Bayesian Estimation of Dynamic Discrete Choice Models and its (potential) applications

Andrew Ching
Rotman School of Management
University of Toronto

November 10, 2010

Introduction

- My presentation will be heavily drawn from the following papers:
  - "Bayesian Estimation of Dynamic Discrete Choice Models," 2009 (with Susumu Imai and Neelam Jain).
  - "A Practitioner's Guide to Bayesian Estimation of Discrete Choice Dynamic Programming Models," 2010 (with Susumu Imai, Masakazu Ishihara and Neelam Jain).
  - "Dynamics of Consumer Adoption of New Technology: The Case of ATM Cards," 2010 (with Botao Yang).

Motivation

- A common framework to study consumers' or firms' forward-looking behavior is **the discrete choice dynamic programming (DDP) models**.
- DDP models have been applied to many interesting dynamic problems:
  - medical treatment or work absence decisions (e.g., Gilleskie 1998)
  - equipment replacement decisions (e.g., Rust 1987)
  - consumer learning and brand choice (e.g., Erdem and Keane 1996; Gönül and Srinivasan 1996; Crawford and Shum 2005)
  - quantity choice with stockpiling behavior (e.g., Erdem et al. 2003; Sun 2005; Hendel and Nevo 2006)
  - new product/technology adoption decisions (e.g., Ackerberg 2003)
  - career or retirement decisions (e.g., Keane and Wolpin 1997; Blau and Gilleskie 2008)

Motivation (cont'd)

- Maximum likelihood/GMM is a common way to estimate DDP models. However, it is hard to implement due to
  - the curse of dimensionality in the state space.
  - the complexity of the likelihood/GMM objective function makes it difficult to search for the global maximum/minimum.
- Standard Bayesian MCMC approach is also computationally burdensome – it typically requires many more iterations than the classical approach to achieve convergence.
- This motivates my coauthors and I to propose a new MCMC algorithm that could alleviate the burden of estimating infinite horizon DDP models.
  - It makes use of the past outcome of the algorithm.
  - It estimates and solves the DDP model simultaneously.
  - It can incorporate random coefficients.

Roadmap

1. A simple dynamic store choice model with reward program.
2. Brief review of the stationary dynamic programming (DP) problem.
3. Estimation methods.
   (a) Conventional Bayesian approach
   (b) IJC algorithm
4. Estimation Results.
5. Extension 1: continuous state space.
6. Extension 2: finite horizon non-stationary DP problems.
7. Conclusion.

Model (An Example)

A simple dynamic model of store choice with reward programs.

- There are two supermarkets in a city $(j = 1, 2)$.
- Each store offers a stamp card, which can be exchanged for a gift upon completion.
- Consumers get one stamp for each visit with a purchase.
- Reward programs at the two supermarkets differ in terms of: (i) the number of stamps required for a reward $(\bar{S}_j)$, and (ii) the mean value of the reward (G_j).
- $G_{ij} \sim N(G_j, \sigma^2_{G_j})$, i indexes consumers.
- Once consumers receive a gift, they will start with a blank stamp card again in the next period.

Model (cont'd)

- $p_{ijt} \sim i.i.d.\ N(\underline{p}, \sigma_p^2)$ (known to consumers).
- Let $s_i = (s_{i1}, s_{i2})$ be the number of stamps collected.
- Consumer i's single period utility of visiting supermarket j is:

$$U_{ijt}(s_{it}, p_{it}; \theta) = \begin{cases} \alpha_j - \gamma p_{ijt} + G_{ij} + \epsilon_{ijt} & \text{if } s_{ijt} = \bar{S}_j - 1, \\ \alpha_j - \gamma p_{ijt} + \epsilon_{ijt} & \text{otherwise.} \end{cases}$$

- The mean utility of "not shopping" is normalized to zero, i.e., $U_{i0t}(s) = \epsilon_{i0t}$.
- We assume that ϵ follows *i.i.d.* extreme value distribution.

Model (cont'd)

- The consumer i's objective is to choose a sequence of store choices to maximize the sum of the present discounted future utility.

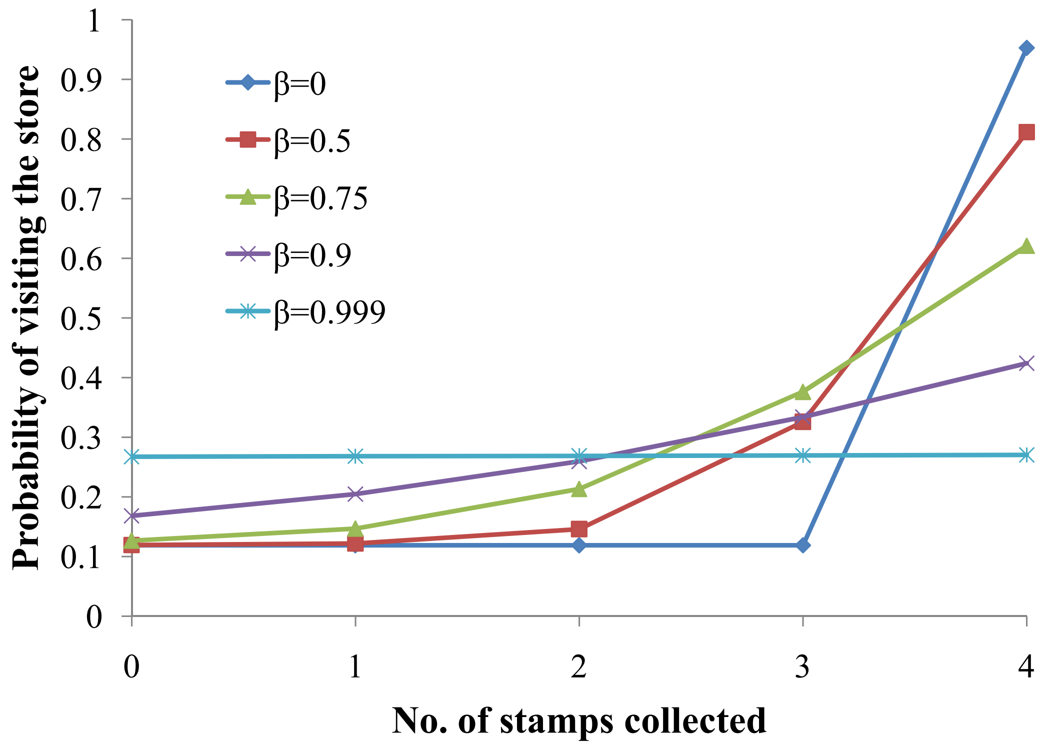
$$\max_{\{b_{ijt}\}_{t=1}^{\infty}} E\left[\sum_{t=1}^{\infty} \beta^{t-1} \sum_{j=0}^{2} b_{ijt} \cdot U_{ijt}(s_{it}, p_{it}; \theta)\right]$$

  where $b_{ijt} = 1$ if consumer i chooses store j in period t and $b_{ijt} = 0$ otherwise, and β is the discount factor.

- The evolution of the state, s_{it}, is as follows.

$$s_{ijt+1} = \begin{cases} s_{ijt} + 1 & \text{if } b_{ijt} = 1 \text{ and } s_{ijt} < \bar{S}_j - 1 \\ 0 & \text{if } b_{ijt} = 1 \text{ and } s_{ijt} = \bar{S}_j - 1 \\ s_{ijt} & \text{if } b_{ijt} = 0. \end{cases}$$

Choice Probabilities



- Only one store and the outside option.
- Other parameters are set as $\bar{S}_1 = 5$, $\alpha_1 = -2.0$, $G_1 = 3.0$, $\sigma_{G_1} = 0$, and $\gamma = -1.0$.

Bellman Equation

The Bellman equation for the above dynamic optimization problem:

$$
\begin{aligned}
& V(s_i; p_i, G_i, \theta) \\
& \quad \equiv \quad E_\epsilon \max\{V_0(s_i; G_i, \theta) + \epsilon_{i0}, V_1(s_i; p_{i1}, G_i, \theta) + \epsilon_{i1}, V_2(s_i; p_{i2}, G_i, \theta) + \epsilon_{i2}\} \\
& \quad = \quad \log(\exp(V_0(s_i; G_i, \theta)) + \exp(V_1(s_i; p_{i1}, G_i, \theta)) + \exp(V_2(s_i; p_{i2}, G_i, \theta))).
\end{aligned}
$$

The second equality follows from the extreme value assumption on ϵ.

Alternative specific value functions are given by

$$
\begin{aligned}
V_j(s_i; p_{ij}, G_i, \theta) &= \begin{cases} \alpha_j - \gamma p_{ij} + G_{ij} + \beta E_{p'} V(s'; p', G_i, \theta) & \text{if } s_{ij} = \bar{S}_j - 1, \\ \alpha_j - \gamma p_{ij} + \beta E_{p'} V(s'; p', G_i, \theta) & \text{otherwise,} \end{cases} \\
V_0(s_i; G_i, \theta) &= \beta E_{p'} V(s'; p', G_i, \theta)
\end{aligned}
$$

The expectation with respect to p' is defined by

$$
E_{p'} V(s'; p', G_i, \theta) = \int V(s'; p', G_i; \theta) dF(p').
$$

Method of Successive Approximation

- Let Γ be the Bellman operator.
- For an arbitrary function, f,

$$\Gamma f(s_i, p_i) = E_\epsilon \max_j \{\bar{U}_j(s_i, p_i; G_i, \theta) + \epsilon_{ijt} + \beta E_{p'}[f(s_i', p')]\}.$$

  where $\bar{U}_j(s_i, p_i; G_i, \theta) = U_{ijt} - \epsilon_{ijt}$.

- The value function, $V(s_i, p_i; G_i, \theta)$, is a fixed point to Γ, i.e., $V = \Gamma V$.
- Note that Γ is a contraction mapping if $\beta < 1$:
  - there is a unique fixed point to Γ.
  - starting with an initial guess of the value function, V^0, and recursively applying Γ to it, i.e., $V^{n+1} = \Gamma V^n$, then $V^n \to V$ uniformly.

Conventional Bayesian Approach (CB)

- Outer loop (MCMC algorithm)
  - Use Metropolis-Hastings algorithm and simulate a markov-chain of $\{\theta^l\}$, which converges to the true posterior distribution.
  - Given the candidate parameter value, θ^{*r}, in MCMC iteration r, the computation of the likelihood requires the value functions, $V(s, p; \theta^{*r})$.
- Inner loop (Method of successive approximation)
  - Given θ^{*r}, solve for the value functions, $V(s, p; \theta^{*r})$, by the method of successive approximation.

CB: Outer loop

- Posterior distribution of θ is proportional to $\pi(\theta)\rho(\mathbf{b^d}|\theta)$ where

$$\rho(\mathbf{b^d}|\theta) = \prod_{i=1}^{I}\prod_{t=1}^{T_i}\prod_{j=0}^{2}\left(\frac{\exp(V_j(s_{it}^d, p_{ijt}^d;\theta))}{\sum_{k=0}^{2}\exp(V_k(s_{it}^d, p_{ijt}^d;\theta))}\right)^{b_{ijt}^d}.$$

- Metropolis-Hastings (in iteration r)
  - Draw θ^{*r} (candidate value) from a proposal distribution, $q(\theta^{r-1}, \theta^{*r})$ (e.g., $\theta^{*r} \sim N(\theta^{r-1}, \sigma^2)$).
  - We accept θ^{*r} (i.e., set $\theta^r = \theta^{*r}$) with probability

$$\lambda = \min\left(\frac{\pi(\theta^{*r})\rho(\mathbf{b^d}|\theta^{*r})q(\theta^{*r}, \theta^{r-1})}{\pi(\theta^{r-1})\rho(\mathbf{b^d}|\theta^{r-1})q(\theta^{r-1}, \theta^{*r})}, 1\right),$$

  and we reject θ^{*r} (i.e., set $\theta^r = \theta^{r-1}$) with probability $1 - \lambda$.

CB: Inner loop

Solve for the value functions V $\forall s$ by the method of successive approximation.

1. Make M draws of $\{p^m\}_{m=1}^M$ ($p^m = (p_1^m, p_2^m)$).
2. Starting with an initial guess, say, $V^0(s, p^m; \theta^{*r}) = 0 \ \forall s, p$. Suppose we know V^l.
3. For each s, substitute $\{p^m\}$ into $V^l(s, p; \theta^{*r})$, and compute the Monte Carlo approximation of the expectation:

$$\bar{E}_{p'} V^l(s, p'; \theta^{*r}) = \frac{1}{M} \sum_{m=1}^{M} V^l(s, p^m; \theta^{*r}).$$

4. Apply the Bellman operator (Γ) once and get V^{l+1}, i.e.,

$$V^{l+1} = \Gamma V^l = E_\epsilon \max_j \{\bar{U}(s, p^m; \theta^{*r}) + \epsilon_{ijt} + \beta \bar{E}_{p'}[V^l(s, p'; \theta^{*r})]\}.$$

5. Repeat 3&4 until $\bar{E}_{p'} V^{l+1}(s, p; \theta^{*r})$ converges.

Note that this is computationally costly (needs to be done in all iterations), and the computational cost increases with the size of the state space.

Highlights of IJC method

- In the conventional approach, the value functions need to be solved at every trial parameter vector (θ^{*r}).
- The value functions computed at past parameter vectors are simply thrown away!
- Imai, Jain and Ching (2009) (IJC) algorithm:
  - In each MCMC iteration, the value function is only partially solved (at the minimum, only apply the Bellman operator once). We call them **pseudo-value functions**.
  - Store those pseudo-value functions evaluated at past parameter vectors, and use them to approximate the value functions at the current parameter vector nonparametrically.
  - This nonparametric approximation can be computationally much cheaper than the method of successive approximation.

IJC Algorithm

- Outer loop (MCMC algorithm)
  - Similar to the conventional Bayesian approach.
  - Use the likelihood constructed based on **pseudo alternative specific value functions**, $\tilde{V}_j^r$. (thus, we also call the likelihood the pseudo-likelihood).
- Inner loop (Key innovation of the IJC algorithm)
  - Approximate the expected future value at θ^{*r} by the weighted average of the past pseudo-value functions.
  - Apply the Bellman operator once to get pseudo-value function evaluated at θ^{*r}, and store it.

IJC: Inner loop

- Let $H^r = \{\theta^{*l}; \tilde{V}^l(s, p^l; \theta^{*l}), \forall s\}_{l=r-N}^{r-1}$ be the outcome of the algorithm to iteration $r-1$.

- For each s, the expected future value at the current parameter value (θ^{*r}) is approximated as

$$\hat{E}^r_{p'}[V(s, p'; \theta^{*r})] = \sum_{l=r-N}^{r-1} \tilde{V}^l(s', p^l, \theta^{*l}) \frac{K_h(\theta^{*r} - \theta^{*l})}{\sum_{k=r-N}^{r-1} K_h(\theta^{*r} - \theta^{*l})},$$

  where $K_h()$ is a kernel with bandwidth $h > 0$.

- Pseudo alternative specific value functions are then

$$\tilde{V}^r_j(s, p_j; \theta^{*r}) = \begin{cases} \alpha_j - \gamma p_j + \beta \hat{E}^r_{p'}[V(s, p'; \theta^{*r})] & \text{if } s_j < \bar{S}_j - 1, \\ \alpha_j - \gamma p_j + G_j + \beta \hat{E}^r_{p'}[V(s, p'; \theta^{*r})] & \text{if } s_j = \bar{S}_j - 1, \end{cases}$$

$$\tilde{V}^r_0(s, p_j; \theta^{*r}) = \beta \hat{E}^r_{p'}[V(s, p'; \theta^{*r})].$$

IJC: Inner loop (cont'd)

- Simulate one draw of price vector, p^r, from the known price distribution. Apply the Bellman operator once and obtain the pseudo-value function:

$$\tilde{V}^r(s, p^r; \theta^{*r}) = E_\epsilon \max_j \{\bar{U}_{ijt}(s, p^r; \theta^{*r}) + \epsilon_{ijt} + \hat{E}_{p'}[V^l(s, p'; \theta^{*r})]\}.$$

  We store $\{\theta^{*r}; \tilde{V}^r(s, p^r; \theta^{*r}), \forall s\}$ and update the outcome to H^{r+1}.

Remark 1: Theoretical Results

When S is discrete, and N increases at an appropriate rate as l increases, IJC show that:

- Theorem 1: The sequence of approximated value function V^l converges to V in probability uniformly over (S, ϵ, Θ).
- Theorem 2: θ^l converges to the true posterior distribution in total variation norm.
- Note that in practice, we need to fix N.

Remark 1: Intuition

- Why pseudo-value functions converge to the true ones?
  - For an arbitrary parameter value, consider its neighborhood.
  - As the number of MCMC iterations increases, the number of times this neighborhood is visited increases (i.e., candidate parameter vector falls within the neighborhood).
  - Thus, the "effective" number of times we apply the Bellman operator to the neighborhood increases
    $\Rightarrow$ pseudo-value functions will converge to the true ones.
- The nonparametric approximation becomes better as
  - pseudo-value functions approach the true ones.
  - we have enough past pseudo-value functions evaluated at various parameter values.

How to choose N and h

- How should we choose N? (N: no. of past pseudo-value functions used for the approximation of the expected future value)
  - The rate of convergence is usually faster for smaller N, because the set of past pseudo-value functions used for the approximation are more recent and accurate.
  - However, smaller N also leads to the weighted average of the pseudo-value functions more sensitive to the change in the set. As a result, we may get larger posterior standard deviation.
- How should we choose h? (h: bandwidth)

How to choose N and h (cont'd)

A general strategy to choose N and h (bandwidth): check how good the approximation for the value function is during the estimation.

- For instance, in every 1,000 iterations, compute the means of the MCMC draws, $\bar{\tilde{\theta}}$, and compare the distance between the pseudo-value function and the exact value function at $\bar{\tilde{\theta}}$.
- If the distance is larger than the tolerance level set by the researcher, increase N and/or reduce h (bandwidth), and vice versa.
- If memory is not a constraint, store a large N past pseudo-value functions and use the most recent $N' < N$ to do the Emax approximation.
- One can increase N' as soon as the researcher discovers that the approximation is not good enough.
- Then set a new h based on standard optimal bandwidth formula, e.g., Silverman's rule of thumb.

Estimation 1

- We simulate samples based on the solution of the dynamic model with and without unobserved consumer heterogeneity.
- True parameters of the simulation
  - Store intercepts: $\alpha_1 = \alpha_2 = 0$.
  - Population mean of rewards: $G_1 = 1.0$, $G_2 = 5.0$.
  - Population variance of rewards: $\sigma_{G_1} = 0$, and $\sigma_{G_2} = 0$ (no heterogeneity) or 1.0 (heterogeneity).
  - Number of stamps for rewards: $\bar{S}_1 = 2$, $\bar{S}_2 = 4$.
  - Price coefficient: $\gamma = -1.0$.
  - Discount factor: $\beta = 0.6$ or 0.8.

Results: Homogeneous Model

		β = 0.6		β = 0.8	
parameter	TRUE	mean	sd	mean	sd
α_1 (intercept for store 1)	0.0	-0.001	0.019	-0.030	0.022
α_2 (intercept for store 2)	0.0	-0.002	0.019	-0.018	0.028
G_1 (reward for store 1)	1.0	0.998	0.017	1.052	0.021
G_2 (reward for store 2)	5.0	5.032	0.048	5.088	0.085
γ (price coefficient)	-1.0	-0.999	0.016	-0.996	0.019
β (discount factor)	0.6/0.8	0.601	0.008	0.800	0.010

Sample size: 100,000 (1,000 consumers for 100 periods)
Fixed parameters: $\bar{S}_1 = 2$, $\bar{S}_2 = 4$, $\bar{p} = 1.0$, $\sigma_p = 0.3$, $\sigma_{G_j} = 0$ for $j = 1, 2$
Tuning parameters: $N = 1,000$ (no. of past pseudo-value functions used for Emax function approximation), $h = 0.01$ (bandwidth).

Results: Heterogeneous Model

		$\beta = 0.6$		$\beta = 0.8$	
parameter	TRUE	mean	sd	mean	sd
α_1 (intercept for store 1)	0.0	-0.005	0.019	-0.022	0.022
α_2 (intercept for store 2)	0.0	0.010	0.021	0.005	0.037
G_1 (reward for store 1)	1.0	1.017	0.017	1.010	0.019
G_2 (reward for store 2)	5.0	5.066	0.065	4.945	0.130
σ_{G2} (sd of G_2)	1.0	1.034	0.046	1.029	0.040
γ (price coefficient)	-1.0	-1.004	0.016	-0.985	0.019
β (discount factor)	0.6/0.8	0.595	0.005	0.798	0.006

Sample size: 100,000 (1,000 consumers for 100 periods)
Fixed parameters: $\bar{S}_1 = 2$, $\bar{S}_2 = 4$, $\bar{p} = 1.0$, $\sigma_p = 0.3$, $\sigma_{G_1} = 0$
Tuning parameters: $N = 1,000$ (no. of past pseudo-value functions used for Emax function approximation), $h = 0.01$ (bandwidth).

Comparison of Computation Time

Average seconds per iteration

	Homogeneous Model			Heterogeneous Model		
algorithm	$\beta = 0.6$	$\beta = 0.8$	$\beta = 0.98$	$\beta = 0.6$	$\beta = 0.8$	$\beta = 0.98$
Full solution based Bayesian	0.782	0.807	1.410	31.526	65.380	613.26
IJC with N=1000	1.071	1.049	1.006	19.300	19.599	18.387

- For the IJC algorithm, every time a candidate parameter value is rejected, we re-computed the likelihood function at previously accepted parameter value, which will be used to evaluate the likelihood ratio in the Metropolis-Hastings step.
- For the full solution based Bayesian algorithm, we set the number of draws for the integration of prices to be 100.
- Note that the average time per iteration for the full-solution based MCMC increases exponentially with β when estimating a model with unobserved heterogeneity, while it remains unchanged for IJC.
- For $\beta = 0.98$ and no. of iterations $=$ 10,000, full-solution based MCMC would run for 70 days, while IJC would run for 2.5 days.

Estimation 2: Impact of N

- We simulate samples based on the solution of the dynamic model. We then estimate the parameters of the model by using different N's.
- True parameters of the simulation
  - Store intercepts: $\alpha_1 = \alpha_2 = 0$.
  - Population mean of rewards: $G_1 = 1.0$, $G_2 = 10.0$.
  - Population variance of rewards: $\sigma_{G_1} = \sigma_{G_2} = 0$ (no heterogeneity).
  - Number of stamps for rewards: $\bar{S}_1 = 5$, $\bar{S}_2 = 10$.
  - Price coefficient: $\gamma = -1.0$.
  - Discount factor: $\beta = 0.98$.
- We use gaussian kernel with bandwidth=0.02, and estimate α_1, α_2, G_1, G_2, γ.

Estimation 2: Impact of N (cont'd)

Parameter estimates

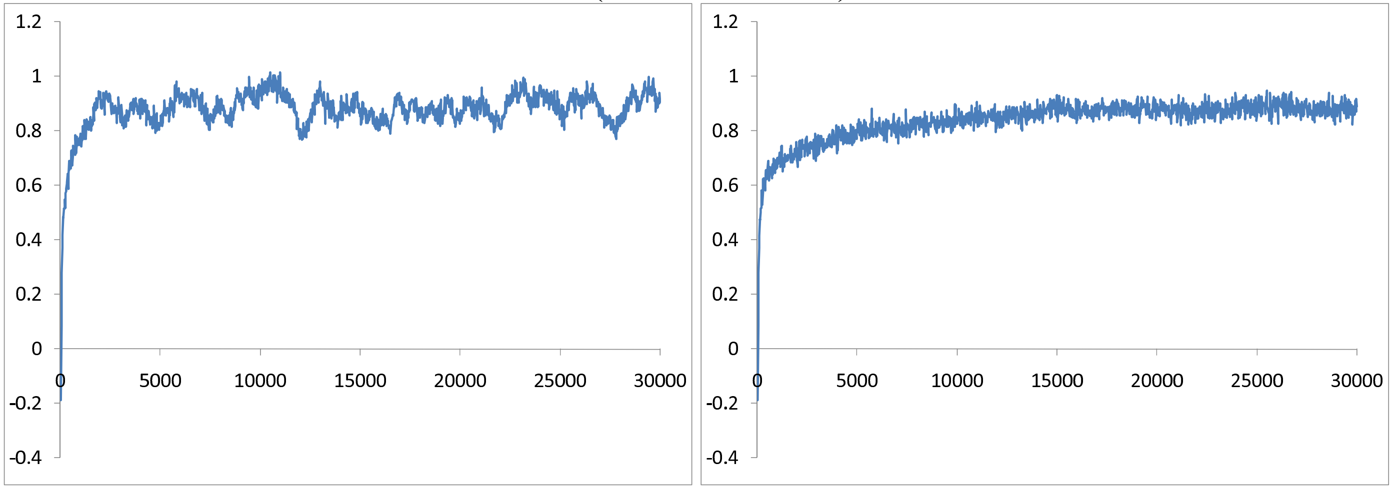
		N=100		N=1000	
parameter	TRUE	mean	sd	mean	sd
α_1 (intercept for store 1)	0.0	-0.049	0.020	-0.061	0.020
α_2 (intercept for store 2)	0.0	0.032	0.019	0.022	0.019
G_1 (reward for store 1)	1.0	1.234	0.034	1.246	0.021
G_2 (reward for store 2)	10.0	9.740	0.063	9.751	0.028
γ (price coefficient)	-1.0	-1.000	0.018	-0.991	0.018

Sample size: 100,000 (1,000 consumers for 100 periods)
Fixed parameters: $\bar{S}_1 = 5$, $\bar{S}_2 = 10$, $\bar{p} = 1.0$, $\sigma_p = 0.3$, $\sigma_{G_j} = 0$ for $j = 1, 2$, $\beta = 0.98$
Tuning parameters: $h = 0.01$ (bandwidth).

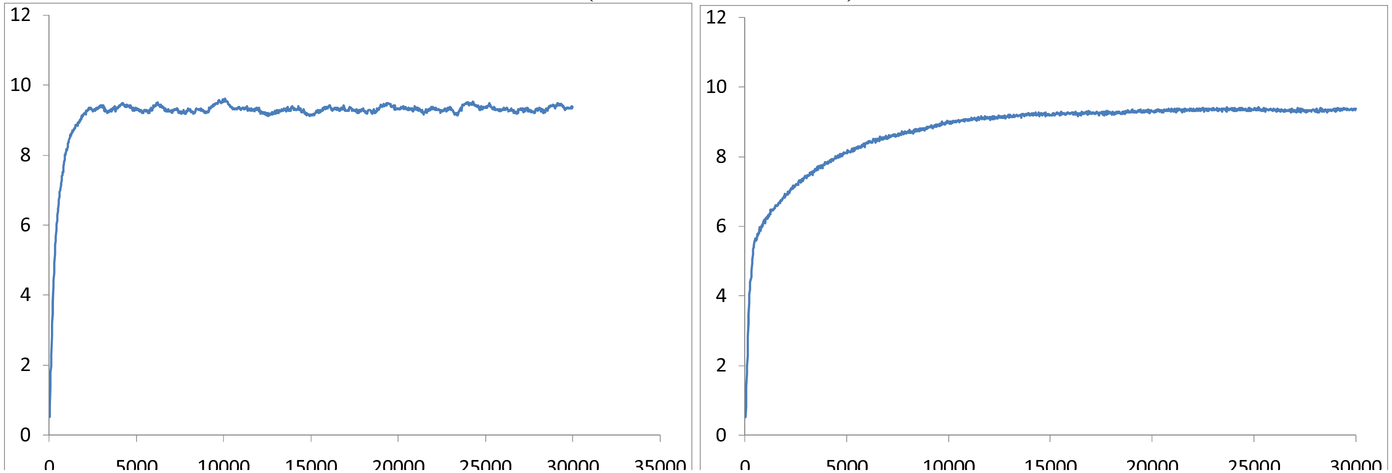
Estimation 2: Impact of N (cont'd)

N = 100 | N = 1000

G_1 (true value = 1.0)



G_2 (true value = 10.0)



Remarks

- Curse of Dimensionality of parameter space – in this case, the problem can be mitigated because we can control N.
- If the size of data set is large, and the model is well-identified, MCMC draws would concentrate in a small neighborhood of the posterior means – reduce the need to use a large N.
- There are many kernels that one could use. Norets (2009) extends IJC by using "nearest" neighbors, and consider serially correlated error terms.
- One can also weigh the past pseudo-value functions depending how recent they are.
- If size of the state space is large, memory constraints could be an issue.

Continuous State Space

- What if we allow prices to be serially correlated? (e.g., $f(p'|p)$)
- Modify the IJC algorithm above and combine it with the random grid approximation by Rust (1997).
  - In each iteration r, make a draw of prices, $p^r = (p_1^r, p_2^r)$, from $U[\underline{p}, \bar{p}]^2$, and compute the pseudo-value function at p^r.
  - Outputs of the algorithm become (in the homogeneous model)

$$H^r = \{\theta^{*l}, p^l, \tilde{V}^l(., p^l; \theta^{*l})\}_{l=r-N}^{r-1}$$

  - The expected future value at $(s, p^r; \theta^{*r})$ is then approximated as

$$\hat{E}_{p'}^r[V(s, p'; \theta^{*r})|p^r] =$$

$$\sum_{l=r-N}^{r-1} \tilde{V}^l(s, p^l; \theta^{*l}) \frac{K_h(\theta^{*r} - \theta^{*l}) f(p^l|p^r)}{\sum_{k=r-N}^{r-1} K_h(\theta^{*r} - \theta^{*l}) f(p^l|p^r)}.$$

Finite Horizon DDP models

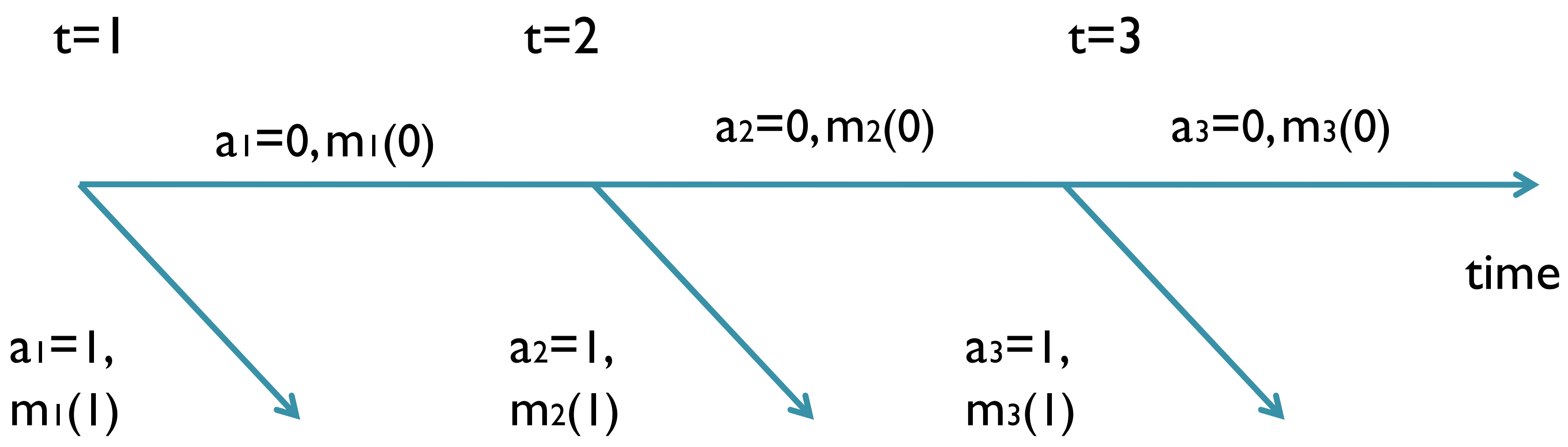
- In many situations, non-stationary, finite horizon DDP models is a more reasonable framework to use.
    - IO: Demand for products that exhibit a product life-cycle, e.g., movies (Ching and Imai (work-in-progress)), video games (Nair 2005; Ishihara 2010), prescription drugs (Ching 2010), high-tech products, etc.
    - Labor: Career Decisions (e.g., Keane and Wolpin 1995), Retirement decisions (e.g., Blau and Gilleskie 2008), Health investment decisions (e.g., Khwaja 2010), etc.
- It is possible to extend the IJC algorithm to estimate finite horizon DDP models as long as some state variables evolves stochastically. Let's use the model in Yang and Ching (2010) to illustrate.

Yang and Ching (2010)

- Yang and Ching (2010) "Dynamics of Consumer Adoption of New Technology: The Case of ATM Cards"
- Use a consumer life-cycle model to measure the adoption costs of a new technology at different ages.
- Adopting ATM cards allows consumers to reduce the transaction costs of withdrawing cash, and hence they can leave more savings at their bank account and earn more interests.
- Stylized fact: Elderly are slow adopters.
  Traditional explanation: They have higher adoption costs (incl. learning costs, psychological costs, etc.)
- But heterogeneous life-span for consumers at different ages would imply that they also face heterogeneous discounted benefits of adopting the new technology. This explanation has been largely neglected in the technology adoption literature.

Model: Basic Structure

- Model the adoption decision as an optimal stopping problem.
- $a_t \in \{0, 1\}$: the adoption decision at time t.
- $m_t(0)$: average amount of cash withdrawal for non-adopters at time t.
- $m_t(1)$: average amount of cash withdrawal for ATM adopters at time t.



Bellman's equations

- Using a Baumol-Tobin cash demand model, together with data on $m_{it}(j)$, #ATMs per 1000 population (N_{it}), household income (y_{it}) and consumption of non-durable (c_{it}), we can estimate the per period benefits of adoption, $B_t(N_{it}, y_{it}, c_{it})$.
- Data source: SHIW from Bank of Italy from 1991 to 2004, 387 households.
- $S_{it} = (N_{it}, y_{it}, c_{it})$ are continuous state variables, and evolve stochastically over time, e.g., each of them follows a AR(1) process.
- The Bellman equation of adopting an ATM card at age t is:

$$V_{1t}(s_{it};\theta) = B_t(s_{it};\theta_1) - F_t + \beta E[V_{t+1}(s_{it+1};\theta_2)|a_{it} = 1, s_{it}] + e_{i1t};$$

$$V_{0t}(s_{it};\theta) = \beta E[V_{t+1}(s_{it+1};\theta_2)|a_{it} = 0, s_{it}] + e_{i0t}.$$

Remarks

- Note that we need to integrate out s_{it+1} when calculating EV for all s_{it} and t, a computationally intensive task (we need them to create the choice probabilities).
- Moreover, we need to do these integrations for every trial parameter vector during estimation.

Extend IJC to non-stationary DDP problems (Ching and Ishihara)

- In each iteration r, make a draw of a sequence of the state variables, $\{s_t^r\}_{t=1}^T$, from $U[\underline{s}, \bar{s}]$, and compute a sequence of pseudo-value functions at $\{s_t^r\}_{t=1}^T$.

- Outputs of the algorithm become

$$H^r = \{\theta^{*l}, (s_1^l, ..., s_T^l), \tilde{V}_1^l(s_1^l; \theta^{*l}), ..., \tilde{V}_T^l(s_T^l; \theta^{*l})\}_{l=r-N}^{r-1}$$

- For each t, the expected future value at $(s_t^r; \theta^{*r})$ is then approximated as

$$\hat{E}^r[V_{t+1}(s_{t+1}; \theta^{*r}) | s_t^r] = \sum_{l=r-N}^{r-1} \tilde{V}_{t+1}^l(s_{t+1}^l; \theta^{*l}) \frac{K_h(\theta^{*r} - \theta^{*l}) f(s_{t+1}^l | s_t^r)}{\sum_{k=r-N}^{r-1} K_h(\theta^{*r} - \theta^{*l}) f(s_{t+1}^l | s_t^r)}.$$

- Substitute this into the Bellman's equations to obtain $\tilde{V}_t^r(s_t^r; \theta^{*r})$.

Conclusion

- The original IJC algorithm is applicable to infinite horizon stationary DDP models with i.i.d. unobserved errors.
- It can also be extended to estimate finite horizon non-stationary DDP models with continuous state space and random transition.
- Computational gain is more significant for models with unobserved heterogeneity.
- Osborne (2010) applies our algorithm to estimate a model with consumer learning, switching costs, and unobserved consumer heterogeneity.
- Ishihara (2010) combine IJC with the pseudo-policy function approach (Ching 2010) to estimate demand for new and used video games using aggregate level data.
- Our Matlab and C programs which implement the IJC method are available upon request.