Econometrica, Vol. 77, No. 6 (November, 2009), 1865–1899

BAYESIAN ESTIMATION OF DYNAMIC DISCRETE CHOICE MODELS

BY SUSUMU IMAI, NEELAM JAIN, AND ANDREW CHING[1]

We propose a new methodology for structural estimation of infinite horizon dynamic discrete choice models. We combine the dynamic programming (DP) solution algorithm with the Bayesian Markov chain Monte Carlo algorithm into a single algorithm that solves the DP problem and estimates the parameters simultaneously. As a result, the computational burden of estimating a dynamic model becomes comparable to that of a static model. Another feature of our algorithm is that even though the number of grid points on the state variable is small per solution-estimation iteration, the number of effective grid points increases with the number of estimation iterations. This is how we help ease the "curse of dimensionality." We simulate and estimate several versions of a simple model of entry and exit to illustrate our methodology. We also prove that under standard conditions, the parameters converge in probability to the true posterior distribution, regardless of the starting values.

KEYWORDS: Bayesian estimation, dynamic programming, discrete choice models, Markov chain Monte Carlo.

1. INTRODUCTION

STRUCTURAL ESTIMATION OF DYNAMIC DISCRETE CHOICE (DDC) models has become increasingly popular in empirical economics. Examples include Keane and Wolpin (1997) on labor economics, Erdem and Keane (1996) on marketing, Imai and Krishna (2004) on crime, and Rust (1987) on empirical industrial organization. Structural estimation of DDC models is appealing because it captures the dynamic forward-looking behavior of individuals. This is important in understanding agents' behavior in various settings. For example, in the labor market, individuals carefully consider future prospects when they decide whether to change jobs. Moreover, since structural estimation allows us to obtain estimates of parameters that have economic interpretations, based on these interpretations and the solution of the model, we can assess the effect of fundamental changes in policy regimes by simply changing the estimated value of "policy" parameters and simulating the model. However, one major obstacle in adopting the structural estimation method has been its computational burden, which is mainly due to the following two reasons.

First, the likelihood or the moment conditions are based on the explicit solution of a dynamic programming (DP) model. For instance, solving an infinite

[1]We are very grateful to the editor and the anonymous referees for insightful comments. Thanks also go to Chris Ferrall, Chris Flinn, Wesley Hartmann, Mike Keane, Justin McCrary, Andriy Norets, Matthew Osborne, Peter Rossi, John Rust, and seminar participants at the UIUC, NYU, Ohio State University, University of Kansas, University of Michigan, University of Minnesota, SBIES, 2006 Quantitative Marketing and Economics Conference, and 2005 Econometrics Society World Congress for helpful comments on the earlier draft of the paper. We also thank SSHRC and FQRSC for financial support. All remaining errors are our own.

 DOI: 10.3982/ECTA5658

horizon DP problem requires us to obtain the fixed point of a Bellman operator for each possible point in the state space. Second, the possible number of points in the state space increases exponentially with the dimensionality of the state space. This is commonly referred to as the curse of dimensionality, which makes the estimation of DDC models infeasible even in a relatively simple setting.

In this paper, we propose an estimator that helps overcome the two computational difficulties of structural estimation of infinite horizon DP models. Our estimator is based on the Bayesian Markov chain Monte Carlo (MCMC) estimation algorithm, where we simulate the posterior distribution by repeatedly drawing parameters from a pseudo-Markov chain until convergence. In contrast to the conventional MCMC estimation approach, we combine the Bellman equation step and the MCMC algorithm step into a single hybrid solution-estimation step, which we iterate until convergence. The key innovation in our algorithm is that, for a given state space point, we only need to conduct a single iteration of the Bellman operator during each estimation step. Since evaluating a Bellman operator once is as computationally demanding as computing a static model, the computational burden of estimating a DP model is in order of magnitude comparable to that of estimating a static model.[2] This is in contrast to conventional estimation methods that "estimate" the model only after solving the DP problem.

Our estimation method is related to the algorithm advocated by Aguirregabiria and Mira (2002) and others, which is an extension of the method developed by Hotz and Miller (1993) and Hotz, Miller, Sanders, and Smith (1994).[3] However, their estimation algorithms, which are not based on the full solution of the model, have difficulties dealing with unobserved heterogeneity. This is because they essentially recover the value function from the observed choices of individuals at each point of the state space by conditioning on observed state variables. In contrast, our estimation algorithm is based on the full solution of the DP problem and, therefore, it can accommodate a rich specification of both observed and unobserved heterogeneity.[4]

[2]Ferrall (2005) also considered an optimal mix of model solution and estimation algorithms. Arcidiacono and Jones (2003) adopted the expectation–maximization (EM) algorithm to estimate different parts of a dynamic model with latent types sequentially rather than jointly. Using a Monte Carlo experiment, they showed that their method could potentially result in significant computational gain compared with the full information maximum likelihood.

[3]See also Aguirregabiria and Mira (2007) and Arcidiacono and Miller (2009) for extensions of the work of Hotz et al. to estimate models with dynamic games and finite mixture models.

[4]In contrast to Ackerberg (2004), where the entire DP problem needs to be solved for each parameter simulation, in our algorithm, the Bellman operator needs to be evaluated only once for each parameter value. Furthermore, there is an additional computational gain because our pseudo-MCMC algorithm guarantees that, except for the initial burn-in simulations, most of the parameter draws are from a distribution close to the true posterior distribution. In Ackerberg's case, the initial parameter simulation and, therefore, the DP solution would be inefficient because

We avoid the computational burden of the full solution by approximating the expected value function (that is, the emax function) at a state space point using the average of value functions of past iterations in which the parameter vector is "close" to the current parameter vector and the state variables are either exactly the same as the current state variables (if the state space is finite) or close to the current state variables (if the state space is continuous). This method of updating the emax function is similar to Pakes and McGuire (2001) except in the important respect that we also include the parameter vector in determining the set of iterations over which averaging occurs.

Note that the probability function that determines the next period parameter values is not a Markov transition function because our updated emax function depends on the past simulations of parameter vectors and value functions. We prove that under mild conditions, the probability function converges to the true MCMC transition function as the number of iterations of our Bayesian MCMC algorithm increases. That is, as the number of iterations increases, our algorithm becomes closer to the standard MCMC algorithm.

Our algorithm also helps in the "curse of dimensionality" situation where the dimension of the state space is high. In most DP solution exercises involving a continuous state variable, the state space grid points, once determined, are fixed over the entire algorithm, as in Rust (1997). In our Bayesian DP algorithm, the state space grid points do not have to be the same for each solution-estimation iteration. In fact, by varying the state space grid points at each solution-estimation iteration, our algorithm allows for an arbitrarily large number of state space grid points by increasing the number of iterations. This is how our estimation method reduces the computational burden in high-dimensional cases.

We demonstrate the performance of our algorithm by estimating a dynamic, infinite horizon model of firm entry and exit choice with observed and unobserved heterogeneity. The unobserved random effects coefficients are assumed to have a continuous distribution function, and the observed characteristics are assumed to be continuous as well. It is well known that for a conventional dynamic programming simulated maximum likelihood estimation strategy, this setup imposes a severe computational burden. The computational burden is due to the fact that during each estimation step, the DP problem has to be solved for each firm hundreds of times. Because of the observed heterogeneity, each firm has a different parameter value. Furthermore, because the random effects term has to be integrated out numerically via Monte Carlo integration, for each firm, one has to simulate the random effects parameter hundreds of times, and for each simulation, solve for the DP problem. This is why most

at the initial stage, true parameter distribution is not known. On the other hand, if prior to the estimation, one has a fairly accurate prior about the location of the parameter estimates, and thus the model needs to be solved at only very few parameter values up front, then the algorithm could be computationally efficient.

practitioners of structural estimation follow Heckman and Singer (1984), and assume discrete distributions for random effects and only allow for discrete types as observed characteristics.

We show that the computational burden of the estimation exercise above, using our algorithm, becomes quite similar in difficulty to the Bayesian estimation of a static discrete choice model with random effects (see McCulloch and Rossi (1994) for details). Indeed, through simulation-estimation exercises, we show that the computing time for our estimation exercise is around five times as fast and significantly more accurate than the conventional random effects simulated maximum likelihood estimation algorithm. In addition to the experiments, we formally prove that under very mild conditions, the distribution of parameter estimates simulated from our solution-estimation algorithm converges in probability to the true posterior distribution as we increase the number of iterations.

Our algorithm shows that the Bayesian methods of estimation, suitably modified, can be used effectively to conduct full-solution-based estimation of structural DDC models. Thus far, application of Bayesian methods to estimate such models has been particularly difficult. The main reason is that the solution of the DP problem, that is, the repeated calculation of the Bellman operator, is computationally so demanding that the MCMC, which typically involves far more iterations than the standard maximum likelihood (ML) routine, becomes infeasible quickly with a relatively small increase in model complexity. One of the few examples of Bayesian estimation is Lancaster (1997). He successfully estimated the equilibrium search model where the Bellman equation can be transformed into an equation where all the information on optimal choice of the individual can be summarized in the reservation wage and, hence, there is no need to solve the value function. Another line of research is Geweke and Keane (2000) and Houser (2003), who estimated the DDC model without solving the DP problem. In contrast, our paper accomplishes Bayesian estimation based on full solution of the infinite horizon DP problem by simultaneously solving for the DP problem and iterating on the pseudo-MCMC algorithm. The difference turns out to be important because their estimation algorithms can only accommodate limited specification of unobserved heterogeneity.[5]

Our estimation method makes Bayesian application to DDC models not only computationally feasible, but possibly even superior to the existing (non-Bayesian) methods, by reducing the computational burden of estimating a dynamic model to that of estimating a static one. Furthermore, the usually cited

[5]Since the working paper version of this paper has been circulated, several authors have used the Bayesian DP algorithm and made some important extensions. Osborne (2007) applied the Bayesian DP algorithm to the estimation of dynamic discrete choice model with random effects and estimated the dynamic consumer brand choice model. Norets (2007) applied it to the DDC model with serially correlated state variables. Also see Brown and Flinn (2006) for a classical econometric application of the idea.

advantages of Bayesian estimation over classical estimation methods apply here as well. That is, first, the conditions for the convergence of the pseudo-MCMC algorithm are in general weaker than the conditions for the global maximum of the ML estimator, as we show in this paper. Second, in MCMC, standard deviations of parameter estimates are simply the sample standard errors of parameter draws, whereas in ML estimation, standard errors have to be computed, usually either by inverting the numerically calculated information matrix, which is valid only in a large sample world, or by repeatedly bootstrapping and reestimating the model, which is computationally demanding.

The organization of the paper is as follows. In Section 2, we present a general version of the DDC model and discuss conventional estimation methods as well as our Bayesian DP algorithm. In Section 3, we state theorems and corollaries on the convergence of our algorithm under some mild conditions. In Section 4, we present a simple model of entry and exit. In Section 5, we present the simulation and estimation results of several experiments applied to the model of entry and exit. Finally, in Section 6, we conclude and briefly discuss the future direction of this research. Appendices are provided as a supplement on the *Econometrica* website (Imai, Jain, and Ching (2009)). Appendix A contains some results of the simulation-estimation exercises of the basic model and the random effects model. Appendix B contains all proofs. Appendix C contains plots of the MCMC estimation of the random effects model.

2. THE FRAMEWORK

We estimate an infinite horizon dynamic model of a forward-looking agent. Let θ be the J-dimensional parameter vector. Let S be the set of state space points and let s be an element of S. We assume that S is finite. Let A be the set of all possible actions and let a be an element of A. We assume A to be finite to study discrete choice models.

Let $R(s, a, \epsilon_a, \theta)$ be the current period return function of choosing action a, where s is the state variable and ϵ is a vector whose ath element ϵ_a is a random shock to current returns to choice a. We further assume that ϵ follows a multivariate distribution $F(\epsilon|\theta)$ with density function $dF(\epsilon, \theta)$ and is independent over time. We assume that the transition probability of next period state s', given current period state s and action a, is $f(s'|s, a, \theta)$, where θ is the parameter vector. Then the time invariant value function can be defined to be the maximum of the discounted sum of expected revenues as

$$V(s_t, \epsilon_t, \theta) \equiv \max_{\{a_t, a_{t+1}, \ldots\}} E\left[\sum_{\tau=t}^{\infty} \beta^{\tau} R\big(s_\tau, a_\tau, \epsilon_{a_\tau}, \theta\big) \Big| s_t, \epsilon_t\right],$$

where β is the discount factor. This value function is known to be the unique solution to the Bellman equation

$$V(s, \epsilon, \theta) = \max_{a \in A} \big\{R(s, a, \epsilon_a, \theta) + \beta E_{s', \epsilon'}[V(s', \epsilon', \theta)|s, a]\big\}, \tag{1}$$

where s' is the next period's state variable. The expectation is taken with respect to the next period shock ϵ' and the next period state s'.

If we define $\mathcal{V}(s, a, \epsilon_a, \theta)$ to be the expected value of choosing action a, then

$$\mathcal{V}(s, a, \epsilon_a, \theta) = R(s, a, \epsilon_a, \theta) + \beta E_{s',\epsilon'}[V(s', \epsilon', \theta)|s, a]$$

and the value function can be written as

$$V(s, \epsilon, \theta) = \max_{a \in A} \mathcal{V}(s, a, \epsilon_a, \theta).$$

We assume that the data set for estimation includes variables which correspond to state vector s and choice a in our model but the choice shock ϵ is not observed. That is, the observed data are $Y_{N^d,T^d} \equiv \{s^d_{i,\tau}, a^d_{i,\tau}, G^d_{i,\tau}\}_{i=1,\tau=1}^{N^d,T^d}$, where N^d is the number of firms and T^d is the number of time periods.[6] Furthermore,

$$a^d_{i,\tau} = \arg\max_{a \in A} \mathcal{V}(s^d_{i,\tau}, a, \epsilon_a, \theta),$$

$$G^d_{i,\tau} = \begin{cases} R\big(s^d_{i,\tau}, a^d_{i,\tau}, \epsilon_{a^d_{i,\tau}}, \theta\big), & \text{if } (s^d_{i,\tau}, a^d_{i,\tau}) \in \Psi, \\ 0, & \text{otherwise.} \end{cases}$$

The current period return is observable in the data only when the pair of state and choice variables belongs to the set Ψ. In the entry–exit problem of firms that we discuss later, profit of a firm is only observed when the incumbent firm stays in. In this case, Ψ is a set whose state variable is being an incumbent (and the capital stock) and the choice variable is staying in.

Let $\pi(\cdot)$ be the prior distribution of θ. Furthermore, let $L(Y_{N^d,T^d}|\theta)$ be the likelihood of the model, given the parameter θ and the value function $V(\cdot, \cdot, \theta)$, which is the solution of the DP problem. Then, we have the posterior distribution function of θ:

$$P(\theta|Y_{N^d,T^d}) \propto \pi(\theta) L(Y_{N^d,T^d}|\theta). \tag{2}$$

Let $\epsilon \equiv \{\epsilon_{i,\tau}\}_{i=1,\tau=1}^{N^d,T^d}$. Because ϵ is unobserved to the econometrician, the likelihood is an integral over it. That is, if we define $L(Y_{N^d,T^d}|\epsilon, \theta)$ to be the likelihood conditional on (ϵ, θ), then

$$L(Y_{N^d,T^d}|\theta) = \int L(Y_{N^d,T^d}|\epsilon, \theta)\, dF_\epsilon(\epsilon|\theta).$$

[6]We denote any variables with d superscript to be the data.

The value function enters into the likelihood through choice probability, which is a component of the likelihood. That is,[7]

$$P[a = a_{i,\tau}^d | s_{i,\tau}^d, V, \theta] = \Pr\Big[\epsilon : a_{i,\tau}^d = \arg\max_{a \in A}\big(R(s_{i,\tau}^d, a, \epsilon_a, \theta) + \beta E_{s',\epsilon'}[V(s', \epsilon', \theta) | s_{i,\tau}^d, a]\big)\Big]. \tag{3}$$

Below we briefly describe the conventional estimation approaches and then the Bayesian dynamic programming algorithm we propose.

2.1. *The Maximum Likelihood Estimation*

The conventional ML estimation procedure of the DP problem consists of two main steps. First is the solution of the DP problem and the subsequent construction of the likelihood, which is called the *inner loop*; second is the estimation of the parameter vector, which is called the *outer loop*.

DP Step (Inner Loop)

Given parameter vector θ, we solve for the fixed point $V(\cdot, \cdot, \theta)$ of the Bellman operator T_θ:

$$T_\theta V(s, \epsilon, \theta) \equiv \max_a \big\{R(s, a, \epsilon_a, \theta) + \beta E_{s',\epsilon'}[V(s', \epsilon', \theta) | s, a]\big\}.$$

This typically involves several steps.

Step a. The random choice shock ϵ is drawn a fixed number of times, say, M_ϵ, generating $\epsilon^{(m)}, m = 1, \ldots, M_\epsilon$. At iteration 0, we let the expected value function be 0, that is, $E_{\epsilon'}[V^{(0)}(s, \epsilon', \theta)] = 0$ for every $s \in S$. Then we set initial guess of the value function at iteration 1 to be the current period payoff. That is,

$$V^{(1)}\big(s, \epsilon^{(m)}, \theta\big) = \max_{a \in A}\big\{R\big(s, a, \epsilon_a^{(m)}, \theta\big)\big\}$$

for every $s \in S$, $\epsilon^{(m)}$.

Step b. Assume we are at iteration t of the Bellman operator. Given $s \in S$ and $\epsilon^{(m)}$, the value of every choice $a \in A$ is calculated. For the emax function, we use the approximated expected value function $\widehat{E}_{\epsilon'}[V^{(t-1)}(s', \epsilon', \theta)]$ computed at the previous iteration $t - 1$ for every $s' \in S$. Hence, the iteration t

[7]Notice that it is not necessary that we have a random choice shock ϵ_a for each choice a. What is important for the feasibility of estimation is that the likelihood, which is based on the choice probability $P[a = a_{i,\tau}^d | s_{i,\tau}^d, V, \theta]$, is well defined and bounded for all $\{a_{i,\tau}^d, s_{i,\tau}^d\}$, and for uniformly bounded V and $\theta \in \Theta$.

value of choice a is

$$\mathcal{V}^{(t)}\big(s, a, \epsilon_a^{(m)}, \theta\big) = R\big(s, a, \epsilon_a^{(m)}, \theta\big) + \beta \sum_{s'} \widehat{E}_{\epsilon'}\big[V^{(t-1)}(s', \epsilon', \theta)\big] f(s'|s, a, \theta).$$

Then we compute the value function, $V^{(t)}(s, \epsilon^{(m)}, \theta) = \max_{a \in A}\{\mathcal{V}^{(t)}(s, a, \epsilon_a^{(m)}, \theta)\}$. This calculation is done for every $s \in S$ and $\epsilon^{(m)}$, $m = 1, \ldots, M_\epsilon$.

Step c. The approximation for the expected value function is computed by taking the average of value functions over simulated choice shocks as

$$\widehat{E}_{\epsilon'}\big[V^{(t)}(s', \epsilon', \theta)\big] \equiv \frac{1}{M_\epsilon} \sum_{m=1}^{M_\epsilon} V^{(t)}\big(s', \epsilon^{(m)}, \theta\big). \tag{4}$$

Steps b and c have to be repeated for every state space point $s \in S$. Furthermore, the two steps have to be repeated until the value function converges. That is, for a small $\delta > 0$, $|V^{(t)}(s, \epsilon^{(m)}, \theta) - V^{(t-1)}(s, \epsilon^{(m)}, \theta)| < \delta$ for all $s \in S$ and $m = 1, \ldots, M_\epsilon$.

Likelihood Construction

The important increment of the likelihood is the conditional choice probability $P[a = a_{i,\tau}^d | s_{i,\tau}^d, V, \theta]$ given the state $s_{i,\tau}^d$, value function V and the parameter θ. For example, suppose that the per period return function is specified as

$$R(s, a, \epsilon_a, \theta) = \widehat{R}(s, a, \theta) + \epsilon_a,$$

where $\widehat{R}(s, a, \theta)$ is the deterministic component of the per period return function. Also, denote

$$\widehat{\mathcal{V}}(s, a, \theta) = \widehat{R}(s, a, \theta) + \beta \sum_{s'} \widehat{E}_{\epsilon'}[V(s', \epsilon', \theta)] f(s'|s, a, \theta)$$

to be the deterministic component of the value of choosing action a. Then

$$P[a_{i,\tau}^d | s_{i,\tau}^d, V, \theta] = P\big[\epsilon_a - \epsilon_{a_{i,\tau}^d} \leq \widehat{\mathcal{V}}(s, a_{i,\tau}^d, \theta) - \widehat{\mathcal{V}}(s, a, \theta); \forall a \neq a_{i,\tau}^d | s_{i,\tau}^d, V, \theta\big],$$

which becomes a multinomial probit specification when the error term ϵ is assumed to follow a joint normal distribution.[8]

[8]As long as the choice probability is well defined, the error term does not have to be additive. The dynamic discrete choice model is essentially a multinomial discrete choice model, where the right hand side includes future expected value functions.

Likelihood Maximization Routine (Outer Loop)

Suppose we have J parameters to estimate. In a typical ML estimation routine, where one uses the Newton hill climbing algorithm, at iteration t, likelihood is derived under the original parameter vector $\theta^{(t)}$ and under the perturbed parameter vector $\theta^{(t)} + \Delta\theta_j$, $j = 1, \ldots, J$. The perturbed likelihood is used together with the original likelihood to derive the new direction of the hill climbing algorithm. This is done to derive the parameters for the iteration $t + 1$, $\theta^{(t+1)}$. That is, during a single ML estimation routine, the DP problem needs to be solved in full $J + 1$ times. Furthermore, often the ML estimation routine has to be repeated many times until convergence is achieved. During a single iteration of the maximization routine, the inner loop algorithm needs to be executed at least as many times as the number of parameters plus 1. Since the estimation requires many iterations of the maximization routine, the entire algorithm is usually computationally extremely burdensome.

2.2. *The Conventional Bayesian MCMC Estimation*

A major computational issue in the Bayesian estimation method is that the posterior distribution, given by equation (2), is a high-dimensional and complex function of the parameters. Instead of directly simulating the posterior, we adopt the MCMC strategy and construct a transition density from current parameter θ to the next iteration parameter θ', $f(\theta, \theta')$, which satisfies, among other more technical conditions, the equality

$$P(\theta|Y_{N^d,T^d}) = \int f(\theta, \theta')P(\theta'|Y_{N^d,T^d})\, d\theta',$$

where $P(\theta|Y_{N^d,T^d})$ is the posterior distribution of θ given Y_{N^d,T^d}. From the transition density, we simulate the sequence of parameters $\{\theta^{(s)}\}_{s=1}^{t}$, which is known to converge to the correct posterior. The conventional Bayesian estimation method applied to the DDC model proceeds in the following two main steps.[9]

Metropolis–Hastings (M–H) Step

The M–H algorithm is a Markov chain simulation algorithm used to draw from a complex target distribution.[10] In our case, the target density is proportional to $\pi(\theta)L(Y_{N^d,T^d}|\theta)$. Given $\theta^{(t)}$, the parameter vector at iteration t, we draw the new parameter vector $\theta^{(t+1)}$ as follows: First, we draw the candidate

[9]See Tierney (1994) and Tanner and Wong (1987) for details on Bayesian estimation.
[10]See Robert and Casella (2004) for more details on the M–H algorithm.

parameter vector $\theta^{*(t)}$ from a candidate generating density (or proposal density) $q(\theta^{(t)}, \theta^{*(t)})$. Then we accept $\theta^{*(t)}$, that is, set $\theta^{(t+1)} = \theta^{*(t)}$ with probability,

$$\lambda\big(\theta^{(t)}, \theta^{*(t)}\big) = \min\left\{\frac{\pi(\theta^{*(t)})L(Y_{N^d,T^d}|\theta^{*(t)})q(\theta^{*(t)}, \theta^{(t)})}{\pi(\theta^{(t)})L(Y_{N^d,T^d}|\theta^{(t)})q(\theta^{(t)}, \theta^{*(t)})}, 1\right\}, \tag{5}$$

and we reject $\theta^{*(t)}$, that is, set $\theta^{(t+1)} = \theta^{(t)}$ with probability $1 - \lambda$.

Since the likelihood is a function of the value function, the DP problem needs to be solved for each $\theta^{*(t)}$. Hence, similar to the maximum likelihood estimation procedure, one can interpret the M–H step as the outer loop of the estimation algorithm and interpret the DP step involved in constructing the likelihood for each candidate parameter vector as the inner loop. This DP step is the same as the one described in the previous subsection. The full-solution-based Bayesian MCMC method turns out to be even more burdensome computationally than the full-solution-based ML method because MCMC typically requires a lot more iterations than the ML routine.

We next present our algorithm for estimating the parameter vector θ. We call it the Bayesian dynamic programming (Bayesian DP) algorithm. The key innovation of our algorithm is that we solve the DP problem and estimate the parameters simultaneously rather than sequentially as in the conventional methods described above.

2.3. *The Bayesian Dynamic Programming Estimation*

The main difference between the Bayesian DP algorithm and the conventional algorithm is that during each estimation step, we do not solve for the fixed point of the Bellman operator. In fact, during each modified M–H step, we iterate the Bellman operator only once. This operator can be expressed as

$$\begin{aligned} &T_\theta^{(t)}V(\cdot,\cdot,\theta) \\ &\quad\equiv \max_a\left\{R(s,a,\epsilon_a,\theta) + \beta\sum_{s'}\widehat{E}_{\epsilon'}^{(t)}[V(s',\epsilon',\theta)]f(s'|s,a,\theta)\right\}. \end{aligned}$$

Our Bellman operator $T_\theta^{(t)}$ depends on t because our approximation of the expected value function, $\widehat{E}_{\epsilon'}^{(t)}$, depends on t. In conventional methods, this approximation is given by equation (4). In contrast, we approximate it by averaging over a subset of past iterations. Let $\mathcal{H}^{(t)} \equiv \{\epsilon^{(s)}, \theta^{*(s)}, V^{(s)}\}_{s=1}^t$ be the history of shocks, candidate parameters,[11] and value functions up to iteration t. Let $\mathcal{V}^{(t)}(s, a, \epsilon_a^{(t)}, \theta^{*(t)}, \mathcal{H}^{(t-1)})$ be the value of choice a and let $V^{(t)}(s, \epsilon^{(t)}, \theta^{*(t)}, \mathcal{H}^{(t-1)})$ be the value function derived at iteration t of our

[11]We do not use past history of $\theta^{(s)}$; hence it is not included in $\mathcal{H}^{(t)}$.

solution–estimation algorithm. Then the value function and the approximation $\widehat{E}_{\epsilon'}^{(t)}[V(s', \epsilon', \theta)|\mathcal{H}^{(t-1)}]$ for the expected value function $E_{\epsilon'}[V(s', \epsilon', \theta)]$ at iteration t are defined recursively as

$$
\begin{aligned}
(6)\qquad & \widehat{E}_{\epsilon'}^{(t)}\big[V(s', \epsilon', \theta)|\mathcal{H}^{(t-1)}\big] \\
& \quad \equiv \sum_{n=1}^{N(t)} V^{(t-n)}\big(s', \epsilon^{(t-n)}, \theta^{*(t-n)}, \mathcal{H}^{(t-n-1)}\big) \frac{K_h(\theta - \theta^{*(t-n)})}{\sum_{k=1}^{N(t)} K_h(\theta - \theta^{*(t-k)})}, \\
& \mathcal{V}^{(t-n)}\big(s, a, \epsilon_a^{(t-n)}, \theta^{*(t-n)}, \mathcal{H}^{(t-n-1)}\big) \\
& \quad = R\big(s, a, \epsilon_a^{(t-n)}, \theta^{*(t-n)}\big) \\
& \qquad + \beta \sum_{s'} \widehat{E}_{\epsilon'}^{(t-n)}\big[V\big(s', \epsilon', \theta^{*(t-n)}\big)\big|\mathcal{H}^{(t-n-1)}\big] f(s'|s, a, \theta), \\
& V^{(t-n)}\big(s, \epsilon^{(t-n)}, \theta^{*(t-n)}, \mathcal{H}^{(t-n-1)}\big) \\
& \quad = \max_{a \in A} \mathcal{V}^{(t-n)}\big(s, a, \epsilon_a^{(t-n)}, \theta^{*(t-n)}, \mathcal{H}^{(t-n-1)}\big),
\end{aligned}
$$

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

The approximated expected value function given by equation (6) is the weighted average of value functions of $N(t)$ most recent iterations. The sample size of the average, $N(t)$, increases with t, that is, $N(t) \to \infty$ as $t \to \infty$. Furthermore, we let $t - N(t) \to \infty$ as $t \to \infty$. The weights are high for the value functions at iterations with parameters close to the current parameter vector $\theta^{(t)}$. This is similar to the idea of Pakes and McGuire (2001), where the expected value function is the average of the past N iterations. In their algorithm, averages are taken only over the value functions that have the same state space point as the current state space point s. In our case, averages are taken over the value functions that have the same state as the current state s as well as parameters that are close to the current parameter $\theta^{(t)}$.

We now describe the complete Bayesian DP algorithm at iteration t. Suppose that $\{\epsilon^{(l)}\}_{l=1}^t$, $\{\theta^{*(l)}\}_{l=1}^t$, and $\{V^{(l)}(s, \epsilon^{(l)}, \theta^{*(l)}, \mathcal{H}^{(l-1)})\}_{l=1}^t$ are given for all discrete $s \in S$. Then we update the value function and the parameters as follows.

Modified M–H Step[12]

We draw the new parameters $\theta^{(t+1)}$ as follows: First, we draw the candidate parameter vector $\theta^{*(t)}$ from the proposal density $q(\theta^{(t)}, \theta^{*(t)})$. Then we

[12]We are grateful to Andriy Norets for pointing out a flaw in the Gibbs sampling scheme adopted in the earlier draft. We follow Norets (2007) and Osborne (2007), and adopt the modified Metropolis–Hastings algorithm for the MCMC sampling. Notice that in a linear model (see

accept $\theta^{*(t)}$, that is, set $\theta^{(t+1)} = \theta^{*(t)}$ with probability $\lambda(\theta^{(t)}, \theta^{*(t)}|\mathcal{H}^{(t-1)})$, defined in the same way as before, in equation (5), and we reject $\theta^{*(t)}$, that is, we set $\theta^{(t+1)} = \theta^{(t)}$ with probability $1 - \lambda$.

Modified DP Step

As explained in the conventional Bayesian MCMC algorithm, during each Metropolis–Hastings step, we need to update the expected value function $\widehat{E}^{(t)}_{\epsilon'}[V(\cdot,\cdot,\cdot)|\mathcal{H}^{(t-1)}]$ for parameters $\theta^{(t)}$ and $\theta^{*(t)}$. To do so for all $s \in S$, without iterating on the Bellman operator until convergence, we follow equation (6). For use in future iterations, we simulate the value function by drawing $\epsilon^{(t)}$ to derive $\mathcal{V}^{(t)}(s, a, \epsilon^{(t)}_a, \theta^{*(t)}, \mathcal{H}^{(t-1)})$ and $V^{(t)}(s, \epsilon^{(t)}, \theta^{*(t)}, \mathcal{H}^{(t-1)})$.

We repeat these two steps until the sequence of parameter simulations converges to a stationary distribution. In our algorithm, in addition to the DP and Bayesian MCMC methods, nonparametric kernel techniques are also used to approximate the value function. Notice that the convergence of kernel-based approximation is not based on the large sample size of the data, but on the number of Bayesian DP iterations. Moreover, the Bellman operator is evaluated only once during each estimation iteration. Hence, the Bayesian DP algorithm avoids the computational burden of solving for the DP problem during each estimation step, which involves repeated evaluation of the Bellman operator.[13]

It is important to note that the modified Metropolis–Hastings algorithm is not a Markov chain.[14] This is because it involves value functions calculated in past iterations. Hence, convergence of our algorithm is by no means trivial.

McCulloch and Rossi (1994) for an example), the equations are

$$z_{it} = R_{it}\gamma + u_{it}, \quad u_{it} \sim N(0, \Sigma),$$

$$y_{ijt} = \begin{cases} 1, & \text{if } z_{ijt} \geq \max\{z_{it}\}, \\ 0, & \text{otherwise.} \end{cases}$$

Then, once z_{it} is derived by data augmentation, the first equation is linear in parameter γ and thus can be estimated by linear Gibbs sampling. However, in our case,

$$z_{it} = R_{it}\gamma + \beta EV(R_{it+1}|R_{it}, \theta) + u_{it}, \quad u_{it} \sim N(0, \Sigma),$$

where $EV(R_{it+1}|R_{it}, \theta)$ is a nonlinear function of the state variable R_{it} and, thus, linear Gibbs sampling algorithms cannot be applied.

[13]Both Osborne (2007) and Norets (2007) approximated the expected value function using the value functions computed in the past iterations evaluated at the past parameter draws $\theta^{(t-n)}$. Here, we use the value functions evaluated at the past proposal parameter draws $\theta^{*(t-n)}$. We chose to do so because given $\theta^{(t)}$, it is easier to control the random movement of $\theta^{*(t)}$ than the random movement of $\theta^{(t+1)}$, since $\theta^{*(t)}$ is drawn from a known distribution function. This simplifies both the proofs and the empirical example. By keeping the conditional variance of the proposal density given $\theta^{(t)}$ small, we can guarantee that the invariant distribution of $\theta^{*(t)}$ is not very different from that of θ.

[14]We are grateful to Peter Rossi for emphasizing this.

In the next section, we state theorems and corollaries which show that under some mild assumptions, the distribution of the parameters generated by our algorithm converges to the true posterior in probability. All proofs are in Appendix B.

3. THEORETICAL RESULTS

In this section, we prove the convergence of the Bayesian DP algorithm in the basic model. We then present the random effects model and the continuous state space model in two subsections. To facilitate the proof, we modify the DP step slightly to calculate the expected value function. That is, we simulate the value function by drawing $\boldsymbol{\epsilon}^{(t)}$ to derive

$$\begin{aligned}&\mathcal{V}^{(t)}\big(s, a, \boldsymbol{\epsilon}_a^{(t)}, \theta^{*(t)}, \mathcal{H}^{(t-1)}\big)\\&\quad = \widetilde{R}\big(s, a, \boldsymbol{\epsilon}_a^{(t)}, \theta^{*(t)}\big) + \beta \sum_{s'} \widehat{E}_{\boldsymbol{\epsilon}'}^{(t)}\big[V\big(s', \boldsymbol{\epsilon}', \theta^{*(t)}\big)\big|\mathcal{H}^{(t-1)}\big] f(s'|s, a, \theta),\end{aligned}$$

where

$$\widetilde{R}\big(s, a, \boldsymbol{\epsilon}_a^{(t)}, \theta^{*(t)}\big) = \min\big\{\max\big\{R\big(s, a, \boldsymbol{\epsilon}_a^{(t)}, \theta^{*(t)}\big), -M_R\big\}, M_R\big\}$$

for a large positive M_R. This makes the current period return function used in equation (1) uniformly bounded, which simplifies the proof. This modification does not make any difference in practice because M_R can be set arbitrarily large. Let V denote the solution of the Bellman equation:[15]

$$V(s, \boldsymbol{\epsilon}, \theta) = \max\big\{\widetilde{R}(s, a, \boldsymbol{\epsilon}_a, \theta) + \beta E_{s', \boldsymbol{\epsilon}'}[V(s', \boldsymbol{\epsilon}', \theta)|s, a]\big\}.$$

Next we show that under some mild assumptions, our algorithm generates a sequence of parameters $\theta^{(1)}, \theta^{(2)}, \ldots$ that converges in probability to the correct posterior distribution.

ASSUMPTION 1: *Parameter space $\Theta \subseteq R^J$ is compact, that is, closed and bounded in the Euclidean space R^J. The proposal density $q(\theta, \cdot)$ is continuously differentiable, strictly positive, and uniformly bounded in the parameter space given any $\theta \in \Theta$.*[16]

[15]Given the expected value function, per period return in the likelihood construction is set to be R not $\widetilde{R}$. See equation (3).

[16]Compactness of the parameter space is a standard assumption used in proving the convergence of the MCMC algorithm. It is often not necessary but simplifies the proofs. An example of the proposal density that satisfies Assumption 1 is the multivariate normal density, truncated to only cover the compact parameter space.

ASSUMPTION 2: *For any* $s \in S$, $a \in A$, *and* ϵ, $\theta \in \Theta$, $|\widetilde{R}(s, a, \epsilon_a, \theta)| < M_R$ *for some* $M_R > 0$. *Also*, $\widetilde{R}(s, a, \cdot, \theta)$ *is a nondecreasing function in* ϵ *and* $\widetilde{R}(s, a, \cdot, \cdot)$ *satisfies the Lipschitz condition in terms of* ϵ *and* θ. *Also*, *the density function* $dF(\epsilon, \theta)$ *and the transition function* $f(\cdot|\cdot, a, \cdot)$ *given a satisfy the Lipschitz condition*.

ASSUMPTION 3: β *is known*[17] *and* $\beta < 1$.

ASSUMPTION 4: *For any* $s \in S$, ϵ, *and* $\theta \in \Theta$, $V^{(0)}(s, \epsilon, \theta) < M_I$ *for some* $M_I > 0$. *Furthermore*, $V^{(0)}(s, \cdot, \cdot)$ *also satisfies the Lipschitz condition in terms of* ϵ *and* θ.

Assumptions 2, 3, and 4 jointly make $V^{(t)}(s, \epsilon, \theta)$ and hence $\widehat{E}^{(t)}_{\epsilon'}[V(s', \epsilon', \theta)]$, $t = 1, \ldots,$ uniformly bounded, measurable, and continuous, and satisfy the Lipschitz condition as well.

ASSUMPTION 5: $\pi(\theta)$ *is positive and bounded for any* $\theta \in \Theta$. *Similarly*, *for any* $\theta \in \Theta$ *and* V *uniformly bounded*, $L(Y_{N^d, T^d}|\theta, V(\cdot, \theta)) > 0$ *and is bounded and uniformly continuous in* $\theta \in \Theta$.

Define the sequence $t(l)$, $\widetilde{N}(l)$ as follows. For some $s > 0$, define $t(1) = s$ and $\widetilde{N}(1) = N(s)$. Let $t(2)$ be such that $t(2) - N(t(2)) = t(1) + 1$. Such $t(2)$ exists from the assumption on $N(s)$ stated in Assumption 6. Also, let $\widetilde{N}(2) = N(t(2))$. Similarly, for any $l > 2$, let $t(l+1)$ be such that $t(l+1) - N(t(l+1)) = t(l) + 1$ and let $\widetilde{N}(l+1) = N(t(l+1))$.

ASSUMPTION 6: $N(t)$ *is nondecreasing in* t, *increases at most by one for a unit increase in* t, *and* $N(t) \to \infty$. *Furthermore*, $t - N(t) \to \infty$ *and there exists a finite constant* $A > 0$ *such that* $\widetilde{N}(l+1) < A\widetilde{N}(l)$ *for all* $l > 1$, *and*, *for any* $l = 2, \ldots,$ $N(t(l) + 1) = N(t(l)) + 1$.

An example of a sequence that satisfies Assumption 6 is $t(l) \equiv s + \frac{(l+1)(l+2)}{2}$, $\widetilde{N}(l) = l$ for $l > 0$, and $N(t) = l + 1$ for $t(l) < t \leq t(l+1)$, $l > 1$.

ASSUMPTION 7: *The bandwidth* h *is a nonincreasing function of* N *and as* $N \to \infty$, $h(N) \to 0$ *and* $Nh(N)^{9J} \to \infty$. *Further*, $h(N)$ *is constant for* $N(t(l)) < N \leq N(t(l+1))$.

ASSUMPTION 8: $K_h(\cdot)$ *is a multivariate kernel with bandwidth* $h > 0$. *That is*, $K_h(z) = (1/h^J)K(z/h)$, *where* K *is a nonnegative*, *continuous*, *bounded real*

[17]The assumption that β is known may not be necessary but greatly simplifies the proofs. However, we show later that β can be successfully estimated as long as its prior is restricted to be strictly less than 1.

function which is symmetric around 0 *and integrates to* 1, *that is,* $\int K(z)\,dz = 1$. *Furthermore,* $\int zK(z)\,dz < \infty$ *and* $\int_{|z|>1/h} K(z)\,dz \leq Ah^{4J}$ *for some positive constant* A, *where for a vector* z, $|z| = \sup_{j=1,\ldots,J} |z_j|$, *and* K *has an absolutely integrable Fourier transform*.

THEOREM 1: *Suppose Assumptions* 1–8 *are satisfied for* $V^{(t)}$, π, L, ϵ, *and* θ. *Then, the sequence of approximated expected value functions* $\widehat{E}^{(t)}_{\epsilon'}[V(s', \epsilon', \theta)|\mathcal{H}^{(t-1)}]$ *converges to* $E_{\epsilon'}[V(s', \epsilon', \theta)]$ *in probability uniformly over* $s' \in S$, $\theta \in \Theta$ *as* $t \to \infty$. *Similarly, the sequence of value functions* $V^{(t)}(s, \epsilon, \theta, \mathcal{H}^{(t-1)})$ *converges to* $V(s, \epsilon, \theta)$ *in probability uniformly over* s, ϵ, *and* $\theta \in \Theta$ *as* $t \to \infty$.

COROLLARY 1: *Suppose Assumptions* 1–8 *are satisfied. Then* $\lambda(\theta^{(t)}, \theta^{*(t)}|\mathcal{H}^{(t-1)})$ *converges to* $\lambda(\theta^{(t)}, \theta^{*(t)})$ *in probability uniformly in* Θ.

THEOREM 2: *Suppose Assumptions* 1–8 *are satisfied for* $V^{(t)}$, $t = 1, \ldots,$ π, L, ϵ, *and* θ. *Then* $\theta^{(t)}$ *converges to* $\widetilde{\theta}^{(t)}$ *in probability, where* $\widetilde{\theta}^{(t)}$ *is a Markov chain generated by the Metropolis–Hastings algorithm with proposal density* $q(\theta, \theta^{(*)})$ *and acceptance probability function* $\lambda(\theta, \theta^{(*)})$.

COROLLARY 2: *The sequence of parameter simulations generated by the Metropolis–Hastings algorithm with proposal density* $q(\theta, \theta^*)$ *and acceptance probability* $\lambda(\theta, \theta^*)$ *converges to the true posterior in total variation norm. That is,*

$$\lim_{n\to\infty} \left\| \int K^n(\theta, \cdot)\mu_0(d\theta) - \mu \right\|_{\mathrm{TV}} = 0$$

for arbitrary initial distribution μ_0, *where* μ *is the true posterior distribution and* $K^n(\theta, \cdot)$ *is the transition kernel for* n *iterations*.

By Corollary 2, we can conclude that the distribution of the sequence of parameters $\theta^{(t)}$ generated by the Bayesian DP algorithm converges in probability to the true posterior distribution.

To understand the basic logic of the proof of Theorem 1, suppose that parameter $\theta^{(t)}$ stays fixed at a value θ^* for all iterations t. Then equation (6) reduces to

$$\widehat{E}^{(t)}_{\epsilon'}\left[V(s', \epsilon', \theta^*)|\mathcal{H}^{(t-1)}\right] = \frac{1}{N(t)} \sum_{n=1}^{N(t)} V^{(t-n)}\left(s', \epsilon^{(t-n)}, \theta^*, \mathcal{H}^{(t-n-1)}\right).$$

Then our algorithm boils down to a simple version of the machine learning algorithm discussed by Pakes and McGuire (2001) and Bertsekas and Tsitsiklis (1996). They approximated the expected value function by taking the average over all past value function iterations whose state space point is the same

as the state space point s'. Bertsekas and Tsitsiklis (1996) discussed the convergence issues and showed that under some assumptions, the sequence of the value functions from the machine learning algorithm converges to the true value function almost surely. The difficulty of the proofs lies in extending the logic of the convergence of the machine learning algorithm to the framework of estimation, where the parameter vector moves around as well. Our answer to this issue is simple: for a parameter vector $\theta \in \Theta$ at iteration t, we look at the past iterations and use value functions at parameters $\theta^{*(t-n)}$ that are very close to θ. Then the convergence is very similar to the case where the parameter vector is fixed, as long as the number of past value functions used can be made arbitrarily large. This is guaranteed by Assumption 1, since every neighborhood in the compact parameter space Θ will be visited infinitely often. It is important to note that for convergence of the value function, the estimation algorithm does not have to be Markov. The only requirement is that during the iteration, each neighborhood in Θ has a strictly positive probability of being drawn.

3.1. *Random Effects*

Consider a model where, for a subset of parameters, each agent has a different value $\widetilde{\theta}_i$, which is randomly drawn from a density $f(\widetilde{\theta}_i|\theta_{(1)})$. The parameter vector of the model is $\theta \equiv (\theta_{(1)}, \theta_{(2)})$, where $\theta_{(1)}$ is the parameter vector for the distribution of the random coefficients and $\theta_{(2)}$ is the vector of other parameters. The parameter vector of firm i is $(\widetilde{\theta}_i, \theta_{(2)})$. Instead of explicitly integrating the likelihood over $\widetilde{\theta}_i$, we follow the commonly adopted and computationally efficient procedure of treating each $\widetilde{\theta}_i$ as a parameter and drawing it from its density. It is known (see McCulloch and Rossi (1994), Albert and Chib (1993), and Chib and Greenberg (1996)) that instead of drawing the entire parameter vector $(\{\widetilde{\theta}_i\}_{i=1}^{N^d}, \theta_{(1)}, \theta_{(2)})$ at once, it is often simpler to partition it into several blocks and draw the parameters of each block separately given the other parameters. Here, we propose to draw them in the following three blocks. At iteration t the blocks are

Block 1: Draw $\{\widetilde{\theta}_i^{(t+1)}\}_{i=1}^{N^d}$ given $\theta_{(1)}^{(t)}, \theta_{(2)}^{(t)}$.
Block 2: Draw $\theta_{(1)}^{(t+1)}$ given $\{\widetilde{\theta}_i^{(t+1)}\}_{i=1}^{N^d}, \theta_{(2)}^{(t)}$.
Block 3: Draw $\theta_{(2)}^{(t+1)}$ given $\{\widetilde{\theta}_i^{(t+1)}\}_{i=1}^{N^d}, \theta_{(1)}^{(t+1)}$.

Below we describe in detail the algorithm at each block.

Block 1—Modified M–H Step for Drawing $\widetilde{\theta}_i$

For firm i, we draw the new random effects parameters $\widetilde{\theta}_i^{(t+1)}$ as follows: We set the proposal density as the distribution function of $\widetilde{\theta}_i$, that is, $f(\widetilde{\theta}_i|\theta_{(1)})$. Notice that the prior is a function of $\theta_{(1)}$ and $\theta_{(2)}$, and not of $\widetilde{\theta}_i$. Hence for drawing $\widetilde{\theta}_i$ given $\theta_{(1)}$ and $\theta_{(2)}$, the prior is irrelevant. Similarly, given $\theta_{(1)}$, the likelihood

increment of firms other than i is also irrelevant in drawing $\widetilde{\theta}_i$. Therefore, we draw $\widetilde{\theta}_i$ using the likelihood increment of firm i, which can be written as

$$L_i\big(Y_{i,T^d}|\big(\widetilde{\theta}_i, \theta_{(2)}\big)\big)f\big(\widetilde{\theta}_i|\theta_{(1)}\big),$$

where

$$L_i\big(Y_{i,T^d}|\big(\widetilde{\theta}_i, \theta_{(2)}\big)\big) \equiv L\big(Y_{i,T^d}|\big(\widetilde{\theta}_i, \theta_{(2)}\big), V^{(t)}\big(\cdot,\cdot,\cdot,\widetilde{\theta}_i, \theta_{(2)}, \mathcal{H}^{(t-1)}\big)\big),$$

because the likelihood depends on the value function. Now, we draw the candidate parameter $\widetilde{\theta}_i^{*(t)}$ from the proposal density $f(\widetilde{\theta}_i^{*(t)}|\theta_{(1)})$. Then we accept $\widetilde{\theta}_i^{*(t)}$, that is, set $\widetilde{\theta}_i^{(t+1)} = \widetilde{\theta}_i^{*(t)}$ with probability

$$\begin{aligned}
&\lambda_1\big(\theta^{(t)}, \widetilde{\theta}_i^{*(t)}\big|\mathcal{H}^{(t-1)}\big)\\
&\quad= \min\left\{\frac{L_i(Y_{i,T^d}|(\widetilde{\theta}_i^{*(t)}, \theta_{(2)}^{(t)}))f(\widetilde{\theta}_i^{*(t)}|\theta_{(1)}^{(t)})f(\widetilde{\theta}_i^{(t)}|\theta_{(1)}^{(t)})}{L_i(Y_{i,T^d}|(\widetilde{\theta}_i^{(t)}, \theta_{(2)}^{(t)}))f(\widetilde{\theta}_i^{(t)}|\theta_{(1)}^{(t)})f(\widetilde{\theta}_i^{*(t)}|\theta_{(1)}^{(t)})}, 1\right\}\\
&\quad= \min\left\{\frac{L_i(Y_{i,T^d}|(\widetilde{\theta}_i^{*(t)}, \theta_{(2)}^{(t)}))}{L_i(Y_{i,T^d}|(\widetilde{\theta}_i^{(t)}, \theta_{(2)}^{(t)}))}, 1\right\};
\end{aligned}$$

otherwise, reject $\widetilde{\theta}_i^{*(t)}$, that is, set $\widetilde{\theta}_i^{(t+1)} = \widetilde{\theta}_i^{(t)}$ with probability $1 - \lambda_1$.

Block 2—Drawing $\theta_{(1)}^{(t+1)}$

Conditional on $\{\widetilde{\theta}_i^{(t+1)}\}_{i=1}^{N^d}$, the density of $\theta_{(1)}^{(t+1)}$ is proportional to $\prod_{i=1}^{N^d} f(\widetilde{\theta}_i^{(t+1)}|\theta_{(1)})$. Drawing from this density is straightforward as it does not involve the solution of the DP problem.[18]

Block 3—Modified M–H Algorithm for Drawing $\theta_{(2)}$

We draw the new parameters $\theta_{(2)}^{(t+1)}$ as follows: First, we draw the candidate parameter $\theta_{(2)}^{*(t)}$ from the proposal density $q(\theta_{(2)}^{(t)}, \theta_{(2)}^{*(t)})$. Then we accept $\theta_{(2)}^{*(t)}$,

[18]As pointed out by a referee, a potential issue could be serial correlation of θ_i, because the new θ_i's could be dependent on $\theta_{(1)}$ from the past iteration. Furthermore, draws of new θ_i could be heavily centered around the past mean of θ_i, which may suppress sufficient movement of $\theta_{(1)}$. MCMC plots in Appendix C for the example discussed later show that there is sufficient movement of the parameters $\theta_{(1)}$ and that serial correlation is small. An important statistic to look at in this case is the acceptance probability of the M–H draw of $\theta_i^{(t)}$. If the acceptance probability is too low, then there is insufficient movement of $\theta_i^{(t)}$ over iteration t and thus their hyperparameter $\theta_1^{(t)}$ will exhibit high correlation across t. On the other hand, if it is too high, that is, close to 1, then their mean $\theta_1^{(t)}$ will not change much if N is large, and thus will result in high serial correlation of $\theta_1^{(t)}$. In our random effects example, the acceptance probability of $\theta_i^{(t)}$ is around 15–25%, which is considered to be quite appropriate in the MCMC literature. If the acceptance rate is either too high or too low, then a different procedure such as the ones proposed by Osborne (2007) or Norets (2007) is recommended.

that is, set $\theta_{(2)}^{(t+1)} = \theta_{(2)}^{*(t)}$ with probability

$$\lambda_2\big(\theta_{(1)}^{(t+1)}, \theta_{(2)}^{*(t)} \big| \mathcal{H}^{(t-1)}\big) = \min\left\{ \frac{\pi(\theta_{(1)}^{(t+1)}, \theta_{(2)}^{*(t)}) \left[\prod_{i=1}^{N^d} L_i(Y_{i,T^d} | \widetilde{\theta}_i^{(t+1)}, \theta_{(2)}^{*(t)})\right] q(\theta_{(2)}^{*(t)}, \theta_{(2)}^{(t)})}{\pi(\theta_{(1)}^{(t+1)}, \theta_{(2)}^{(t)}) \left[\prod_{i=1}^{N^d} L_i(Y_{i,T^d} | \widetilde{\theta}_i^{(t+1)}, \theta_{(2)}^{(t)})\right] q(\theta_{(2)}^{(t)}, \theta_{(2)}^{*(t)})}, 1 \right\};$$

otherwise, we reject $\theta_{(2)}^{*(t)}$, that is, set $\theta_{(2)}^{(t+1)} = \theta_{(2)}^{(t)}$ with probability $1 - \lambda_2$.

Bellman Equation Step

During each M–H step, for each agent i we evaluate the expected value function $\widehat{E}_{\epsilon'}^{(t)}[V(\cdot, \cdot, \widetilde{\theta}_i, \theta_{(2)}) | \mathcal{H}^{(t-1)}]$. To do so for each agent, for all $s \in S$, we follow equation (6) as before.[19] For use in future iterations, we simulate the value function by drawing $\epsilon^{(t)}$ to derive

$$\begin{aligned}
&\mathcal{V}^{(t)}\big(s, a, \epsilon_a^{(t)}, \widetilde{\theta}_i, \theta_{(2)}, \mathcal{H}^{(t-1)}\big) \\
&\quad = \widetilde{R}\big(s, a, \epsilon_a^{(t)}, \widetilde{\theta}_i, \theta_{(2)}\big) \\
&\qquad + \beta \sum_{s'} \widehat{E}_{\epsilon'}^{(t)}\big[V\big(s', \epsilon', \widetilde{\theta}_i, \theta_{(2)}\big) \big| \mathcal{H}^{(t-1)}\big] f(s' | s, a, \theta), \\
&V^{(t)}\big(s, \epsilon^{(t)}, \widetilde{\theta}_i, \theta_{(2)}, \mathcal{H}^{(t-1)}\big) = \max_{a \in A} \mathcal{V}^{(t)}\big(s, a, \epsilon_a^{(t)}, \widetilde{\theta}_i, \theta_{(2)}, \mathcal{H}^{(t-1)}\big).
\end{aligned}$$

The additional computational burden necessary to estimate the random coefficient model is the computation of the value function which has to be done separately for each firm i, because each firm has a different random effects parameter vector. In this case, adoption of the Bayesian DP algorithm results in a large reduction in computational cost.

3.2. *Continuous State Space*

So far, we assumed a finite state space. However, the Bayesian DP algorithm can also be applied, with minor modifications, in a straightforward manner to other settings of dynamic discrete choice models. One example is the random grid approximation of Rust (1997).

Conventionally, randomly generated state vector grid points are fixed throughout the solution-estimation algorithm. If we follow this procedure and

[19] For more details, see Experiment 1 in Section 5.1.

let s_m, $m = 1, \ldots, M$, be the random grids that are generated before the start of the solution-estimation algorithm, then, given parameter θ, the expected value function approximation at iteration t of the DP solution algorithm using the Rust random grids method would be

$$
\begin{aligned}
(7)\qquad & \widehat{E}^{(t)}_{s',\epsilon'}\big[V(s',\epsilon',\theta)|s,a,\mathcal{H}^{(t-1)}\big] \\
&\equiv \sum_{m=1}^{M}\left[\sum_{n=1}^{N(t)} V^{(t-n)}\big(s_m,\epsilon^{(t-n)},\theta^{*(t-n)},\mathcal{H}^{(t-n-1)}\big)\frac{K_h(\theta-\theta^{*(t-n)})}{\sum_{k=1}^{N(t)}K_h(\theta-\theta^{*(t-k)})}\right] \\
&\quad\times\frac{f(s_m|s,a,\theta)}{\sum_{l=1}^{M}f(s_l|s,a,\theta)}.
\end{aligned}
$$

Notice that in this definition of emax function approximation, the grid points remain fixed over all iterations. In contrast, in our Bayesian DP algorithm, random grids can be changed at each solution-estimation iteration. Let $s^{(t)}$ be the random grid point generated at iteration t. Here $s^{(\tau)}$, $\tau = 1, 2, \ldots$, are drawn independently from a distribution. Then the expected value function can be approximated as

$$
\begin{aligned}
&\widehat{E}^{(t)}_{s',\epsilon'}\big[V(s',\epsilon',\theta)|s,a,\mathcal{H}^{(t-1)}\big] \\
&\equiv \sum_{n=1}^{N(t)} V^{(t-n)}\big(s^{(t-n)},\epsilon^{(t-n)},\theta^{*(t-n)},\mathcal{H}^{(t-n-1)}\big) \\
&\quad\times\frac{K_h(\theta-\theta^{*(t-n)})f(s^{(t-n)}|s,a,\theta)}{\sum_{k=1}^{N(t)}K_h(\theta-\theta^{*(t-k)})f(s^{(t-k)}|s,a,\theta)}.
\end{aligned}
$$

In the Rust method, if the total number of random grids is M, then the number of computations required for each iteration of the Bellman operator is M. Hence, at iteration τ, the number of DP computations that is required is $M\tau$. If a single DP solution step requires τ iterations of the Bellman operator and if each Newton ML step requires K DP solution steps, then to iterate the Newton ML algorithm once, we need to compute a single DP iteration $M\tau K$ times.

In contrast, in our Bayesian DP algorithm, at iteration t we only need to draw one state vector $s^{(t)}$ (so that $M = 1$) and only iterate on the Bellman operator once on that state vector (so that $\tau = 1$ and $K = 1$). Still, at iteration t,

the number of random grid points is $N(t)$, which can be made arbitrarily large when we increase the number of iterations. In other words, in contrast to the Rust method, the accuracy of the DP computation in our algorithm automatically increases with iterations.

Another issue that arises in application of the Rust random grid method is that the method assumes that the transition density function $f(s'|s, a, \theta)$ is not degenerate. That is, we cannot use the random grid algorithm if the transition from s to s', given a and θ, is deterministic. It is also well known that the random grid algorithm becomes inaccurate if the transition density has a small variance. In these cases, several versions of polynomial-based, expected value function approximation have been used. Keane and Wolpin (1994) approximated the emax function using polynomials of the deterministic part of the value functions for each choice and state space point. Imai and Keane (2004) used Chebychev polynomials of state variables. It is known that in some cases, global approximation using polynomials can be numerically unstable and exhibit "wiggling." Here, we propose a kernel-based local interpolation approach to emax function approximation. The main problem behind the local approximation has been the computational burden of having a large number of grid points. As pointed out earlier, in our solution-estimation algorithm, we can make the number of grid points arbitrarily large by increasing the total number of iterations, even though the number of grid points per iteration is 1. Thus, if the continuous state variable evolves deterministically, we approximate the emax function $\widehat{E}_{s',\epsilon'}[V(s', \epsilon', \theta)|s, a]$ as follows. Let $K_{h_s}(\cdot)$ be the kernel function with bandwidth h_s for the state variable and $K_{h_\theta}(\cdot)$ for the parameter vector θ. Then

$$\begin{aligned}
&\widehat{E}^{(t)}_{\epsilon'}\left[V(s', \epsilon', \theta)|s, a, \mathcal{H}^{(t-1)}\right] \\
&\quad \equiv \sum_{n=1}^{N(t)} V^{(t-n)}\left(s^{(t-n)}, \epsilon^{(t-n)}, \theta^{*(t-n)}, \mathcal{H}^{(t-n-1)}\right) \\
&\qquad \times \frac{K_{h_s}(s' - s^{(t-n)})K_{h_\theta}(\theta - \theta^{*(t-n)})}{\sum_{k=1}^{N(t)} K_{h_s}(s' - s^{(t-k)})K_{h_\theta}(\theta - \theta^{*(t-k)})}.
\end{aligned}$$

4. EXAMPLES

We estimate a simple, infinite horizon, dynamic discrete choice model of entry and exit, where firms are in a competitive environment.[20] We describe the model in general first and then consider two simplifications. In the general model, there are firm-specific random effects and the state variable evolves

[20]For an estimation exercise based on this model, see Roberts and Tybout (1997).

stochastically. In the first model, which we term the basic model, there is no observed or unobserved heterogeneity. In the second model, in addition, the state variable evolves deterministically.

The firm is either an incumbent (I) or a potential entrant (O). If the incumbent firm chooses to stay, its per period return is

$$R_{I,\mathrm{IN}}(K_t, \epsilon_t, \theta_i) = \alpha_i K_t + \epsilon_{1t},$$

where K_t is the capital of the firm, ϵ_{1t} is the independent and identically distributed (i.i.d.) random shock, and θ_i is the vector of parameters, including firm-specific parameter α_i, which is distributed according to $N(\alpha, \sigma_\alpha)$. When there are no random effects, $\alpha_i = \alpha$ for all i and $\sigma_\alpha = 0$. If the firm chooses to exit, its per period return is

$$R_{I,\mathrm{OUT}}(K_t, \epsilon_t, \theta_i) = \epsilon_{2t}.$$

Similarly, if the potential entrant chooses to enter, its per period return is

$$R_{O,\mathrm{IN}}(K_t, \epsilon_t, \theta_i) = -\delta + \epsilon_{1t},$$

and if it decides to stay out, its per period return is

$$R_{O,\mathrm{OUT}}(K_t, \epsilon_t, \theta_i) = \epsilon_{2t},$$

where ϵ_{2t} is an i.i.d. shock. We assume that the random components of current period returns are i.i.d. and normally distributed, that is, $\epsilon_{lt} \sim N(0, \sigma_{\epsilon_l})$, $l = 1, 2$.

The level of capital K_t evolves as follows. If the incumbent firm stays in, then

$$\ln K_{t+1} = b_0 + b_1 X_i^d + b_2 \ln K_t + u_{t+1},$$

where $u_{t+1} \sim N(0, \sigma_u)$ and X_i^d is a firm-specific characteristic vector observable to the econometrician. In the simple specification without firm-specific heterogeneity, b_1 is set to zero. In the specification where we allow for heterogeneity, we set b_0 to zero. In the specification where we assume the capital transition for the incumbent who stays to be deterministic, we simply set $K_{t+1} = K_t$, in other words, $b_0 = 0$, $b_1 = 0$, $b_2 = 1$, and $\sigma_u = 0$. If the potential entrant enters, then

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

Now consider a firm that is an incumbent at the beginning of period t. Let $V_I(K_t, \epsilon_t, \theta_i)$ be the value function of the incumbent with capital stock K_t and let $V_O(0, \epsilon_t, \theta_i)$ be the value function of the potential entrant, who has capital stock 0. The Bellman equation for the optimal choice of the incumbent is

$$V_I(K_t, \epsilon_t, \theta_i) = \max\{V_{I,\mathrm{IN}}(K_t, \epsilon_t, \theta_i), V_{I,\mathrm{OUT}}(K_t, \epsilon_t, \theta_i)\},$$

where

$$V_{I,\mathrm{IN}}(K_t, \epsilon_t, \theta_i) = R_{I,\mathrm{IN}}(K_t, \epsilon_{1t}, \theta_i) + \beta E_{t+1} V_I(K_{t+1}(K_t, u_{t+1}, \theta_i), \epsilon_{t+1}, \theta_i)$$

is the value of staying in during period t. Similarly,

$$V_{I,\mathrm{OUT}}(K_t, \epsilon_t, \theta_i) = R_{I,\mathrm{OUT}}(K_t, \epsilon_{2t}, \theta_i) + \beta E_{t+1} V_O(0, \epsilon_{t+1}, \theta_i)$$

is the value of exiting during period t. The Bellman equation for the optimal choice of the potential entrant is

$$V_O(0, \epsilon_t, \theta_i) = \max\{V_{O,\mathrm{IN}}(0, \epsilon_t, \theta_i), V_{O,\mathrm{OUT}}(0, \epsilon_t, \theta_i)\},$$

where

$$V_{O,\mathrm{IN}}(0, \epsilon_t, \theta_i) = R_{O,\mathrm{IN}}(0, \epsilon_{1t}, \theta_i) + \beta E_{t+1} V_I(K_{t+1}(0, u_{t+1}, \theta_i), \epsilon_{t+1}, \theta_i)$$

is the value of entering during period t and

$$V_{O,\mathrm{OUT}}(0, \epsilon_t, \theta_i) = R_{O,\mathrm{OUT}}(0, \epsilon_{2t}, \theta_i) + \beta E_{t+1} V_O(0, \epsilon_{t+1}, \theta_i)$$

is the value of staying out during period t. Notice that the capital stock of a potential entrant is always 0.

Notice that if we assume ϵ_{lt} to be extreme value distributed (see Rust (1987) for details on dynamic discrete choice models based on extreme valued error term), then the deterministic component of the value function can be expressed analytically, greatly simplifying the solution of the dynamic programming problem. To allow for correlation of the revenue function across different choices, one can adopt the random coefficient logit specification, where the random coefficient term is added to the per period revenue function. Then, in the basic model, the underlying latent per period revenue would be

$$R_{I,\mathrm{IN}}(K_t, \epsilon, \theta_i, \theta),$$

where ϵ_a, $a \in A$ is assumed to be i.i.d. extreme value distributed and the distribution of θ_i is assumed to be $G(d\theta_i; \theta)$. McFadden and Train (2000) showed that any choice probabilities can be approximated by the random coefficient multinomial logit model. Since θ_i is only introduced to allow for correlation of the revenues across different choices, and not to add serial correlation of choices, we assume θ_i to be i.i.d. over time as well. $R_{I,\mathrm{OUT}}(K_t, \epsilon, \theta_i, \theta)$, $R_{O,\mathrm{IN}}(K_t, \epsilon, \theta_i, \theta)$, and $R_{O,\mathrm{OUT}}(K_t, \epsilon, \theta_i, \theta)$ are similarly defined.

To derive the expected value function at iteration $t+1$, we would first draw $\theta_m \sim G(d\theta_i; \theta)$, $m = 1, \ldots, M$, and then use the analytic formula for the ex-

pected value function proposed by Rust (1987) to evaluate

$$
\begin{aligned}
&E_\epsilon[V_I(K,\epsilon,\theta_m,\theta)]\\
&\quad=\log\bigl[\exp\bigl(R_{I,\mathrm{IN}}(K,0,\theta_m,\theta)+\beta E^{(t)}V_I(K',\epsilon',\theta)\bigr)\\
&\qquad+\exp\bigl(R_{I,\mathrm{OUT}}(K,0,\theta_m,\theta)+\beta E^{(t)}V_O(K',\epsilon',\theta)\bigr)\bigr].
\end{aligned}
$$

These two steps are repeated to derive

$$
E^{(t+1)}[V_I(K,\epsilon,\theta)|K]=\frac{1}{M}\sum_{m=1}^{M}E_\epsilon V_I(K,\epsilon,\theta_m,\theta).
$$

Similarly, $E^{(t+1)}[V_O(K,\epsilon,\theta)|K]$ is computed. This algorithm involves Monte Carlo integration over θ_m and thus is very similar to the DP step of the Bayesian DP algorithm for the probit case. Hence, our algorithm would straightforwardly apply to the case of mixed logit where the mixture distribution is continuous and would result in a computational gain.

We now discuss estimation of the basic model. The parameter vector θ of the model is $(\delta,\alpha,\beta,\sigma_{\epsilon_1},\sigma_{\epsilon_2},\sigma_u,b_0,b_2,b_e)$. The state variables are the capital stock K and the status of the firm $\Gamma\in\{I,O\}$, that is, whether the firm is an incumbent or a potential entrant. We assume that for each firm, we only observe the capital stock, the profit of the firm that stays in, and the entry–exit status over T^d periods. That is, we know $\{K^d_{i,t},R^d_{i,t},\Gamma^d_{i,t}\}^{t=1,T^d}_{i=1,N^d}$, where $R^d_{i,t}\equiv\alpha K^d_{i,t}+\epsilon_{1t}$ if the firm stays in. We assume the prior distribution of all parameters to be diffuse. That is, we set $\pi(\theta)=1$. Below, we explain the estimation steps in detail.

Assume we start with the initial guess of expected value functions being zero, that is,

$$
\widehat{E}^{(0)}_\epsilon\bigl[V_I\bigl(K,\epsilon,\theta^{(0)}\bigr)\bigr]=0,\quad \widehat{E}^{(0)}_\epsilon\bigl[V_O\bigl(0,\epsilon,\theta^{(0)}\bigr)\bigr]=0.
$$

We employ the modified random walk M–H algorithm, where at iteration s the proposal density $q(\theta^{(s)},\theta^{*(s)})$, given iteration s parameter $\theta^{(s)}$, is[21] $\delta^{*(s)}\sim N(\delta^{(s)},\sigma^2_\delta)$, $\alpha^{*(s)}\sim N(\alpha^{(s)},\sigma^2_\alpha)$, $\ln\sigma^{*(s)}_{\epsilon_1}\sim N(\ln\sigma^{(s)}_{\epsilon_1},\sigma^2_{\ln\sigma_{\epsilon_1}})$, $\ln\sigma^{*(s)}_{\epsilon_2}\sim N(\ln\sigma^{(s)}_{\epsilon_2},\sigma^2_{\ln\sigma_{\epsilon_2}})$, $b^{*(s)}_0\sim N(b^{(s)}_0,\sigma^2_{b_0})$, $b^{*(s)}_2\sim N(b^{(s)}_2,\sigma^2_{b_2})$, $b^{*(s)}_e\sim N(b^{(s)}_e,\sigma^2_{b_e})$, and $\ln\sigma^{*(s)}_u\sim N(\ln\sigma^{(s)}_u,\sigma^2_{\ln\sigma_u})$, and when we estimate β, $\beta^{*(s)}\sim N(\beta^{(s)},\sigma^2_\beta)$. Given the parameters of iteration s, $\theta^{(s)}=(\delta^{(s)},\alpha^{(s)},\sigma^{(s)}_{\epsilon_1},\sigma^{(s)}_{\epsilon_2},b^{(s)}_0,b^{(s)}_2,b^{(s)}_e,\sigma^{(s)}_u)$, we draw the candidate parameter $\theta^{*(s)}$ from these normal densities.

[21]The standard errors of the innovations of the random walk M–H are all set to be 0.004, except for β, for which it is set to be 0.001.

Expected Value Function Iteration Step

We update the expected value function for parameter $\theta^{(s)}$ and $\theta^{*(s)}$. First, we derive $E_{\epsilon}^{(s)}[V_I(K, \epsilon, \theta|\mathcal{H}^{(s)})]$ for $\theta = \theta^{(s)}$ and $\theta^{*(s)}$, using the Gaussian kernel[22] $K_h(\theta - \theta') = (2\pi)^{-L/2} \prod_{j=1}^{J} h_j^{-1} \exp[-\frac{1}{2}((\theta_j - \theta_j')/h_j)^2]$, as follows:

$$\begin{aligned}
&\widehat{E}_{\epsilon'}^{(s)}\big[V_I(K, \epsilon', \theta)|\mathcal{H}^{(s-1)}\big] \\
&\quad \equiv \sum_{n=1}^{N(s)} \left[\frac{1}{M_\epsilon} \sum_{j=1}^{M_\epsilon} V_I^{(s-n)}\big(K, \epsilon^{j,(s-n)}, \theta^{*(s-n)}, \mathcal{H}^{(s-n-1)}\big) \right] \\
&\qquad \times \frac{K_h(\theta - \theta^{*(s-n)}) I_{L\theta}(\theta^{*(s-n)})}{\sum_{k=1}^{N(s)} K_h(\theta - \theta^{*(s-k)}) I_{L\theta}(\theta^{*(s-k)})}, \\
&\widehat{E}_{\epsilon'}^{(s)}\big[V_O(0, \epsilon', \theta)|\mathcal{H}^{(s-1)}\big] \\
&\quad \equiv \sum_{n=1}^{N(s)} \left[\frac{1}{M_\epsilon} \sum_{j=1}^{M_\epsilon} V_O^{(s-n)}\big(0, \epsilon^{j,(s-n)}, \theta^{*(s-n)}, \mathcal{H}^{(s-n-1)}\big) \right] \\
&\qquad \times \frac{K_h(\theta - \theta^{*(s-n)}) I_{L\theta}(\theta^{*(s-n)})}{\sum_{k=1}^{N(s)} K_h(\theta - \theta^{*(s-k)}) I_{L\theta}(\theta^{*(s-k)})}.
\end{aligned}$$

The expected value function is updated by taking the weighted average over L of the past $N(s)$ iterations where the parameter vector $\theta^{*(l)}$ was closest to θ (we denote this as $I_{L\theta}(\theta^{*(l)}) = 1$), where L is set to be 2,000 and $N(s)$ increases to 3,000. As discussed before, in principle, only one simulation of ϵ is needed during each solution-estimation iteration. But that requires the number of past iterations for averaging, that is, requires $N(s)$ to be large, which adds to the computational burden. Instead, in our example, we draw ϵ ten times and take an average. Hence, when we derive the expected value function, instead of averaging past value functions, we average over past average value functions, that is, $(1/M_\epsilon) \sum_{m=1}^{M_\epsilon} V_I(K, \epsilon_m^{(j)}, \theta^{(j)})$, where $M_\epsilon = 10$. This obviously increases the accuracy per iteration and reduces the need to have a large $N(s)$.

It is important to notice that as the algorithm proceeds, and t and $N(t)$ become sufficiently large, the computational burden of our nonparametric approximation of the expected value functions could become more than that of solving the DP problem.[23] In our examples, we have set the number of MCMC iterations and the maximum of $N(t)$ arbitrarily, but at these values, the al-

[22] Kernel bandwidth h_j is set to be 0.02 for all $j = 1, 2, \ldots, J$.

[23] We thank an anonymous referee for emphasizing this point.

gorithm is computationally much superior to the full-solution-based MCMC, without experiencing any noticeable loss in accuracy in posterior distribution estimation. To avoid this arbitrariness, and still avoid the computational burden of large $N(t)$, one could initially set $N(t)$ at a fixed number $\widehat{N}$ and occasionally conduct one-step Bellman updates using the expected value $E^{(t)}V(s')$ computed using the past values as an initial value for the DP iteration. If the newly iterated expected value function is sufficiently close to $E^{(t)}V(s')$, then there is no need for an increase in $\widehat{N}$. Norets (2008) considered other ways to combine the Bayesian DP algorithm with the standard DP steps to gain further computational efficiency.

To further integrate the value function over the capital shock u, we can use the Rust random grid integration method which uses a fixed grid. Since the state space has only one dimension, we use equally spaced K_m, $m = 1, \ldots, M$, capital grid points and apply equation (7). That is, for the incumbent,

$$\begin{aligned}
&\widehat{E}^{(s)}\big[V_I(K'(K_{i,t}^d, u, \theta), \epsilon, \theta)|K_{i,t}^d, \mathcal{H}^{(s-1)}\big] \\
&\quad = \sum_m \widehat{E}_\epsilon^{(s)}\big[V_I(K_m, \epsilon, \theta)|\mathcal{H}^{(s-1)}\big] \\
&\qquad \times \frac{[K_m\sigma_u]^{-1}\exp(-(\ln K_m - b_0 - b_1 \ln K_{i,t}^d)^2/(2\sigma_u^2))}{\sum_{l=1}^M [K_l\sigma_u]^{-1}\exp(-(\ln K_l - b_0 - b_1 \ln K_{i,t}^d)^2/(2\sigma_u^2))}
\end{aligned}$$

for $\theta = \theta^{(s)}, \theta^{*(s)}$. For the entrant,

$$\begin{aligned}
&\widehat{E}^{(s)}\big[V_I\big(K'\big(0, u, \theta^{(s)}\big), \epsilon, \theta\big)\big|\mathcal{H}^{(s-1)}\big] \\
&\quad = \sum_m \widehat{E}_\epsilon^{(s)}\big[V_I(K_m, \epsilon, \theta)|\mathcal{H}^{(s-1)}\big] \\
&\qquad \times \frac{[K_m\sigma_u]^{-1}\exp(-(\ln K_m - b_e)^2/(2\sigma_u^2))}{\sum_{l=1}^M [K_l\sigma_u]^{-1}\exp(-(\ln K_l - b_e)^2/(2\sigma_u^2))}
\end{aligned}$$

for $\theta = \theta^{(s)}, \theta^{*(s)}$.

Modified DP Step

We draw $\epsilon_l^{j,(s)} \sim N(0, \sigma_{\epsilon_l})$, $l = 1, 2$; $j = 1, \ldots, M_\epsilon$, and compute

$$\begin{aligned}
&V_{I,\mathrm{IN}}\big(K_m, \epsilon^{j,(s)}, \theta^{*(s)}, \mathcal{H}^{(s-1)}\big) \\
&\quad = R_{I,\mathrm{IN}}\big(K_m, \epsilon_1^{j,(s)}, \theta^{*(s)}\big) \\
&\qquad + \beta\widehat{E}^{(s)}\big[V_I\big(K'\big(K_m, u_{t+1}, \theta^{*(s)}\big), \epsilon, \theta^{*(s)}\big)\big|K_m, \mathcal{H}^{(s-1)}\big],
\end{aligned}$$

$$V_{I,\mathrm{OUT}}\big(K_m, \boldsymbol{\epsilon}^{j,(s)}, \theta^{*(s)}, \mathcal{H}^{(s-1)}\big)$$
$$= R_{I,\mathrm{OUT}}\big(K_m, \boldsymbol{\epsilon}_2^{j,(s)}, \theta^{*(s)}\big) + \beta\widehat{E}^{(s)}\big[V_O\big(0, \boldsymbol{\epsilon}, \theta^{*(s)}\big)\big|\mathcal{H}^{(s-1)}\big],$$
$$V_I\big(K_m, \boldsymbol{\epsilon}^{j,(s)}, \theta^{*(s)}, \mathcal{H}^{(s-1)}\big)$$
$$= \max\big\{V_{I,\mathrm{IN}}\big(K_m, \boldsymbol{\epsilon}^{j,(s)}, \theta^{*(s)}, \mathcal{H}^{(s-1)}\big), V_{I,\mathrm{OUT}}\big(K_m, \boldsymbol{\epsilon}^{j,(s)}, \theta^{*(s)}, \mathcal{H}^{(s-1)}\big)\big\},$$

to derive

$$\frac{1}{M_\epsilon}\sum_{j=1}^{M_\epsilon} V_I^{(s)}\big(K_m, \boldsymbol{\epsilon}^{j,(s)}, \theta^{*(s)}, \mathcal{H}^{(s-1)}\big)$$

and

$$\frac{1}{M_\epsilon}\sum_{j=1}^{M_\epsilon} V_O^{(s)}\big(0, \boldsymbol{\epsilon}^{j,(s)}, \theta^{*(s)}, \mathcal{H}^{(s-1)}\big).$$

Modified M–H Step

We draw the new parameter vector $\theta^{(s+1)}$ from the posterior distribution. Let

$$\mathbf{I}_i = [I_{i,1}^d(\mathrm{IN}), \ldots, I_{i,t}^d(\mathrm{IN}), \ldots, I_{i,T}^d(\mathrm{IN})],$$

where $I_{i,t}^d(\mathrm{IN}) = 1$ if the firm either enters or decides to stay in, and $=0$ otherwise. Similarly, we use $\mathbf{K}_i$ and $\mathbf{R}_i$ to denote vectors of $K_{i,t}^d$ and $R_{i,t}^d$. The likelihood increment for firm i at time t is (suppressing the superscript $(s-1)$ on $\mathcal{H}$ and denoting $\phi(\cdot)$ to be the standard normal density for convenience)

$$L_i(\mathbf{I}_i, \mathbf{K}_i, \mathbf{R}_i|\theta)$$
$$= \Pr\big[\boldsymbol{\epsilon}_{2t} \le R_{it}^d + \beta\big\{\widehat{E}^{(s)}\big[V_I(K'(K_{it}^d, u, \theta), \boldsymbol{\epsilon}, \theta)|K_{it}^d, \mathcal{H}\big]$$
$$- \widehat{E}^{(s)}[V_O(0, \boldsymbol{\epsilon}, \theta)|\mathcal{H}]\big\}\big]$$
$$\times \frac{1}{\sigma_{\epsilon_1}}\phi\left(\frac{R_{it}^d - \alpha K_{it}^d}{\sigma_{\epsilon_1}}\right)\frac{1}{K_{i,t+1}^d\sigma_u}\phi\left(\frac{\ln K_{i,t+1}^d - b_1 - b_2\ln K_{it}^d}{\sigma_u}\right)$$
$$\times I_{it}^d(\mathrm{IN})I_{i,t+1}^d(\mathrm{IN})$$
$$+ \Pr\big[\boldsymbol{\epsilon}_{2t} - \boldsymbol{\epsilon}_{1t} > \alpha K_{it}^d + \beta\big\{\widehat{E}^{(s)}\big[V_I\big(K'\big(K_{it}^d, u, \theta^{(s)}\big), \boldsymbol{\epsilon}, \theta\big)|K_{it}^d, \mathcal{H}\big]$$
$$- \widehat{E}^{(s)}[V_O(0, \boldsymbol{\epsilon}, \theta)|\mathcal{H}]\big\}\big]$$
$$\times I_{it}^d(\mathrm{IN})(1 - I_{i,t+1}^d(\mathrm{IN}))$$
$$+ \Pr\big[\boldsymbol{\epsilon}_{2t} - \boldsymbol{\epsilon}_{1t} \le -\delta + \beta\big\{\widehat{E}^{(s)}\big[V_I\big(K'\big(0, u, \theta^{(s)}\big), \boldsymbol{\epsilon}, \theta\big)|\mathcal{H}\big]$$
$$- \widehat{E}^{(s)}[V_O(0, \boldsymbol{\epsilon}, \theta)|\mathcal{H}]\big\}\big]$$

$$\times \frac{1}{K_{i,t+1}^d \sigma_u} \phi\left(\frac{\ln K_{i,t+1}^d - b_e}{\sigma_u}\right)(1 - I_{it}^d(\mathrm{IN}))I_{i,t+1}^d(\mathrm{IN})$$
$$+ \Pr\big[\epsilon_{2t} - \epsilon_{1t} > -\delta + \beta\big\{\widehat{E}^{(s)}\big[V_I\big(K'\big(0, u, \theta^{(s)}\big), \epsilon, \theta\big)|\mathcal{H}\big]$$
$$- \widehat{E}^{(s)}[V_O(0, \epsilon, \theta)|\mathcal{H}]\big\}\big]$$
$$\times (1 - I_{it}^d(\mathrm{IN}))(1 - I_{i,t+1}^d(\mathrm{IN})).$$

The algorithm sets $\theta^{(s+1)} = \theta^{*(s)}$ with probability $\lambda(\theta^{(s)}, \theta^{*(s)}|\mathcal{H}^{(s-1)})$, where the random walk proposal density satisfies $q(\theta^{*(s)}, \theta^{(s)}) = q(\theta^{(s)}, \theta^{*(s)})$. Thus,

$$\lambda\big(\theta^{(s)}, \theta^{*(s)}|\mathcal{H}^{(s-1)}\big)$$
$$= \min\left\{\frac{\pi(\theta^{*(s)})\prod_i L_i(\mathbf{I}_i, \mathbf{K}_i, \mathbf{R}_i|\theta^{*(s)})q(\theta^{*(s)}, \theta^{(s)})}{\pi(\theta^{(s)})\prod_i L_i(\mathbf{I}_i, \mathbf{K}_i, \mathbf{R}_i|\theta^{(s)})q(\theta^{(s)}, \theta^{*(s)})}, 1\right\}$$
$$= \min\left\{\frac{\pi(\theta^{*(s)})\prod_i L_i(\mathbf{I}_i, \mathbf{K}_i, \mathbf{R}_i|\theta^{*(s)})}{\pi(\theta^{(s)})\prod_i L_i(\mathbf{I}_i, \mathbf{K}_i, \mathbf{R}_i|\theta^{(s)})}, 1\right\}.$$

Notice that if the firm stays out or exits, then its future capital stock is zero. Therefore, no averaging over capital grid points is required to derive the emax function, which is simply $E_\epsilon^{(s)}[V_O(0, \epsilon, \theta)|\mathcal{H}^{(s-1)}]$.

In the next section, we present the results of several Monte Carlo studies we conducted using our Bayesian DP algorithm. The first experiment is for the model that incorporates observed and unobserved heterogeneity; the second incorporates deterministic capital transition.[24]

5. SIMULATION AND ESTIMATION

Denote the true values of θ by θ^0, that is, for the basic model, $\theta^0 = (\delta^0, \sigma_{\epsilon_1}^0, \sigma_{\epsilon_2}^0, \sigma_u^0, \alpha^0, b_0^0, b_2^0, b_e^0, \beta^0)$. We set them as $\delta^0 = 0.4$, $\sigma_{\epsilon_1}^0 = 0.3$, $\sigma_{\epsilon_2}^0 = 0.3$, $\sigma_u^0 = 0.4$, $\alpha^0 = 0.1$, $b_0^0 = 0.0$, $b_2^0 = 0.4$, $b_e^0 = 0.5$, and $\beta^0 = 0.98$.

We first solve the DP problem numerically using the conventional full-solution method described earlier in detail. Next, we generate artificial data based on this DP solution. All estimation exercises are done on a 2.8 GHz Pentium 4 Linux workstation. For data generation, we solved the DP problem, where during each iteration we set capital grid points $M_K = 200$ to be equally

[24]The results of the experiment, where we estimate the basic model (without heterogeneity), are shown in Appendix A, Table AI.

spaced between 0 and $\overline{K}$, which we set to be 5.0. We draw $M_\epsilon = 1{,}000$ revenue shocks, ϵ^m, $m = 1, \ldots, M_\epsilon$, and calculate the value function $V(K_i, \epsilon^m)$. Then we compute the expected value function as their sample average.

$$\widehat{E}_\epsilon[V(K_i, \epsilon)] = \sum_{m=1}^{M} V(K_i, \epsilon^m)/1{,}000.$$

We simulate artificial data of capital stock, profit, and entry–exit choice sequences $\{K_{i,t}^d, R_{i,t}^d, I_{i,t}^d\}_{i=1,t=1}^{N^d,T^d}$ using this DP solution. We then estimate the model using the simulated data with our Bayesian DP routine. We either set the discount factor at the true value $\beta^0 = 0.98$ or estimate it, but with its prior being $\pi(\beta) \sim N(\overline{\beta}, \sigma_\beta)$ if $\beta \leq \overline{\beta}$ and $\pi(\beta) = 0$ otherwise, where $\sigma_\beta = 0.2$ and $\overline{\beta} = 0.995$. We imposed restrictions on the prior to guarantee that $\beta < 1$ so that the Bellman operator is a contraction mapping.[25]

Next, we report results of the experiments mentioned above.[26]

5.1. *Experiment 1: Random Effects*

We now report estimation results of a model that includes observed and unobserved heterogeneity. For data generation, we assume that the profit coefficient for each firm i, α_i, is distributed normally with mean $\alpha = 0.2$ and standard error $\sigma_\alpha = 0.1$. For the transition of capital, we simulate X_i^d from $N(0.0, 1.0)$ and set $b_1 = 0.1$. All other parameters are set at true values given by the vector θ^0.

Notice that if we use the conventional simulated ML estimation to estimate the model, for each firm i, we need to draw α_i many times, say M_α times, and for each draw, we need to solve the DP problem. If the number of firms in the data is N^d, then for a single simulated likelihood evaluation, we need to solve the DP problem $N^d M_\alpha$ times. This process is computationally demanding and most researchers use only a finite number of types, typically less than 10, as an approximation of the observed heterogeneity and the random effect.[27] Since in our Bayesian DP estimation exercise, the computational burden of estimating the dynamic model is similar to that of a static model, we can easily accommodate random effects estimation.

[25]Notice that in DDC models, the discount factor is nonparametrically unidentified (see Rust (1994) and Magnac and Thesmar (2002)). Hence, estimation of the discount factor relies on functional form assumptions.

[26]The results reported in Appendix A, Table AI, show that the full-solution-based ML outperforms the Bayesian DP in the basic model (without heterogeneity).

[27]The only exceptions are economists who have access to supercomputers or large PC clusters. Bound, Stinebrickner, and Waidmann (2007) used interpolation methods to evaluate the expected value functions where the unobserved health status was continuous. They used PC clusters for their estimation.

In contrast to the solution-estimation algorithm of the basic model, we iterate the Bellman operator once for each firm i separately. Let $\theta_{-\alpha}$ be the parameter vector except for the random effects term α_i. Then, for any given K, we derive

$$\begin{aligned}\widehat{E}_{\epsilon}^{(s)}\left[V_{\Gamma}(K,\epsilon,\theta_{-\alpha},\alpha_i)|\mathcal{H}^{(s-1)}\right] &= \sum_{n=1}^{N(s)}\left[\frac{1}{M_{\epsilon}}\sum_{l=1}^{M_{\epsilon}} V_{\Gamma}^{(s-n)}\left(K,\epsilon_l^{(s-n)},\theta_{-\alpha}^{*(s-n)},\alpha_i^{*(s-n)},\mathcal{H}^{(s-n-1)}\right)\right] \\ &\quad\times\frac{K_h(\theta_{-\alpha}-\theta_{-\alpha}^{*(s-n)})K_h(\alpha_i-\alpha_i^{*(s-n)})}{\sum_{n=1}^{N(s)}K_h(\theta_{-\alpha}-\theta_{-\alpha}^{*(s-n)})K_h(\alpha_i-\alpha_i^{*(s-n)})}.\end{aligned}$$

As pointed out by Heckman (1981) and others, the missing initial state vector is likely to be correlated with the unobserved heterogeneity α_i, which would result in bias in the parameter estimates. To deal with this problem, for each firm i, given parameters $(\theta_{-\alpha},\alpha_i)$, we simulate the model for 100 initial periods to derive the initial capital and the initial status of the firm. Then we proceed to construct the likelihood increment for firm i.

We set $N(s)$ to go up to 1,000 iterations. The one-step Bellman operator for each firm i is the part where we have an increase in computational burden, but it turns out that the additional burden is far lighter than that of computing the fixed point of the Bellman operator for each firm M_{α} times to integrate out the random effects α_i, as would be done in the simulated ML estimation strategy.

We set the sample size to be 100 firms for 100 periods. All priors are diffuse. We set the initial guess of the expected value function to be 0. The Bayesian DP iteration was conducted 10,000 times. We only use the draws from the 5,001st iteration up to the 10,000th iteration to derive the posterior means and standard deviations. We conducted 50 replications in this experiment. Table I reports the average of the posterior means ($\overline{\text{PM}}$), the posterior standard deviations ($\overline{\text{PSD}}$), and the standard deviation of the posterior means (sd(PM)) for the 50 replications. There are three sets of results. To obtain the first and second set of results (Bayesian DP 1, Bayesian DP 2, respectively), we set the initial parameter values equal to the true ones. We fix the discount factor β in Bayesian DP 1, while we estimate it in Bayesian DP 2. To obtain the third set of results (Bayesian DP 3), we fix β at the true value and set the other initial parameter values to be half of the true ones. All these results show that the posterior means are very close to—indeed within 1 standard deviation of—the true parameter values. In particular, the results presented in Bayesian DP 3

TABLE I

POSTERIOR MEANS AND STANDARD DEVIATIONS[a]

	Bayesian DP 1			Bayesian DP 2			Bayesian DP 3			
	$\overline{\text{PM}}$	$\overline{\text{PSD}}$	sd(PM)	$\overline{\text{PM}}$	$\overline{\text{PSD}}$	sd(PM)	$\overline{\text{PM}}$	$\overline{\text{PSD}}$	sd(PM)	True
δ	0.4005	0.0162	0.0224	0.3957	0.0165	0.0239	0.4011	0.0163	0.0226	0.4
α	0.2013	0.0104	0.0104	0.2012	0.0105	0.0103	0.2013	0.0104	0.0105	0.2
σ_α	0.1006	0.00736	0.00655	0.1005	0.00736	0.00651	0.1006	0.00735	0.00656	0.1
σ_{ϵ_1}	0.3005	0.00284	0.00261	0.3006	0.00292	0.00271	0.3005	0.00286	0.0264	0.3
σ_{ϵ_2}	0.3034	0.0113	0.0184	0.2995	0.0124	0.0208	0.3036	0.0116	0.0190	0.3
b_1	0.0993	0.00481	0.00425	0.0994	0.00490	0.00434	0.0993	0.00482	0.00420	0.1
b_2	0.3975	0.00943	0.00941	0.3977	0.00952	0.00924	0.3975	0.00943	0.00939	0.4
b_e	0.4954	0.0125	0.0146	0.4954	0.0127	0.0149	0.4954	0.0126	0.0146	0.5
σ_u	0.4014	0.00314	0.00316	0.4014	0.00318	0.00312	0.4014	0.00314	0.00318	0.4
β				0.9689	0.0109	0.0170				0.98
CPU		4 h 0 min			3 h 58 min			4 h 0 min		

[a] $\overline{\text{PM}}$ is the average of the posterior means across 50 replications; $\overline{\text{PSD}}$ is the average of the posterior standard deviations across 50 replications; sd(PM) is the standard deviation of the posterior means across 50 replications.

confirm the robustness of the Bayesian DP algorithm to the initial parameter values.[28]

On the other hand, there is a fairly large bias in the parameters estimated by simulated ML with $M_\alpha = 100$. The point estimate of entry cost parameter $\delta = 0.3795$, the mean of profit coefficient $\alpha = 0.1701$ and its standard error $\sigma_\alpha = 0.09326$, and the standard error of the choice shock $\sigma_{\epsilon_2} = 0.2805$ are all downwardly biased, and except for σ_α, the magnitude of the bias is larger than the standard error.[29] The downward bias seems to be especially large for α, which leads us to conclude that the simulation size of $M_\alpha = 100$ is not enough to integrate out the unobserved heterogeneity sufficiently accurately. The CPU time required for the Bayesian DP algorithm is about 4 hours, whereas for the full-solution-based Bayesian MCMC estimation, we needed about 31 hours, and for the full-solution-based ML estimation, 21 hours. That is, the Bayesian DP is about 8 times as fast as the full-solution-based Bayesian MCMC algorithm and about 5 times as fast as the simulated ML algorithm.[30]

[28] The standard deviations of the posterior means across the 50 replications sd(PM) reflect the data uncertainty of the 50 simulation-estimation exercises. Note that they are very close to the mean of the posterior standard deviations $\overline{\text{PSD}}$. Thus, the standard deviation of the Bayesian DP draws captures the data uncertainty well.

[29] These values are the averages of 10 simulation-estimation exercises. The detailed results are shown in Tables AIII and AIV of Appendix A.

[30] When we solve for the DP problem, both for the full-solution-based Bayesian estimation and the simulated ML estimation (details in Appendix A), we set $M_\epsilon = 100$. If we were to set $M_\epsilon = 1{,}000$, then a single Newton iteration would take about 4 hours and 20 minutes, which is about the same CPU time as required for the entire Bayesian DP algorithm.

We also tried to reduce the computational time for the full-solution-based ML algorithm by reducing the number of draws for α_i from 100 to 20. Then the CPU time reduced to 8 hours and 43 minutes, which is still about twice as much time as required for the Bayesian DP algorithm. However, the point estimate of α is 0.145, having larger downward bias than the estimate with $M_\alpha = 100$.

If we were to try to reduce the bias of the full-solution-based ML method by increasing the simulation size of unobserved heterogeneity from $M_\alpha = 100$ to, say $M_\alpha = 1{,}000$, then the CPU time would be at least 200 hours. We also tried the ML estimation where the simulation size for ϵ draws is reduced from 100 to 20 while keeping $M_\alpha = 100$. The parameter estimates and their standard errors are very similar to those of the 100 ϵ draws. However, the total CPU time of the ML estimation with 20 ϵ draws is 18 hours and 15 minutes, hardly different from that of the original 100 ϵ draws.

Another estimation strategy for the simulated ML could be to expand the state variables of the DP problem to include both X and α_i. Then we have to assign grid points for the three-dimensional state space points (K, X, α_i). If we assign 100 grid points per dimension, then we end up having 10,000 times more grid points than before. Hence, the overall computational burden would be quite similar to the previous simulated ML estimation strategy. Thus, our Bayesian DP algorithm outperforms the full-solution-based conventional methods when the model includes observed and unobserved heterogeneity.

Furthermore, the computational advantage of the Bayesian DP algorithm over the conventional full-solution-based Bayesian or ML estimation grows as the discount factor becomes closer to 1. Ching, Imai, Ishihara, and Jain (2009) showed that while the time required to estimate the model under full-solution-based conventional methods becomes twice as much when β changes from 0.6 to 0.8, and becomes 20 times as much when β changes from 0.6 to 0.98, there is no difference in the overall computational performance of the Bayesian DP algorithm. This is because in full-solution-based algorithms, the closer the discount factor is to 1, the more time is required for the DP algorithm to converge. However, in our algorithm, DP iteration is done only once during each parameter estimation step, regardless of the value of the discount factor.

5.2. *Experiment 2: Deterministic Transition*

At iteration t we use $K_1^{(t)}, \ldots, K_{M_K}^{(t)}$ as grid points. We set $M_K = 10$, hence the total number of grid points increases over iterations up to $M_K \times N(t) = 10 \times 1{,}000 = 10{,}000$.

The formula for the expected value function for the incumbent who stays in is

$$\begin{aligned}&\widehat{E}^{(t)}\left[V_I(K, \epsilon', \theta)|\mathcal{H}^{(t-n-1)}\right]\\&\quad\equiv \sum_{n=1}^{N(t)}\sum_{m=1}^{M_K}\left[\frac{1}{M_\epsilon}\sum_{j=1}^{M_\epsilon} V_I^{(t-n)}\left(K_m^{(t-n)}, \epsilon_j^{(t-n)}, \theta^{*(t-n)}, \mathcal{H}^{(t-n-1)}\right)\right]\end{aligned}$$

TABLE II

POSTERIOR MEANS AND STANDARD DEVIATIONS[a]

Parameter	$\overline{\text{PM}}$	$\overline{\text{PSD}}$	sd(PM)	True Value
δ	0.1905	0.0124	0.0175	0.2
α	0.1019	0.00477	0.00428	0.1
σ_{ϵ_1}	0.3980	0.00506	0.00532	0.4
σ_{ϵ_2}	0.3961	0.0126	0.0187	0.4
b_1	0.2004	0.00466	0.00401	0.2
σ_u	0.2000	0.00322	0.00366	0.2
Sample size		10,000		
CPU time		48 min 3 s		

[a] $\overline{\text{PM}}$ is the average of the posterior means across 50 replications; $\overline{\text{PSD}}$ is the average of the posterior standard deviations across 50 replications; sd(PM) is the standard deviation of the posterior means across 50 replications.

$$\times \frac{K_{h_K}(K - K_m^{(t-n)})K_{h_\theta}(\theta - \theta^{*(t-n)})}{\sum_{k=1}^{N(t)}\sum_{m=1}^{M_K} K_{h_K}(K - K_m^{(t-k)})K_{h_\theta}(\theta - \theta^{*(t-k)})},$$

where K_{h_K} is the kernel for the capital stock with bandwidth h_K. The formulas for the expected value functions for the incumbent who exits and the potential entrant who stays out or enters are the same as those in the basic model.

Table II shows the posterior means and the posterior standard deviations of the parameter estimates. They are the average of 50 replications of the simulation-estimation exercises. We can see that the parameter estimates are close to the true values. We can also see that the posterior standard deviations closely reflect the data uncertainty. The entire exercise took about 48 minutes.

6. CONCLUSION

We have proposed a Bayesian estimation algorithm where the infinite horizon DP problem is solved and parameters are estimated at the same time. This dramatically increases the speed of estimation, particularly in models with observed and unobserved heterogeneity. We have demonstrated the effectiveness of our approach by estimating a simple infinite horizon, dynamic model of entry–exit choice. We find that the computational time required for estimating this dynamic model is in line with the time required for Bayesian estimation of static models. The additional computational cost of our algorithm is the cost of using information obtained in past iterations. Our Monte Carlo experiments show that the more complex a model becomes, the smaller is this cost relative to the cost of computing the full solution.

We have also shown that our algorithm may help reduce the computational burden when the dimension of the state space is high. As is well known, the

computational burden increases exponentially with an increase in the dimension of the state space. In our algorithm, even though at each iteration, the number of state space points on which we calculate the expected value function is small, the total number of "effective" state space points over the entire solution-estimation iteration grows with the number of Bayesian DP iterations. This number can be made arbitrarily large without much additional computational cost, and it is the total number of effective state space points that determines accuracy. This explains why our nonparametric approximation of the expected value function works well under the assumption of a continuous state space with deterministic transition function of the state variable. In this case, as is discussed in the main body of the paper, the Rust random grid method may face computational difficulties.

It is worth mentioning that since we are locally approximating the expected value function nonparametrically, as we increase the number of parameters, we may face the curse of dimensionality in terms of the number of parameters to be estimated. So far, in our examples, this issue does not seem to have made a difference. The reason could be that most dynamic models specify per period return function and transition functions to be smooth and well behaved. Hence, we know in advance that the value functions we need to approximate are smooth and, hence, are well suited for nonparametric approximation. Furthermore, the simulation exercises show that with a reasonably large sample size, the MCMC simulations are tightly centered around the posterior mean. Hence, the actual multidimensional area where we need to apply nonparametric approximation is small. But in empirical exercises that involve many more parameters, one probably needs to adopt an iterative MCMC strategy where only up to four or five parameters are moved at once, which is also commonly done in conventional ML estimation.

REFERENCES

ACKERBERG, D. (2004): "A New Use of Importance Sampling to Reduce Computational Burden in Simulation Estimation," Unpublished Manuscript, University of Arizona. [1866]

AGUIRREGABIRIA, V., AND P. MIRA (2002): "Swapping the Nested Fixed Point Algorithm: A Class of Estimators for Discrete Markov Decision Models," *Econometrica*, 70, 1519–1543. [1866]

——— (2007): "Sequential Estimation of Dynamic Discrete Games," *Econometrica*, 75, 1–53. [1866]

ALBERT, J., AND S. CHIB (1993): "Bayesian Analysis of Binary and Polychotomous Data," *Journal of the American Statistical Association*, 88, 669–679. [1880]

ARCIDIACONO, P., AND J. B. JONES (2003): "Finite Mixture Distributions, Sequential Likelihood and the EM Algorithm," *Econometrica*, 71, 933–946. [1866]

ARCIDIACONO, P., AND R. MILLER (2009): "CCP Estimation of Dynamic Discrete Choice Models With Unobserved Heterogeneity," Unpublished Manuscript, Duke University. [1866]

BERTSEKAS, D. P., AND J. TSITSIKLIS (1996): *Neuro-Dynamic Programming*. Cambridge, MA: Athena Scientific Press. [1879,1880]

BOUND, J., T. STINEBRICKNER, AND T. WAIDMANN (2007): "Health, Economic Resources and the Work Decisions of Older Men," Working Paper 13657, NBER. [1892]

BROWN, M., AND C. FLINN (2006): "Investment in Child Quality Over Marital States," Unpublished Manuscript, University of Wisconsin. [1868]

CHIB, S., AND E. GREENBERG (1996): "Markov Chain Monte Carlo Simulation Methods in Econometrics," *Econometric Theory*, 12, 409–431. [1880]

CHING, A., S. IMAI, M. ISHIHARA, AND N. JAIN (2009): "A Practitioner's Guide to Bayesian Estimation of Discrete Choice Dynamic Programming Models," Working Paper, Rotman School of Management, University of Toronto. [1895]

ERDEM, T., AND M. P. KEANE (1996): "Decision Making Under Uncertainty: Capturing Dynamic Brand Choice Processes in Turbulent Consumer Goods Markets," *Marketing Science*, 15, 1–20. [1865]

FERRALL, C. (2005): "Solving Finite Mixture Models: Efficient Computation in Economics Under Serial and Parallel Execution," *Computational Economics*, 25, 343–379. [1866]

GEWEKE, J., AND M. P. KEANE (2000): "Bayesian Inference for Dynamic Discrete Choice Models Without the Need for Dynamic Programming," in *Simulation Based Inference and Econometrics: Methods and Applications*, ed. by R. Mariano, T. Schuermann, and M. J. Weeks. Cambridge, MA: Cambridge University Press. [1868]

HECKMAN, J. J. (1981): "The Incidental Parameters Problem and the Problem of Initial Conditions in Estimating a Discrete Time-Discrete Data Stochastic Process," in *Structural Analysis of Discrete Data With Econometric Applications*, ed. by C. F. Manski and D. McFadden. Cambridge, MA: MIT Press, 179–195. [1893]

HECKMAN, J. J., AND B. SINGER (1984): "A Method for Minimizing the Impact of Distributional Assumptions in Econometric Models for Duration Data," *Econometrica*, 52, 271–320. [1868]

HOTZ, J. V., AND R. MILLER (1993): "Conditional Choice Probabilities and the Estimation of Dynamic Models," *Review of Economic Studies*, 60, 497–529. [1866]

HOTZ, J. V., R. A. MILLER, S. SANDERS, AND J. SMITH (1994): "A Simulation Estimator for Dynamic Models of Discrete Choice," *Review of Economic Studies*, 61, 265–289. [1866]

HOUSER, D. (2003): "Bayesian Analysis of Dynamic Stochastic Model of Labor Supply and Savings," *Journal of Econometrics*, 113, 289–335. [1868]

IMAI, S., AND M. P. KEANE (2004): "Intertemporal Labor Supply and Human Capital Accumulation," *International Economic Review*, 45, 601–641. [1884]

IMAI, S., AND K. KRISHNA (2004): "Employment, Deterrence and Crime in a Dynamic Model," *International Economic Review*, 45, 845–872. [1865]

IMAI, S., N. JAIN, AND A. CHING (2009): "Supplement to 'Bayesian Estimation of Dynamic Discrete Choice Models'," *Econometrica Supplemental Material*, 77, http://www.econometricsociety.org/ecta/Supmat/5658_proofs.pdf; http://www.econometricsociety.org/ecta/Supmat/5658_data and programs.zip. [1869]

KEANE, M. P., AND K. I. WOLPIN (1994): "The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence," *The Review of Economics and Statistics*, 76, 648–672. [1884]

——— (1997): "The Career Decisions of Young Men," *Journal of Political Economy*, 105, 473–521. [1865]

LANCASTER, T. (1997): "Exact Structural Inference in Optimal Job Search Models," *Journal of Business & Economic Statistics*, 15, 165–179. [1868]

MAGNAC, T., AND D. THESMAR (2002): "Identifying Dynamic Decision Processes," *Econometrica*, 70, 801–816. [1892]

MCCULLOCH, R., AND P. ROSSI (1994): "An Exact Likelihood Analysis of the Multinomial Probit Model," *Journal of Econometrics*, 64, 207–240. [1868,1876,1880]

MCFADDEN, D., AND K. TRAIN (2000): "Mixed MNL Models for Discrete Response," *Journal of Applied Econometrics*, 15, 447–470. [1886]

NORETS, A. (2007): "Inference in Dynamic Discrete Choice Models With Serially Correlated Unobserved State Variables," Unpublished Manuscript, University of Iowa. [1868,1875,1876, 1881]

——— (2008): "Implementation of Bayesian Inference in Dynamic Discrete Choice Models," Unpublished Manuscript, Princeton University. [1889]

OSBORNE, M. J. (2007): "Consumer Learning, Switching Costs, and Heterogeneity: A Structural Estimation," Discussion Paper EAG 07-10, Dept. of Justice, Antitrust Division, EAG. [1868,1875,1876,1881]

PAKES, A., AND P. MCGUIRE (2001): "Stochastic Algorithms, Symmetric Markov Perfect Equilibrium, and the 'Curse' of Dimensionality," *Econometrica*, 69, 1261–1281. [1867,1875,1879]

ROBERT, C. P., AND G. CASELLA (2004): *Monte Carlo Statistical Methods*. Springer Texts in Statistics. New York: Springer-Verlag. [1873]

ROBERTS, M., AND J. TYBOUT (1997): "The Decision to Export in Columbia: An Empirical Model of Entry With Sunk Costs," *American Economic Review*, 87, 545–564. [1884]

RUST, J. (1987): "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher," *Econometrica*, 55, 999–1033. [1865,1886,1887]

——— (1994): "Structural Estimation of Markov Decision Processes," in *Handbook of Econometrics*, Vol. IV, ed. by R. F. Engle and D. L. McFadden. Amsterdam, Netherlands: Elsevier, 3082–3139. [1892]

——— (1997): "Using Randomization to Break the Curse of Dimensionality," *Econometrica*, 65, 487–516. [1867,1882]

TANNER, M. A., AND W. H. WONG (1987): "The Calculation of Posterior Distributions by Data Augmentation," *Journal of the American Statistical Association*, 82, 528–550. [1873]

TIERNEY, L. (1994): "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, 22, 1701–1762. [1873]

Dept. of Economics, Queen's University, 233 Dunning Hall, 94 University Avenue, Kingston, ON K7L 5M2, Canada; imais@econ.gueensu.ca,

Dept. of Economics, City University London, Northampton Square, London EC1V 0HB, U.K. and Dept. of Economics, Northern Illinois University, 508 Zulauf Hall, DeKalb, IL 60115, U.S.A.; njain@niu.edu,

and

Rotman School of Management, University of Toronto, 105 St. George Street, Toronto, ON M5S 3E6, Canada; andrew.ching@rotman.utoronto.ca.

Manuscript received January, 2005; final revision received May, 2009.