Constrained Optimization Approaches to Structural Estimation

CHE-LIN SU
The University of Chicago
Booth School of Business
Che-Lin.Su@ChicagoBooth.edu

ICE 2012
July 16 – 28, 2012
Outline

1. Estimation of Dynamic Programming Models of Individual Behavior
Outline

1. Estimation of Dynamic Programming Models of Individual Behavior
2. Estimation of Demand Systems
Outline

1. Estimation of Dynamic Programming Models of Individual Behavior
2. Estimation of Demand Systems
3. Estimation of Games of Incomplete Information
Part I

Optimization Overview
Unconstrained Optimization: Background

\[
\min \{ f(x) : x \in \mathbb{R}^n \}
\]

- \( f : \mathbb{R}^n \to \mathbb{R} \) smooth (typically \( C^2 \))
- \( x \in \mathbb{R}^n \) finite dimensional (may be large)

Optimality conditions: \( x^* \) local minimizer:

\[
\nabla f(x^*) = 0
\]

Numerical methods: generate a sequence of iterates \( x_k \) such that the gradient test

\[
\|\nabla f(x_k)\| \leq \tau
\]

is eventually satisfied; usually \( \tau = 1.e^{-6} \)

Warning: Any point \( x \) that does NOT satisfy \( \|\nabla f(x)\| \leq \tau \) should NOT be considered as a “solution” or a candidate for the solution.
## Did the solver Find a Solution?

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Func-count</th>
<th>f(x)</th>
<th>Step-size</th>
<th>First-order optimality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>51770.3</td>
<td></td>
<td>5.53e+004</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5165.79</td>
<td>1.80917e-005</td>
<td>1.26e+004</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3604.44</td>
<td>1</td>
<td>9.05e+003</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2482.01</td>
<td>1</td>
<td>6.01e+003</td>
</tr>
<tr>
<td>20</td>
<td>22</td>
<td>209.458</td>
<td>1</td>
<td>150</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>207.888</td>
<td>1</td>
<td>151</td>
</tr>
<tr>
<td>22</td>
<td>24</td>
<td>199.115</td>
<td>1</td>
<td>166</td>
</tr>
<tr>
<td>23</td>
<td>25</td>
<td>188.692</td>
<td>1</td>
<td>217</td>
</tr>
<tr>
<td>24</td>
<td>26</td>
<td>162.908</td>
<td>1</td>
<td>325</td>
</tr>
<tr>
<td>25</td>
<td>27</td>
<td>143.074</td>
<td>1</td>
<td>614</td>
</tr>
<tr>
<td>26</td>
<td>28</td>
<td>129.016</td>
<td>1</td>
<td>320</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>113.675</td>
<td>1</td>
<td>205</td>
</tr>
<tr>
<td>28</td>
<td>30</td>
<td>94.7791</td>
<td>1</td>
<td>184</td>
</tr>
<tr>
<td>29</td>
<td>32</td>
<td>75.7777</td>
<td>0.431713</td>
<td>166</td>
</tr>
<tr>
<td>30</td>
<td>33</td>
<td>71.4657</td>
<td>1</td>
<td>110</td>
</tr>
<tr>
<td>31</td>
<td>34</td>
<td>71.0592</td>
<td>1</td>
<td>55</td>
</tr>
</tbody>
</table>

Optimization terminated: relative infinity-norm of gradient less than options.TolFun.
Generic Nonlinear Optimization Problem

Nonlinear Programming (NLP) problem

\[
\begin{align*}
\text{minimize} & \quad f(x) & \text{objective} \\
\text{subject to} & \quad c(x) = 0 & \text{constraints} \\
& \quad x \geq 0 & \text{variables}
\end{align*}
\]

- \( f : \mathbb{R}^n \to \mathbb{R} \), \( c : \mathbb{R}^n \to \mathbb{R}^m \) smooth (typically \( C^2 \))
- \( x \in \mathbb{R}^n \) finite dimensional (may be large)
- more general \( l \leq c(x) \leq u \) possible
Optimality Conditions for NLP

Constraint qualification (CQ)
Linearizations of $c(x) = 0$ characterize all feasible perturbations

$x^*$ local minimizer & CQ holds $\Rightarrow \exists$ multipliers $y^*$, $z^*$:

$$
\nabla f(x^*) - \nabla c(x^*)^T y^* - z^* = 0
$$
$$
c(x^*) = 0
$$
$$
X^* z^* = 0
$$
$$
x^* \geq 0, \ z^* \geq 0
$$

where $X^* = \text{diag}(x^*)$, thus $X^* z^* = 0 \iff x_i^* z_i^* = 0$
Solving the FOC for NLP

• Nonlinear equations: \( F(w) = 0 \), where \( w = (x, y, z) \) with \( x, z \geq 0 \).
• NLP solvers: generate a sequence of iterates \( w_k \) such that the test

\[ \| \nabla F(w_k) \| \leq \tau \quad \text{with} \quad x_k \geq 0, \; z_k \geq 0 \]

is eventually satisfied; usually \( \tau = 1.e - 6 \). Same warning applies.
• Supply exact derivatives: \( \nabla f(x), \nabla c(x), \nabla^2 L(x, y, z) \),
where is the Lagrangian: \( L(x, y, z) := f(x) - y^T c(x) - z^T x \)
• Concerns: NLP is difficult to solve when \# of variables and \# of constraints are large
• In many applied models, constraint Jacobian \( \nabla c(x) \) and Hessian of the Lagrangian \( \nabla^2 L(x) \) are sparse
• Modern solvers exploit the sparsity structure of \( \nabla c(x) \) and \( \nabla^2 L(x) \)
Structural Estimation

- Great interest in estimating models based on economic structure
  - DP models of individual behavior: Rust (1987)
  - Dynamic stochastic general equilibrium
  - Popularity of structural models in empirical IO and marketing

- Model sophistication introduces computational difficulties
- General belief: Estimation is a major computational challenge because it involves solving the model many times
- Our approach: Formulate structural estimation models as constrained optimization problems and use modern constrained optimization methods and software to solve the models for you
Structural Estimation

- Single-Agent Dynamic Discrete Choice Models
  - Rust (1987): Bus-Engine Replacement Problem
  - Nested-Fixed Point Problem (NFXP)
  - Su and Judd (2012): Constrained Optimization Approach

- Random-Coefficients Logit Demand Models
  - Nested-Fixed Point Problem (NFXP)
  - Dubé, Fox and Su (2012): Constrained Optimization Approach

- Estimating Discrete-Choice Games of Incomplete Information
  - Aguirregabiria and Mira (2007): NPL (Recursive 2-Step)
  - Bajari, Benkard and Levin (2007): 2-Step
  - Pakes, Ostrovsky and Berry (2007): 2-Step
  - Pesendorfer and Schmidt-Dengler (2008): 2-Step
  - Kasahara and Shimotsu (2012): Modified NPL
Optimization and Computation in Structural Estimation

- Optimization and computation often perceived as 2nd-order importance to research agenda
- Typical computational method is Nested Fixed-Point procedure: fixed-point calculation embedded in calculation of objective function
  - compute an “equilibrium”
  - invert a model (e.g. non-linearity in disturbance)
  - compute a value function (i.e. dynamic model)
- Mis-use of optimization can lead to the “wrong answer”
  - naively use canned optimization algorithms – e.g., Matlab’s fminsearch
  - adjust default-settings of solvers to improve speed not accuracy
  - assume there is a unique fixed-point
  - do NOT check or understand solver’s output message!!
  - KNITRO: Locally Optimal Solution Found.
  - FilterSQP: Optimal Solution Found.
  - SNOPT: Optimal Solution Found.
  - Matlab Optimization Toolbox: Optimization terminated ... does NOT tell you much about what happened at the end
First Step in Solving an Estimation Model

- Make sure you have a smooth formulation for the model
  - *smooth* objective function
  - *smooth* constraints

- Use the best available NLP solvers!
  - Many free NLP solvers are crappy; they often fail or even worse, can give you *wrong solutions*
  - Do not attempt to develop numerical algorithms/solvers by yourself
  - You should use solvers developed by “professionals”, i.e., numerical optimization people
  - Best NLP solvers: SNOPT (Stanford), KNITRO (Northwestern), Filter-SQP (Argonne), IPOPT (IBM), PATH (UW-Madison)

- Keys to efficient implementation
  - Supply exact 1st and 2nd order derivatives
  - Supply sparsity pattern for constraint Jacobian and Hessian of the Lagragian
Part II

Estimation of Dynamic Programming Models
Rust (1987): Zurcher’s Data

### Bus #: 5297

<table>
<thead>
<tr>
<th>events</th>
<th>year</th>
<th>month</th>
<th>odometer at replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st engine replacement</td>
<td>1979</td>
<td>June</td>
<td>242400</td>
</tr>
<tr>
<td>2nd engine replacement</td>
<td>1984</td>
<td>August</td>
<td>384900</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>year</th>
<th>month</th>
<th>odometer reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>1974</td>
<td>Dec</td>
<td>112031</td>
</tr>
<tr>
<td>1975</td>
<td>Jan</td>
<td>115223</td>
</tr>
<tr>
<td>1975</td>
<td>Feb</td>
<td>118322</td>
</tr>
<tr>
<td>1975</td>
<td>Mar</td>
<td>120630</td>
</tr>
<tr>
<td>1975</td>
<td>Apr</td>
<td>123918</td>
</tr>
<tr>
<td>1975</td>
<td>May</td>
<td>127329</td>
</tr>
<tr>
<td>1975</td>
<td>Jun</td>
<td>130100</td>
</tr>
<tr>
<td>1975</td>
<td>Jul</td>
<td>133184</td>
</tr>
<tr>
<td>1975</td>
<td>Aug</td>
<td>136480</td>
</tr>
<tr>
<td>1975</td>
<td>Sep</td>
<td>139429</td>
</tr>
</tbody>
</table>
Zurcher’s Bus Engine Replacement Problem

- Each bus comes in for repair once a month
- Bus manager sees
  - $x_t$: mileage at time $t$ since last engine overhaul
  - $\varepsilon_t = [\varepsilon_t(d_t = 0), \varepsilon_t(d_t = 1)]:$ other state variable
- Bus manager chooses between overhaul and ordinary maintenance
  $$d_t = \begin{cases} 
  1, & \text{replacing the engine;} \\
  0, & \text{performing regular maintenance.}
  \end{cases}$$
- Utility per period
  $$u(x_t, d_t, \varepsilon_t; \theta^c, RC) = \nu(x_t, d_t; \theta^c, RC) + \varepsilon_t(d_t)$$
  where
  $$\nu(x_t, d_t, \theta^c, RC) = \begin{cases} 
  -c(x_t, \theta^c) & \text{if } d_t = 0 \\
  -(RC + c(0, \theta^c)) & \text{if } d_t = 1
  \end{cases}$$
- $c(x; \theta^c):$ expected operating costs per period at mileage $x$
- $RC:$ the expected replacement cost to install a new engine, net of any scrap value of the old engine
- The mileage $x$ is reset to 0 after the engine replacement
Zurcher’s Bus Engine Replacement Problem

- Given \((x_t, \varepsilon_t)\), the bus manager solves the DP:

\[
\max \left\{ d_t, d_{t+1}, d_{t+2}, \ldots \right\} \mathbb{E} \left[ \sum_{\tau=t}^{\infty} \beta^{\tau-t} u(x_\tau, d_\tau, \varepsilon_\tau; \theta^c, RC) \right]
\]

- The expectation \(\mathbb{E}\) is taken over the state transition probability

\[p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, d_t; \theta^p)\]

- Value function

\[V(x_t, \varepsilon_t) = \max \left\{ d_t, d_{t+1}, d_{t+2}, \ldots \right\} \mathbb{E} \left[ \sum_{\tau=t}^{\infty} \beta^{\tau-t} u(x_\tau, d_\tau, \varepsilon_\tau; \theta^c, RC) \right]\]

- Econometrician
  - Observes mileage \(x_t\) and decision \(d_t\), but not cost
  - Assumes extreme value distribution for \(\varepsilon_t(d_t)\)

- Structural parameters to be estimated: \(\theta = (\theta^c, RC, \theta^p)\)
  - Coefficients of operating cost function; e.g., \(c(x, \theta^c) = \theta^c_1 x + \theta^c_2 x^2\)
  - Overhaul cost \(RC\)
  - state transition probability \(p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, d_t; \theta^p)\)
Zurcher’s Bus Engine Replacement Problem

- Bellman equation

\[ V(x, \varepsilon) = \max_d \left\{ \nu(x, d; \theta^c, RC') + \varepsilon(d) + \beta \int_{x'} \int_{\varepsilon'} V(x', \varepsilon') p(x', \varepsilon'|x, \varepsilon, d; \theta^p) dx' d\varepsilon' \right\} \]

- Conditional Independence (CI) Assumption:

\[ p(x', \varepsilon'|x, \varepsilon, d; \theta^p) = p_2(\varepsilon'|x'; \theta^p_2)p_3(x'|x, d; \theta^p_3) \]

- Expected value function

\[ EV(x) = \int_{\varepsilon} V(x, \varepsilon)p_2(\varepsilon'|x'; \theta^p_2) \]

- Choice-specific expected value function

\[ EV(x, d) = \nu(x, d; \theta^c, RC) + \varepsilon(d) + \beta \int_{x'} EV(x)p_3(x'|x, d; \theta^p_3) dx' \]
Zurcher’s Bus Engine Replacement Problem

- Assume type-1 extreme value distribution for $\varepsilon = [\varepsilon(0), \varepsilon(1)]$
- Conditional choice probability

$$P(d|x; \theta) = \frac{\exp \left[ \nu(x, d; \theta^c, RC) + \beta EV(x, d) \right]}{\sum_{d' \in \{0, 1\}} \exp \left[ \nu(x, d'; \theta^c, RC) + \beta EV(x, d') \right]}$$

- Choice-specific expected value function

$$EV(x, d) = \int_{x'=0}^{\infty} \log \left\{ \sum_{d' \in \{0, 1\}} \exp \left[ \nu(x', d'; \theta^c, RC) + \beta EV(x', d') \right] \right\} p_3(dx'|x, d, \theta^P_3)$$
Zurcher’s Bus Engine Replacement Problem

- Discretize the mileage state space $x$ into $K$ grid points
  \[ \hat{x} = \{\hat{x}_1, \ldots, \hat{x}_K\} \text{ with } \hat{x}_1 = 0 \]

- Mileage transition probability: for $j = 1, \ldots, J$
  \[
p_3(x' | \hat{x}_k, d, \theta_P^3) = \begin{cases} 
    \Pr\{x' = \hat{x}_{k+j} | \theta_P^3\}, & \text{if } d = 0 \\
    \Pr\{x' = \hat{x}_{1+j} | \theta_P^3\}, & \text{if } d = 1 
  \end{cases}
\]

- Mileage in the next period $x'$ can move up at most $J$ grid points

- Choice-specific expected value function for $\hat{x} \in \hat{x}$
  \[
  EV(\hat{x}, d) = \sum_{j=0}^{J} \log \left\{ \sum_{d' \in \{0, 1\}} \exp \left[ \nu(x', d'; \theta^c, RC) + \beta EV(x', d') \right] \right\} p_3(x' | \hat{x}, d, \theta_P^3)
  \]
Zurcher’s Bus Engine Replacement Problem

- Data: time series \((x_t, d_t)^T_{t=1}\)
- Likelihood function

\[
L(\theta) = \prod_{t=2}^{T} P(d_t|x_t, \theta^c, RC)p_3(x_t|x_{t-1}, d_{t-1}, \theta^p_3)
\]

with

\[
P(d|x, \theta^c, RC) = \frac{\exp\{\nu(x, d; \theta^c, RC) + \beta EV_\theta(x, d)\}}{\sum_{d' \in \{0,1\}} \exp\{\nu(x, d'; \theta^c, RC) + \beta EV_\theta(x', d)\}}
\]

\[
EV_\theta(x, d) = T_\theta(EV_\theta)(x, d)
\]

\[
\equiv \sum_{j=0}^{J} \log \left\{ \sum_{d' \in \{0,1\}} \exp \left[ \nu(x', d'; \theta^c, RC) + \beta EV(x', d') \right] \right\} p_3(x'|\hat{x}, d, \theta^p_3)
\]
Nested Fixed Point Algo: Rust (1987)

- Outer loop: Solve likelihood

\[
\max_{\theta \geq 0} L(\theta) = \prod_{t=2}^{T} P(d_t|x_t, \theta^c, RC)p_3(x_t|x_{t-1}, d_{t-1}, \theta^p)
\]

- Convergence test: \( \|\nabla_\theta L(\theta)\| \leq \epsilon_{out} \)

- Inner loop: Compute expected value function \( EV_\theta \) for a given \( \theta \)
  - \( EV_\theta \) is the implicit expected value function defined by the Bellman equation or the fixed point function

\[
EV_\theta = T_\theta(EV_\theta)
\]

- Convergence test: \( \|EV^{k+1}_\theta - EV^k_\theta\| \leq \epsilon_{in} \)
- Rust started with contraction iterations and then switched to Newton iterations
Smooth Objective Function?

- Recall first step in solving an estimation problem
- Is the ML objective function $L(\theta)$ smooth (differentiable w.r.t. $\theta$)?

  \[ L(\theta) = \prod_{t=2}^{T} P(d_t|x_t, \theta^c, RC)p_3(x_t|x_{t-1}, d_{t-1}, \theta^p_3) \]

  \[ P(d|x, \theta^c, RC) = \frac{\exp\{\nu(x, d; \theta^c, RC) + \beta EV_\theta(x, d)\}}{\sum_{d'\in\{0,1\}} \exp\{\nu(x, d'; \theta^c, RC) + \beta EV_\theta(x', d)\}} \]

- Is $EV_\theta$ differentiable w.r.t. $\theta$?
  Yes, because $T_\theta(EV_\theta)$ is a contraction mapping(!)

- Is the “approximated” ML objective function $L(\theta, \epsilon_{in})$ smooth (differentiable w.r.t. $\theta$)?
  Is $EV_\theta(\epsilon_{in})$ differentiable w.r.t. $\theta$ or w.r.t. $\epsilon_{in}$?
Concerns with NFXP – Dubé Fox and Su (2011)

- Inner-loop error propagates into outer-loop function and derivatives
- NFXP needs to solve inner-loop exactly for each vector of parameters
  - to accurately compute the search direction for the outer loop
  - to accurately evaluate derivatives for the outer loop
  - for the outer loop to converge
- Stopping rules: choosing inner-loop and outer-loop tolerances
  - inner-loop can be slow: contraction mapping is linearly convergent
  - tempting to loosen inner loop tolerance $\epsilon_{in}$ used
    - often see $\epsilon_{in} = 1.e - 6$ or higher
  - outer loop may not converge with loose inner loop tolerance
    - check solver output message
    - tempting to loosen outer loop tolerance $\epsilon_{in}$ to promote convergence
    - often see $\epsilon_{out} = 1.e - 3$ or higher
- Rust’s implementation of NFXP was correct
  - $\epsilon_{in} = 1.e - 13$
  - finished the inner-loop with Newton’s method
Stopping Rules – Dubé Fox and Su (2011)

- Notations:
  - \( L(\theta, \epsilon_{in}) \): the programmed outer loop objective function with \( \epsilon_{in} \)
  - Analytic derivatives \( \nabla_{\theta} L(\theta, \epsilon_{in}) \) are provided: \( \epsilon_{out} = O\left(\frac{\beta}{1-\beta} \epsilon_{in}\right) \)
  - Finite-difference derivatives are used: \( \epsilon_{out} = O\left(\sqrt{\frac{\beta}{1-\beta}} \epsilon_{in}\right) \)
Constrained Optimization for Solving Zucher Model

- Form augmented likelihood function for data $X = (x_t, d_t)_{t=1}^T$

$$L(\theta, EV; X) = \prod_{t=2}^T P(d_t|x_t, \theta^c, RC)p(x_t|x_{t-1}, d_{t-1}, \theta^p)$$

with

$$P(d|x, \theta^c, RC) = \frac{\exp\{\nu(x, d; \theta^c, RC) + \beta EV(x, d)\}}{\sum_{d' \in \{0, 1\}} \exp\{\nu(x, d'; \theta^c, RC) + \beta EV(x, d')\}}$$

- Rationality and Bellman equation imposes a relationship between $\theta$ and $EV$

$$EV = T(EV, \theta)$$

- Solve the constrained optimization problem

$$\max_{(\theta, EV)} L(\theta, EV; X)$$

subject to

$$EV = T(EV, \theta)$$


**Equivalent Reformulation?**

- **ML-NFXP:**
  
  \[
  \max_{\theta \geq 0} L(\theta) = \prod_{t=2}^{T} P(d_t|x_t, \theta^c, RC)p_3(x_t|x_{t-1}, d_{t-1}, \theta^p_3)
  \]

- **ML-Constrained Optimization:**
  
  \[
  \max_{(\theta, EV)} \mathcal{L}(\theta, EV; X) \quad \text{subject to} \quad EV = T(EV, \theta)
  \]

- Are these two formulations equivalent? Proof?
- Are the first-order conditions of these two formulations equivalent? Proof?
Monte Carlo: Rust’s Table X - Group 1,2, 3

- Fixed point dimension: 175
- Maintenance cost function: \( c(x, \theta^c) = 0.001 \times \theta_1^c \times x \)
- Mileage transition: stay or move up at most 4 grid points
- True parameter values:
  - \( \theta_1^c = 2.457 \)
  - \( RC = 11.726 \)
  - \( (\theta_{p30}^p, \theta_{p31}^p, \theta_{p32}^p, \theta_{p33}^p) = (0.0937, 0.4475, 0.4459, 0.0127) \)
- Solve for EV at the true parameter values
- Simulate 250 datasets of monthly data for 10 years and 50 buses
- Estimation implementations
  - MPEC1: AMPL/Knitro (with 1st- and 2nd-order derivative)
  - MPEC2: Matlab/ktrlink (with 1st-order derivatives)
  - NFXP: Matlab/ktrlink (with 1st-order derivatives)
  - 5 re-start in each of 250 replications
Monte Carlo: $\beta = 0.975$ and $0.980$

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>Iple.</th>
<th>Parameters</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$RC$</td>
<td>$\theta_1^c$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(true)</td>
</tr>
<tr>
<td>true</td>
<td>11.726</td>
<td>2.457</td>
<td>0.0937</td>
</tr>
<tr>
<td>0.975</td>
<td>MPEC1</td>
<td>12.212</td>
<td>2.607</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.613)</td>
<td>(0.500)</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>12.212</td>
<td>2.607</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.613)</td>
<td>(0.500)</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>12.213</td>
<td>2.606</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.617)</td>
<td>(0.500)</td>
</tr>
<tr>
<td>0.980</td>
<td>MPEC1</td>
<td>12.134</td>
<td>2.578</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.570)</td>
<td>(0.458)</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>12.134</td>
<td>2.578</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.570)</td>
<td>(0.458)</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>12.139</td>
<td>2.579</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.571)</td>
<td>(0.459)</td>
</tr>
</tbody>
</table>
Monte Carlo: $\beta = 0.985$ and 0.990

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>Iple.</th>
<th>$RC$</th>
<th>$\theta_1^c$</th>
<th>$\theta_{31}^p$</th>
<th>$\theta_{32}^p$</th>
<th>$\theta_{33}^p$</th>
<th>$\theta_{34}^p$</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td></td>
<td>11.726</td>
<td>2.457</td>
<td>0.0937</td>
<td>0.4475</td>
<td>0.4459</td>
<td>0.0127</td>
<td></td>
</tr>
<tr>
<td>0.985</td>
<td>MPEC1</td>
<td>12.013 (1.371)</td>
<td>2.541 (0.413)</td>
<td>0.0943 (0.0037)</td>
<td>0.4473 (0.0057)</td>
<td>0.4455 (0.0060)</td>
<td>0.0127</td>
<td>2.140</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>12.013 (1.371)</td>
<td>2.541 (0.413)</td>
<td>0.0943 (0.0037)</td>
<td>0.4473 (0.0057)</td>
<td>0.4455 (0.0060)</td>
<td>0.0127</td>
<td>2.140</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>12.021 (1.368)</td>
<td>2.544 (0.411)</td>
<td>0.0943 (0.0037)</td>
<td>0.4473 (0.0057)</td>
<td>0.4455 (0.0060)</td>
<td>0.0127</td>
<td>2.136</td>
</tr>
<tr>
<td>0.990</td>
<td>MPEC1</td>
<td>11.830 (1.305)</td>
<td>2.486 (0.407)</td>
<td>0.0943 (0.0036)</td>
<td>0.4473 (0.0057)</td>
<td>0.4455 (0.0060)</td>
<td>0.0127</td>
<td>1.880</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>11.830 (1.305)</td>
<td>2.486 (0.407)</td>
<td>0.0943 (0.0036)</td>
<td>0.4473 (0.0057)</td>
<td>0.4455 (0.0060)</td>
<td>0.0127</td>
<td>1.880</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>11.830 (1.305)</td>
<td>2.486 (0.407)</td>
<td>0.0943 (0.0036)</td>
<td>0.4473 (0.0057)</td>
<td>0.4455 (0.0060)</td>
<td>0.0127</td>
<td>1.880</td>
</tr>
</tbody>
</table>
## Monte Carlo: $\beta = 0.995$

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>Iple.</th>
<th>Parameters</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$RC$</td>
<td>$\theta_1^c$</td>
<td>$\theta_{31}^p$</td>
</tr>
<tr>
<td>true</td>
<td>11.726</td>
<td>2.457</td>
<td>0.0937</td>
</tr>
<tr>
<td>0.995</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPEC1</td>
<td>11.819</td>
<td>2.492</td>
<td>0.0942</td>
</tr>
<tr>
<td></td>
<td>(1.308)</td>
<td>(0.414)</td>
<td>(0.0036)</td>
</tr>
<tr>
<td>MPEC2</td>
<td>11.819</td>
<td>2.492</td>
<td>0.0942</td>
</tr>
<tr>
<td></td>
<td>(1.308)</td>
<td>(0.414)</td>
<td>(0.0036)</td>
</tr>
<tr>
<td>NFXP</td>
<td>11.819</td>
<td>2.492</td>
<td>0.0942</td>
</tr>
<tr>
<td></td>
<td>(1.308)</td>
<td>(0.414)</td>
<td>(0.0036)</td>
</tr>
</tbody>
</table>

Che-Lin Su
## Monte Carlo: Numerical Performance

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>Imple.</th>
<th>Runs</th>
<th>CPU Time</th>
<th># of Major</th>
<th># of Func.</th>
<th># of Contrac.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.975</td>
<td>MPEC1</td>
<td>1240</td>
<td>0.13</td>
<td>12.8</td>
<td>17.6</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>1247</td>
<td>7.9</td>
<td>53.0</td>
<td>62.0</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>998</td>
<td>24.6</td>
<td>55.9</td>
<td>189.4</td>
<td>134,748</td>
</tr>
<tr>
<td>0.980</td>
<td>MPEC1</td>
<td>1236</td>
<td>0.15</td>
<td>14.5</td>
<td>21.8</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>1241</td>
<td>8.1</td>
<td>57.4</td>
<td>70.6</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>1000</td>
<td>27.9</td>
<td>55.0</td>
<td>183.8</td>
<td>162,505</td>
</tr>
<tr>
<td>0.985</td>
<td>MPEC1</td>
<td>1235</td>
<td>0.13</td>
<td>13.2</td>
<td>19.7</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>1250</td>
<td>7.5</td>
<td>55.0</td>
<td>62.3</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>952</td>
<td>42.2</td>
<td>61.7</td>
<td>227.3</td>
<td>265,827</td>
</tr>
<tr>
<td>0.990</td>
<td>MPEC1</td>
<td>1161</td>
<td>0.19</td>
<td>18.3</td>
<td>42.2</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>1248</td>
<td>7.5</td>
<td>56.5</td>
<td>65.8</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>935</td>
<td>70.1</td>
<td>66.9</td>
<td>253.8</td>
<td>452,347</td>
</tr>
<tr>
<td>0.995</td>
<td>MPEC1</td>
<td>965</td>
<td>0.14</td>
<td>13.4</td>
<td>21.3</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>MPEC2</td>
<td>1246</td>
<td>7.9</td>
<td>59.6</td>
<td>70.7</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>NFXP</td>
<td>950</td>
<td>111.6</td>
<td>58.8</td>
<td>214.7</td>
<td>748,487</td>
</tr>
</tbody>
</table>
Observations

- **MPEC**
  - In MPEC/AMPL, problems are solved very quickly.
  - The likelihood function, the constraints, and their first-order and second-order derivatives are evaluated only around 20 times.
  - Constraints (Bellman Eqs) are NOT solved exactly in most iterations.
    - No need to resolve the fixed-point equations for every guess of structural parameters.
  - Quadratic convergence is observed in the last few iterations; in contrast, NFXP is linearly convergent (or super-linear at best).

- In NFXP, the Bellman equations are solved around 200 times and evaluated between 134,000 and 750,000 times.
Advantages of Constrained Optimization

- Newton-based methods are locally quadratic convergent
- Two key factors in efficient implementations:
  - Provide analytic-derivatives – huge improvement in speed
  - Exploit sparsity pattern in constraint Jacobian – huge saving in memory requirement
Part III

Random-Coefficients Demand Estimation

- Berry, Levinsohn and Pakes (BLP, 1995) consists of an economic model and a GMM estimator
- Demand estimation with a large number of differentiated products
  - characteristics approach
  - applicable when only aggregate market share data available
  - flexible substitution patterns / price elasticities
  - control for price endogeneity
- Computational algorithm to construct moment conditions from a non-linear model
- Useful for measuring market power, welfare, optimal pricing, etc.
Random-Coefficients Logit Demand

- Utility of consumer $i$ from purchasing product $j$ in market $t$

$$u_{ijt} = \beta^0_i + x_{jt} \beta^x_i - \beta^p_i p_{jt} + \xi_{jt} + \epsilon_{ijt}$$

- product characteristics: $x_{jt}, p_{jt}, \xi_{jt}$
  - $x_{jt}, p_{jt}$ observed; $\text{cov}(\xi_{jt}, p_{jt}) \neq 0$
  - $\xi_{jt}$: not observed – not in data
- $\beta_i$: random coefficients/individual-specific taste to be estimated
  - Distribution: $\beta_i \sim F_\beta(\beta; \theta)$
  - BLP’s statistical goal: estimate $\theta$ in parametric distribution
- error term $\epsilon_{ijt}$: Type I E.V. shock (i.e., Logit)
- Consumer $i$ picks product $j$ if $u_{ijt} \geq u_{ij't}$, $\forall j' \neq j$
Market Share Equations

- Predicted market shares

\[ s_j(x_t, p_t, \xi_t, ; \theta) = \int_{\{\beta_i, \varepsilon_j | u_{ijt} \geq u_{ij't}, \forall j' \neq j\}} dF_\beta(\beta; \theta)dF_\varepsilon(\varepsilon) \]

- With logit errors \( \varepsilon \)

\[ s_j(x_t, p_t, \xi_t, ; \theta) = \int \frac{\exp(\beta^0 + x_{jt}\beta^x - \beta^pp_{jt} + \xi_{jt})}{1 + \sum_{k=1}^J \exp(\beta^0 + x_{kt}\beta^x - \beta^pp_{kt} + \xi_{kt})} dF_\beta(\beta; \theta) \]

- Simulate numerical integral

\[ \hat{s}_j(x_t, p_t, \xi_t, ; \theta) = \frac{1}{ns} \sum_{r=1}^{ns} \frac{\exp(\beta^{0r} + x_{jt}\beta^{xr} - \beta^{pr}p_{jt} + \xi_{jt})}{1 + \sum_{k=1}^J \exp(\beta^{0r} + x_{kt}\beta^{xr} - \beta^{pr}p_{kt} + \xi_{kt})} \]

- Market share equations

\[ \hat{s}_j(x_t, p_t, \xi_t, ; \theta) = S_{jt}, \forall j \in J, t \in T \]
Random-Coefficients Logit Demand: GMM Estimator

- Assume $E [\xi_{jt} z_{jt} | z_{jt}] = 0$ for some vector of instruments $z_{jt}$
  - Empirical analog $g (\theta) = \frac{1}{TJ} \sum_{t,j} \xi_{jt} (\theta)' z_{jt}$

- Data: $\{(x_{jt}, p_{jt}, S_{jt}, z_{jt}) | j \in J, t \in T\}$

- Minimize GMM objective function

$$Q(\theta) = g (\theta)' Wg (\theta)$$

- Cannot compute $\xi_{jt} (\theta)$ analytically
  - “Invert” $\xi_t$ from system of predicted market shares numerically

$$S_t = s (x_t, p_t, \xi_t; \theta)$$

$$\Rightarrow \xi_t (\theta) = s^{-1} (x_t, p_t, S_t; \theta)$$

- BLP show the inversion of share equations for $\xi(\theta)$ is a contraction-mapping
BLP/NFXP Estimation Algorithm

- Outer loop: \( \min_\theta g(\theta)' W g(\theta) \)
  
  - Guess \( \theta \) parameters to compute \( g(\theta) = \frac{1}{TJ} \sum_{t=1}^{T} \sum_{j=1}^{J} \xi_{jt}(\theta)' z_{jt} \)
  
  - Stop when \( \| \nabla_\theta (g(\theta)' W g(\theta)) \| \leq \epsilon_{out} \)
BLP/NFXP Estimation Algorithm

- **Outer loop:** \( \min_{\theta} \, g(\theta)' \, W \, g(\theta) \)
  - Guess \( \theta \) parameters to compute \( g(\theta) = \frac{1}{TJ} \sum_{t=1}^{T} \sum_{j=1}^{J} \xi_{jt}(\theta)' \, z_{jt} \)
  - Stop when \( \| \nabla_{\theta}(g(\theta)' \, W \, g(\theta)) \| \leq \epsilon_{\text{out}} \)

- **Inner loop:** compute \( \xi_t(\theta) \) for a given \( \theta \)
  - Solve \( s(x_t, p_t, \xi_t; \theta) = S_t \) for \( \xi \) by contraction mapping:
    \[
    \xi^{h+1}_t = \xi^h_t + \log S_t - \log s(x_t, p_t, \xi_t; \theta)
    \]
  - Stop when \( \| \xi^{h+1}_t - \xi^h_t \| \leq \epsilon_{\text{in}} \)
  - Denote the approximated demand shock by \( \xi(\theta, \epsilon_{\text{in}}) \)

- **Stopping rules:** need to choose tolerance/stopping criterion for both inner loop (\( \epsilon_{\text{in}} \)) and outer loop (\( \epsilon_{\text{out}} \))
Smooth Objective Function?

Is the GMM objective function \( Q(\xi(\theta)) \) smooth (differentiable w.r.t. \( \theta \))?

- \( Q(\xi(\theta)) = \xi(\theta)'ZWZ'\xi(\theta) \)
- Is \( \xi(\theta) \) differentiable w.r.t. \( \theta \)?

Is the “approximated” GMM objective function \( Q(\xi(\theta; \epsilon_{in})) \) smooth (differentiable w.r.t. \( \theta \))?

- \( Q(\xi(\theta, \epsilon_{in})) = \xi(\theta, \epsilon_{in})'ZWZ'\xi(\theta, \epsilon_{in}) \)
- Is \( \xi(\theta, \epsilon_{in}) \) differentiable w.r.t. \( \theta \) or w.r.t. \( \epsilon_{in} \)?
Our Concerns with NFP/BLP

- Inefficient amount of computation
  - we only need to know $\xi(\theta)$ at the true $\theta$
  - NFP solves inner-loop exactly each stage of parameter search
  - evaluating $s(x_t, p_t, \xi_t; \theta)$ thousands of times in the contraction mapping
- Stopping rules: choosing inner-loop and outer-loop tolerances
  - inner-loop can be slow (especially for bad guesses of $\theta$): linear convergence at best
  - tempting to loosen inner loop tolerance $\epsilon_{in}$ used
    - often see $\epsilon_{in} = 1.e - 6$ or higher
  - outer loop may not converge with loose inner loop tolerance
    - check solver output message; see Knittel and Metaxoglou (2008)
    - tempting to loosen outer loop tolerance $\epsilon_{in}$ to promote convergence
      - often see $\epsilon_{out} = 1.e - 3$ or higher
- Inner-loop error propagates into outer-loop
Knittel and Metaxoglou (2010)

- Perform extensive numerical studies on BLP/NFXP algorithms with two data sets
  - 10 free solvers and 50 starting points for each solver
- Find that convergence may occur at a number of local extrema, at saddles and in regions of the objective function where the First-Order Conditions are not satisfied.
- Furthermore, parameter estimates and measures of market performance, such as price elasticities, exhibit notable variation (two orders of magnitude) depending on the combination of the algorithm and starting values in the optimization exercise at hand
- Recall the optimization output that you saw earlier
Analyzing BLP/NFXP Algorithm

- Let $L$ be the Lipschitz constant of the inner-loop contraction mapping.
- Numerical Errors in GMM function and gradient:

\[
|Q(\xi(\theta, \epsilon_{in})) - Q(\xi(\theta, 0))| = O\left(\frac{L}{1-L} \epsilon_{in}\right)
\]
\[
\|\nabla_{\theta}Q(\xi(\theta))|_{\xi=\xi(\theta, \epsilon_{in})} - \nabla_{\theta}Q(\xi(\theta))|_{\xi=\xi(\theta, 0)}\| = O\left(\frac{L}{1-L} \epsilon_{in}\right)
\]

- Ensuring convergence: $\epsilon_{out} = O\left(\frac{L}{1-L}\right) \epsilon_{in}$
Errors in Parameter Estimates

\[ \theta^* = \arg \max_{\theta} \{ Q(\xi(\theta, 0)) \} \]

\[ \hat{\theta} = \arg \max_{\theta} \{ Q(\xi(\theta, \epsilon_{in})) \} \]

- Finite sample error in parameter estimates

\[ O\left(\| \hat{\theta} - \theta^* \|^2 \right) \leq \left| Q(\xi(\hat{\theta}, \epsilon_{in})) - Q(\xi(\theta^*, 0)) \right| + O\left(\frac{L}{1 - L} \epsilon_{in} \right) \]

- Large sample error in parameter estimates

\[ \| \hat{\theta} - \theta^0 \| \leq \| \hat{\theta} - \theta^* \| + \| \theta^* - \theta^0 \| \]

\[ \leq \sqrt{\left| Q(\xi(\hat{\theta}, \epsilon_{in})) - Q(\xi(\theta^*, 0)) \right| + O\left(\frac{L}{1 - L} \epsilon_{in} \right) + O\left(1/\sqrt{T} \right) } \]
Numerical Experiment: 100 different starting points

- 1 dataset: 75 markets, 25 products, 10 structural parameters
  - NFP tight: $\epsilon_{in} = 1.e-10; \epsilon_{out} = 1.e-6$
  - NFP loose inner: $\epsilon_{in} = 1.e-4; \epsilon_{out} = 1.e-6$
  - NFP loose both: $\epsilon_{in} = 1.e-4; \epsilon_{out} = 1.e-2$

<table>
<thead>
<tr>
<th>Starting point</th>
<th>NFXP tight</th>
<th>NFXP loose inner</th>
<th>NFXP loose both</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$7.9967e+01$</td>
</tr>
<tr>
<td>2</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$9.7130e-02$</td>
</tr>
<tr>
<td>3</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$1.1873e-01$</td>
</tr>
<tr>
<td>4</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$1.3308e-01$</td>
</tr>
<tr>
<td>5</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$7.3024e-02$</td>
</tr>
<tr>
<td>6</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$6.0614e+01$</td>
</tr>
<tr>
<td>7</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$1.5909e+02$</td>
</tr>
<tr>
<td>8</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$2.1087e-01$</td>
</tr>
<tr>
<td>9</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$6.4803e+00$</td>
</tr>
<tr>
<td>10</td>
<td>$4.3084e-02$</td>
<td>Fail</td>
<td>$1.2271e+03$</td>
</tr>
</tbody>
</table>

Main findings: Loosening tolerance leads to non-convergence
- Check optimization exit flags!
- Solver does NOT produce a local optimum with loose tolerances!
Constrained Optimization Applied to BLP

• Constrained optimization formulation

$$\min_{(\theta, \xi)} \xi^T ZW Z^T \xi$$
subject to \( s(\xi, \theta) = S \)

• Advantages:
  • No need to worry about setting up two tolerance levels
  • No inner-loop errors propagated into parameter estimates
  • Easy to code in AMPL and to access good NLP solvers
  • AMPL provides analytic derivatives
  • AMPL analyzes sparsity structure of constraint Jacobian
  • Fewer iterations/function evaluations with first-order and second-order derivatives information
  • Share equations only need to be hold at the solution

• Bad news: Hessian of the Lagrangian is dense
Exploiting Symmetry and Sparsity in the Hessian

- By adding additional variable $g$ and constraint $Z^T \xi = g$

$$\min_{(\theta, \xi, g,)} \quad g^T W g$$

subject to

$$s(\delta; \theta_2) = S$$

$$Z^T \xi = g$$

- Advantages:
  - The Hessian of the objective function is now sparse
  - Increasing the sparsity $\Rightarrow$ huge saving on memory
Sparsity Pattern of Constraint Jacobian $\nabla c(x)$
Sparsity Pattern of Hessian $\nabla^2 \mathcal{L}(x, y, z)$
param ns ;  # := 20 ;   # number of simulated "individuals" per market
param nmkt ;  # := 94 ;  # number of markets
param nbrn ;  # := 24 ;  # number of brands per market
param nbrnPLUS1 := nbrn+1;  # number of products plus outside good
param nk1 ;  # := 25;   # of observable characteristics
param nk2 ;  # := 4 ;  # of observable characteristics
param niv ;  # := 21 ;  # of instrument variables
param nz := niv-1 + nk1 -1;  # of instruments including iv and X1
param nd ;  # := 4 ;   # of demographic characteristics

set S := 1..ns ;  # index set of individuals
set M := 1..nmkt ;  # index set of market
set J := 1..nbrn ;  # index set of brand (products), including outside good
set MJ := 1..nmkt*nbrn;  # index of market and brand
set K1 := 1..nk1 ;  # index set of product observable characteristics
set K2 := 1..nk2 ;  # index set of product observable characteristics
set Demogr := 1..nd;
set DS := 1..nd*ns;
set K2S := 1..nk2*ns;

set H := 1..nz ;  # index set of instrument including iv and X1
## Define input data format:

param X1 {mj in MJ, k in K1} ;
param X2 {mj in MJ, k in K2} ;
param ActuShare {m in MJ} ;
param Z {mj in MJ, h in H} ;
param D {m in M, di in DS} ;
param v {m in M, k2i in K2S} ;
param invA {i in H, j in H} ;  # optimal weighting matrix = inv(Z’Z);
param OutShare {m in M} := 1 - sum {mj in (nbrn*(m-1)+1)..(nbrn*m)} ActuShare[mj];
## Define variables

```plaintext
var theta1 {k in K1};
var SIGMA {k in K2};
var PI {k in K2, d in Demogr};
var delta {mj in MJ};

var EstShareIndivTop {mj in MJ, i in S} = exp( delta[mj] + sum {k in K2} (X2[mj,k]*SIGMA[k]*v[ceil(mj/nbrn), i+(k-1)*ns]) + sum{k in K2, d in Demogr} (X2[mj,k]*PI[k,d]*D[ceil(mj/nbrn),i+(d-1)*ns]) );

var EstShareIndiv{mj in MJ, i in S} = EstShareIndivTop[mj,i] / (1+ sum{l in ((ceil(mj/nbrn)-1)*nbrn+1)..(ceil(mj/nbrn)*nbrn)} EstShareIndivTop[l, i]);

var EstShare {mj in MJ} = 1/ns * (sum{i in S} EstShareIndiv[mj,i]) ;

var w {mj in MJ} = delta[mj] - sum {k in K1} (X1[mj,k]*theta1[k]);

var Zw {h in H};  ## Zw{h in H} = sum {mj in MJ} Z[mj,h]*w[mj];
```
minimize GMM : sum\{h1 in H, h2 in H\} Zw[h1]*invA[h1, h2]*Zw[h2];

subject to

\hspace{1cm} \text{conZw} \{h in H\}: Zw[h] = \text{sum}\{mj in MJ\} Z[mj,h]*w[mj] ;

\hspace{1cm} \text{Shares} \{mj in MJ\}: \log(\text{EstShare}[mj]) = \log(\text{ActuShare}[mj]) ;
AMPL/KNITRO Output

KNITRO 6.0.0: alg=1
opttol=1.0e-6
feastol=1.0e-6

Problem Characteristics
-----------------------
Objective goal: Minimize
Number of variables: 2338
  bounded below: 0
  bounded above: 0
  bounded below and above: 0
  fixed: 0
  free: 2338
Number of constraints: 2300
  linear equalities: 44
  nonlinear equalities: 2256
  linear inequalities: 0
  nonlinear inequalities: 0
  range: 0
Number of nonzeros in Jacobian: 131440
Number of nonzeros in Hessian: 58609
### AMPL/KNITRO Output

| Iter | Objective     | FeasError    | OptError     | ||Step|| | CGits |
|------|---------------|--------------|--------------|---------|-------|
| 0    | 2.936110e+01  | 1.041e-04    |              |         |       |
| 1    | 1.557550e+01  | 3.813e-01    | 4.561e-02    | 4.835e+01 | 9     |
| 2    | 6.289721e+00  | 6.157e-01    | 2.605e+01    | 3.416e+02 | 0     |
| 3    | 4.646499e+00  | 1.145e-01    | 3.041e+00    | 1.901e+02 | 0     |
| 4    | 4.527042e+00  | 4.951e-02    | 5.887e-01    | 1.071e+02 | 0     |
| 5    | 4.562016e+00  | 8.379e-03    | 4.865e-02    | 4.243e+01 | 0     |
| 6    | 4.564521e+00  | 8.874e-05    | 6.051e-04    | 4.660e+00 | 0     |
| 7    | 4.564553e+00  | 1.196e-08    | 6.356e-08    | 5.280e-02 | 0     |

EXIT: Locally optimal solution found.
Final Statistics

Final objective value = 4.56455310841869e+00
Final feasibility error (abs / rel) = 1.20e-08 / 1.20e-08
Final optimality error (abs / rel) = 6.36e-08 / 3.21e-09
# of iterations = 7
# of CG iterations = 9
# of function evaluations = 8
# of gradient evaluations = 8
# of Hessian evaluations = 7
Total program time (secs) = 10.48621 ( 10.278 CPU time)
Time spent in evaluations (secs) = 8.62244

KNITRO 6.0.0: Locally optimal solution.
objective 4.564553108; feasibility error 1.2e-08
7 iterations; 8 function evaluations
Monte Carlo in DFS11: Simulated Data Setup

- \[
\begin{bmatrix}
  x_{1,j,t} \\
  x_{2,j,t} \\
  x_{3,j,t}
\end{bmatrix}
\sim \mathcal{N} \left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.8 & 0.3 \\ -0.8 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix} \right)
\]

- \(\xi_{j,t} \sim \mathcal{N}(0, 1)\)

- \(p_{j,t} = |0.5 \cdot \xi_{j,t} + e_{j,t}| + 1.1 \cdot |\sum_{k=1}^{3} x_{k,j,t}|\)

- \(z_{j,t,d} \sim \mathcal{N}(\frac{1}{4} p_{j,t}, 1), D = 6\) instruments

- \(F_{\beta}(\beta; \theta): 5\) independent normal distributions (\(K = 3\) attributes, price and the intercept)

- \(\beta_{i} = \{\beta_{i}^{0}, \beta_{i}^{1}, \beta_{i}^{2}, \beta_{i}^{3}, \beta_{i}^{p}\}: E[\beta_{i}] = \{0.1, 1.5, 1.5, 0.5, -3\}\) and \(\text{Var}[\beta_{i}] = \{0.5, 0.5, 0.5, 0.5, 0.2\}\)
Implementation Details

- MATLAB, highly vectorized code, available at
  http://faculty.chicagobooth.edu/jean-pierre.dube/vita/MPEC%20code.htm
- Optimization software KNITRO
  - Professional quality optimization program
  - Can be called directly from R2008a version of MATLAB
  - We call from TOMLAB
- We provide sparsity pattern for $\nabla c(x)$ and $\nabla^2 \mathcal{L}(x)$ for MPEC
- We code exact **first-order** and **second-order** derivatives
  - Important for performance of smooth optimizers
  - With both 1st and 2nd derivatives, NFP is 3 to 10 times faster than using only 1st order derivatives
  - Same component functions for derivatives
  - Helpful for standard errors
Loose v.s. Tight Tolerances for NFXP

<table>
<thead>
<tr>
<th></th>
<th>NFXP Loose Inner</th>
<th>NFXP Loose Both</th>
<th>NFXP Tight</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fraction Convergence</td>
<td>0.0</td>
<td>0.54</td>
<td>0.95</td>
<td></td>
</tr>
<tr>
<td>Frac.&lt; 1% &gt; “Global” Min.</td>
<td>0.0</td>
<td>0.0</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Mean Own Price Elasticity</td>
<td>-7.24</td>
<td>-7.49</td>
<td>-5.77</td>
<td>-5.68</td>
</tr>
<tr>
<td>Std. Dev. Own Price Elasticity</td>
<td>5.48</td>
<td>5.55</td>
<td>~0</td>
<td></td>
</tr>
<tr>
<td>Lowest Objective</td>
<td>0.0176</td>
<td>0.0198</td>
<td>0.0169</td>
<td></td>
</tr>
<tr>
<td>Elasticity for Lowest Obj.</td>
<td>-5.76</td>
<td>-5.73</td>
<td>-5.77</td>
<td>-5.68</td>
</tr>
</tbody>
</table>

- 100 starting values for one dataset
- NFXP loose inner loop: $\epsilon_{in} = 10^{-4}$, $\epsilon_{out} = 10^{-6}$
- NFXP loose both: $\epsilon_{in} = 10^{-4}$, $\epsilon_{out} = 10^{-2}$
- NFXP tight: $\epsilon_{in} = 10^{-14}$, $\epsilon_{out} = 10^{-6}$
Lessons Learned

- Loose inner loop causes numerical error in gradient
  - Failure to diagnose convergence of outer loop
  - Leads to false estimates
- Making outer loop tolerance loose allows “convergence”
  - But to false solution
**Speeds, # Convergences and Finite-Sample Performance**

\( T = 50, J = 25, nn = 1000, 20 \) replications, 5 starting points/replication

<table>
<thead>
<tr>
<th>Intercept ( E [\beta_i^0] )</th>
<th>Lipsch.</th>
<th>Alg.</th>
<th>CPU (min)</th>
<th>Elasticities</th>
<th>Out. Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>( -2 )</td>
<td>0.891</td>
<td>NFP</td>
<td>21.7</td>
<td>-0.077</td>
<td>-10.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEC</td>
<td>18.3</td>
<td>-0.076</td>
<td></td>
</tr>
<tr>
<td>( -1 )</td>
<td>0.928</td>
<td>NFP</td>
<td>28.3</td>
<td>-0.078</td>
<td>-10.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEC</td>
<td>16.3</td>
<td>-0.077</td>
<td></td>
</tr>
<tr>
<td>( 0 )</td>
<td>0.955</td>
<td>NFP</td>
<td>41.7</td>
<td>-0.079</td>
<td>-10.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEC</td>
<td>15.2</td>
<td>-0.079</td>
<td></td>
</tr>
<tr>
<td>( 1 )</td>
<td>0.974</td>
<td>NFP</td>
<td>71.7</td>
<td>-0.083</td>
<td>-10.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEC</td>
<td>11.8</td>
<td>-0.083</td>
<td></td>
</tr>
<tr>
<td>( 2 )</td>
<td>0.986</td>
<td>NFP</td>
<td>103.3</td>
<td>-0.085</td>
<td>-10.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEC</td>
<td>13.5</td>
<td>-0.085</td>
<td></td>
</tr>
<tr>
<td>( 3 )</td>
<td>0.993</td>
<td>NFP</td>
<td>166.7</td>
<td>-0.088</td>
<td>-11.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEC</td>
<td>10.7</td>
<td>-0.088</td>
<td></td>
</tr>
<tr>
<td>( 4 )</td>
<td>0.997</td>
<td>NFP</td>
<td>300.0</td>
<td>-0.091</td>
<td>-11.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEC</td>
<td>12.7</td>
<td>-0.090</td>
<td></td>
</tr>
</tbody>
</table>
# of Function/Gradient/Hessian Evals and # Contraction Mapping Iterations

<table>
<thead>
<tr>
<th>Intercept $E \left[ \beta_i^0 \right]$</th>
<th>Alg.</th>
<th>Func Eval</th>
<th>Grad/Hess Eval</th>
<th>Contraction Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>NFP MPEC</td>
<td>80 184</td>
<td>58 126</td>
<td>10,400</td>
</tr>
<tr>
<td>-1</td>
<td>NFP MPEC</td>
<td>82 274</td>
<td>60 144</td>
<td>17,100</td>
</tr>
<tr>
<td>0</td>
<td>NFP MPEC</td>
<td>77 195</td>
<td>56 113</td>
<td>29,200</td>
</tr>
<tr>
<td>1</td>
<td>NFP MPEC</td>
<td>71 148</td>
<td>54 94</td>
<td>55,000</td>
</tr>
<tr>
<td>2</td>
<td>NFP MPEC</td>
<td>68 188</td>
<td>50 107</td>
<td>84,000</td>
</tr>
<tr>
<td>3</td>
<td>NFP MPEC</td>
<td>68 144</td>
<td>49 85</td>
<td>146,000</td>
</tr>
<tr>
<td>4</td>
<td>NFP MPEC</td>
<td>81 158</td>
<td>50 100</td>
<td>262,000</td>
</tr>
</tbody>
</table>
Lessons Learned

- For low Lipschitz constant, NFXP and MPEC about the same speed
- For high Lipschitz constant, NFXP becomes very slow
  - 1 hour per run for Intercept = 4
  - Reminder: you need to use more starting points if you want to find a good solution
- MPEC speed relatively invariant to Lipschitz constant
  - No contraction mapping in MPEC
### Speed for Varying # of Markets, Products, Draws

<table>
<thead>
<tr>
<th>$T$</th>
<th>$J$</th>
<th>$nn$</th>
<th>Lipsch. Const.</th>
<th>Alg</th>
<th>Runs</th>
<th>CPU (hr)</th>
<th>Outside Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>25</td>
<td>1000</td>
<td>0.999</td>
<td>NFP</td>
<td>80%</td>
<td>10.9</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>100%</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>25</td>
<td>1000</td>
<td>0.997</td>
<td>NFP</td>
<td>100%</td>
<td>22.3</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>100%</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>25</td>
<td>1000</td>
<td>0.998</td>
<td>NFP</td>
<td>80%</td>
<td>65.6</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>100%</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>25</td>
<td>3000</td>
<td>0.999</td>
<td>NFP</td>
<td>80%</td>
<td>42.3</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>100%</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>25</td>
<td>3000</td>
<td>0.997</td>
<td>NFP</td>
<td>100%</td>
<td>80.0</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>100%</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td>1000</td>
<td>0.993</td>
<td>NFP</td>
<td>100%</td>
<td>5.7</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>100%</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>250</td>
<td>1000</td>
<td>0.999</td>
<td>NFP</td>
<td>100%</td>
<td>28.4</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>100%</td>
<td>2.3</td>
<td></td>
</tr>
</tbody>
</table>
# of Function/Gradient/Hessian Evals and # Contraction Mapping Iterations

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>25</td>
<td>1000</td>
<td>NFP</td>
<td>68</td>
<td>130</td>
<td>69</td>
<td>372,278</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>84</td>
<td>98</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>25</td>
<td>1000</td>
<td>NFP</td>
<td>58</td>
<td>82</td>
<td>59</td>
<td>246,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>118</td>
<td>172</td>
<td>119</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>25</td>
<td>1000</td>
<td>NFP</td>
<td>52</td>
<td>99</td>
<td>53</td>
<td>280,980</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>123</td>
<td>195</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>25</td>
<td>3000</td>
<td>NFP</td>
<td>60</td>
<td>171</td>
<td>61</td>
<td>479,578</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>83</td>
<td>114</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>25</td>
<td>3000</td>
<td>NFP</td>
<td>55</td>
<td>68</td>
<td>56</td>
<td>204,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>102</td>
<td>135</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td>1000</td>
<td>NFP</td>
<td>54</td>
<td>71</td>
<td>55</td>
<td>198,114</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>97</td>
<td>145</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>250</td>
<td>1000</td>
<td>NFP</td>
<td>60</td>
<td>126</td>
<td>61</td>
<td>359,741</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MPEC</td>
<td>75</td>
<td>103</td>
<td>76</td>
<td></td>
</tr>
</tbody>
</table>
Summary

• Constrained optimization formulation for the random-coefficients demand estimation model is

$$\min_{(\theta, \xi, g, \gamma)} g^T W g$$

subject to

$$s(\delta; \theta_2) = S$$

$$Z^T \xi = g$$

• The constrained optimization approach (with good solvers) is reliable and has speed advantage

• It allows researchers to access best optimization solvers
Part IV

General Formulations
Standard Problem and Current Approach

- Individual solves an optimization problem
- Econometrician observes states and decisions
- Want to estimate structural parameters and equilibrium solutions that are consistent with structural parameters
- Current standard approach
  - Structural parameters: \( \theta \)
  - Behavior (decision rule, strategy, price): \( \sigma \)
  - Equilibrium (optimality or competitive or Nash) imposes
    \[
    G(\theta, \sigma) = 0
    \]
  - Likelihood function for data \( X \) and parameters \( \theta \)
    \[
    \max_{\theta} L(\theta; X)
    \]
  where equilibrium can be presented by \( \sigma = \Sigma(\theta) \)

- $G(\theta, \sigma) = 0$ represents the Bellman equations
- $\Sigma(\theta)$ is the expected value function and is single-valued as a function of $\theta$
- Outline of NFXP
  - Given $\theta$, compute $\sigma = \Sigma(\theta)$ by solving $G(\theta, \sigma) = 0$
  - For each $\theta$, define $L(\theta; X)$: likelihood given $\sigma = \Sigma(\theta)$
  - Solve $\max_\theta L(\theta; X)$
• $G(\theta, \sigma) = 0$ represents the demand system of share equations

• $\Sigma(\theta)$ is the unobserved demand shock and is single-valued as a function of $\theta$

• Outline of NFXP
  
  • Given $\theta$, compute $\sigma = \Sigma(\theta)$ by solving $G(\theta, \sigma) = 0$
  
  • For each $\theta$, define

    $Q(\theta; X) : \text{GMM given } \sigma = \Sigma(\theta)$

  
  • Solve

    $\min_{\theta} Q(\theta; X)$
NFXP Applied to Games with Multiple Equilibria

- \( G(\theta, \sigma) = 0 \) characterizes Nash equilibrium for a given \( \theta \)
- \( \Sigma(\theta) \) is set of Nash equilibria given \( \theta \) and can be multi-valued
- Outline of NFXP
  - Given \( \theta \), compute all \( \sigma \in \Sigma(\theta) \)
  - For each \( \theta \), define
    \[
    L(\theta; X) = \max \text{ likelihood over all } \sigma \in \Sigma(\theta)
    \]
  - Solve
    \[
    \max_{\theta} L(\theta; X)
    \]
  - If \( \Sigma(\theta) \) is multi-valued, then \( L \) can be nondifferentiable and/or discontinuous
NFXP Applied to Games with Multiple Equilibria

\[ L(\theta) \]

\[ L^* \]

\[ \theta_1 \quad \theta_2 \quad \theta^* \]
NFXP and Related Methods to Games

- NFXP requires finding all $\sigma$ that solve $G(\theta, \sigma) = 0$, compute the likelihood at each such $\sigma$, and report the max as the likelihood value $L(\theta)$
- Finding all equilibria for arbitrary games is an essentially intractable problem - see Judd and Schmedders (2006)
- One fundamental issue: G-S or G-J type methods are often used to solve for an equilibrium. This implicitly imposes an undesired equilibrium selection rule: converge only to equilibria that are stable under best reply
- Two-step estimator: Computationally light, very popular, biased in small samples
- NPL – Ag-M(2007): Iterating over the two-step estimator in an attempt to improve the small-sample bias
Constrained Optimization Applied to ML Estimation

- Suppose the game has parameters $\theta$.
- Let $\sigma$ denote the equilibrium strategy given $\theta$; equilibrium conditions impose

$$G(\theta, \sigma) = 0$$

- For a data set, $X$, Denote the augmented likelihood by $\mathcal{L}(\theta, \sigma; X)$

  - $\mathcal{L}(\theta, \sigma; X)$ decomposes $L(\theta; X)$ so as to highlight the separate dependence of likelihood on $\theta$ and $\sigma$
  - In fact, $L(\theta; X) = \mathcal{L}(\theta, \Sigma(\theta); X)$

- Therefore, maximum likelihood estimation is

$$\max_{(\theta, \sigma)} \mathcal{L}(\theta, \sigma; X)$$

subject to $G(\theta, \sigma) = 0$
Equivalent Reformulation for Games (or Models) with Multiple Equilibria?

- ML-NFXP:
  
  \[
  \max_{\theta} L(\theta; X),
  \]

  where for each \( \theta \),

  \[
  L(\theta; X) = \max \text{ likelihood over all } \sigma \in \Sigma(\theta)
  \]

- ML-Constrained Optimization:

  \[
  \max_{(\theta, \sigma)} \mathcal{L}(\theta, \sigma; X)
  \]

  subject to \( G(\theta, \sigma) = 0 \)

- Are these two formulations equivalent? Proof?
- Are the first-order conditions of these two formulations equivalent?
Advantages of Constrained Optimization

- Both $\mathcal{L}$ and $G$ are smooth functions
- Do not require that equilibria are stable under best-reply iteration
- Do not need to solve for all equilibria $\sigma$ for every $\theta$
- Use multi-start to attempt to find the global solution
- Using a constrained optimization approach allows one to take advantage of the best available methods and software
So ... What is NFXP?

- NFXP is equivalent to nonlinear elimination of variables
- Consider

\[
\max_{(x,y)} f(x, y)
\]
\[
\text{subject to } g(x, y) = 0
\]

- Define \( Y(x) \) implicitly by \( g(x, Y(x)) = 0 \)
- Solve the unconstrained problem

\[
\max_x f(x, Y(x))
\]

- Used only when memory demands are too large
- Often creates very difficult unconstrained optimization problems
Part V

Estimation of Games
Structural Estimation of Games of Incomplete Information

• An active research topic in Applied Econometrics/Empirical Industrial Organization
  • Statis entry/exit games of incomplete information: Seim (2006), Su (2012)

• Two main econometric issues appear in the estimation of these models
  • the existence of multiple equilibria – need to find all of them
  • computational burden in the solution of the game – repeated solving for equilibria for every guessed of structural parameters
Example: Static Game of Incomplete Information - due to John Rust

- Two firms: \( a \) and \( b \)
- Actions: each firm has two possible actions:

\[
\begin{align*}
    d_a &= \begin{cases} 
        1 & \text{if firm } a \text{ choose to enter the market} \\
        0 & \text{if firm } a \text{ choose not to enter the market}
    \end{cases} \\
    d_b &= \begin{cases} 
        1 & \text{if firm } b \text{ choose to enter the market} \\
        0 & \text{if firm } b \text{ choose not to enter the market}
    \end{cases}
\end{align*}
\]
Example: Static Game of Incomplete Information

- Utility: Ex-post payoff to firms

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

- \((\alpha, \beta)\): 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**
- \(\epsilon_a, \epsilon_b\) are observed only by each firm, but not by their opponent firm nor by the econometrician
Example: Static Game of Incomplete Information

• Assume the error terms \((\varepsilon_a, \varepsilon_b)\) have a standardized type III extreme value distribution

• A Bayesian Nash equilibrium \((p_a, p_b)\) satisfies

\[
p_a = \frac{\exp[p_b\beta x_a + (1 - p_b)\alpha x_a]}{1 + \exp[p_b\beta x_a + (1 - p_b)\alpha x_a]}
\]

\[
= \frac{1}{1 + \exp[-x_a \alpha + p_b x_a (\alpha - \beta)]}
\]

\[
\equiv \Psi_a(p_b, x_a; \alpha, \beta).
\]

\[
p_b = \frac{1}{1 + \exp[-x_b \alpha + p_a x_b (\alpha - \beta)]}
\]

\[
\equiv \Psi_b(p_a, x_b; \alpha, \beta).
\]
Static Game Example with One Market: Solving for Equilibria

- The true values of the structural parameters are
  \[(\alpha, \beta) = (5, -11)\]

- There is only 1 market with observed types \((x_a, x_b) = (0.52, 0.22)\)

\[
p_a = \frac{1}{1 + \exp\left\{0.52(-5) + p_b0.52(16)\right\}}
\]

\[
p_b = \frac{1}{1 + \exp\left\{0.22(-5) + p_a0.22(16)\right\}}
\]
Static Game Example: Three Bayesian Nash Equilibria

Eq1: \((p_a, p_b) = (0.030100, 0.729886)\) \text{ stable under BR}

Eq2: \((p_a, p_b) = (0.616162, 0.255615)\) \text{ unstable under BR}

Eq3: \((p_a, p_b) = (0.773758, 0.164705)\) \text{ stable under BR}
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: $X = \{(d^i_a, d^i_b)_{i=1}^{1000}, (x_a, x_b) = (0.52, 0.22)\}$
- Given data $X$, we want to recover structural parameters $\alpha$ and $\beta$
Static Game Example: Maximum Likelihood Estimation

- Maximize the likelihood function

$$
\max_{(\alpha, \beta)} \ log \mathcal{L} \ (p_a(\alpha, \beta), p_b(\alpha, \beta); X) \\
= \sum_{i=1}^{1000} \left( d_a^i \ast \log(p_a(\alpha, \beta)) + (1 - d_a^i) \ast \log(1 - p_a(\alpha, \beta)) \right) \\
+ \sum_{i=1}^{1000} \left( d_b^i \ast \log(p_b(\alpha, \beta)) + (1 - d_b^i) \ast \log(1 - p_b(\alpha, \beta)) \right)
$$

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

$$
p_a = \frac{1}{1 + \exp\{-0.52(\alpha) + p_b 0.52(\alpha) - \beta\}} = \Psi_a(p_b, x_a, \alpha, \beta)$$

$$
p_b = \frac{1}{1 + \exp\{-0.22(\alpha) + p_a 0.22(\alpha) - \beta\}} = \Psi_b(p_a, x_b, \alpha, \beta)
$$
Static Game Example: MLE via NFXP

- **Outer loop:**
  - Choose \((\alpha, \beta)\) to maximize the likelihood function
    \[
    \log \mathcal{L} \left( p_a(\alpha, \beta), p_b(\alpha, \beta); X \right)
    \]

- **Inner loop:**
  - For a given \((\alpha, \beta)\), solve the BNE equations for **ALL** equilibria:
    \[
    (p_a^k(\alpha, \beta), p_b^k(\alpha, \beta)), \quad k = 1, \ldots, K
    \]
  - Choose the equilibrium that gives the highest likelihood value:
    \[
    k^* = \arg\max_{\{k=1,\ldots,K\}} \log \mathcal{L} \left( p_a^k(\alpha, \beta), p_b^k(\alpha, \beta); X \right)
    \]
    \[
    (p_a(\alpha, \beta), p_b(\alpha, \beta)) = (p_a^{k^*}(\alpha, \beta), p_b^{k^*}(\alpha, \beta))
    \]
NFXP’s Likelihood as a Function of $(\alpha, \beta) – Eq 1$
NFXP’s Likelihood as a Function of $(\alpha, \beta) – Eq 2$
NFXP’s Likelihood as a Function of $(\alpha, \beta) – \text{Eq 3}$
Constrained Optimization Formulation for Maximum Likelihood Estimation

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

\[
= \sum_{i=1}^{1000} (d_a^i \log(p_a) + (1 - d_a^i) \log(1 - p_a))
\]

\[
+ \sum_{i=1}^{1000} (d_b^i \log(p_b) + (1 - d_b^i) \log(1 - p_b))
\]

subject to

\[
p_a = \frac{1}{1 + \exp\{0.52(\alpha) + p_b 0.52(\alpha - \beta)\}}
\]

\[
p_b = \frac{1}{1 + \exp\{-0.22(\alpha) + p_a 0.22(\alpha - \beta)\}}
\]

\[
0 \leq p_a, p_b \leq 1
\]

Log-likelihood function is a smooth function of \((p_a, p_b)\).
Monte Carlo Results with Eq2

Truth for alpha is $-5$

Truth for beta is 11

Truth for $p_1$ in Equilibrium #2 is 0.61616

Truth for $p_2$ in Equilibrium #2 is 0.25561
Monte Carlo Results with Eq1

Truth for alpha is $-5$

Truth for beta is 11

Truth for $p_1$ in Equilibrium #1 is 0.0301

Truth for $p_2$ in Equilibrium #1 is 0.72989
Monte Carlo Results with Eq3

Truth for alpha is −5

Truth for beta is 11

Truth for p1 in Equilibrium #3 is 0.77376

Truth for p2 in Equilibrium #3 is 0.1647
Estimation with Multiple Markets

- There 256 different markets, i.e., 256 pairs of observed types \((x^m_a, x^m_b), m = 1, \ldots, 256\)
- The grid on \(x_a\) has 16 points equally distributed between the interval \([0.12, 0.87]\), and similarly for \(x_b\)
- Use the same true parameter values: \((\alpha^0, \beta^0) = (-5, 11)\)
- For each market with \((x^m_a, x^m_b)\), solve BNE conditions for \((p^m_a, p^m_b)\).
- There are multiple equilibria in most of 256 markets
- For each market, we (randomly) choose an equilibrium to generate 250 data points for that market
- The equilibrium used to generate data can be different in different markets
# of Equilibria with Different \((x^m_a, x^m_b)\)
Estimation with Multiple Markets

• Constrained optimization formulation for MLE

\[
\max_{(\alpha, \beta, \{p^m_a, p^m_b\})} \mathcal{L}(\{p^m_a, p^m_b\}, X)
\]

subject to

\[
p^m_a = \Psi_a(p^m_b, x^m_a, \alpha, \beta)
\]

\[
p^m_b = \Psi_b(p^m_a, x^m_b, \alpha, \beta)
\]

\[
0 \leq p^m_a, p^m_b \leq 1, \quad m = 1, \ldots, 256.
\]
Static Game Example: Monte Carlo Results with Multiple Markets

Truth for alpha is -5

Truth for beta is 11
2-Step Methods

- Recall the constrained optimization formulation for FIML is:

\[
\max_{\{\alpha, \beta, p_a, p_b\}} \mathcal{L}(p_a, p_b, X)
\]

subject to

\[
p_a = \Psi_a(p_b, x_a, \alpha, \beta)
\]

\[
p_b = \Psi_b(p_a, x_b, \alpha, \beta)
\]

\[
0 \leq p_a, p_b \leq 1
\]

- Denote the solution as \((\alpha^*, \beta^*, p_a^*, p_b^*)\)

- Suppose we know \((p_a^*, p_b^*)\), how do we recover \((\alpha^*, \beta^*)\)?
2-Step Methods: Recovering \((\alpha^*, \beta^*)\)

• Idea 1: Solve the BNE equations for \((\alpha^*, \beta^*)\):

\[
\begin{align*}
p_a^* &= \Psi_a(p_b^*, x_a, \alpha, \beta) \\
p_b^* &= \Psi_b(p_a^*, x_b, \alpha, \beta)
\end{align*}
\]

• Idea 2: Choose \((\alpha, \beta)\) to

\[
\max_{(\alpha, \beta)} \mathcal{L}(\Psi_a(p_b^*, x_a, \alpha, \beta), \Psi_b(p_a^*, x_b, \alpha, \beta), X)
\]
2-Step Methods

- **Idea 1**
  - Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$ from the data
  - Step 2: Solve
    
    \[
    \hat{p}_a = \Psi_a(\hat{p}_b, x_b, \alpha, \beta) \\
    \hat{p}_b = \Psi_b(\hat{p}_b, x_b, \alpha, \beta)
    \]

- **Idea 2**
  - Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$ from the data
  - Step 2:
    
    \[
    \max_{(\alpha, \beta)} \mathcal{L}(\Psi_a(\hat{p}_b, x_a, \alpha, \beta), \Psi_b(\hat{p}_a, x_b, \alpha, \beta), X)
    \]
2-Step Methods: Potential Issues to be Addressed

- How do we estimate \( \hat{p} = (\hat{p}_a, \hat{p}_b) \)?
  - Different methods give different \( \hat{p} \)
  - One method is the frequency estimator:
    \[
    \hat{p}_a = \frac{1}{N} \sum_{i=1}^{N} I\{d_{ia}=1\}
    \]
    \[
    \hat{p}_b = \frac{1}{N} \sum_{i=1}^{N} I\{d_{ib}=1\}
    \]

- If \( (\hat{p}_a, \hat{p}_b) \neq (p^*_a, p^*_b) \), then \( (\hat{\alpha}, \hat{\beta}) \neq (\alpha^*, \beta^*) \)

- For a given \( (\hat{p}_a, \hat{p}_b) \), there might not be a solution to the BNE equations
  \[
  \hat{p}_a = \Psi_a(\hat{p}_b, x_a, \alpha, \beta)
  \]
  \[
  \hat{p}_b = \Psi_b(\hat{p}_a, x_b, \alpha, \beta)
  \]
2-Step Methods: Pseudo Maximum Likelihood

- In 2-tep methods
  - Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$
  - Step 2: Solve

$$\max_{\{\alpha, \beta, p_a, p_b\}} \mathcal{L}(p_a, p_b, X)$$
subject to
$$p_a = \Psi_a(\hat{p}_b, x_a, \alpha, \beta)$$
$$p_b = \Psi_b(\hat{p}_a, x_b, \alpha, \beta)$$
$$0 \leq p_a, p_b \leq 1$$

- Or equivalently
  - Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$
  - Step 2: Solve

$$\max_{\{\alpha, \beta\}} \mathcal{L}(\Psi_a(\hat{p}_b, x_a, \alpha, \beta), \Psi_b(\hat{p}_a, x_b, \alpha, \beta), X)$$
2-Step Methods: Least Square Estimators

- Pesendofer and Schmidt-Dengler (2008)
  - Step 1: Estimate $\hat{p} = (\hat{p}_a, \hat{p}_b)$ from the data
  - Step 2:
    \[
    \min_{(\alpha, \beta)} \left\{ (\hat{p}_a - \Psi_a(\hat{p}_b, x_a, \alpha, \beta))^2 + (\hat{p}_b - \Psi_b(\hat{p}_b, x_b, \alpha, \beta))^2 \right\}
    \]

- For dynamic games, Markov perfect equilibrium conditions are characterized by
  \[
  p = \Psi(p, \theta)
  \]

- Step 1: Estimate $\hat{p}$ from the data
- Step 2:
  \[
  \min_{\theta} \quad [\hat{p} - \Psi(\hat{p}, \theta)]'W[\hat{p} - \Psi(\hat{p}, \theta)]'
  \]
Static Game Example: ML v.s. 2-Step PML

- Pakes, Ostrovsky, and Berry (2007): pseudo likelihood function is not a suitable criterion function in a 2-step estimator and can lead to large bias.
Nested Pseudo Likelihood (NPL): Aguirregabiria and Mira (2007)

- NPL iterates on the 2-step methods

1. Estimate $\hat{p}^0 = (\hat{p}_a^0, \hat{p}_b^0)$, set $k = 0$

2. REPEAT
   2.1 Solve 
   $$(\alpha^{k+1}, \beta^{k+1}) = \arg \max_{(\alpha, \beta)} \mathcal{L} \left( \Psi_a(\hat{p}_b^k, x_a, \alpha, \beta), \Psi_b(\hat{p}_a^k, x_b, \alpha, \beta), X \right)$$

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

   $${\hat{p}_a}^{k+1} = \Psi_a(\hat{p}_b^k, x_a, \alpha^{k+1}, \beta^{k+1})$$

   $${\hat{p}_b}^{k+1} = \Psi_b(\hat{p}_a^k, x_b, \alpha^{k+1}, \beta^{k+1})$$

2.3 Let $k := k + 1$;

UNTIL convergence in $(\alpha^k, \beta^k)$ and $(\hat{p}_a^k, \hat{p}_b^k)$
Static Game Example: ML, 2-Step PML and NPL

- Some equilibria in the data are **NOT** best-reply (Lyapunov) stable
- Pesendofer (2010): Best-reply stable is not a reasonable equilibrium selection rule in games of incomplete information
Design of Data Generating Process in Monte Carlo Experiments

- **Monte Carlo 1**: Randomly selected equilibrium in each market
  - In each market, we randomly choose an equilibrium to generate data

- **Monte Carlo 2**: Best-response stable equilibrium with lowest probabilities of entering for firm $a$
  - In each market, we choose the equilibrium that results in the lower probability of entering for firm $a$ to generate data. These equilibria are stable under Best-Reply iteration.

- **Monte Carlo 3**: Best-response stable equilibrium with lowest probabilities of entering for firm $a$
  - In each market, we randomly choose an equilibrium that is stable under Best-Reply iteration to generate data.

- In each experiment, we vary the number of repeated observations $T$. For each experiment and each $T$, we generate 100 datasets and estimate the model using each of the 100 datasets.
Monte Carlo 1: Random Equilibrium in Each Market

- In each market, we randomly choose an equilibrium to generate data
## Monte Carlo 1: $T = 5$ and 10 for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Truth</strong></td>
<td>5</td>
<td>-11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ML (Cons. Opt.)</td>
<td>5.027 (0.179)</td>
<td>-10.743 (0.585)</td>
<td>0.661</td>
<td>1.346</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>2-Step PML</td>
<td>3.068 (0.208)</td>
<td>-7.279 (0.512)</td>
<td>4.228</td>
<td>0.043</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>2-Step LS</td>
<td>2.918 (0.203)</td>
<td>-7.597 (0.654)</td>
<td>4.047</td>
<td>0.048</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>NPL (freq. prob.)</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>N/A</td>
<td>31.527</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>ML (Cons. Opt.)</td>
<td>5.029 (0.126)</td>
<td>-10.816 (0.326)</td>
<td>0.394</td>
<td>0.641</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>2-Step PML</td>
<td>3.719 (0.165)</td>
<td>-8.535 (0.403)</td>
<td>2.812</td>
<td>0.042</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>2-Step LS</td>
<td>3.459 (0.164)</td>
<td>-8.499 (0.531)</td>
<td>2.990</td>
<td>0.049</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>NPL (freq. prob.)</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>N/A</td>
<td>35.756</td>
<td>0</td>
</tr>
</tbody>
</table>
Monte Carlo 1: $T = 25$ and 50 for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Truth</strong></td>
<td>5</td>
<td>-11</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>25</td>
<td>ML</td>
<td>5.018 (0.084)</td>
<td>-10.964 (0.166)</td>
<td>0.189</td>
<td>0.512</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(Cons. Opt.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>2-Step PML</td>
<td>4.302 (0.122)</td>
<td>-9.663 (0.268)</td>
<td>1.537</td>
<td>0.060</td>
<td>100</td>
</tr>
<tr>
<td>25</td>
<td>2-Step LS</td>
<td>3.959 (0.134)</td>
<td>-9.311 (0.354)</td>
<td>2.019</td>
<td>0.050</td>
<td>100</td>
</tr>
<tr>
<td>25</td>
<td>NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>N/A</td>
<td>52.268</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(freq. prob.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>ML</td>
<td>5.005 (0.056)</td>
<td>-11.007 (0.139)</td>
<td>0.150</td>
<td>0.669</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(Cons. Opt.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>2-Step PML</td>
<td>4.590 (0.099)</td>
<td>-10.280 (0.230)</td>
<td>0.865</td>
<td>0.093</td>
<td>100</td>
</tr>
<tr>
<td>50</td>
<td>2-Step LS</td>
<td>4.279 (0.109)</td>
<td>-9.895 (0.283)</td>
<td>1.354</td>
<td>0.052</td>
<td>100</td>
</tr>
<tr>
<td>50</td>
<td>NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>N/A</td>
<td>82.390</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(freq. prob.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Monte Carlo 1: $T = 100$ and $250$ for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Truth</td>
<td>5</td>
<td>-11</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>100</td>
<td>ML</td>
<td>5.006</td>
<td>-10.997</td>
<td>0.102</td>
<td>1.252</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(Cons. Opt.)</td>
<td>(0.045)</td>
<td>(0.092)</td>
<td>(0.045)</td>
<td>(0.092)</td>
<td>–</td>
</tr>
<tr>
<td>100</td>
<td>2-Step PML</td>
<td>4.773</td>
<td>-10.607</td>
<td>0.487</td>
<td>0.174</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(0.067)</td>
<td>(0.165)</td>
<td>(0.067)</td>
<td>(0.165)</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td>100</td>
<td>2-Step LS</td>
<td>4.533</td>
<td>-10.285</td>
<td>0.881</td>
<td>0.053</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(0.084)</td>
<td>(0.200)</td>
<td>(0.084)</td>
<td>(0.200)</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td>100</td>
<td>NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>N/A</td>
<td>150.220</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(freq. prob.)</td>
<td></td>
<td></td>
<td>(N/A)</td>
<td>(N/A)</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>ML</td>
<td>5.000</td>
<td>-10.999</td>
<td>0.063</td>
<td>2.512</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(Cons. Opt.)</td>
<td>(0.028)</td>
<td>(0.057)</td>
<td>(0.028)</td>
<td>(0.057)</td>
<td>–</td>
</tr>
<tr>
<td>250</td>
<td>2-Step PML</td>
<td>4.905</td>
<td>-10.828</td>
<td>0.231</td>
<td>0.410</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(0.043)</td>
<td>(0.114)</td>
<td>(0.043)</td>
<td>(0.114)</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td>250</td>
<td>2-Step LS</td>
<td>4.905</td>
<td>-10.624</td>
<td>0.472</td>
<td>0.054</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(0.051)</td>
<td>(0.157)</td>
<td>(0.051)</td>
<td>(0.157)</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td>250</td>
<td>NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>N/A</td>
<td>351.990</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(freq. prob.)</td>
<td></td>
<td></td>
<td>(N/A)</td>
<td>(N/A)</td>
<td></td>
</tr>
</tbody>
</table>
Monte Carlo 2: B-R Stable Equilibrium with Lowest Probabilities of Entering for Firm $a$

- In each market, we choose the equilibrium that results in the lower probability of entering for firm $a$ to generate data.
- These equilibria are stable under Best-Reply iteration.
Monte Carlo 2: \( T = 5 \) and 10 for Each Market

<table>
<thead>
<tr>
<th>( T )</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\textbf{Truth}</td>
<td>( \alpha = 5 )</td>
<td>( \beta = -11 )</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>ML (Cons. Opt.)</td>
<td>5.234 (0.278)</td>
<td>–11.238 (0.506)</td>
<td>0.665</td>
<td>0.692</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>2-Step PML</td>
<td>4.459 (0.276)</td>
<td>–10.646 (0.796)</td>
<td>1.058</td>
<td>0.040</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>2-Step LS</td>
<td>4.514 (0.347)</td>
<td>–11.369 (1.100)</td>
<td>1.300</td>
<td>0.053</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>NPL (freq. prob.)</td>
<td>4.863 (0.241)</td>
<td>–10.019 (1.830)</td>
<td>1.639</td>
<td>36.051</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>ML (Cons. Opt.)</td>
<td>5.065 (0.143)</td>
<td>–11.111 (0.345)</td>
<td>0.393</td>
<td>0.441</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>2-Step PML</td>
<td>4.787 (0.165)</td>
<td>–10.886 (0.529)</td>
<td>0.602</td>
<td>0.043</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>2-Step LS</td>
<td>4.914 (0.238)</td>
<td>–11.473 (0.852)</td>
<td>1.002</td>
<td>0.055</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>NPL (freq. prob.)</td>
<td>5.054 (0.241)</td>
<td>–10.411 (1.830)</td>
<td>0.958</td>
<td>33.153</td>
<td>29</td>
</tr>
</tbody>
</table>
Monte Carlo 2: $T = 25$ and $50$ for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Truth</td>
<td>5</td>
<td>-11</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>25</td>
<td>ML (Cons. Opt.)</td>
<td>5.018 (0.076)</td>
<td>-11.022 (0.181)</td>
<td>0.197</td>
<td>0.417</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step PML</td>
<td>4.926 (0.114)</td>
<td>-11.040 (0.264)</td>
<td>0.298</td>
<td>0.058</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step LS</td>
<td>5.014 (0.147)</td>
<td>-11.387 (0.479)</td>
<td>0.632</td>
<td>0.057</td>
<td>100</td>
</tr>
<tr>
<td>25</td>
<td>NPL (freq. prob.)</td>
<td>4.995 (0.081)</td>
<td>-10.607 (0.563)</td>
<td>0.688</td>
<td>29.122</td>
<td>71</td>
</tr>
<tr>
<td>50</td>
<td>ML (Cons. Opt.)</td>
<td>5.000 (0.061)</td>
<td>-11.000 (0.133)</td>
<td>0.146</td>
<td>0.398</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step PML</td>
<td>4.956 (0.080)</td>
<td>-10.983 (0.198)</td>
<td>0.218</td>
<td>0.090</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step LS</td>
<td>5.007 (0.109)</td>
<td>-11.119 (0.329)</td>
<td>0.365</td>
<td>0.056</td>
<td>100</td>
</tr>
<tr>
<td>50</td>
<td>NPL (freq. prob.)</td>
<td>4.998 (0.070)</td>
<td>-10.665 (0.472)</td>
<td>0.581</td>
<td>32.133</td>
<td>86</td>
</tr>
</tbody>
</table>
Monte Carlo 2: $T = 100$ and 250 for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Truth</td>
<td></td>
<td>5</td>
<td>-11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>ML (Cons. Opt.)</td>
<td>5.005 (0.046)</td>
<td>-10.996 (0.103)</td>
<td>0.112</td>
<td>0.858</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>2-Step PML</td>
<td>4.985 (0.060)</td>
<td>-11.011 (0.164)</td>
<td>0.175</td>
<td>0.164</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>2-Step LS</td>
<td>5.011 (0.077)</td>
<td>-11.090 (0.238)</td>
<td>0.265</td>
<td>0.056</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>NPL (freq. prob.)</td>
<td>5.005 (0.051)</td>
<td>-10.908 (0.283)</td>
<td>0.301</td>
<td>34.516</td>
<td>96</td>
</tr>
<tr>
<td>250</td>
<td>ML (Cons. Opt.)</td>
<td>5.000 (0.031)</td>
<td>-10.995 (0.062)</td>
<td>0.069</td>
<td>1.798</td>
<td>100</td>
</tr>
<tr>
<td>250</td>
<td>2-Step PML</td>
<td>4.994 (0.037)</td>
<td>-11.002 (0.092)</td>
<td>0.099</td>
<td>0.379</td>
<td>100</td>
</tr>
<tr>
<td>250</td>
<td>2-Step LS</td>
<td>5.005 (0.042)</td>
<td>-11.025 (0.152)</td>
<td>0.160</td>
<td>0.057</td>
<td>100</td>
</tr>
<tr>
<td>250</td>
<td>NPL (freq. prob.)</td>
<td>5.002 (0.051)</td>
<td>-10.955 (0.283)</td>
<td>0.198</td>
<td>57.083</td>
<td>100</td>
</tr>
</tbody>
</table>
Monte Carlo 3: B-R Stable Equilibrium in Each Market

- In each market, we randomly choose an equilibrium that is stable under Best-Reply iteration to generate data.
# Monte Carlo 3: $T = 5$ and 10 for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\text{Truth}$</td>
<td>5</td>
<td>-11</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>ML (Cons. Opt.)</td>
<td>5.197</td>
<td>-11.189</td>
<td>0.588</td>
<td>0.803</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.245)</td>
<td>(0.463)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2-Step PML</td>
<td>4.380</td>
<td>-10.427</td>
<td>1.132</td>
<td>0.040</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.263)</td>
<td>(0.711)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2-Step LS</td>
<td>4.395</td>
<td>-11.131</td>
<td>1.278</td>
<td>0.053</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.318)</td>
<td>(1.078)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NPL (freq. prob.)</td>
<td>4.707</td>
<td>-8.534</td>
<td>2.574</td>
<td>34.847</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.241)</td>
<td>(1.830)</td>
<td></td>
<td></td>
<td>975</td>
</tr>
<tr>
<td>10</td>
<td>ML (Cons. Opt.)</td>
<td>5.104</td>
<td>-11.038</td>
<td>0.354</td>
<td>0.472</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.149)</td>
<td>(0.304)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2-Step PML</td>
<td>4.787</td>
<td>-10.831</td>
<td>0.615</td>
<td>0.043</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.181)</td>
<td>(0.523)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2-Step LS</td>
<td>4.893</td>
<td>-11.418</td>
<td>0.942</td>
<td>0.055</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.243)</td>
<td>(0.805)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>NPL (freq. prob.)</td>
<td>5.019</td>
<td>-9.732</td>
<td>1.534</td>
<td>28.135</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.148)</td>
<td>(0.753)</td>
<td></td>
<td></td>
<td>682</td>
</tr>
</tbody>
</table>
Monte Carlo 3: $T = 25$ and 50 for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Truth</td>
<td>5</td>
<td>-11</td>
<td>0.214</td>
<td>0.245</td>
<td>100</td>
</tr>
<tr>
<td>25</td>
<td>ML (Cons. Opt.)</td>
<td>5.040</td>
<td>-10.992</td>
<td>0.214</td>
<td>0.245</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step PML</td>
<td>4.945</td>
<td>-10.943</td>
<td>0.335</td>
<td>0.059</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step LS</td>
<td>5.022</td>
<td>-11.175</td>
<td>0.553</td>
<td>0.057</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>NPL (freq. prob.)</td>
<td>5.032</td>
<td>-10.087</td>
<td>1.229</td>
<td>25.661</td>
<td>75</td>
</tr>
<tr>
<td>50</td>
<td>ML (Cons. Opt.)</td>
<td>5.009</td>
<td>-10.999</td>
<td>0.160</td>
<td>0.380</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step PML</td>
<td>4.967</td>
<td>-10.990</td>
<td>0.223</td>
<td>0.091</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2-Step LS</td>
<td>5.016</td>
<td>-11.106</td>
<td>0.374</td>
<td>0.058</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>NPL (freq. prob.)</td>
<td>5.018</td>
<td>-10.243</td>
<td>1.087</td>
<td>30.148</td>
<td>86</td>
</tr>
</tbody>
</table>
Monte Carlo 3: $T = 100$ and 250 for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Truth</strong></td>
<td>5</td>
<td>-11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>ML (Cons. Opt.)</td>
<td>5.011</td>
<td>-10.982</td>
<td>0.107</td>
<td>0.821</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.046)</td>
<td>(0.095)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>2-Step PML</td>
<td>4.995</td>
<td>-11.011</td>
<td>0.176</td>
<td>0.164</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.060)</td>
<td>(0.164)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>2-Step LS</td>
<td>5.022</td>
<td>-11.090</td>
<td>0.275</td>
<td>0.059</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.077)</td>
<td>(0.249)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>NPL (freq. prob.)</td>
<td>5.024</td>
<td>-10.661</td>
<td>0.733</td>
<td>30.406</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.060)</td>
<td>(0.650)</td>
<td></td>
<td></td>
<td>225</td>
</tr>
<tr>
<td>250</td>
<td>ML (Cons. Opt.)</td>
<td>5.003</td>
<td>-10.993</td>
<td>0.062</td>
<td>1.838</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.025)</td>
<td>(0.057)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>2-Step PML</td>
<td>4.9957</td>
<td>-11.000</td>
<td>0.108</td>
<td>0.377</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.034)</td>
<td>(0.103)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>2-Step LS</td>
<td>5.008</td>
<td>-11.025</td>
<td>0.176</td>
<td>0.060</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.040)</td>
<td>(0.171)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>NPL (freq. prob.)</td>
<td>5.010</td>
<td>-10.854</td>
<td>0.470</td>
<td>53.572</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.060)</td>
<td>(0.650)</td>
<td></td>
<td></td>
<td>168</td>
</tr>
</tbody>
</table>
Monte Carlo 3: $T = 100$ and 250 for Each Market

<table>
<thead>
<tr>
<th>$T$</th>
<th>Estimator</th>
<th>Estimates</th>
<th>RMSE</th>
<th>CPU (sec.)</th>
<th># of Data Sets</th>
<th>Avg. NPL Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Truth</td>
<td>5</td>
<td>-11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>ML (Cons. Opt.)</td>
<td>5.011</td>
<td>-10.982</td>
<td>0.107</td>
<td>0.821</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>2-Step PML</td>
<td>4.995</td>
<td>-11.011</td>
<td>0.176</td>
<td>0.164</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>2-Step LS</td>
<td>5.022</td>
<td>-11.090</td>
<td>0.275</td>
<td>0.059</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>NPL (freq. prob.)</td>
<td>5.024</td>
<td>-10.661</td>
<td>0.733</td>
<td>30.406</td>
<td>99</td>
</tr>
<tr>
<td>250</td>
<td>ML (Cons. Opt.)</td>
<td>5.003</td>
<td>-10.993</td>
<td>0.062</td>
<td>1.838</td>
<td>100</td>
</tr>
<tr>
<td>250</td>
<td>2-Step PML</td>
<td>4.9957</td>
<td>-11.000</td>
<td>0.108</td>
<td>0.377</td>
<td>100</td>
</tr>
<tr>
<td>250</td>
<td>2-Step LS</td>
<td>5.008</td>
<td>-11.025</td>
<td>0.176</td>
<td>0.060</td>
<td>100</td>
</tr>
<tr>
<td>250</td>
<td>NPL (freq. prob.)</td>
<td>5.010</td>
<td>-10.854</td>
<td>0.470</td>
<td>53.572</td>
<td>100</td>
</tr>
</tbody>
</table>
Dynamic Game: Egesdal, Lai and Su (2012)

The Example in Kasahara and Shimotsu (2012) with $\theta_{RN} = 2$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$T$</th>
<th>Estimator</th>
<th>$\theta_{RN}$</th>
<th>$\theta_{RS}$</th>
<th>CPU Time Per Run (sec.)</th>
<th># of Data Sets Converged</th>
<th>Avg. NPL(-$\Lambda$)</th>
<th>Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>1</td>
<td>MLE</td>
<td>1.895 (0.580)</td>
<td>0.961 (0.156)</td>
<td>0.27</td>
<td>100</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>2S-PML</td>
<td>1.134 (0.616)</td>
<td>0.753 (0.171)</td>
<td>0.02</td>
<td>100</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>NPL</td>
<td>1.909 (0.628)</td>
<td>0.964 (0.168)</td>
<td>0.45</td>
<td>100</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>$\Lambda$-NPL</td>
<td>1.909 (0.628)</td>
<td>0.964 (0.168)</td>
<td>0.42</td>
<td>100</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>MLE</td>
<td>1.970 (0.158)</td>
<td>0.992 (0.042)</td>
<td>0.16</td>
<td>100</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>2S-PML</td>
<td>1.819 (0.236)</td>
<td>0.951 (0.062)</td>
<td>0.03</td>
<td>100</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>NPL</td>
<td>1.963 (0.191)</td>
<td>0.991 (0.050)</td>
<td>0.61</td>
<td>100</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>$\Lambda$-NPL</td>
<td>1.963 (0.191)</td>
<td>0.991 (0.050)</td>
<td>0.56</td>
<td>100</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>MLE</td>
<td>2.001 (0.118)</td>
<td>1.000 (0.033)</td>
<td>0.15</td>
<td>100</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>2S-PML</td>
<td>1.923 (0.158)</td>
<td>0.979 (0.042)</td>
<td>0.06</td>
<td>100</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>NPL</td>
<td>1.999 (0.129)</td>
<td>0.999 (0.036)</td>
<td>1.01</td>
<td>100</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>$\Lambda$-NPL</td>
<td>1.999 (0.129)</td>
<td>0.999 (0.036)</td>
<td>0.91</td>
<td>100</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Game: Egesdal, Lai and Su (2012)

The Example in Kasahara and Shimotsu (2012) with $\theta_{RN} = 4$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$T$</th>
<th>Estimator</th>
<th>$\theta_{RN}$</th>
<th>$\theta_{RS}$</th>
<th>CPU Time Per Run (sec.)</th>
<th># of Data Sets Converged</th>
<th>Avg. NPL(-$\Lambda$) Iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Truth</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>MLE</td>
<td>4.055 (0.613)</td>
<td>1.003 (0.158)</td>
<td>0.61</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>2S-PML</td>
<td>3.107 (0.442)</td>
<td>0.839 (0.099)</td>
<td>0.02</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>1.68</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>$\Lambda$-NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>1.68</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>MLE</td>
<td>4.003 (0.039)</td>
<td>1.000 (0.016)</td>
<td>0.50</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>2S-PML</td>
<td>3.902 (0.099)</td>
<td>0.983 (0.025)</td>
<td>0.04</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>7.61</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>400</td>
<td>10</td>
<td>$\Lambda$-NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>7.54</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>MLE</td>
<td>4.003 (0.032)</td>
<td>1.001 (0.011)</td>
<td>0.47</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>2S-PML</td>
<td>3.954 (0.084)</td>
<td>0.992 (0.019)</td>
<td>0.06</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>12.38</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>$\Lambda$-NPL</td>
<td>N/A (N/A)</td>
<td>N/A (N/A)</td>
<td>12.41</td>
<td>0</td>
<td>250</td>
</tr>
</tbody>
</table>
Conclusion

• NPL (Aguirregabiria and Mira 2007) is not an appropriate method for estimating games
• Estimation of dynamic games is an interesting but challenging computational optimization problem
  • Exploring sparsity patterns in constraint Jacobian and Hessian in numerical implementation
• Ongoing research
  • Estimation of dynamic discrete choice games of incomplete information – Egesdal, Lai and Su (2012)