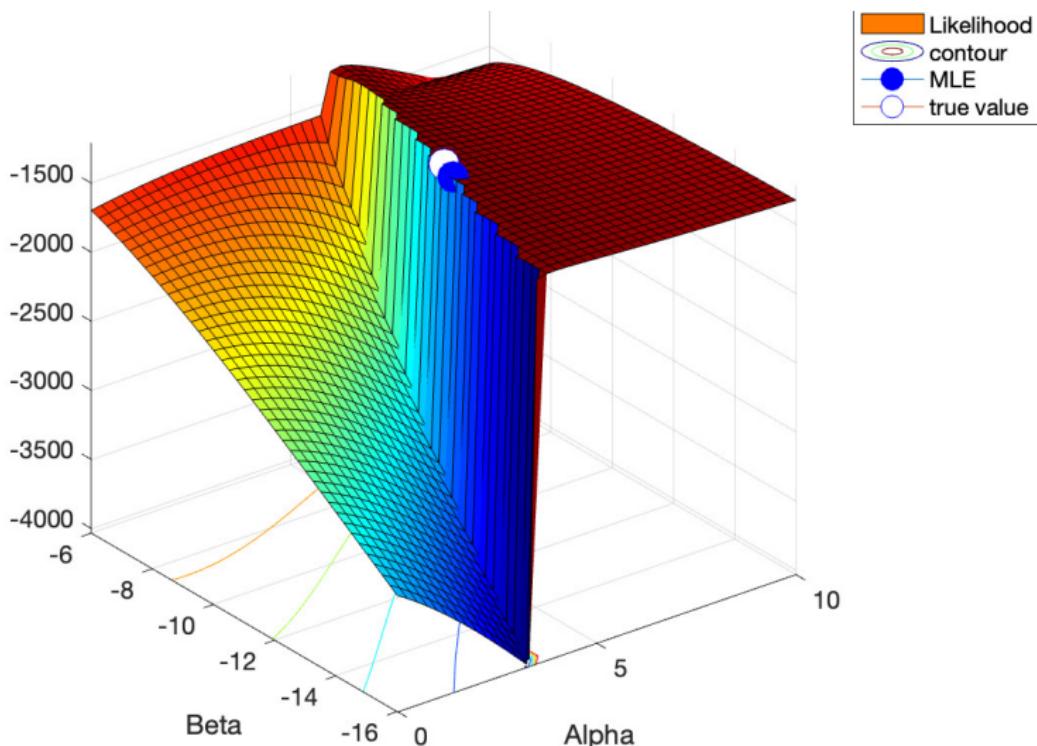
NFXP's Likelihood as a Function of (α, β) - Eq 1

Figure: Data generated from equilibrium 1



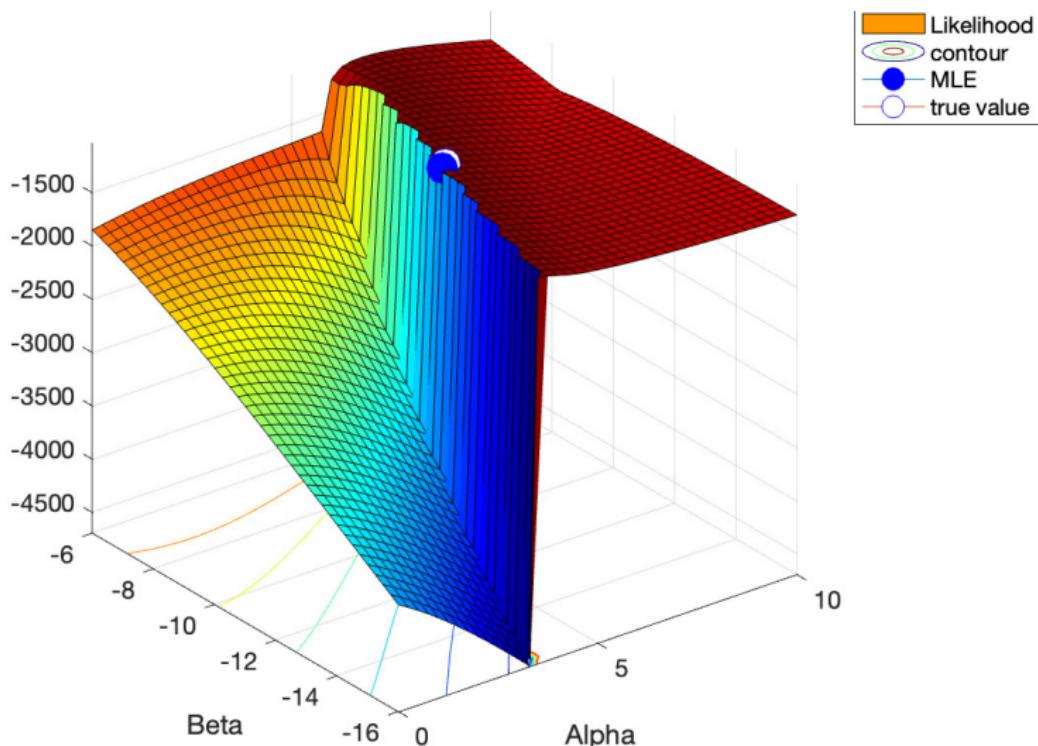
NFXP's Likelihood as a Function of (α, β) - Eq 2

Figure: Data generated from equilibrium 2



NFXP's Likelihood as a Function of (α, β) - Eq 3

Figure: Data generated from equilibrium 3



Static Game Example: MLE via MPEC

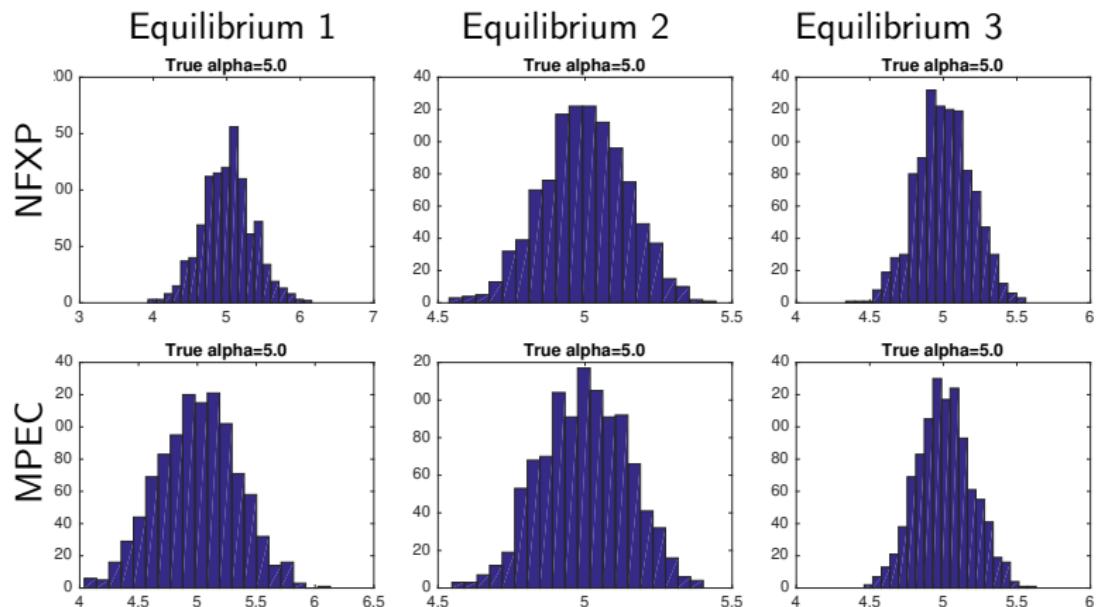
- Maximize the likelihood function

$$\begin{aligned} \max_{\alpha, \beta, p_a, p_b} & \log \mathcal{L}(p_a; X) \\ = & \sum_{i=1}^N (d_a^i * \log(p_a) + (1 - d_a^i) \log(1 - p_a)) \\ + & \sum_{i=1}^N (d_b^i * \log(p_b) + (1 - d_b^i) \log(1 - p_b)) \end{aligned}$$

- Subject to p_a and p_b are the solutions of the Bayesian-Nash Equilibrium equations

$$\begin{aligned} p_a &= \frac{1}{1 + \exp[-\alpha x_a + p_b x_a (\alpha - \beta)]} \\ p_b &= \frac{1}{1 + \exp[-\alpha x_b + p_a x_b (\alpha - \beta)]} \\ 0 &\leq p_a, p_b \leq 1 \end{aligned}$$

Monte Carlo Results



Estimates of parameter α

Data generated from each of the three equilibria

Least Square Estimators

Pesendofer and Schmidt-Dengler (2008)

- ▶ Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$ from the data
- ▶ Step 2: Solve

$$\min_{\alpha, \beta} \left\{ (\hat{p}_a - \Psi_a(\hat{p}_b, x_a; \alpha, \beta))^2 + (\hat{p}_b - \Psi_b(\hat{p}_a, x_b; \alpha, \beta); X))^2 \right\}$$

For dynamic games, Markov perfect equilibrium conditions are characterized by

$$p = \Psi(p, \theta)$$

- ▶ Step 1: Estimate \hat{p} from the data
- ▶ Step 2: Solve

$$\min_{\alpha, \beta} [\hat{p} - \Psi(\hat{p}; \theta)]' W [\hat{p} - \Psi(\hat{p}; \theta)]'$$

2-Step Methods: Pseudo Maximum Likelihood

In 2-step methods

- ▶ Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$ from the data
- ▶ Step 2: Solve

$$\max_{\alpha, \beta, p_a, p_b} \log \mathcal{L}(p_a, p_b; X)$$

subject to

$$\begin{aligned} p_a &= \Psi_a(\hat{p}_a, x_a; \alpha, \beta) \\ p_b &= \Psi_b(\hat{p}_b, x_b; \alpha, \beta) \\ 0 &\leq p_a^m, p_b^m \leq 1, m = 1, \dots, M \end{aligned}$$

Or equivalently

- ▶ Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$ from the data
- ▶ Step 2: Solve

$$\max_{\alpha, \beta} \log \mathcal{L}(\Psi_a(\hat{p}_b, x_a; \alpha, \beta), \Psi_b(\hat{p}_a, x_b; \alpha, \beta); X)$$

Nested Pseudo Likelihood (NPL)

 Aguirregabiria and Mira (2007)

NPL iterates on the 2-step methods

1. Step 1: Estimate $\hat{p}^0 = (\hat{p}_a^0, \hat{p}_b^0)$ from the data, set $k = 0$
2. Step 2:

REPEAT

2.1 Solve

$$\alpha^{k+1}, \beta^{k+1} = \arg \max_{\alpha, \beta} \log \mathcal{L}(\Psi_a(\hat{p}_b^k, x_a; \alpha, \beta), \Psi_b(\hat{p}_a^k, x_b; \alpha, \beta); X)$$

2.2 One best-reply iteration on \hat{p}^k

$$\begin{aligned}\hat{p}_a^{k+1} &= \Psi_a(\hat{p}_a^k, x_a; \alpha^{k+1}, \beta^{k+1}) \\ \hat{p}_b^{k+1} &= \Psi_b(\hat{p}_b^k, x_b; \alpha^{k+1}, \beta^{k+1})\end{aligned}$$

2.3 Let $k := k + 1$;

UNTIL convergence in (α^k, β^k) and $(\hat{p}_a^k, \hat{p}_b^k)$

Some remarks

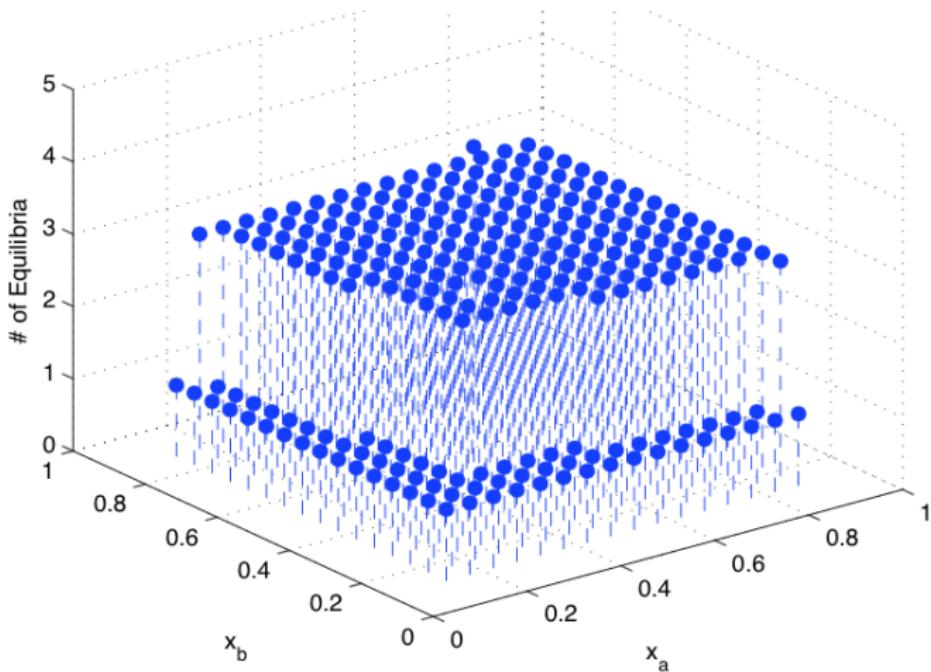
- ▶ NFXP/MPEC implementations of MLE is statistically efficient
 - ▶ NFXP computationally daunting
 - need to **compute ALL equilibria** for each $\theta = (\alpha, \beta)$
 - find maxima of **discontinuous likelihood**
 - ▶ MPEC may be computational lighter
 - objective function and constraints are **smooth** in α, β, p_a, p_b
 - only need to satisfy equilibrium constraints in optimum
 - may get stuck in **local minima** (equilibria not played in the data).
 - ▶ MLE and Multiple equilibria
 - discontinuous likelihood function → non-standard inference
- ▶ Two step estimators
 - ▶ Computationally fast
 - ▶ Inefficient and biased in small samples
(best response mapping not satisfied)
- ▶ NPL and EPL should bridge this gap
 - ▶ NLP can be unstable in games with multiple equilibria
(NPL mapping not contraction)
 - ▶ EPL does not rely on contraction property, but still an iterative algorithm that depends on starting values and estimates of p_a, p_b
- ▶ Problems amplified by orders of magnitude in dynamic games

Estimation with Multiple Markets

- ▶ There 25 different markets, i.e., 25 pairs of observed types (x_a^m, x_b^m) , $m = 1, \dots, 25$
- ▶ The grid on x_a has 5 points equally distributed between the interval $[0.12, 0.87]$, and similarly for x_b
- ▶ Use the same true parameter values: (α_0, β_0)
- ▶ For each market with (x_a^m, x_b^m) , solve BNE conditions for (p_a^m, p_b^m) .
- ▶ There are multiple equilibria in most of 25 markets
- ▶ For each market, we (randomly) choose an equilibrium to generate T data points for that market
- ▶ The equilibrium used to generate data can be different in different markets - we flip a coin at each market.

of Equilibria with Different (x_a^m, x_b^m)

Figure: Number of equilibria



NFXP - Estimation with Multiple Markets

Outer loop:

$$\max_{\alpha, \beta} \log \mathcal{L}(p_a^m(\alpha, \beta), p_b^m(\alpha, \beta); X)$$

Inner loop: For a given values of (α, β) solve BNE equations for ALL equilibria, $k = 1, \dots, K$ at each market, $m = 1, \dots, M$: That is, $p_a^{m,k}(\alpha, \beta)$ and $p_b^{m,k}(\alpha, \beta)$ are the solutions to

$$\begin{aligned} p_a^m &= \Psi_a(p_b^m, x_a^m; \alpha, \beta) \\ p_b^m &= \Psi_b(p_a^m, x_b^m; \alpha, \beta) \\ m &= 1, \dots, M \end{aligned}$$

where we again choose the equilibrium, that gives the highest likelihood value at each market m

$$k^* = \arg \max_{k=1, \dots, K} \log \mathcal{L}(p_a^{m,k}(\alpha, \beta), p_b^{m,k}(\alpha, \beta); X)$$

such that

$$(p_a^m(\alpha, \beta), p_b^m(\alpha, \beta)) = (p_a^{m,k^*}(\alpha, \beta), p_b^{m,k^*}(\alpha, \beta))$$

Estimation with Multiple Markets - MPEC

Constrained optimization formulation

$$\max_{\alpha, \beta, p_a^m, p_b^m} \log \mathcal{L}(p_a^m, p_b^m; X)$$

subject to

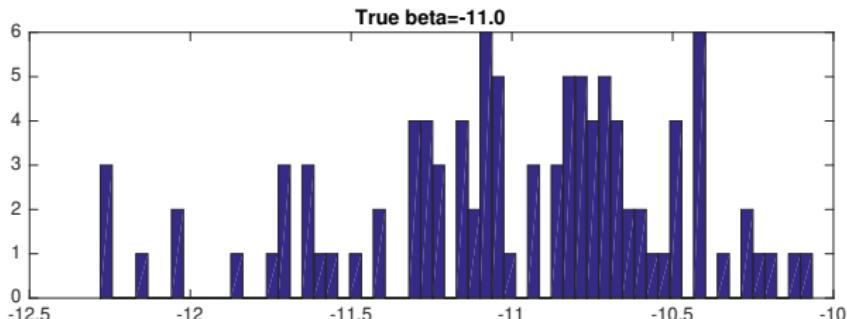
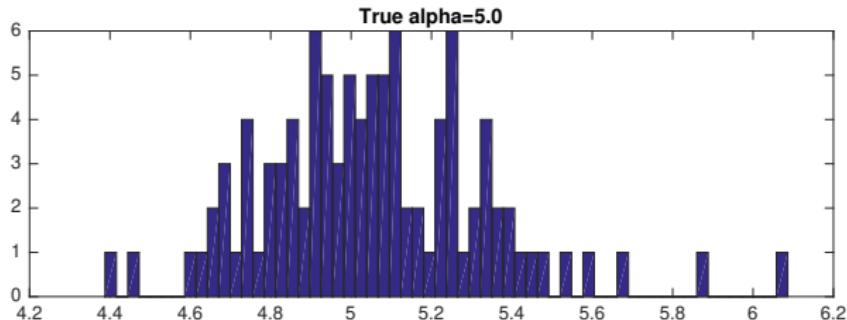
$$\begin{aligned} p_a^m &= \Psi_a(p_b^m, x_a^m; \alpha, \beta) \\ p_b^m &= \Psi_b(p_a^m, x_b^m; \alpha, \beta) \\ 0 &\leq p_a^m, p_b^m \leq 1, \\ m &= 1, \dots, M \end{aligned}$$

- ▶ MPEC does not explicitly solve the BNE equations to find ALL equilibria at each market - for every trial value of parameters.
- ▶ But the number of parameters is much larger.
- ▶ Both MPEC and NFXP are based on Full Information Maximum Likelihood (FIML) estimators.

NFXP: Monte Carlo - Multiple Markets (M=25, T=50)

Starting values $\alpha_0 = \alpha$, $\beta_0 = \beta$

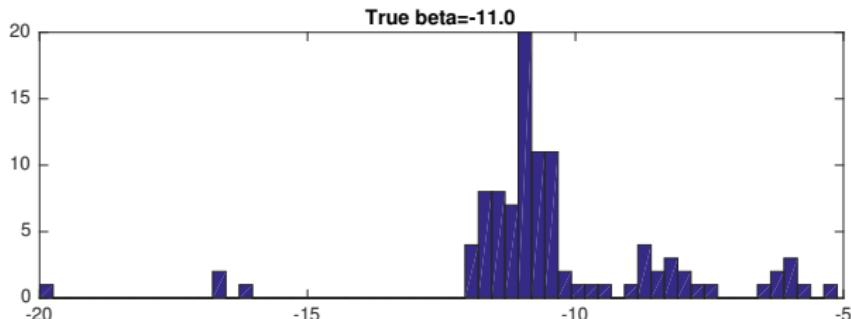
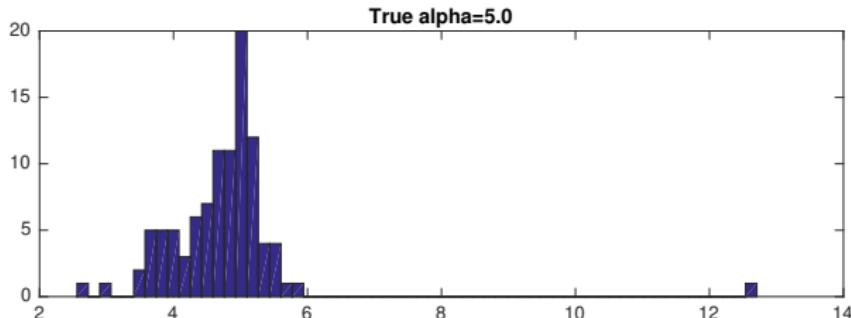
Random equilibrium selection in different markets



MPEC: Monte Carlo - Multiple Markets (M=25, T=50)

Starting values $\alpha_0 = \alpha$, $\beta_0 = \beta$

Random equilibrium selection in different markets



MPEC and NFXP: multiple markets

NFXP:

- ▶ 2 parameters in optimization problem
- ▶ Needs to find ALL equilibria at each market
(very hard in more complex problems)
- ▶ Good full solution methods required

MPEC:

- ▶ $2 + 2M$ parameters in optimization problem
- ▶ Local minima with many markets:
MPEC equilibrium constraints are satisfied and gradients of objective function is within tolerance - but wrong equilibrium selected