

Bayesian Estimation of Dynamic Discrete
Choice Models

Susumu Imai
Queen's University

Neelam Jain
Northern Illinois University

Andrew Ching
Rotman School of Management
University of Toronto

Motivation:

- Applications of Discrete Choice Dynamic Programming (DDP) Models has been growing in economics and marketing, e.g., Erdem and Keane (1996), Gonul and Srinivasan (1996), Keane and Wolpin (1997), Rust and Phelan (1997), Erdem, Imai, Keane (2003), Crawford and Shum (2003), Hartmann (2005), Nevo and Hendel (2005).
- Maximum likelihood is a common way to estimate DDP models. However, it is hard to implement. In particular, it is very hard to estimate a DDP model with continuous unobserved heterogeneities.
- Bayesian Econometrics, which uses Markov Chain Monte Carlo (MCMC) method, provides a feasible way to estimate static discrete choice models with unobserved heterogeneity.

- Due to the computational burdens, there has been no works on estimating a full solution DDP model using bayesian econometrics.
- In this research, we propose a new MCMC algorithm that alleviate the burden of estimating stationary DDP.

Roadmap

- Literature
- Key features of our algorithm.
- Brief review of the dynamic programming and Metropolis-Hastings sampler.
- Our new step to obtain the expected future value.
- Monte Carlo Experiments.
- Conclude.

Literature

- Lancaster (1997) estimates a job search model.
- Geweke and Keane (2002) estimates a DDP model without solving the dynamic programming problem.
- Hotz and Miller (1993), Hotz, Miller, Sanders, Smith (1994), Aguirreagabiria and Mira (2002).
- Akerberg (2001).

Advantages of our method:

- Applicable to general stationary DDP model.
- It estimates and solves the DDP model simultaneously.
- It can incorporate unobserved heterogeneities (random coefficients).

Key features of the Bayesian DP algorithm:

- Markov Chain Monte Carlo algorithm that bases on a proxy of the value function.
- Each iteration generates $\{\theta^l, \epsilon^l, V^l(k, \epsilon^l; \theta^l)\}$. Store $\Omega^l = \{\theta^\tau, \epsilon^\tau, V^\tau(k, \epsilon^\tau; \theta^\tau)\}_{\tau=1}^{l-1}$
- Use Metropolis-Hastings within Gibbs to obtain θ^{l+1} .
- Given a draw θ^* , use $\{V^j(k, \epsilon^j; \theta^j)\}_{j=1}^l$ to obtain an estimate of $\tilde{E}_\epsilon[V(k', \epsilon'; \theta^*)]$.
- By iterating the Bellman equation only once to obtain $V^{l+1}(k, \epsilon^{l+1}; \theta^{l+1})$.

Framework: Suppose that the state space is discrete, k' evolves deterministically.

- $V(k, a, \epsilon; \theta) = R(k, a, \epsilon_a; \theta) + \beta E_\epsilon[V(k', \epsilon'; \theta)]$, where $k' = f(k, a; \theta)$; a is the action.
- $V(k, \epsilon; \theta) = \max_{a \in A} V(k, a, \epsilon; \theta)$.
- Use contraction mapping to solve for V . Start with an initial guess, V^0 . Then use Bellman equation to update V^n recursively until it converges.

$$\hat{E}_\epsilon[V^{n-1}(k', \epsilon'; \theta)] = \frac{1}{M} \sum_{m=1}^M V^{n-1}(k', \epsilon^m; \theta)$$

$$V^n(k, \epsilon; \theta) = \max_a \{R(k, a, \epsilon_a; \theta) + \beta \hat{E}_\epsilon[V^{n-1}(k', \epsilon'; \theta)]\}, \quad (1)$$

where $\{\epsilon^m\}_{m=1}^M$ are drawn from $F_\epsilon(\epsilon)$.

- The observed data is $Y_T = \{k_\tau^d, a_\tau^d, R_\tau^d\}_{\tau=1}^T$

- The likelihood, $L(Y_T|\theta)$, depends on θ and $E[V(k', \cdot; \theta)]$

- The posterior distribution function:

$$P(\theta|Y_T) \propto \pi(\theta)L(Y_T|\theta).$$

- Random-walk Metropolis-Hastings algorithm. Draw θ^{l+1*} from $N(\theta^l, W)$.

- Accept, i.e., $\theta^{l+1} = \theta^{l+1*}$ with probability

$$\alpha(\theta^l, \theta^{l+1*}) = \min\left\{\frac{\pi(\theta^{l+1*})L(Y_T|\theta^{l+1*})}{\pi(\theta^l)L(Y_T|\theta^l)}, 1\right\}$$

- Otherwise, set $\theta^{l+1} = \theta^l$.

The Bayesian DP algorithm replaces the standard dynamic programming step (1) with the following step:

- Let $\Omega^l = \{\epsilon^s, \theta^{s*}, V^s(\cdot, \epsilon^s; \theta^{s*})\}_{s=l-N(l)}^{l-1}$ be the history of simulated draws up to the previous iteration $l - 1$.

- For each k, a ,

$$\begin{aligned} \tilde{E}_\epsilon[V(k', \epsilon'; \theta^{l*})|\Omega^l] \\ = \sum_{n=1}^{N(l)} V^{l-n}(k', \epsilon^{l-n}, \theta^{l-n*}) \frac{K_h(\theta^{l*} - \theta^{l-n*})}{\sum_{k=1}^{N(l)} K_h(\theta^{l*} - \theta^{l-k*})} \end{aligned}$$

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

- Simulate one ϵ^l ,

$$\begin{aligned} V^l(k, \epsilon^l; \theta^{l*}) = \\ \max_{a \in A} \{R(k, a, \epsilon_a^l; \theta^{l*}) + \beta \tilde{E}_\epsilon[V(k', \epsilon'; \theta^{l*})|\Omega^l]\} \end{aligned}$$

- $N(l)$ increases with l .

Highlights:

- Computational burden of our algorithm of each iteration is comparable to that of estimating a standard static discrete choice model.
- We make use of all the past iterated $V^l(., \epsilon^l; \theta^l)$ to obtain $E[V(., \epsilon; \theta)]$. But the conventional method simply throws away this information.

Theoretical Analysis: When S is discrete.

- 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 probability.

Continuous observed and unobserved Heterogeneity (random effects)

- Suppose that there are N firms. Each firm has different observed and unobserved heterogeneities.
- In Simulated ML, one needs to integrate out the unobserved heterogeneities when forming the likelihood.
- Suppose we make N_s draws for coefficients with random effects. Then we need to solve $N * N_s$ dynamic programming problems in order to evaluate the likelihood once.
- Very high computational burden!
- Bayesian DP algorithm: The computational burden for each iteration is similar to a standard static random effects model.

- $\theta = (\theta_{1i}, \theta_2)$, where θ_{1i} is the parameters with random effects for $i = 1, \dots, N$.
- Step 1: Use Metropolis-Hastings algorithm to draw the random effects parameter $\theta_{1i}^l, i = 1, \dots, N$. Draw θ_{1i}^{l+1*} based on a random walk.
- Accept, i.e., $\theta_{1i}^{l+1} = \theta_{1i}^{l+1*}$ with probability

$$\min\left\{\frac{\pi(\theta_{1i}^{l+1*})L(Y_{iT}|\theta_{1i}^{l+1*}, \theta_2^l)}{\pi(\theta_{1i}^l)L(Y_{iT}|\theta_{1i}^l, \theta_2^l)}, 1\right\}$$

Note that only the likelihood of firm i is used here.

- Otherwise, $\theta_{1i}^{l+1} = \theta_{1i}^l$.

- Step 2: Draw θ_2^{l+1*} based on a random walk. Accept, i.e., set $\theta_2^{l+1} = \theta_2^{l+1*}$ with probability

$$\min\left\{\frac{\pi(\theta_2^{l+1*})L(Y_T|\theta_1^{l+1}, \theta_2^{l+1*})}{\pi(\theta_2^l)L(Y_T|\theta_1^{l+1}, \theta_2^l)}, 1\right\}$$

where $\theta_1 = (\theta_{1i})_{i=1}^N$.

- Otherwise, set $\theta_2^{l+1} = \theta_2^l$.
- Expected value function approximation is similar.
- But the computational burden will be significantly higher because we need to evaluate the kernel N more times in each iteration.

How to combine our Bayesian DP algorithm with Rust's (1997) random grid method in continuous state space problem

- Rust (1997) proposed to use

$$\bar{E}_{s'}[V(s')] = \sum_{k=1}^N V(\tilde{s}_k) \frac{p(\tilde{s}_k|s, a)}{\sum_{j=1}^N p(\tilde{s}_j|s, a)}$$

where $\{\tilde{s}_1, \dots, \tilde{s}_N\}$ are *i.i.d.* draws w.r.t. Lebesgue measure from S . Need to solve the Bellman equation for each grid point when doing ML.

- In Bayesian DP algorithm, we simulate K new grid points in each iteration. WOLG, assume $K = 1$.
- $\Omega^l = \{s^\tau, \epsilon^\tau, \theta^{\tau*}, V^\tau(s^\tau, \epsilon^\tau; \theta^{\tau*})\}_{\tau=l-N(l)}^{l-1}$, where s^τ 's are the random grids obtained during the estimation process.

- Simulate one ϵ^l , and one s^l ,

$$V^l(s^l, \epsilon^l; \theta^{l*}) = \max_{a \in A} \{R(s^l, a, \epsilon_a^l; \theta^{l*}) + \beta \tilde{E}_{s', \epsilon} [V(s', \epsilon'; \theta^{l*}) | \Omega^l]\}$$

- $\tilde{E}_{s', \epsilon'} [V(s', \epsilon', \theta^{l*}) | \Omega^l]$ is a weighted average of $\{V^\tau(s^\tau, \epsilon^\tau; \theta^{\tau*})\}_{\tau=l-N(l)}^{l-1}$. For each τ , the weight is

$$\frac{p(s^\tau | s^l, a, \theta^{l*}) K_h(\theta^{l*} - \theta^{\tau*})}{\sum_{k=1}^N p(\tilde{s}^{\tau-k} | s^l, a, \theta^{l*}) K_h(\theta^{l*} - \theta^{l-k*})}$$

where $p(\cdot | s, a, \theta)$ is the transition density for the state variables.

- Number of effective grid points is $N(l)K$.
- One can also extend our algorithm to the case of continuous state space with deterministic transition function.

An example: A simple dynamic discrete choice model of entry and exit (decision to export), in a perfectly competitive environment (Roberts and Tybout (1997)).

- Two types of firms: incumbent (I) and potential entrant (O).

- If the incumbent firm chooses to stay, its per period return is,

$$R_{I,IN}(k_t, \epsilon_t; \theta) = \alpha k_t + \epsilon_{1t},$$

where k_t is the capital of the firm.

- If it chooses to exit, its per period return is,

$$R_{I,OUT}(\epsilon_t; \theta) = \delta_x + \epsilon_{2t},$$

where δ_x is the exit value to the firm.

- If the potential entrant chooses to enter, its per period return is,

$$R_{O,IN}(\epsilon_{1t}; \theta) = -\delta_e + \epsilon_{1t},$$

- If it stays out, its per period return is,

$$R_{O,OUT}(\epsilon_{2t}; \theta) = \epsilon_{2t},$$

- Assume the random component of the current period returns to be *i.i.d.* normal,

$$\epsilon_{lt} \sim N(0, \sigma_l)$$

- If $k_t > 0$,

$$\ln k_{t+1} = b_1 + b_2 \ln k_t + u_{t+1}.$$

If $k_t = 0$ (for potential entrants),

$$\ln k_{t+1} = b_e + u_{t+1}.$$

-

$$u_t \sim N(0, \sigma_u),$$

- The value function of the incumbent:

$$V_I(k, \epsilon; \theta) = \max\{V_{I,IN}(k, \epsilon; \theta), V_{I,OUT}(k, \epsilon; \theta)\}$$

where

$$V_{I,OUT}(k, \epsilon; \theta) = R_{I,OUT}(\epsilon_2; \theta) + \beta E[V_I(\epsilon'; \theta)],$$

$$V_{I,IN}(k, \epsilon; \theta) = R_{I,IN}(k, \epsilon_1; \theta) + \beta E[V_I(k', \epsilon'; \theta)]$$

- The value function of the potential entrant:

$$V_O(\epsilon; \theta) = \max\{V_{O,IN}(\epsilon; \theta), V_{O,OUT}(\epsilon; \theta)\}$$

where

$$V_{O,IN}(\epsilon; \theta) = R_{O,IN}(\epsilon_1; \theta) + \beta E[V_I(k', \epsilon'; \theta)]$$

$$V_{O,OUT}(k, \epsilon; \theta) = R_{O,OUT}(\epsilon_2; \theta) + \beta E[V_O(\epsilon'; \theta)]$$

- Data: $\{k_{i,t}^d, R_{i,t}^d, a_{i,t}^d\}$, for $i = 1, \dots, N_d$, $t = 1, \dots, T$. i indexes firms, t indexes time.
- Observe $R_{i,t}^d$ only if firms stay in.

- Use Gaussian kernel:

$$K_h(\theta^s - \theta^j) = (2\pi)^{-L/2} \prod_{l=1}^L h_l^{-1} \exp\left[-\frac{1}{2} \left(\frac{\theta_l^s - \theta_l^j}{h_l}\right)^2\right],$$

and use our new step to obtain,

$$\begin{aligned} & \tilde{E}_\epsilon[V_\Gamma(k', \epsilon'; \theta^s) | \Omega^s] \\ &= \sum_{n=1}^{N(s)} V_\Gamma^{s-n}(k', \epsilon^{s-n}, \theta^{s-n}) \frac{K_h(\theta^s - \theta^{s-n})}{\sum_{k=1}^{N(s)} K_h(\theta^l - \theta^{s-k})} \end{aligned}$$

Now substitute $\tilde{E}_\epsilon(\cdot, \cdot; \theta^s)$ into the bellman equations above. Then evaluate the value functions at ϵ^s and θ^s to obtain $V_\Gamma^s(\cdot, \epsilon^s; \theta^s)$, for $\Gamma \in \{I, O\}$.

- Note that k' is stochastic. We integrate out the value function over the capital shock using Rust's(1997) random grid integration method.
- Now we have $(V_\Gamma^s(\cdot, \epsilon^s; \theta^s), \epsilon^s, \theta^s)$.

Example: A Dynamic Model of Entry and Exit

- One firm – either an incumbent (I) or an entrant (O)
- Choices – Stay or Enter (IN) and exit or stay out (Out)

Per period return function of the incumbent

$$R_{I,IN}(K_t, \varepsilon_t, \theta) = \alpha K_t + \varepsilon_{1t}$$

$$R_{I,OUT}(K_t, \varepsilon_t, \theta) = \varepsilon_{2t}$$

$$\varepsilon_{lt} \sim N(0, \sigma_l^2): \text{i.i.d normal}$$

Per period return function of potential entrant

$$R_{O,IN}(K_t, \varepsilon_t, \theta) = -\delta_E + \varepsilon_{1t}$$

$$R_{O,OUT}(K_t, \varepsilon_t, \theta) = \varepsilon_{2t}$$

Evolution of capital stock

- If the incumbent stays:

$$\ln K_{t+1} = b_1 + b_2 \ln K_t + u_{t+1}$$

$$u_t \sim N(0, \sigma_u)$$

- If the incumbent exits, or the entrant stays out, its capital stock is 0.
- If the firm enters, its initial capital stock is.

$$\ln K_{t+1} = b_e + u_{t+1}$$

$$u_t \sim N(0, \sigma_u)$$

Value of being in

$$\begin{aligned} & .V_I(K_t, \varepsilon_t, \theta) \\ & = \text{Max} \{ R_{I,IN}(K_t, \varepsilon_{1t}, \theta) + \beta E_{t+1} V_I(K_{t+1}, \varepsilon_{t+1}, \theta | K_t), \\ & \quad R_{I,OUT}(K_t, \varepsilon_{1t}, \theta) + \beta E_{t+1} V_O(K_{t+1}, \varepsilon_{t+1}, \theta | K_t) \} \end{aligned}$$

Value of being out

$$\begin{aligned} & .V_O(K_t, \varepsilon_t, \theta) \\ & = \text{Max} \{ R_{O,IN}(K_t, \varepsilon_{1t}, \theta) + \beta E_{t+1} V_I(K_{t+1}, \varepsilon_{t+1}, \theta | K_t), \\ & \quad R_{O,OUT}(K_t, \varepsilon_{1t}, \theta) + \beta E_{t+1} V_O(K_{t+1}, \varepsilon_{t+1}, \theta | K_t) \} \end{aligned}$$

Estimating With Maximum Likelihood Method

- *Steps for Solving the DP Problem*

Start with initial guess of the expected value functions: M Rust random Grid points.

$$E_{\varepsilon} V(K_m, \varepsilon, \theta) = 0, \quad m = 0, 1, 2, \dots, M$$

Step 1: Draw ε^l .

Derive $.V_I(K_m, \varepsilon^l, \theta)$, $.V_O(K_m, \varepsilon^l, \theta)$.

Step 2: Repeat step 1 for $l = 1, 2, \dots, L$ times to derive the expected value function.

$$E_{\varepsilon} V_I(K_m, \varepsilon, \theta) = \frac{1}{L} \sum_{l=1}^L V_I(K_m, \varepsilon^l, \theta)$$

Step 3: Repeat Step 1 and Step 2 to derive

$EV_I(K_m, \varepsilon, \theta)$ for each capital grid point, $m = 0, 1, 2, \dots, M$.

- Repeat the iteration until convergence. That is,

$$\max_l |E_\varepsilon V_s(K_l, \varepsilon, \theta) - E_\varepsilon V_{s+1}(K_l, \varepsilon, \theta)| < \delta$$

for some small $\delta > 0$

- *Steps for Estimating with Maximum Likelihood*

Step 1: Given $\theta^{(s)}$, solve the DP problem and derive the expected value function $E_{\varepsilon}V(K, \varepsilon, \theta^{(s)})$

Step 2: Given $\theta^{(s)}$ and $E_{\varepsilon}V(K, \varepsilon, \theta^{(s)})$, derive the log likelihood based on the probability of entry and exit.

Step 3: Get another parameter $\theta^{(s+1)}$, repeat Step 1 and 2.

Repeat Steps 1, 2, 3 until you find parameters that maximize the likelihood.

Computational Burden

Burden 1: For each possible state space point (capital stock $K_m, m = 0, 1, 2, \dots, M$, and the simulated errors $\varepsilon^1, \dots, \varepsilon^L$), we solve for the value function, to derive the expected value function $EV()$.

Burden 2: The above calculation of the expected value function has to be repeated many times (30-50 times, or even more) until the DP algorithm converges. Only then we can form the log likelihood.

Burden 3: During the maximum likelihood routine, whenever we have a new parameter value, we have to do all those steps over again to derive the new likelihood.

Bayesian DP Estimation Algorithm

Prior:

$$\text{Exit value: } \textit{exit} \sim N(\underline{\delta}_x, A_x^{-1})$$

$$\text{Entry cost: } \textit{enter} \sim N(\underline{\delta}_E, A_E^{-1})$$

Precision of shocks

$$s_1^2 h_{\varepsilon_1} \sim \chi^2(\nu_1), s_\eta^2 h_\eta \sim \chi^2(\nu_\eta), \eta = \varepsilon_1 - \varepsilon_2$$

$$s_u^2 h_u \sim \chi^2(\nu_u)$$

$$\text{Profit function coefficient: } \alpha \sim N(\underline{\alpha}, A_\alpha^{-1})$$

$$\text{Capital coefficients: } b_1 \sim N(\underline{b}_1, A_{b_1}^{-1})$$

$$b_2 \sim N(\underline{b}_2, A_{b_2}^{-1})$$

$$b_e \sim N(\underline{b}_e, A_{b_e}^{-1})$$

Calculate the expected value function.

Here, simple Rust (1997) Random grids.

$$E^{(s)} \left[V_{\Gamma} \left(K'(K, u, \theta^{(s)}), \varepsilon, \theta^{(s)} \right) \mid \Omega^{(s)} \right]$$

$$= \sum_{m=1}^M E_{\varepsilon}^{(s)} \left[V_{\Gamma} \left(K_m, \varepsilon, \theta^{(s)} \right) \mid \Omega^{(s)} \right] f \left(K_m \mid K, \theta^{(s)} \right)$$

$$E_{\varepsilon}^{(s)} V_{\Gamma} \left(K_m, \varepsilon, \theta^{(s)} \right)$$

$$= \sum_{j=s-1-N(s)}^{s-1} \left[\frac{1}{M} \sum_{l=1}^M V_{\Gamma}^{(j)} \left(K_m, \varepsilon_l^{(j)}, \theta^{(j)}, \Omega^{(j)} \right) \right] W_j^{(s)}$$

$$W_j^{(s)} = \frac{K_h \left(\theta^{(s)} - \theta^{(j)} \right)}{\sum_{j=s-1-N(s)}^{s-1} K_h \left(\theta^{(s)} - \theta^{(j)} \right)}$$

M is set to be 20. $N(s)$ increases to 2000 at iteration 10,000.

At Iteration s , given $\theta^{(s)}$ Calculate

- Value of being in at the beginning of period t

$$= V_I^{(s)}(K_m, \varepsilon^{(s)}, \theta^{(s)}) \quad m = 1, \dots, M$$

- Value of being out at the beginning of period t

$$= V_O^{(s)}(K_m, \varepsilon^{(s)}, \theta^{(s)}) \quad m = 1, \dots, M$$

Notice parameters change over iterations.

Deterministic values of staying in and out

If the firm is an incumbent:

$$\bar{V}_{I,IN}(K_m, \theta^{(s)}) = \alpha^{(s)} K_m + \beta EV_I^{(s)}(K', \varepsilon, \theta^{(s)})$$

$$\bar{V}_{I,OUT}(K_t, \theta^{(s)}) = \delta_x^{(s)} + \beta EV_O^{(s)}(0, \varepsilon, \theta^{(s)})$$

If the firm is a potential entrant:

$$\bar{V}_{O,IN}(K_t, \theta^{(s)}) = -\delta_E^{(s)} + \beta EV_I^{(s)}(K', \varepsilon, \theta^{(s)})$$

$$\bar{V}_{O,OUT}(K_t, \theta^{(s)}) = \beta EV_O^{(s)}(0, \varepsilon, \theta^{(s)})$$

MCMC Hastings-Metropolis algorithm

As discussed above, we adopt the random walk H-M algorithm with the errors being normal, i.e. for each element of the parameter vector:

$$\theta_l^{(*)} = \theta_l^{(t)} + \varepsilon_l^{(t)}, \varepsilon_l^{(t)} \sim N(0, \sigma_l), \sigma_l = 0.004$$

ESTIMATION RESULTS

We simulate samples based on the solution of the dynamic model.

We then estimate back the parameters of the model.

True parameters of the simulation:

$$\delta_E^* = 0.4, \sigma_1^* = 0.3, \sigma_2^* = 0.3$$

$$\alpha^* = 0.1, b_1^* = 0.0, b_2^* = 0.4, b_e = 0.5, \sigma_u^* = 0.4,$$

$$\beta^* = 0.98$$

We estimate 9 parameters:

$$\delta_E, \sigma_1, \sigma_2, \alpha, b_1, b_2, b_e, \sigma_u$$

In Bayesian DP estimation, initial guess of the Expected value function is set to be zero. That is,

$$E_\varepsilon V_\Gamma^{(0)}(K_m, \varepsilon, \theta^{(0)}) = 0 \text{ for } m = 1, \dots, M$$

Priors:

- **Entry-exit parameters:** uninformative prior

$$\underline{\delta}_x = 0.4, A_x = 0.0 \text{ (inverse of std. error)}$$

$$\underline{\delta}_E = 0.4, A_E = 0.0$$

- **Precision of shocks**

$$\underline{s_{\varepsilon_1}^2} h_{\varepsilon_1} \sim \chi^2(v_{\varepsilon_1})$$

$$\left(\underline{s_{\varepsilon_1}^2}\right)^{-1/2} = 0.4, v_{\varepsilon_1} = 10$$

$$\underline{s_{\eta}^2} h_{\eta} \sim \chi^2(v_{\eta}), \left(\underline{s_{\eta}^2}\right)^{-1} = 0.32, v_{\eta} = 10$$

$$\underline{s_u^2} h_u \sim \chi^2(v_u) \quad \left(\underline{s_u^2}\right)^{-1/2} = 0.4, v_u = 10$$

- **Profit function coefficient:**

α : noninformative prior

- **Capital coefficients:**

b_1, b_2, b_e : noninformative prior.

Experiment 1:

- Starting values of the parameters are set to be the true value.
- 10,000 iterations

Table 1: Posterior Means and Standard Errors
(Rust random grids with 200 grid points)
(Calculated using 5001-10000th iteration)

Parameter	Estimate	Estimate	True
δ_E (entry cost)	0.3786 (0.0275)	0.3861 (0.0179)	0.4
α (capital coef.)	0.1002 (0.0061)	0.0995 (0.0040)	0.1
σ_1 (profit shock)	0.2983 (0.0072)	0.2983 (0.0048)	0.3
σ_2 (choice shock)	0.3076 (0.0194)	0.3039 (0.0156)	0.3
b_1 (capital transit.)	0.0018 (0.0121)	0.0011 (0.0076)	0.0
b_2 (capital transit.)	0.4012 (0.0246)	0.4027 (0.0153)	0.4
b_e (capital transit.)	0.4737 (0.0246)	0.4903 (0.0156)	0.5
σ_u (capital trans.)	0.4012 (0.0077)	0.3976 (0.0048)	0.4
Sample size	2,000	5,000	2800 MHz
CPU time	6 min. 29 sec.	10 min. 56 sec.	1.03GBytes

Parameter	Estimate	True
δ_E (entry cost)	0.3991 (0.0126)	0.4
α (capital coef.)	0.1004 (0.0028)	0.1
σ_1 (profit shock)	0.2994 (0.0034)	0.3
σ_2 (choice shock)	0.3056 (0.0108)	0.3
b_1 (capital transit.)	-0.0006 (0.0055)	0.0
b_2 (capital transit.)	0.4033 (0.0110)	0.4
b_e (capital transit.)	0.5033 (0.0110)	0.5
σ_u (capital trans.)	0.3987 (0.0034)	0.4
Sample size	10,000	2800 MHz
CPU time	19 min. 32 sec.	1.03GBytes

The results are the average over 10 simulation/estimation trials

Main Points

- Posterior mean is very close to the true value.
- Both bias and std. errors decrease with sample size.
- The computational speed is on the order of the estimation of static models.

- **Experiment 2:** Same as Experiment 1 except initial value of the parameter estimate θ^0 is set to be half of the true value.

Table 2: Posterior Means and Standard Errors

Parameter	Estimate	True value
δ_E	0.3955 (0.0090)	0.4
α	0.1001 (0.0027)	0.1
σ_1	0.3001 (0.0035)	0.3
σ_2	0.3004 (0.0191)	0.3
b_1	-0.0005 (0.0103)	0.0
b_2	0.4031 (0.0062)	0.4
b_e	0.5032 (0.0060)	0.5
σ_u	0.3987 (0.0022)	0.4
Sample size	10,000	2800 MHz
CPU time	19 min. 08 sec.	1.03GBytes

The results are the average over 10 simulation/estimation trials

The results and the Gibbs sampler output are very similar to those of experiment 1.

Bayesian DP algorithm is shown to be robust to the starting values of the parameters.

Experiment 3: Same as Experiment 1, but we introduce random effects heterogeneity into the profit coefficient α , and observed heterogeneity.

$$\alpha_i \sim N(\mu, \tau^2)$$

$$\mu \sim N(\underline{\mu}, h_\alpha^{-1})$$

$$\underline{s}_\tau^2 \tau^{-2} \sim \chi^2(v_\tau)$$

$$\hat{b}_{1i} = b_1 X$$

: Continuous observed covariate affecting capital transition.

Expected value calculation:

$$E_{\varepsilon}^{(s)} V_{\Gamma} \left(K_m, \varepsilon, \theta_{-\alpha}^{(s)}, \alpha_i^{(s)} \right) =$$

$$\sum_{j=\max\{s-1-N(s),1\}}^{s-1} \left[\frac{1}{M} \sum_{l=1}^M V_{\Gamma}^{(j)} \left(K_m, \varepsilon_l^{(j)}, \theta_{-\alpha}^{(j)}, \alpha_i^{(j)}, \Omega^{(j)} \right) \right] W_j^{(s)}$$

$$W_j^{(s)} = \frac{K_h \left(\theta_{-\alpha}^{(s)} - \theta_{-\alpha}^{(j)} \right) K_h \left(\alpha_i^{(s)} - \alpha_i^{(j)} \right)}{\sum_{j=\max\{s-1-N(s),1\}}^{s-1} K_h \left(\theta_{-\alpha}^{(s)} - \theta_{-\alpha}^{(j)} \right) K_h \left(\alpha_i^{(s)} - \alpha_i^{(j)} \right)}$$

This is computationally burdensome because the nonparametric kernel smoothing has to be done for each firm.

Sample: 100 firms for 100 periods (10,000)

Initial conditions problem is dealt with by simulating the model for 20 initial periods until steady state is achieved for each firm.

Parameter	Random Effects	True
δ_E (entry cost)	0.3964 (0.0163)	0.4
μ (capital coef.)	0.1976 (0.0104)	0.2
τ (capital coef.)	0.1010 (0.0075)	0.1
σ_1 (profit shock)	0.3018 (0.0029)	0.3
σ_2 (choice shock)	0.3019 (0.0130)	0.3
b_1 (capital transit.)	0.0998 (0.0049)	0.1
b_2 (capital transit.)	0.3973 (0.0093)	0.4
b_e	0.4969 (0.0129)	0.5
σ_u (capital trans.)	0.4004 (0.0031)	0.4
Sample size	100 × 100	2800 MHz
CPU time	4hr. 2min. 1sec.	1.03GBytes

Bayesian DP Estimation Results

- Posterior mean is very close to the true value.

Simulated ML with 100 simulation size:

- Step 1: For each firm, solve the DP problem and evaluate the likelihood with different parameters $\alpha_i^m \sim N(\mu, \tau^2)$ for $m=1, \dots, 100$
- Step 2: Take the average of the likelihood.
 - Step 3: Repeat Step 1 and Step 2 for each of the 100 firms.

For each simulated likelihood calculation, you need to solve the DP problem 10,000 times!

Example in an earlier draft: A single ML iteration (evaluation of the likelihood and the numerical derivative w.r.t. parameters: 11 likelihood evaluations): 6 hrs. 20 min.

Bayesian DP algorithm : 6,744 iterations in 6 hrs. 20 min.

would have already converged.

Full Solution based Bayesian estimation.

With 1000 simulation used in integration of shocks per state: It would take more than a week.

With 100 simulation used.

Parameter	Random Effects	True
δ_E (entry cost)	0.4021 (0.0187)	0.4
μ (capital coef.)	0.2005 (0.0108)	0.2
τ (capital coef.)	0.1043 (0.0075)	0.1
σ_1 (profit shock)	0.3005 (0.0031)	0.3
σ_2 (choice shock)	0.3099 (0.0156)	0.3
b_1 (capital transit.)	0.0997 (0.0050)	0.1
b_2 (capital transit.)	0.3993 (0.0098)	0.4
b_e	0.4899 (0.0141)	0.5
σ_u (capital trans.)	0.4009 (0.0032)	0.4
Sample size	100 \times 100	2800 MHz
CPU time	33 hr. 32min. 9sec.	1.03GBytes

Continuous State Space with Deterministic Transition

Rust random grid method does not work because the capital transition is deterministic.

We use kernel methods to derive the value function over a continuum of state space.

By increasing the estimation iteration, we can make the number of grid points arbitrarily large by increasing the number of iterations, even though the number of grid points per iteration remains to be 10.

Modified model:

If the firm stays: $K_{t+1} = K_t$

If the firm enters: $\ln(K_{t+1}) = b_1 + u_{t+1}$,

$u_{t+1} \sim N(0, \sigma_u)$

Formula for the expected value function:

$$\begin{aligned}
 & E_{\varepsilon}^{(s)} V_{IN} \left(K_m, \varepsilon, \theta^{(s)} \right) \\
 &= \sum_{j=s-1-N(s)}^{s-1} \sum_{m=1}^{M_K} \left[\frac{1}{M} \sum_{l=1}^M V_{IN}^{(j)} \left(K_m, \varepsilon_l^{(j)}, \theta^{(j)}, \Omega^{(j)} \right) \right] W_{jm}^{(s)} \\
 & W_{jm}^{(s)} = \frac{K_{h_K} \left(K^{(s)} - K_m^{(j)} \right) K_{h_{\theta}} \left(\theta^{(s)} - \theta^{(j)} \right)}{\sum_{j=\max\{s-1-N(s), 1\}}^{s-1} \sum_{m=1}^{M_K} K_{h_K} K_{h_{\theta}}}
 \end{aligned}$$

Kernel over parameters and state variables.

Parameter	Infinite Random Grids	True
δ_E (entry cost)	0.1891 (0.0123)	0.2
a (capital coef.)	0.2044 (0.0048)	0.2
σ_1 (profit shock)	0.3956 (0.0051)	0.4
σ_2 (choice shock)	0.3993 (0.0134)	0.4
b_1 (capital transit.)	0.1996 (0.0047)	0.2
σ_u (capital trans.)	0.2017 (0.0030)	0.2
Sample size	10,000	2800 MHz
CPU time	48 min. 50 sec.	1.03GBytes

The results are the average over 10 simulation/estimation trials

10 grid points per iteration.

Original DP problem had 2000 grid points for capital.

Conclusions and Future Research:

- We develop a Bayesian Estimation method for Dynamic Discrete Choice models where we solve for the Dynamic Programming problem and estimate the parameters at the same time.
- Since we solve the Dynamic problem and estimate parameters at the same time, the computational burden of the exercise is essentially of the same order of magnitude as estimating static models.

- **Application 1** (Models with random effects and other heterogeneities)

Our estimation framework allows for much richer heterogeneity (parameters varying with firm size, location, etc for I.O., see Osborne (2006)).

- **Application 2** (Continuous Choice Models)

Imai, Jain and Katayama are now developing similar full solution based Bayesian estimation algorithm for continuous choice models (Consumption, labor supply choice, etc.)

- **Application 3** (Overcoming Curse of Dimensionality in the Parameter Space)

Develop similar full solution based Bayesian estimation algorithm where only a small number of parameters are iterated at one Bayesian DP algorithm. Ferrall, Imai, Jain

- **Application 4** (Equilibrium models)
Develop an estimation method that solved the DP model, computes the equilibrium, and estimates parameters all simultaneously and not subsequently. During each solution/estimation step, each of the above 3 algorithms are done only once.

The Point of the Paper

Suppose you have a very complicated model that is computationally hard to solve (equilibrium models, dynamic programming models, etc. that involve fixed point algorithms, etc.)

Suppose you can do the following.

- 1) Decompose the solution algorithm into small steps that are repeated until convergence.
- 2) Combine the single solution step with the single MCMC estimation step into a single iteration step.

Then, the estimation of the model becomes computationally feasible.

Why does it become feasible?

- In standard estimation algorithm, (for example, MCMC), when you iterate 10,000 times, people **throw away all the computations** until 9999 iterations except for the parameters (or the likelihood, Jacobian or Hessian of iteration 9999)
- We **utilize all the computations** obtained until 9999 iterations for model solutions and other purposes: Those can be reused at very small additional computational cost.