Estimation of Discrete Choice Dynamic Programming Models

Hiroyuki Kasahara*
Department of Economics
University of British Columbia
hkasahar@mail.ubc.com

Katsumi Shimotsu
Faculty of Economics
University of Tokyo
shimotsu@e.u-tokyo.ac.jp

Abstract

This study reviews estimation methods for the infinite horizon discrete choice dynamic programming models and conducts Monte Carlo experiments. We consider: the maximum likelihood estimator, the two-step conditional choice probabilities estimator, sequential estimators based on policy iterations mapping under finite dependence, and sequential estimators based on value iteration mappings. Our simulation result shows that the estimation performance of the sequential estimators based on policy iterations and value iteration mappings is largely comparable to the MLE, while they achieve substantial computation gains over the MLE by a factor of 100 for a model with a moderately large state space.

Keywords: discrete choice dynamic, conditional choice probabilities, policy iteration mapping, value iteration mapping, nested pseudo likelihood.

JEL Classification Numbers: C13, C14, C63.

*Address for correspondence: Hiroyuki Kasahara, Vancouver School of Economics, University of British Columbia, 6000 Iona Dr., Vancouver, BC, V6T 1L4 Canada. Phone: +1-604-822-4814. E-mail: hkasahar@mail.ubc.ca.

Part of this work was done while the first author was a visiting scholar at the Institute of Economic Research, Hitotsubashi University. The authors are grateful for helpful comments from a referee and the participants at the Nakahara Prize Lecture. This research was supported by the SSHRC and JSPS Grant-in-Aid for Scientific Research (B) No. 26285053.

1 Introduction

Many important decisions we make in our daily lives are essentially forward-looking. For example, today's consumption and saving decision crucially depends on expectations of tomorrow's state of the world. Similarly, a firm manager's decision to invest hinges on her anticipation of next year's state. Understanding the dynamic response of individuals and firms is, thus, imperative to best advise policy makers and to assess various policy proposals.

The literature on estimating dynamic models of discrete choice was pioneered by Gotz and McCall (1980), Wolpin (1984), Miller (1984), Pakes (1986), and Rust (1987). Many existing empirical studies illustrate that the estimation of dynamic discrete models enhances our understanding of individual and firm behaviors and provides important policy implications. Here, contributions include Berkovec and Stern (1991), Das (1992), Keane and Wolpin (1997), Rust and Phelan (1997), Rothwell and Rust (1997), Altuğ and Miller (1998), Gilleskie (66), Eckstein and Wolpin (1999), Aguirregabiria (1999), Das et al. (2007), Kasahara (2009), Kennan and Walker (2011), Gowrisankaran and Rysman (2012), Kasahara and Lapham (2013), and Gayle et al. (2014).

The estimation of discrete choice dynamic programming models is complicated. There is no readily available Stata command or R package that can be used to estimate a class of discrete choice dynamic programming models. Given the research questions and the data sets at hand, empirical researchers have to carefully formulate the agent's dynamic programming models represented by the Bellman equation, where per-period utility flows and expectations with regard to future events are explicitly specified and parameterized. Standard maximum likelihood estimators (MLE) for estimating infinite horizon discrete choice dynamic programming models, the so-called "full solution methods," require repeatedly solving the fixed-point problem (i.e., the Bellman equation) during optimization, and can be very costly when the dimensionality of state space is large. The complication in estimating discrete choice dynamic programming models leads to a lack of transparency in how the parameters are identified and estimated. The large computational cost associated with estimation prevents empirical researchers from providing specification tests and various robustness checks, making estimation results less credible.

For these reasons, developing estimation techniques that are simple to implement and easy to understand is important for dynamic programming models. This study reviews recently developed computationally attractive estimation methods for the infinite horizon discrete choice dynamic programming models, and provides a simulation to assess the advantages and disadvantages of the different methods in terms of finite sample biases and computational costs. For readers who are interested in learning more about the solution and estimation methods for discrete choice dynamic programming models, please refer to Rust (1994a,b), Pakes (1994), Aguirregabiria and Mira (2010), Arcidiacono and Ellickson (2011), and Keane et al. (2011), among others.

To reduce the computational burden of full solution methods, Hotz and Miller (1993) developed a simpler two-step estimator, called the *conditional choice probability (CCP) estimator*. Given the initial non-parametric estimator of conditional choice probabilities, the likelihood-based version of the CCP estimator maximizes the pseudo-likelihood function constructed from the policy iterations mapping, which exploits the inverse mapping from the value functions to the conditional choice probabilities. The Hotz and Miller CCP estimator requires the inversion of a potentially very large matrix, and may become time-consuming when the state space is large, even though the inversion of matrix has to be done only once outside of the optimization routine, in many cases. Altuğ and Miller (1998) proposed an alternative policy iteration mapping that is computationally easy to evaluate without involving an inversion of large matrix when the model exhibits *finite dependence*. Under the assumption of additively separable Type-I extreme value unobserved states, the estimation procedure in the second step for CCP estimators is as simple as the estimation of a logit model. These two-step estimators may suffer from substantial finite sample bias, however, when the choice probabilities are poorly estimated in the first step.

To address the limitations of two-step estimators, Aguirregabiria and Mira (2002) proposed sequential estimators based on policy iterations mapping using the *nested pseudo likelihood (NPL) algorithm*, which is a recursive extension of the CCP estimator. Arcidiacono and Miller (2011) generalized the concept of finite dependence and developed estimators that relax some of the limitations of the CCP estimator by combining the expectation-maximization (EM) algorithm with the NPL algorithm to estimate models with unobserved heterogeneity. These sequential estimators utilize the restriction of dynamic programming models better and, hence, are more efficient than the two-step CCP estimator. Our simulation shows that, after only a few iterations, the performance of sequential estimators improves substantially over that of the CCP estimator.

When the model structure is complicated, the Hotz and Miller inverse mapping from the value functions to the conditional choice probabilities is also complicated, and is often difficult to code in computer software. The alternative policy iterations mapping under finite dependence considered by Altuğ and Miller (1998) and Arcidiacono and Miller (2011) are much simpler than the one considered by Hotz and Miller (1993), and are easy to code, although there are many empirically relevant models (e.g., see Example 3 in this paper) that do not exhibit finite dependence. One limitation of using the Hotz and Miller policy iterations mapping and the alternative mapping based on finite dependence is that the assumption of additively separable extreme value-distributed unobserved state variables must hold. The additive separability assumptions are violated, however, in many popular empirical models—for example, in labor economics, it is standard to specify the additively separable errors in log-wage regressions, but additive separability in a log-wage implies that the errors are not additively separable in the wage level (e.g., Keane and Wolpin (1997)).

To address the limitation of the estimator based on policy iteration mapping, Kasahara and Shimotsu (2011) developed a new sequential estimator based on value iteration mapping (i.e., Bellman equation), where, unlike the HotzMiller-type estimators, there is no need to reformulate a Bellman equation as a fixed-point mapping in the space of probability distributions (i.e., policy iteration operator). Kasahara and Shimotsu's sequential estimator based on value iteration mapping potentially permits unobserved state variables that are neither additively separable nor follow a generalized extreme value distribution. Its implementation is straightforward in terms of computer programming once the value iteration mapping is coded in a computer language.

Using a bus engine replacement model studied by Arcidiacono and Miller (2011), we use a simulation to examine how finite sample bias and computational speed vary across the MLE, the two-step CCP estimator, the sequential estimators based on policy iteration mapping under finite dependence, and the sequential estimators based on value iteration mappings. For comparison, we maintain the assumption of additively separable Type-I extreme value unobserved state variables. The result shows that the two-step CCP estimator based on finite dependence may suffer from finite-sample bias relative to the MLE. On the other hand, the estimation performance of the sequential estimators based on policy iterations and value iteration mappings is largely comparable to the MLE after three iterations, while they achieve substantial computation gains over the MLE, by a factor of 100, for a model with a moderately large state space. Overall, our simulation results suggest that the sequential estimators provide a good alternative to the MLE or the two-step CCP by dramatically improving the computational speed with relatively small efficiency loss over the MLE.

The rest of the paper is organized as follows. In Section 2, we present discrete choice dynamic programming models under the assumption of separably additive, conditionally independent, Type-I extreme value distributed unobserved state variables. Section 3 reviews various estimation methods including the MLE, CCP estimator, and sequential estimators based on policy iteration and value iteration mappings. Section 4 presents the simulation results, and section 5 provides concluding remarks.

2 Discrete Choice Dynamic Programming Models

An agent maximizes the expected discounted sum of utilities, $E[\sum_{j=0}^{\infty}\beta^j U_{\theta_u}(a_{t+j}, s_{t+j})|a_t, s_t]$, where s_t is the vector of states, and a_t is a discrete action to be chosen from the constraint set $A \equiv \{1, 2, \ldots, |A|\}$. The parameter β is a discount factor, while the utility function U_{θ_u} is known up to a finite dimensional parameter θ_u. The transition probabilities are given by $p_{\theta_p}(s_{t+1}|s_t, a_t)$. The Bellman equation for this dynamic optimization problem is written as

$$W(s_t) = \max_{a\in A}\ U_{\theta_u}(a, s_t) + \beta \int W(s_{t+1}) p_{\theta_p}(s_{t+1}|s_t, a) ds_{t+1}.$$

From the viewpoint of an econometrician, the state vector can be partitioned as $s_t = (x_t, \epsilon_t)$, where $x_t \in X$ is an observable state variable, and ϵ_t is an unobservable state variable.

We make the following assumptions throughout the paper.

Assumption 1 (Conditional Independence of ϵ_t)**.** *The transition probability function of the state variables can be written as* $p_{\theta_p}(s_{t+1}|s_t, a_t) = g_{\theta_g}(\epsilon_{t+1}|x_{t+1}) f_{\theta_f}(x_{t+1}|x_t, a_t)$.

Assumption 2 (Finite support for x)**.** *The support of x is finite and given by* $X = \{1, \ldots, |X|\}$.

Assumption 1 assumes away the presence of persistent unobserved state variables, which is often referred as "unobserved heterogeneity." Developing the computationally attractive estimation methods for the dynamic programming model with unobserved heterogeneity is an important research agenda (e.g., Arcidiacono and Miller (2011)). In this study, we focus on reviewing the estimation methods for the model without unobserved heterogeneity; understanding the relative computational costs and the relative efficiencies across different estimators for the model without unobserved heterogeneity is an important step in extending the different estimators we consider here to models with unobserved heterogeneity.

We maintain Assumption 2 to compare computational costs across different estimation methods. When the underlying state space is continuous, empirical researchers often discretize the continuous state space and estimate the discretized version of the dynamic programming model. In our simulation, across different estimation methods, we examine how the computational cost increases with the number of grid points by taking the increasingly finer discretization of continuous state space.[1]

Under Assumption 1, define the integrated value function $V(x) = \int W(x,\epsilon) g_{\theta_g}(\epsilon|x) d\epsilon$, and let $B_V \subset \mathbb{R}^{|X|}$ be the space of $V \equiv \{V(x) : x \in X\}$. The Bellman equation can be rewritten in terms of this integrated value function, as follows:

$$V(x) = \int \max_{a \in A} \left\{ U_{\theta_u}(a, x, \epsilon') + \beta \sum_{x' \in X} V(x') f_{\theta_f}(x'|x, a) \right\} g_{\theta_g}(\epsilon'|x) d\epsilon' := [\Gamma(\theta, V)](x), \tag{1}$$

where $\theta = (\beta, \theta_u', \theta_g', \theta_f')' \in \Theta$ and $\Gamma(\theta, V) : \Theta \times B_V \to B_V$ is called the Bellman operator. The Bellman equation is compactly written as $V = \Gamma(\theta, V)$. We denote the fixed point of the Bellman operator given θ by V_θ, so that

$$V_\theta = \Gamma(\theta, V_\theta). \tag{2}$$

Let $P(a|x)$ denote the conditional choice probabilities of the action a, given the state x, and let $B_P \subset \mathbb{R}^{|A||X|}$ be the space of $P \equiv \{P(a|x) : a \in A, x \in X\}$. Given the value function V, $P(a|x)$ is expressed as

$$P(a|x) = \int I \left\{ a = \arg\max_{j \in A} \left(U_{\theta_u}(a, x, \epsilon) + \beta \sum_{x' \in X} V(x') f_{\theta_f}(x'|x, a) \right) \right\} g_{\theta_g}(\epsilon|x) d\epsilon := [\Lambda(\theta, V)](a|x), \tag{3}$$

where $I(\cdot)$ is an indicator function and the right-hand side of (3) defines the mapping $\Lambda(\theta, V) : \Theta \times B_V \to B_P$. Denote the conditional choice probabilities associated with the fixed point of the Bellman operator by

$$P_\theta := \Lambda(\theta, V_\theta), \tag{4}$$

where V_θ is defined in (2).

[1] Kasahara and Shimotsu (2008) and Srisuma and Linton (2012) discuss the estimation procedures for discrete choice dynamic programming models when the state space is continuous.

In general, evaluating the Bellman operator $\Gamma(\theta, V)$ and the conditional choice probabilities defined by the right-hand side of (1) and (3), respectively, involves multi-dimensional integrations, which are computationally costly. The following two assumptions are often made in empirical applications.

Assumption 3 (Additive Separability)**.** *The unobservable state variable ϵ_t is additively separable in the utility function, so that $U_{\theta_u}(s_t, a_t) = u_{\theta_u}(x_t, a_t) + \epsilon_t(a_t)$, where $\epsilon_t(a_t)$ is the a-th element of the unobservable state vector $\epsilon_t = \{\epsilon_t(a) : a \in A\}$.*

Assumption 4 (Type-I extreme value)**.** *$\epsilon_t = (\epsilon_t(1), ..., \epsilon_t(|A|))'$ is independently and identically distributed under a Type-I extreme value distribution, with $g_{\theta_g}(\epsilon_t|x_t) = \prod_{a\in A} \exp\{-\epsilon_t(a) - \exp(-\epsilon_t(a))\}$.*

Together with Assumption 1, Assumptions 3 and 4 are first introduced by Rust (1987), and widely used in existing empirical applications. Under Assumptions 1–4, (1) and (3) yield closed-form expressions given by

$$V(x) = \gamma + \log\left\{\sum_{a\in A} \exp\left(v_\theta(x, a; V)\right)\right\} := [\Gamma(\theta, V)](x), \tag{5}$$

$$P(a|x) = \frac{\exp\left(v_\theta(x, a; V)\right)}{\sum_{j\in A} \exp\left(v_\theta(x, j; V)\right)} := [\Lambda(\theta, V)](a|x), \tag{6}$$

where

$$v_\theta(x, a; V) := u_{\theta_u}(x, a) + \beta \sum_{x'\in X} V(x') f_{\theta_f}(x'|x, a) \tag{7}$$

is the choice-specific value function, and $\gamma \approx 0.5772$ is Euler's constant. Therefore, under Assumption 4, solving the fixed point of the Bellman operator and evaluating the conditional choice probabilities are computationally straightforward, because it is not necessary to compute multi-dimensional integrals numerically.

The parameter for the transition probability of x_t, f_{θ_f}, can be estimated directly from the observed data $\{x_{it}, x_{it+1}\}_{i=1}^n$, which is straightforward and less computationally costly than estimating the parameter associated with the utility function u_{θ_u}. To focus our discussion on the computational cost of estimating the parameter in u_{θ_u}, which involves solving the Bellman equation (1), we simply assume that f_{θ_f} is known. Then, we drop the subscript θ_f from f_{θ_f}, and write $f = f_{\theta_f}$.

Assumption 5 (f_θ is known)**.** *$f_{\theta_f}(x_{t+1}|x_t, a_t)$ is known to an econometrician.*

Under Assumptions 4–5, the unknown parameter is $\theta = (\beta, \theta_u)'$. We are interested in estimating the parameter vector θ from the cross-sectional data set $\{x_{it}, a_{it}\}_{i=1}^n$, where (x_{it}, a_{it}) is randomly drawn across i from the stationary distribution, and n is the sample size.

Let θ^* and P^* denote the true parameter value and the true conditional choice probabilities. Let V^* denote the true integrated value function. Then, P^* and V^* are related as $P^* = \Lambda(\theta^*, V^*)$. Note that V^* is the fixed point of $\Gamma(\theta^*, \cdot)$ and, hence, $V^* = \Gamma(\theta^*, V^*)$.

We consider the following two examples.

Example 1 (The bus engine replacement model)**.** *We consider a version of the bus engine problem studied by Rust (1987) and Arcidiacono and Miller (2011). Let* $a_t \in A := \{0, 1\}$*, where* $a_t = 1$ *represents the bus engine replacement, and* $a_t = 0$ *means no replacement. The utility function is specified as* $u_{\theta_u}(x_t, a_t, \epsilon) = u_{\theta_u}(x_t, a_t) + \epsilon_t(a_t)$*,* $\epsilon_t = (\epsilon_t(0), \epsilon_t(1))$ *is independently drawn from a Type-I extreme value distribution, and*

$$u_{\theta_u}(x_t, a_t) = \begin{cases} 0, & \textit{if } a_t = 1 \\ \theta_0 + \theta_1 x_t, & \textit{if } a_t = 0, \end{cases}$$

The state variable x_t *represents the mileage accumulated over time. We assume that mileage accumulates in increments of* $\delta = 0.25$*, with the maximum mileage given by* $\bar{x}_{\max} = 25$*, so that* $X = \{0, \delta, 2\delta, ..., \bar{x}_{\max}\}$*, with* $|X| = \bar{x}_{\max}/\delta + 1 = 51$*. The transition probability of* x_{t+1} *is given by* x_t*, and* a_t *is specified by a discrete analog of an exponential distribution, where* $x_{t+1} - x_t$ *is approximately distributed according to Exp*(λ)*, as*

$$f(x_{t+1}|x_t, a_t = 0) = \begin{cases} e^{-\lambda(x_{t+1}-x_t)} - e^{-\lambda(x_{t+1}+\delta-x_t)}, & \textit{if } \bar{x}_{\max} > x_{t+1} \geq x_t, \\ e^{-\lambda(\bar{x}_{\max}+\delta-x_t)}, & \textit{if } x_{t+1} = x_{\max} \textit{ and } x_t < \bar{x}_{\max}, \\ 1, & \textit{if } x_{t+1} = x_t = \bar{x}_{\max}, \\ 0, & \textit{otherwise}. \end{cases} \tag{8}$$

while

$$f(x_{t+1}|x_t, a_t = 1) = \begin{cases} e^{-\lambda(x_{t+1})} - e^{-\lambda(x_{t+1}+\delta)}, & \textit{if } \bar{x}_{\max} > x_{t+1}, \\ e^{-\lambda(\bar{x}_{\max}+\delta)}, & \textit{if } x_{t+1} = x_{\max}. \end{cases} \tag{9}$$

Because the parameter λ *can be estimated directly from the observation of* $\{x_t\}$ *in the data set, we assume that* λ *is known, as assumed in Assumption 5. The transition probability of* x_{t+1}*, given* x_t *and* $a_t = 1$*, does not depend on the value of* x_t*. That is, choosing* $a_t = 1$ *"renews" the stochastic process of* x_t *and makes the past decisions irrelevant for the continuation value. This is an example of finite dependence, which we discuss later.*

Example 2 (The dynamic decision to import with sunk cost)**.** *We consider a dynamic model of an import decision. This class of models is studied by Das et al. (2007), Kasahara and Rodrigue (2008), and Kasahara and Lapham (2012). Let* $a_t \in \{0, 1\}$*, where* $a_t = 1$ *represents importing. Let* ω_t *be the current productivity level. The state variable is* $x_t = (\omega_t, a_{t-1})$*.*

The profit function is given by $u_{\theta_u}(x_t, a_t, \epsilon) = u_{\theta_u}(x_t, a_t) + \epsilon_t(a_t)$*,* $\epsilon_t = (\epsilon_t(0), \epsilon_t(1))$ *is independently drawn from a Type-I extreme value distribution, and*

$$\begin{aligned} &u_{\theta_u}(\omega_t, a_{t-1}, a_t) = \pi_{\theta_u}(\omega_t, a_t) - c_{\theta_u}(a_t, a_{t-1}) \quad \textit{with} \\ &\qquad \pi_{\theta_u}(\omega_t, a_t) = \exp(\theta_0 + \theta_1 \omega_t + \theta_2 a_t) \quad \textit{and} \quad c_{\theta_u}(a_t, a_{t-1}) = (\theta_3 + \theta_4(1 - a_{t-1}))a_t, \end{aligned}$$

where $\pi_{\theta_u}(\omega_t, a_t)$ *is the gross profit that depends on the productivity* ω_t *and import decision* a_t*.*

Then, $c_{\theta_u}(a_t, a_{t-1})$ represents the fixed cost of importing materials, where θ_3 is the per-period fixed cost, and θ_4 is a one-time sunk cost of importing.

We assume that the productivity $\omega_t \in \Omega$, with $\Omega = \{-1.5, -1.45, -1.4, ..., -0.05, 0, 0.05, ..., 1.4, 1.45, 1.5\}$. Let $\omega^1 = -1.5$, $\omega^2 = -1.45$, ..., $\omega^{60} = 1.45$, and $\omega^{|\Omega|} = 1.5$. Let $q^j = (\omega^j + \omega^{j+1})/2$ for $j = 1, ..., |\Omega| - 1$ so that q^j is the middle point between ω^j and ω^{j+1}. The transition function of ω_t approximates the AR(1) process, $\omega_{t+1} = \rho\omega_t + \epsilon_t$, with $\epsilon_t|\omega_t \overset{iid}{\sim} N(0, 1)$, and is given by

$$f(\omega_{t+1} = \omega^j|\omega_t = \omega^i) = \begin{cases} \Phi((q^1 - \rho\omega^i)/\sigma), & \text{if } j = 1 \\ \Phi((q^j - \rho\omega^i)/\sigma) - \Phi((q^{j-1} - \rho\omega^i)/\sigma), & \text{if } 1 < j < |\Omega| \\ 1 - \Phi((q^{|\Omega|} - \rho\omega^i)/\sigma), & \text{if } j = |\Omega|, \end{cases}$$

where the values of ρ and σ are assumed to be known. Then, the transition function of $x_t = (\omega_t, a_{t-1})$, given $x_{t-1} = (\omega_{t-1}, a_{t-2})$ and a_{t-1}, is defined as

$$f(\omega_{t+1}, a_t = i|(\omega_t, a_{t-1}), a_t = j) = f(\omega_{t+1}|\omega_t, a_t = j)I\{i = j\}.$$

As in the model of the bus engine replacement, this model exhibits finite dependence.

3 Estimation methods

3.1 Maximum likelihood estimator

The *MLE* solves the following constrained maximization problem:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \ln \{[\Lambda(\theta, V)](a_i|x_i)\} \quad \text{subject to} \quad V = \Gamma(\theta, V). \tag{10}$$

Computation of the MLE by the *nested fixed point (NFXP) algorithm* requires repeatedly solving all fixed points of $V = \Gamma(\theta, V)$ at each parameter value in order to maximize the objective function with respect to θ, as follows:

Initialization: Initialize V_0 as a zero vector.

Inner loop (successive approximation): Given θ, solve the fixed point of $V = \Gamma(\theta, V)$ by iterating $V^j = \Gamma(\theta, V^{j-1})$, starting from V^0, until a pre-specified stopping criterion is satisfied (e.g., $||V^j - V^{j-1}|| < tol$, with, say $tol = 10^{-8}$).

Outer loop: Compute $\hat{\theta}_{MLE} = \arg\max_{\theta\in\Theta} \sum_{i=1}^{n} \ln \{[\Lambda(\theta, V_\theta)](a_i|x_i)\}$, where V_θ is obtained from the inner loop upon convergence.

If evaluating the fixed point of $\Gamma(\theta, \cdot)$ is costly, estimating the MLE is computationally costly, because it is required to repeatedly solve the fixed point of $\Gamma(\theta, \cdot)$ to maximize the likelihood function. The existing literature suggests a few ways to improve the computational speed of estimating the MLE.

First, when the analytical expression of the first-order derivatives of Bellman operator is available, it is possible to speed up solving the fixed point of $V = \Gamma(\theta, V)$ in inner loop by switching from successive approximations to Newton–Kantorovich iterations once the domain of attraction has been reached (Rust, 2000; Iskhakov et al., 2016).[2]

Second, Su and Judd (2012) propose a constrained optimization approach, referred to as the mathematical program with equilibrium constraints (MPEC). In MPEC, a "state-of-art" optimizer such as KNITRO is used to obtain the MLE by solving (10) as a constrained optimization, where repeatedly solving the fixed point of the Bellman equation is avoided because the constraint is imposed only at the end of optimization. The use of MPEC generally requires providing the analytical expression of the first-order derivatives but, if it is coded in a language that allows automatic differentiation, this approach may be promising in future, given the fast technological progress in computing software and hardware.

Third, Imai et al. (2009), Norets (2009), and Ching et al. (2012) propose a Bayesian estimation method that uses a Markov chain Monte Carlo (MCMC) algorithm, in which the parameters are estimated simultaneously while solving the dynamic programming model; in their MCMC algorithm, value iteration mapping is evaluated only once for each draw of the parameters. The growing popularity of the Bayesian approach makes it promising option for dynamic discrete choice models.

Fourth, as we discuss later, the MLE can be obtained using the nested pseudo likelihood algorithm, as suggested by Aguirregabiria and Mira (2002).

Fifth, the choice of stopping criteria in the inner loop matters. Empirical researchers often use norm-based stopping criteria, such as the sup-norm stopping criterion given by

$$\max_{x \in X} |V^j(x) - V^{j-1}(x)| < 10^{-8}. \tag{11}$$

Bray (2017) and Puterman (1994, section 6.6.3) illustrate that the use of an alternative stopping criterion, based on the "span-seminorm" defined by

$$\mathrm{sp}(V^j - V^{j-1}) := \max_{x \in X}\{V^j(x) - V^{j-1}(x)\} - \min_{x \in X}\{V^j(x) - V^{j-1}(x)\} < 10^{-8}, \tag{12}$$

leads to an inner loop termination with far fewer iterations than in the case of the sup-norm stopping criterion (11), without affecting the likelihood value. The span seminorm is invariant to the subtraction of a constant and, hence, the span-seminorm stopping criterion (12) is based on the change in the value function's relative differences across states. Using the span-seminorm stopping criterion (12) leads to the computational gain in implementing the NFXP algorithm, without affecting the property of the MLE, because the conditional choice probabilities depend only on the value function's relative differences across states, rather than its level. Furthermore, the value

[2]The Newton–Kantorovich iteration is defined as $V^j = V^{j-1} - [I_{|X|} - \nabla_{V'}\Gamma(\theta, V^{j-1})]^{-1}(V^{j-1} - \Gamma(\theta, V^{j-1}))$, where $I_{|X|}$ is a $|X| \times |X|$ identity matrix, and $\nabla_{V'}\Gamma(\theta, V^{j-1})$ is a $|X| \times |X|$ matrix defined by differentiating the map $\Gamma(\theta, V)$ with respect to V and evaluating it at $V = V^{j-1}$.

function's relative differences converge faster than its level.[3] In our simulation, we illustrate the computational gain of using the stopping criterion (12) relative to the sup-norm stopping criterion (11).

3.2 The Hotz–Miller conditional choice probabilities (CCP) estimator

While the MLE is efficient, computing the MLE for the discrete choice dynamic programming model is costly, especially when the state space is large. To reduce the computational burden, Hotz and Miller (1993) developed a simpler two-step estimator, called the *conditional choice probability (CCP) estimator*.

Note that the Bellman equation (1) can be rewritten as

$$V(x) = \sum_{a \in A} P(a|x) \left\{ u_{\theta_u}(x,a) + E[\epsilon(a)|x,a;\{\tilde{v}_\theta(x,a)\},P(a|x)] + \beta \sum_{x' \in X} V(x') f(x'|x,a) \right\}, \quad (13)$$

where $E[\epsilon(a)|x,a;\{\tilde{v}_\theta(x,a)\},P(a|x)] = [P(a|x)]^{-1} \int \epsilon(a) I\{\tilde{v}_\theta(x,a)+\epsilon(a) \geq \tilde{v}_\theta(x,j)+\epsilon(j),\ j \in A\} g(d\epsilon|x)$ is the expected value of $\epsilon(a)$ when a is chosen, given x, with $\tilde{v}_\theta(x,a) = v_\theta(x,a) - v_\theta(x,1)$. In general, $E[\epsilon(a)|x,a;\{\tilde{v}_\theta(x,a)\},P(a|x)]$ is a function of conditional choice probabilities because there exists a one-to-one relationship between $\{\tilde{v}_\theta(x,a)\}$ and $\{P(a|x)\}$. When ϵ follows a Type-I extreme value distribution under Assumption 4, this conditional expectation takes a simple form (Hotz and Miller, 1993, eq. (4.12)), given by

$$E[\epsilon(a)|x,a;\{\tilde{v}_\theta(x,a)\},P(a|x)] = \gamma - \ln P(a|x). \quad (14)$$

By substituting (14) into (13), we obtain

$$V(x) = u_{\theta,P}(x) + \beta E_P V(x) \quad \text{for } x \in X, \quad (15)$$

where $u_{\theta,P}(x) = \sum_{a \in A} P(a|x)[u_{\theta_u}(x,a) + \gamma - \ln P(a|x)]$ and $E_P V(x) = \sum_{a \in A} P(a|x) \sum_{x \in X} V(x') f(x'|x,a)$.

It is useful to represent (15) using matrix notation. Let $V = (V(1),...,V(|X|))'$ be a $|X| \times 1$ vector, let $u_{\theta,P}$ be a $|X| \times 1$ vector, and let E_P be a $|X| \times |X|$ matrix, where the i-th element of $u_{\theta,P}$ is given by $\sum_{a \in A} P(a|x=i)[u_{\theta_u}(i,a) + \gamma - \ln P(a|x=i)]$. In addition, the (i,j) element of E_P is given by $\sum_{a \in A} P(a|x=i) f(x'=j|x=i,a)$. Then, (15) is represented by a system of linear equations as $V = u_{\theta,P} + \beta E_P V$. Therefore, the value function implied by the conditional choice probability P is a unique solution to the system of linear equations (15): $V = (I - \beta E_P)^{-1} u_{\theta,P}$, where the right-hand side of this equation defines a mapping from the choice probability space B_P

[3] To see the first claim, note that $\arg\max_{j \in A} u_{\theta_u}(x,a) + \beta \sum_{x' \in X} V(x') f_\theta(x'|x,a) + \epsilon(a) = \arg\max_{j \in A} u_{\theta_u}(x,a) + \beta \sum_{x' \in X} \{V(x') - C\} f_\theta(x'|x,a) + \epsilon(a) + \beta C$, for any constant C, so that the conditional choice probabilities depend only on $V(x) - C$. For the second claim, Puterman (1994, theorem 6.6.6) shows that $\{\text{sp}(V^j - V^{j-1})\}_{j=1}^\infty$ converges faster than $\{\max_{x \in X} |V^j(x) - V^{j-1}(x)|\}_{j=1}^\infty$.

to the value function space B_V, denoted by

$$\varphi(\theta, P) \equiv (I_{|X|} - \beta E_P)^{-1} u_{\theta,P}. \tag{16}$$

Then, we may define a policy iteration operator Ψ_{HM} as a composite operator of $\varphi(\cdot)$ and $\Lambda(\cdot)$:

$$P = \Psi_{HM}(\theta, P) \equiv \Lambda(\theta, \varphi(\theta, P)). \tag{17}$$

The fixed point of Ψ_{HM}, given θ, defined by $P_\theta = \Psi_{HM}(\theta, P_\theta)$ is the same as the conditional choice probabilities associated with the fixed point of the Bellman operator defined by (4). Furthermore, the policy iteration operator $\Psi_{HM}(\theta, P)$ has an important property that its derivative with respect to P, evaluated at the fixed point, is equal to zero; that is,

$$\nabla_P \Psi_{HM}(\theta, P)|_{P=P_\theta} = 0. \tag{18}$$

See Proposition 2 of Aguirregabiria and Mira (2002) and Proposition 1 of Kasahara and Shimotsu (2009). This zero derivative property has an important implication on the efficiency property of the CCP estimator and its recursive extension, as we discuss later.

Given the initial consistent estimator of P, denoted by $\hat{P}^0$, the likelihood-version of the CCP estimator of Hotz and Miller (1993) is defined as[4]

$$\hat{\theta}_{HM} = \arg\max_{\theta} \sum_{i=1}^{n} \log[\Psi_{HM}(\theta, \hat{P}^0)](a_i|x_i). \tag{19}$$

As discussed in Aguirregabiria and Mira (2002), when $u_{\theta_u}(x, a)$ is multiplicatively separable between θ and (a, x) and Assumption 4 is satisfied, the policy iteration operator $\Psi_{HM}(\theta, P)$ takes a simple logistic closed form with respect to θ. Here, the most costly part of evaluating $\Psi_{HM}(\theta, \hat{P}^0)$ (i.e., the evaluation of $(I - \beta E_P)^{-1}$) can be done outside of the likelihood maximization when β and P are known, so that evaluating $\Psi_{HM}(\theta, \hat{P}^0)$ across different values of θ does not involve the repeated evaluations of $(I - \beta E_P)^{-1}$. The CCP estimator (19) does not require repeatedly solving the fixed point, leading to a significant computational gain over the NFXP.

There are some disadvantages of the CCP estimator defined by (19). First, the feasibility of the CCP estimator relies on Assumptions 3 and 4. This is because evaluating $E[\epsilon(a)|x, a; \{\tilde{v}_\theta(x, a)\}, P(a|x)]$ in terms of $\{P(a|x)\}$ is difficult unless ϵ is additively separable and is drawn from a Type-I extreme value distribution or its variant (i.e., the generalized extreme value distribution).[5] For example, if the unobserved state variable ϵ enters multiplicatively to the utility function $u_{\theta_u}(x, a, \epsilon)$, then the CCP estimator is difficult to implement. Second, the CCP estimator requires the inversion of the

[4]The CCP estimator is often defined in terms of the generalized method of moments objective function, rather than the likelihood-based objective function, as in the case of the original paper by Hotz and Miller (1993). Here, to ease the comparison across different estimation methods, we focus on the likelihood-based objective function.

[5]Assumption 4 can be relaxed by considering the generalized extreme value distribution family, as discussed in Arcidiacono and Miller (2011).

matrix $I_{|X|} - \beta E_P$, which could be costly when the dimension of the state space $|X|$ is very large. Third, the CCP estimator (19) could be inefficient when the initial estimator $\hat{P}^0$ is imprecise, which is often the case when the state space is large relative to the sample size.

We now review existing methods that address each of these disadvantages of the Hotz–Miller CCP estimator.

3.3 Finite dependence and the alternative CCP estimator

One disadvantage of the Hotz–Miller CCP estimator is that it involves the inversion of the matrix $I_{|X|} - \beta E_P$, which is computationally costly to implement when the state space is large.

When the model exhibits *finite dependence*, Altuğ and Miller (1998) and Arcidiacono and Miller (2011) show that we may derive an alternative mapping from B_P to B_P that does not involve the inversion of potentially large matrix and, hence, its computational cost is small. We denote this alternative mapping by $\Psi_{FD}(\theta, P)$. Analogous to the Hotz–Miller CCP estimator, we may define an alternative CCP estimator based on finite dependence as

$$\hat{\theta}_{FD} = \arg\max_{\theta} \sum_{i=1}^{n} \log[\Psi_{FD}(\theta, \hat{P}^0)](a_i|x_i), \tag{20}$$

where $\hat{P}^0$ is the initial estimator of P.

Arcidiacono and Miller (2011) and Arcidiacono and Ellickson (2011) provide a detailed discussion on finite dependence. A model exhibits finite dependence when there exists a sequence of future choices such that the subsequent continuation values do not depend on the current choice. For example, in the bus engine replacement model, choosing a replacement for the engine in $t+1$ makes the continuation value after $t+2$ independent of the replacement choice in period t. We now illustrate that both the bus engine replacement model and the dynamic import decision model in Examples 1 and 2 exhibit finite dependence, and then we derive the alternative mapping $\Psi_{FD}(\theta, P)$ in these models.

Example 1 (The bus engine replacement model)**.** *For the bus engine replacement model, we may write (5) as*

$$V(x_t) = \gamma + v_\theta(x_t, 1) + \log\left(\frac{\sum_{a_t \in \{0,1\}} \exp(v_\theta(x_t, a_t))}{\exp(v_\theta(x_t, 1))}\right) = \gamma + v_\theta(x_t, 1) - \log P(1|x_t).$$

Substituting this expression into (7), and then using (7) again gives

$$\begin{aligned} v_\theta(x_t, a_t) = u_{\theta_u}(x_t, a_t) + \beta\gamma + \beta \sum_{x_{t+1} \in X} \left(u_{\theta_u}(x_{t+1}, 1) - \log P(1|x_{t+1})\right) f(x_{t+1}|x_t, a_t) \\ + \beta^2 \sum_{x_{t+1} \in X} \sum_{x_{t+2} \in X} V(x_{t+2}) f(x_{t+2}|x_{t+1}, 1) f(x_{t+1}|x_t, a_t). \end{aligned} \tag{21}$$

Because $f(x_{t+2}|x_{t+1},1)$ does not depend on x_{t+1} in this model, the last term on the right-hand side of (21) is equal to $\beta^2 \sum_{x_{t+2}\in X} V(x_{t+2}) f(x_{t+2}|x_{t+1},1)$, and does not depend on a_t. This is an example of finite dependence. As a result of finite dependence, taking the difference in value functions leads to a cancellation of the last term in (21), yielding

$$v_\theta(x_t,0) - v_\theta(x_t,1) = \theta_0 + \theta_1 x_t - \beta \sum_{x_{t+1}\in X} \log P(1|x_t)\left(f(x_{t+1}|x_t,0) - f(x_{t+1}|x_t,1)\right).$$

Finally, the conditional choice probabilities are determined by the difference in the value functions, rather than the level in the value functions. From (6), we have

$$\begin{aligned} P(a_t=1|x_t) &= \frac{1}{1+\exp\left(\theta_0+\theta_1 x_t - \beta\sum_{x_{t+1}\in X}\log P(1|x_t)\left(f(x_{t+1}|x_t,0)-f(x_{t+1}|x_t,1)\right)\right)} \\ &:= [\Psi_{FD}(\theta,P)](a_t=1|x_t), \end{aligned}$$

where the right-hand side of this equation defines a mapping from B_P to B_P, given θ. When P is fixed so that $\sum_{x_{t+1}\in X}\log P(1|x_t)\left(f(x_{t+1}|x_t,0)-f(x_{t+1}|x_t,1)\right)$ can be computed outside of the optimization routine, evaluating $[\Psi_{FD}(\theta,P)](a|x)$ across different values of β and θ is as easy as the logit specification.

Example 2 (The dynamic model of importing)**.** *In the dynamic model of importing, following a similar argument to that leading to (21), and noting that $x=(\omega,a)$, we obtain*

$$\begin{aligned} v_\theta(\omega_t,a_{t-1},a_t) = u_{\theta_u}(\omega_t,a_{t-1},a_t) + \beta\gamma + \beta\sum_{\omega_{t+1}\in\Omega}\left(u_{\theta_u}((\omega_{t+1},a_t),1) - \log P(1|\omega_{t+1},a_t)\right) f(\omega_{t+1}|\omega_t) \\ + \beta^2 \sum_{\omega_{t+1}\in\Omega}\sum_{\omega_{t+2}\in\Omega} V(\omega_{t+2},1) f(\omega_{t+2}|\omega_{t+1}) f(\omega_{t+1}|\omega_t). \end{aligned}$$

As in the bus engine replacement model, the last term on the right-hand side does not depend on the value of a, leading to finite dependence. Therefore,

$$[\Psi_{FD}(\theta,P)](a_t=1|\omega_t,a_{t-1}) := \frac{1}{1+\exp\left(u_{\theta_u}(\omega_t,a_{t-1},0) - u_{\theta_u}(\omega_t,a_{t-1},1) + \beta\Delta_a EV(\theta,P)\right)}, \quad (22)$$

with

$$\Delta_a EV(\theta,P) := -\beta\theta_4 - \beta\sum_{\omega_{t+1}\in\Omega}\left(\log P(1|\omega_{t+1},0) - \log P(1|\omega_{t+1},1)\right) f(\omega_{t+1}|\omega_t), \quad (23)$$

where $u_{\theta_u}(\omega_{t+1},0,1) - u_{\theta_u}(\omega_{t+1},1,1) = -\theta_4$ is used to derive the first term on the right-hand side of (23). Because we may compute $\sum_{\omega_{t+1}\in\Omega}\left(\log P(1|\omega_{t+1},0) - \log P(1|\omega_{t+1},1)\right) f(\omega_{t+1}|\omega_t)$ outside of the optimization routine, evaluating the log-likelihood function in (20) across different values of β and θ is computationally easy.

The CCP estimator based on finite dependence, using the map $\Psi_{FD}(\theta, P)$, is a promising approach for the following reasons. First, it is extremely easy to compute. Second, it is often easier to code $\Psi_{FD}(\theta, P)$ in a computer language than $\Psi_{HM}(\theta, P)$. Third, as illustrated in the above two examples, the specification becomes "logit," which makes it easier for us to understand the underlying variation of the data to identify the structural parameter.

On the other hand, as in the original Hotz–Miller CCP estimator, the feasibility of the CCP estimator depends crucially on Assumptions 3 and 4. In addition, the CCP estimator based on $\Psi_{FD}(\theta, P)$ is even less efficient than the CCP estimator based on $\Psi_{HM}(\theta, P)$ because $\Psi_{FD}(\theta, P)$ no longer satisfies the zero derivative property (18).

Finite dependence may not hold for some empirically relevant models, even under Assumptions 3 and 4. The following example shows that a model may not exhibit finite dependence, even when Assumptions 3 and 4 hold.

Example 3 (The dynamic model of importing with lagged productivity effect)**.** *Consider the model of Example 2, but now assume that the distribution of ω_{t+1} depends not only on ω_t, but also on a_t. Therefore, the transition function of ω_t approximates $\omega_{t+1} = \rho\omega_t + \gamma a_t + \epsilon_t$ with $\epsilon_t|\omega_t \overset{iid}{\sim} N(0, 1)$, and is given by*

$$f(\omega_{t+1} = \omega^j | \omega_t = \omega^i, a_t) = \begin{cases} \Phi((q^1 - \rho\omega^i - \gamma a_t)/\sigma), & \text{if } j = 1 \\ \Phi((q^j - \rho\omega^i - \gamma a_t)/\sigma) - \Phi((q^{j-1} - \rho\omega^i - \gamma a_t)/\sigma), & \text{if } 1 < j < |\Omega| \\ 1 - \Phi((q^{|\Omega|} - \rho\omega^i - \gamma a_t)/\sigma), & \text{if } j = |\Omega|, \end{cases}$$

where the values of ρ, σ, and γ are assumed to be known. Then, the transition function of $x_t = (\omega_t, a_{t-1})$, given $x_{t-1} = (\omega_{t-1}, a_{t-2})$ and a_{t-1}, is defined as

$$f(\omega_{t+1}, a_t = i | (\omega_t, a_{t-1}), a_t = j) = f(\omega_{t+1} | \omega_t, a_t = j) I\{i = j\}.$$

This model does not exhibit finite dependence because there is no sequence of future choices that makes the subsequent continuation values independent of the current choice.

3.4 The nested pseudo likelihood (NPL) algorithm and the sequential estimator based on the policy iterations mappings

Aguirregabiria and Mira (2002) addressed the issue of inefficiency in the CCP estimator by developing a recursive extension of the CCP estimator based on the *nested pseudo likelihood (NPL) algorithm*.[6] For $\Psi(\theta, P) = \Psi_{HM}(\theta, P)$ or $\Psi_{FD}(\theta, P)$, the NPL algorithm starts from an initial estimate $\tilde{P}^0$, and then iterates the following steps until $j = k$:

Step 1: Given $\tilde{P}^{j-1}$, update θ by $\tilde{\theta}_j = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \log[\Psi(\theta, \tilde{P}^{j-1})](a_i|x_i)$.

Step 2: Update $\tilde{P}^{j-1}$ using the obtained estimate $\tilde{\theta}^j$: $\tilde{P}^j = \Psi(\tilde{\theta}^j, \tilde{P}^{j-1})$.

[6] Aguirregabiria and Mira (2007) extends the NPL algorithm to the model of dynamic games.

The NPL algorithm based on $\Psi(\theta, P) = \Psi_{HM}(\theta, P)$ and $\Psi_{FD}(\theta, P)$ produces a sequence of estimators $\{\tilde{\theta}^j_{HM}, \tilde{P}^j_{HM}\}^k_{j=1}$ and $\{\tilde{\theta}^j_{FD}, \tilde{P}^j_{FD}\}^k_{j=1}$, respectively. Denote the limit of this sequence of estimators as $(\hat{\theta}_{NPL}, \hat{P}_{NPL})$.

When we use the Hotz–Miller policy iteration operator $\Psi_{HM}(\theta, P)$, Aguirregabiria and Mira (2002) show that the limit of this sequence $\hat{\theta}_{NPL}$ gives the MLE, so that $(\hat{\theta}_{NPL}, \hat{P}_{NPL}) = (\hat{\theta}_{MLE}, \hat{P}_{MLE})$. Furthermore, Kasahara and Shimotsu (2009) show that its convergence rate toward the MLE is superlinear, characterized by

$$\tilde{\theta}^j_{HM} - \hat{\theta}_{MLE} = O_p(n^{-1/2}||\tilde{P}^{j-1}_{HM} - \hat{P}_{MLE}|| + ||\tilde{P}^{j-1}_{HM} - \hat{P}_{MLE}||^2) \quad \text{and} \quad \tilde{P}^{j-1}_{HM} - \hat{P}_{MLE} = O_p(\tilde{\theta}^j_{HM} - \hat{\theta}_{MLE}).$$

This suggests that the NPL algorithm using $\Psi_{HM}(\theta, P)$ may work, even with relatively imprecise initial estimates of the conditional choice probabilities, and that a few iterations of the NPL algorithm may lead to an estimator that is asymptotically equivalent to the MLE. The zero derivative property of the policy iteration operator (18) plays the key role in both results.[7]

In contrast, when we use the alternative mapping based on finite dependence, $\Psi_{FD}(\theta, P)$, the NPL algorithm generally does not lead to an asymptotically efficient estimator. Applying the result of Kasahara and Shimotsu (2012), we may characterize the convergence property of $\{\tilde{\theta}^j_{FD}, \tilde{P}^j_{FD}\}^k_{j=1}$ as follows:

$$\begin{aligned}
\tilde{\theta}^j_{FD} - \hat{\theta}_{NPL} &= O_p(||\tilde{P}^{j-1}_{FD} - \hat{P}_{NPL}||), \\
\tilde{P}^j_{FD} - \hat{P}_{NPL} &= M_{\Psi_\theta}\Psi_P(\tilde{P}^{j-1}_{FD} - \hat{P}_{NPL}) + O_p(n^{-1/2}||\tilde{P}^{j-1}_{FD} - \hat{P}_{NPL}|| + ||\tilde{P}^{j-1}_{FD} - \hat{P}_{NPL}||^2),
\end{aligned}$$

where $\Psi_P := \nabla_{P'}\Psi_{FD}(\theta^*, P^*)$ and $M_{\Psi_\theta} \equiv I - \Psi_\theta(\Psi'_\theta \Delta_P \Psi_\theta)^{-1}\Psi'_\theta \Delta_P$, with $\Psi_\theta := \nabla_{\theta'}\Psi_{FD}(\theta^*, P^*)$ and $\Delta_P := \text{diag}(P^*)^{-2}\text{diag}(P^*_{a,x})$, where $P^*_{a,x}$ denote an $|X||A| \times 1$ vector, the elements of which are the probability mass function of (a_i, x_i) arranged conformably with $\Psi_{FD}(a|x)$. Kasahara and Shimotsu (2012) argue that the largest eigenvalue of $M_{\Psi_\theta}\Psi_P$ in absolute value can be well approximated by the largest eigenvalue of Ψ_P in absolute value, implying that the convergence property of the NPL algorithm is determined by the largest eigenvalue of Ψ_P. Therefore, the sequence of estimators converges to a consistent estimator in the neighborhood of (θ^*, P^*) when all of the eigenvalues of Ψ_P are within the unit circle. In our simulation based on two examples, we find that the largest eigenvalue of Ψ_P is much smaller than one in absolute value. The limit of the sequence of estimators generated by the NPL algorithm, $\hat{\theta}_{NPL}$, is not the MLE, but it is typically more efficient than the two-step CCP estimator.

In practice, the largest eigenvalues of Ψ_P are not known because they depend on (θ^*, P^*). We could potentially use the nonparametric estimator $\hat{P}^0$ and the CCP estimator $\hat{\theta}_{FD}$ to evaluate

[7] When $\nabla_P\Psi(\theta, P)|_{P=P_\theta} = 0$, taking the derivative of both sides of $P_\theta = \Psi(\theta, P_\theta)$ with respect to θ, and rearranging the terms, gives $\nabla_\theta P_\theta = (I - \nabla_{P'}\Psi(\theta, P_\theta))^{-1}\nabla_\theta\Psi(\theta, P_\theta) = \nabla_\theta\Psi(\theta, P_\theta)$. As a result, the first-order condition of the MLE is identical to the first-order condition of Step 1 of the NPL algorithm, upon convergence. The zero derivative property (18) is also important to deriving this convergence rate, as shown in the proof of Proposition 2 of Kasahara and Shimotsu (2009).

the largest eigenvalue of $\nabla_{P'}\Psi_{FD}(\hat{\theta}_{FD}, \hat{P}^0)$. If the largest eigenvalues of $\nabla_{P'}\Psi_{FD}(\hat{\theta}_{FD}, \hat{P}^0)$ are larger than one, then the convergence condition is likely to be violated. In such a case, Kasahara and Shimotsu (2012) suggest using an alternative mapping $[\Psi_{FD}(\theta, P)]^{\alpha} P^{1-\alpha}$ to $\Psi_{FD}(\theta, P)$ for an appropriately chosen α. Here, the largest eigenvalue of $\nabla_{P'}\{[\Psi_{FD}(\theta^*, P^*)]^{\alpha}(P^*)^{1-\alpha}\}$ could be less than one, even when that of Ψ_P is not, if the value of α is chosen judiciously. See Kasahara and Shimotsu (2012) for details.

Note that once the mapping $\Psi_{HM}(\theta, P)$ or $\Psi_{FD}(\theta, P)$ is coded in a computer language, it is straightforward to implement the NPL algorithm by adding an outside loop to the code. Given that even a few iterations of the NPL algorithm may lead to substantial efficiency gains over the two-step CCP estimator, implementing the NPL algorithm is recommended to anyone who uses the CCP estimator.

3.5 Sequential estimator based on value iteration mappings

The class of models that can be estimated using the CCP estimator and its recursive version are restrictive in that the structural errors have be additively separable, and must follow the (generalized) extreme value distribution (see Assumptions 3 and 4). For example, in the celebrated paper of Keane and Wolpin (1997), the errors in the utility function are additively separable and normally distributed, because the utility function is directly related to the individual wage-level and the log wage specification has an additively separable error. Similarly, the firm's profit function is often specified in log form with an additively separable error, leading to a multiplicative structural error in the profit function, as in Kasahara (2009).

Kasahara and Shimotsu (2011) propose an alternative sequential algorithm based on a fixed-point mapping defined in the value function space, rather than in the probability space. This is applicable to a wider class of dynamic programming models than the class of models that are applicable to the CCP estimator. Define a q-fold operator of Γ given in (1) as

$$\Gamma^q(\theta, V) \equiv \underbrace{\Gamma(\theta, \Gamma(\theta, \dots \Gamma(\theta, \Gamma}_{q \text{ times}}(\theta, V)) \dots)).$$

Then, the sequential estimators proposed by Kasahara and Shimotsu (2011) are defined as follows. For pre-specified values of q_1 and q_2, starting from an initial value $\tilde{V}_0$, the q-NPL algorithm iterates the following steps until $j = k$:

Step 1: Given $\tilde{V}_{j-1}$, update θ by $\tilde{\theta}_j = \arg\max_{\theta\in\Theta} n^{-1}\sum_{i=1}^n \ln\left\{\left[\Lambda(\theta, \Gamma^{q_1}(\theta, \tilde{V}_{j-1}))\right](a_i|x_i)\right\}$.

Step 2: Update $\tilde{V}_{j-1}$ using the obtained estimate $\tilde{\theta}_j$: $\tilde{V}_j = \Gamma^{q_2}(\tilde{\theta}_j, \tilde{V}_{j-1})$.

Evaluating the objective function for a value of θ involves only q_1 evaluations of the Bellman operator $\Gamma(\theta, \cdot)$ and one evaluation of the probability operator $\Lambda(\theta, \cdot)$. The computational cost of Step 1 is roughly equivalent to that of estimating a model with q_1 periods.

This algorithm generates a sequence of estimators $\{\tilde{\theta}_j, \tilde{V}_j\}_{j=1}^k$. If this sequence converges, its limit satisfies the following conditions:

$$\check{\theta} = \arg\max_{\theta\in\Theta} n^{-1} \sum_{i=1}^{n} \ln \Lambda(\theta, \Gamma^{q_1}(\theta, \check{V}))(a_i|x_i) \quad \text{and} \quad \check{V} = \Gamma^{q_2}(\check{\theta}, \check{V}). \tag{24}$$

Any pair $(\check{\theta}, \check{V})$ that satisfies these two conditions in (24) is called a *q-NPL fixed point*. The *q-NPL estimator*, denoted by $(\hat{\theta}_{qNPL}, \hat{V}_{qNPL})$, is defined as the q-NPL fixed point with the highest value of the pseudo likelihood among all the q-NPL fixed points. Kasahara and Shimotsu (2011) analyze the convergence property of a sequence of estimators $\{\tilde{\theta}_j, \tilde{V}_j\}_{j=1}^k$, and argue that the sequence converges when we choose q_1 and q_2 to be sufficiently large.

The sequential estimator proposed by Kasahara and Shimotsu (2011) has the following advantages. First, this estimator can be applied to a wider class of models than the CCP estimators can. Assumptions 3 and 4 are not required, as long as it is possible to evaluate the Bellman operator (1). This estimator can also be applied to a model that does not exhibit finite dependence. For example, as we show below, we may apply this estimator to the model in Example 3, while the CCP estimator based on finite dependence cannot be applied. Second, the implementation of the estimator is straightforward in terms of computer programming once the Bellman operator is coded in a computer language. It does not require a reformulation of a Bellman equation because a fixed point that maps the space of probability distributions is not necessary. This is a major advantage over the CCP estimator, because coding the Bellman operator $\Gamma(\theta, V)$ is far more straightforward than coding the policy iteration mappings, $\Psi_{HM}(\theta, P)$ or $\Psi_{FD}(\theta, P)$.

When $q_1 \geq 1$, the sequential estimator based on value iterations is costlier to compute than is the sequential estimator based on the CCP estimator with finite dependence. This is because it involves q_1 iterations of the Bellman operator to evaluate the objective function across different values of θ.

When we choose $q_1 = 0$ with $\Gamma^0(\theta, V) := V$, from an initial value $\tilde{V}_0$, the q-NPL algorithm iterates the following steps until $j = k$:

Step 1: Given $\tilde{V}_{j-1}$, update θ by $\tilde{\theta}_j = \arg\max_{\theta\in\Theta} n^{-1} \sum_{i=1}^{n} \ln \left\{ \left[\Lambda(\theta, \tilde{V}_{j-1}) \right] (a_i|x_i) \right\}$.

Step 2: Update $\tilde{V}_{j-1}$ using the obtained estimate $\tilde{\theta}_j$: $\tilde{V}_j = \Gamma^{q_2}(\tilde{\theta}_j, \tilde{V}_{j-1})$.

In this case, the computational cost of the sequential estimator of Kasahara and Shimotsu (2011) in Step 1 is comparable to that of the CCP estimator with finite dependence.

In practice, the choice of q_1 and q_2 depends on the data set and the computational complexity relative to the available computational capacity. As the value of q_1 and q_2 increases, the sequential estimator proposed by Kasahara and Shimotsu (2011) upon convergence will approach the MLE. Therefore, in terms of efficiency, larger values of q_1 and q_2 are better. On the other hand, when the efficiency loss is not an important issue (e.g., because the sample size is large and the identification of model parameters is strong), we may set small values of q_1 and q_2 in order to achieve greater computational gains over the MLE based on the NFXP algorithm. In our simulation example, we show that setting $q_1 = 0$ and $q_2 = 5$ gives substantial computation gains, with relatively small efficiency loss.

Example 1 (The bus engine replacement model)**.** *When $q_1 = 0$, we have*

$$[\Lambda(\theta, \tilde{V}_{j-1})](a_t = 1|x_t) = \frac{1}{1 + \exp\left(\theta_0 + \theta_1 x_t + \beta[\Delta_a E\tilde{V}_{j-1}](x_t)\right)},$$

where $\Delta_a E\tilde{V}_{j-1}$ is computed from $\tilde{V}_{j-1}$ and $f(x_{t+1}|x_t, a_t)$ as

$$[\Delta_a E\tilde{V}_{j-1}](x_t) := \sum_{x_{t+1}\in X} \left(\tilde{V}_{j-1}(x_{t+1}) f(x_{t+1}|x_t, 1) - \tilde{V}_{j-1}(x_{t+1}) f(x_{t+1}|x_t, 0)\right).$$

Therefore, because $\Delta_a E\tilde{V}_{j-1}$ can be computed outside of the optimization routine, it is computationally easy to evaluate the pseudo-likelihood objective function of Step 1 in the q-NPL algorithm across different values of θ.

Example 3 (The dynamic model of importing with lagged productivity effect)**.** *When $q_1 = 0$, we have*

$$[\Lambda(\theta, \tilde{V}_{j-1})](a_t = 1|\omega_t, a_{t-1}) = \frac{1}{1 + \exp\left(u_{\theta_u}(\omega_t, a_{t-1}, 0) - u_{\theta_u}(\omega_t, a_{t-1}, 1) + \beta[\Delta_a E\tilde{V}_{j-1}](\omega_t)\right)}, \tag{25}$$

where $\Delta_a E\tilde{V}_{j-1}$ is computed from $\tilde{V}_{j-1}$ and $f(\omega_{t+1}|\omega_t, a_t)$ as

$$[\Delta_a E\tilde{V}_{j-1}](\omega_t) := \sum_{\omega_{t+1}\in \Omega} \left(\tilde{V}_{j-1}(\omega_{t+1}, 1) f(\omega_{t+1}|\omega_t, 1) - \tilde{V}_{j-1}(\omega_{t+1}, 0) f(\omega_{t+1}|\omega_t, 0)\right),$$

which can be computed outside of the optimization routine.

Comparing the pseudo-likelihood from the CCP estimator with finite dependence (22)–(23) and the pseudo-likelihood defined in (25), we find that both likelihoods are constructed from the difference in static utility functions and the adjustment to the difference in continuation values across different choices. These two estimators are different in their construction of continuation values. However, once appropriate "estimates" for the difference in continuation values are constructed, they are similar, in that maximizing the pseudo-likelihood value is mostly done by changing the difference in static utility functions. As a result, the computational speed of estimating the parameter in each step is comparable to that of estimating the logit model in both estimators.

4 Simulation: The bus engine replacement model

We consider an extended version of the bus engine replacement model in Example 1, as follows. The model is the same as that studied in Section 7.1.1 of Arcidiacono and Miller (2011). Let

$a_t \in A := \{0,1\}$, where $a_t = 1$ the represents bus engine replacement. The utility function is specified as $u_{\theta_u}(a_t, x_t, \epsilon) = u_{\theta_u}(a_t, x_t) + \epsilon_t(a_t)$, where $x_t = (x_{1t}, x_2, x_3)$, and $\epsilon_t = (\epsilon_t(0), \epsilon_t(1))$ are drawn independently from a Type-I extreme value distribution and

$$u_{\theta_u}(a_t, x_t) = \begin{cases} 0, & \text{if } a_t = 1, \\ \theta_0 + \theta_1 x_{1t} + \theta_2 x_2, & \text{if } a_t = 0. \end{cases}$$

The state variable x_{1t} represents the mileage accumulated over time. We assume that mileage accumulates in increments of $\delta_1 \in \{2.5, 0.50, 0.125\}$, depending on the choice of discretization of the state space, with the maximum mileage given by 25. The state variable x_2 captures a time-invariant observed feature of the bus and takes the value of 0 or 1. The state variable x_3 represents a time-invariant route characteristic of the bus. This determines the transition probability of x_{1t}, where a higher value of x_3 implies less frequent use of the bus. The transition probability of $x_{1,t+1}$ is given by $(x_{1,t}, x_3)$ and a_t follows a discrete analog of $\mathrm{Exp}(x_3)$. Furthermore, it is specified analogously to (8)–(9) with $\lambda = x_3$ so that, for example, we have

$$f(x_{1,t+1}|x_{1t}, x_3, a_t = 1) = \begin{cases} e^{-x_3(x_{1,t+1})} - e^{-x_3(x_{1,t+1}+\delta)}, & \text{if } 25 > x_{1,t+1}, \\ e^{-x_3(25+\delta)}, & \text{if } x_{1,t+1} = 25. \end{cases}$$

We assume that the state space of x_3 is given by the equally spaced discretization of the interval between 0.25 and 1.25 with the increment of $\delta_3 \in \{0.20, 0.04, 0.01\}$. The state space of the model is given by $X = X_1 \times X_2 \times X_3$, where $X_1 = \{0, \delta_1, 2\delta_1, ..., 25\}$, $X_2 = \{0, 1\}$, and $X_3 = \{0.25, 0.25 + \delta_3, 0.25 + 2\delta_3, ..., 1.25\}$.

We simulate a cross-sectional data set with the number of observations given by $N = 100$ or 400 for a decision-maker who solves the infinite horizon discrete choice dynamic problem by simulating the choices, given the bus mileage drawn from the stationary mileage distribution. We consider three different discretizations of the state space, with $(|X_1|, |X_3|) = (11, 6)$, $(51, 26)$, and $(201, 101)$, where the sizes of the state space are given by 132, 2652, and 40602, respectively.

In this model, the parameter to be estimated is $\theta := (\theta_0, \theta_1, \theta_2, \beta)'$. We examine how the efficiencies and the computational speed vary across the following nine different estimation methods.[8]

1. FIML-1: The parameter is estimated by MLE, as in Section 3.1, using the sup-norm stopping criterion (11) for the inner loop.

2. FIML-2: The same as FIML-1, except we use the span-seminorm stopping criterion (12) instead of the sup-norm stopping criterion (11).

3. SEQ-P1: This is the two-step CCP estimator based on finite dependence in Section 3.3, where the initial estimator of P is obtained by logit using the second-order polynomials of

[8] In this example, Hotz and Miller's CCP is not applicable, because β also has to be estimated.

(x_{1t}, x_2, x_3). Here, the policy iteration mapping is given by

$$[\Psi_{FD}(\theta, P)](a_t = 1|x_t) = \frac{1}{1 + \theta_0 + \theta_1 x_{1t} + \theta_2 x_2 - \beta \sum_{x_{1t} \in X_1} \log P(1|x_t)[f(x_{1,t+1}|x_{1t}, x_3, 0) - f(x_{1,t+1}|x_{1t}, x_3, 1)]}. \tag{26}$$

4. SEQ-P3: The parameter is estimated by a recursive extension of the CCP estimator by iterating Steps 1 and 2 in Section 3.4 three times using the policy iteration mapping (26), where the initial estimator of P is obtained as in SEQ-P1.

5. SEQ-P5: The same as SEQ-P3 except that we iterate Steps 1 and 2 five times instead of three times.

6. SEQ-V1: The parameter is estimated by $\tilde{\theta}_1 = \arg\max_{\theta \in \Theta} n^{-1} \sum_{i=1}^{n} \ln \left\{ \left[\Lambda(\theta, \tilde{V}_0) \right] (a_{it}|x_{it}) \right\}$, as in Step 1 of the sequential estimator based on value iterations in Section 3.5, where $\tilde{V}_0$ is set to the zero vector, so that $\left[\Lambda(\theta, \tilde{V}_0) \right] (a_t = 1|x_t) = 1/ (1 + \theta_0 + \theta_1 x_{1t} + \theta_2 x_2)$. In other words, we estimate θ_1 by a "static" logit model.

7. SEQ-V3: The parameter is estimated by the sequential estimator based on value iterations by iterating Steps 1 and 2 in Section 3.5 three times, where $\tilde{V}_0$ is set to zero vector. We set $q_1 = 0$ and $q_2 = 5$ so that we use

$$[\Lambda(\theta, \tilde{V}_{j-1})](a_t = 1|x_t) = \frac{1}{1 + \exp\left(\theta_0 + \theta_1 x_{1t} + \theta_2 x_2 + \beta \Delta_a E \tilde{V}_{j-1}\right)}$$

with $\Delta_a E \tilde{V}_{j-1} := \sum_{x_{1,t+1} \in X_1} \left(\tilde{V}_{j-1}(x_{1,t+1}) f(x_{1,t+1}|x_{1t}, x_3, 1) - \tilde{V}_{j-1}(x_{t+1}) f(x_{1,t+1}|x_{1t}, x_3, 0) \right)$ in Step 1. Then, we update $\tilde{V}_j$ by iterating $\Gamma(\tilde{\theta}_j, \tilde{V}_{j-1})$ five times in Step 2.

8. SEQ-V5: The same as SEQ-V3, except that we iterate Steps 1 and 2 five times instead of three times.

9. SEQ-V6-2: Starting from the estimator $(\tilde{\theta}_5, \tilde{V}_5)$ obtained in SEQ-V5, we iterate Step 1 in Section 3.5 once with $q_1 = 1$.

Tables 1 and 2 report the mean and the standard deviation of the estimates from 500 simulations across nine different estimators when the model is estimated with $N = 100$ and 400, respectively. Table 4 reports the CPU time in seconds used by the MATLAB process to complete 500 simulations for each of the estimation methods.

We first examine how the choice of stopping criterion for the inner loop affects the speed of computing the MLE. Comparing columns (2) with (3) in Table 1 or 2, we notice that FIML-1 and FIML-2 give identical estimation results across three discretization specifications, indicating that the estimates are identical between using the sup-norm stopping criterion (11) and using the span-seminorm stopping criterion (12). On the other hand, columns (2)–(3) of Tables 4 and Table 5 show that using the span seminorm stopping criterion (12) instead of the sup-norm stopping criterion (11) leads to substantial computational gains, reducing the average computational time and the average number of iterations by factors of 17 and 9, respectively, for the specification with $(|X_1|, |X_3|) = (201, 101)$. Therefore, using the span seminorm stopping criterion (12) in the inner loop is strongly recommended for estimating the model by MLE.

Columns (4)-(6) of Table 1 and 2 present the estimates using the sequential estimator based on the policy iteration mapping with finite dependence. Here, column (4) is the two-step CCP estimator (SEQ-P1), while columns (5) and (6) are its recursive extensions that iterate Steps 1 and 2 in Section 3.4 three and five times (SEQ-P3 and SEQ-P5), respectively. Except for the parameter θ_0, the estimation bias of the two-step CCP estimator in column (4) is larger than that of the FIML in columns (2) or (3), where the downward bias is especially severe for the parameter β. In columns (4)–(6), the downward estimation bias becomes smaller and the mean estimates get closer to the FIML as we move from SEQ-P1 to SEQ-P3 or SEQ-P5, indicating that the recursive extension of the CCP estimator based on the NPL algorithm help to reduce the finite sample bias.

Columns (7)–(9) of Table 1 and 2 present the estimates using the sequential estimator based on the value iteration mapping. The estimation biases of SEQ-V1, SEQ-V3, and SEQ-V5 in columns (7)–(9) are largely comparable to those of SEQ-P1, SEQ-P3, and SEQ-P5. Here, the bias for the sequential estimator based on value iteration mappings tends to become smaller as we move from SEQ-V1 to SEQ-V3, and then to SEQ-V5, as in the case of the estimator based on policy iteration mappings. For $N = 400$ and $(|X_1|, |X_3|) = (201, 101)$ in Table 2, the estimation results of SEQ-P3, SEQ-P5, SEQ-V3, SEQ-V5, and SEQ-V6-2 in columns (5), (6), (8), (9), and (10), respectively, are similar to those of the FIML in columns (2)–(3). This suggests that the efficiency loss in the sequential estimator with policy and value iteration mappings relative to the FIML is not very large when we consider the sequential estimator with more than three iterations.

Panel A of Table 3 reports the mean of the absolute bias in the estimated probabilities of choosing $a = 1$. These are obtained from the fixed point of the Bellman equation (4) under estimated parameters across states and 500 replications when the state space discretization is given by $(|X_1|, |X_3|) = (201, 101)$. Figures 1 and 2 show the pointwise 90% confidence interval of the estimated probabilities of choosing $a = 1$ across different values of X_1, when $X_2 = 0$ and $X_3 = 1$, for SEQ-P1, SEQ-P3, FIML-2, SEQ-V1, SEQ-V3, and SEQ-V6-2 for $N = 100$ and 400, respectively. Consistent with the results reported in Tables 1 and 2, the 90% confidence intervals of estimated conditional choice probabilities for SEQ-P3 and SEQ-V3 are similar to those of FIML-2, even for $N = 100$. This indicates that the loss of precision and finite sample bias in SEQ-P3 and SEQ-V relative to those of FIML-2 are small.

Panel B of Table 3 reports the mean of the absolute bias in the effect of the government credibly announcing that the bus engine replacement cost in the next period is temporarily subsidized by 1 dollar (so that $u_{\theta_u}(1, x_{t+1}) = 1$ instead of 0) on the probabilities of choosing $a_t = 1$ in the current period. The mean absolute bias of SEQ-P1 and SEQ-V1 reported in Panel B of Table 3 are larger than those of other estimators. Figures 3 and 4 shows the pointwise 90% confidence interval of the estimated announcement effect on the conditional choice probabilities across different values of X_1 when $X_2 = 0$ and $X_3 = 1$, for SEQ-P1, SEQ-P3, FIML-2, SEQ-V1, SEQ-V3, and SEQ-V6-2 when $N = 100$ and 400, respectively. Consistent with the relatively large downward bias of parameter β for SEQ-P1 in column (4) of Table 1, the mean of the estimated announcement effect of the CCP estimator (i.e., SEQ-P1) is substantially lower than that of FIML-2.[9] However, after iterating the NPL algorithm three times, the mean and the 90% confidence interval of SEQ-P3 become very similar to those of FIML-2. Similarly, the mean and the 90% confidence interval of SEQ-V3 are similar to those of FIML-2 in Figures 3 and 4.

As shown in Table 4, the computational gains of using the sequential estimator based on policy or value iteration mappings over the FIML are significant when the state space is large. Comparing column (2) with column (3) or (6) of Table 4 for the specification with $N = 400$ and $(|X_1|, |X_3|) = (201, 101)$, implementing SEQ-P1 and SEQ-V1 is faster than FIML by factors of 543 and 283, respectively. This is because the FIML requires repeatedly solving the Bellman equation at each candidate parameter value to maximize the log-likelihood function, while the computational burden of implementing SEQ-P1 and SEQ-V1 is similar to estimating the logit model. Implementing SEQ-P5 and SEQ-V5 is slower than SEQ-P1 and SEQ-V1, but is still faster than FIML by factors of 87 and 62, respectively.

In Table 4, as the state space increases from $(|X_1|, |X_3|) = (11, 6)$ to $(|X_1|, |X_3|) = (51.26)$, and then to $(|X_1|, |X_3|) = (201, 101)$, the computational time for the FIML increases drastically, while the computational time for SEQ-P1, SEQ-P3, SEQ-P5, SEQ-V1, SEQ-V3, and SEQ-V5 does not change much. As indicated in Figure 5, the FIML suffers from the curse of dimensionality in computational time, while the sequential estimator based on policy or value iteration mappings is not subject to the curse of dimensionality. SEQ-V6-2 also suffers from the curse of dimensionality, although SEQ-V6-2 is about one order of magnitude faster than FIML-2. In fact, Figure 6 shows that the computational time for FIML or SEQ-V6-2 has an approximately log-linear relationship with the size of the state space, while the relationship between the computational time for the sequential estimators and the size of the state space is approximately linear.

Overall, the simulation results suggest that SEQ-P1 and SEQ-V1 tend to suffer from finite sample bias that is substantially larger than FIML. However, by iterating the NPL algorithm three or five times to obtain their recursive extensions (SEQ-P3, SEQ-P5, SEQ-V3, and SEQ-V5), the finite sample bias can be reduced to the extent that their biases and their 90% confidence intervals become comparable to those of the FIML. Because the sequential estimators based on the policy or value iteration mappings do not suffer severely from dimensionality, the computational gains

[9] In this counterfactual policy experiment, an agent who now knows that the replacement cost will be temporarily low in the next period has a higher incentive to postpone replacing until the next period. Furthermore, the higher the value of β, the higher the incentive to postpone because, for example, if the value of β is zero, then the future event is irrelevant to the current choice.

associated with the sequential estimators relative to the FIML can be quite large when the state space of the model is large. Therefore, when the state space of the model is large and the sample size is moderate, the sequential estimators based on the policy or value iteration mappings are good alternatives to the CCP estimator or the MLE, providing large computational gains over MLE, while keeping the loss of efficiency small relative to that of the two-step CCP estimator.

5 Conclusion

This study provides a review of the literature on estimating discrete choice dynamic models, focusing on computational aspects of various estimation methods. Our simulation results suggest that the sequential estimators based on policy iterations under finite dependence, as well as the sequential estimators based on value iterations achieve substantial computational gains over the MLE based on the NFXP algorithm, with a relatively small efficiency loss and finite sample bias.

One important on-going research topic is to develop computationally attractive estimation methods for models with permanent or serially correlated unobserved heterogeneity. Accounting for unobserved heterogeneity is important in empirical applications because not doing so often leads to substantial bias in sunk or switching cost parameter estimates (e.g., Keane and Wolpin (1997), Eckstein and Wolpin (1999), Aguirregabiria and Mira (2007), Kennan and Walker (2011), and Kasahara and Lapham (2013)).

Arcidiacono and Miller (2011) makes an important contribution in this context by proposing a computationally attractive estimation procedure for models with unobserved heterogeneity. However, as discussed in the main text, the class of models that the estimation method of Arcidiacono and Miller (2011) can be applied to is limited practically to models with the i.i.d. additively separable structural errors that follow the (generalized) extreme value distribution, conditional on unobserved heterogeneity. Many empirically relevant models may not exhibit finite dependence. The sequential estimator based on value iterations proposed by Kasahara and Shimotsu (2011) can be applied to a wider class of dynamic programming models than the class of models considered by Arcidiacono and Miller (2011), and is straightforward to implement in practice (e.g., see Yamaguchi (2016)). We are currently working on extending the sequential estimator based on value iterations to models with unobserved heterogeneity, where the EM algorithm is implemented together with value iterations, similarly to the EM algorithm proposed in Arcidiacono and Miller (2011) with policy iterations.

References

Aguirregabiria, V. (1999), "The Dynamics of Markups and Inventories in Retailing Firms," *Review of Economic Studies*, 66, 275–308.

Table 1: Mean and standard deviation for estimated parameters (N = 100)

	DGP	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
Discretization: $(\lvert X_1\rvert, \lvert X_3\rvert) = (11, 6)$										
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
θ_0	2.000	2.100	2.100	2.048	2.053	2.059	2.124	2.078	2.077	2.081
		(0.549)	(0.549)	(0.572)	(0.539)	(0.538)	(0.549)	(0.538)	(0.540)	(0.534)
θ_1	-0.150	-0.229	-0.229	-0.285	-0.226	-0.226	-0.246	-0.233	-0.232	-0.235
		(0.129)	(0.129)	(0.145)	(0.122)	(0.131)	(0.145)	(0.141)	(0.140)	(0.130)
θ_2	1.000	1.133	1.133	1.028	1.064	1.047	0.962	1.006	1.018	1.070
		(1.049)	(1.049)	(0.998)	(0.907)	(0.900)	(0.852)	(0.931)	(0.951)	(0.968)
β	0.900	0.637	0.637	0.384	0.682	0.640	0.523	0.573	0.575	0.599
		(0.417)	(0.417)	(0.342)	(0.380)	(0.407)	(0.401)	(0.428)	(0.428)	(0.424)
Discretization: $(\lvert X_1\rvert, \lvert X_3\rvert) = (51, 26)$										
	DGP	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
θ_0	2.000	2.172	2.172	2.116	2.126	2.123	2.202	2.128	2.126	2.156
		(0.641)	(0.641)	(0.613)	(0.601)	(0.609)	(0.626)	(0.621)	(0.614)	(0.631)
θ_1	-0.150	-0.224	-0.224	-0.276	-0.227	-0.223	-0.243	-0.224	-0.224	-0.226
		(0.109)	(0.109)	(0.111)	(0.108)	(0.112)	(0.118)	(0.116)	(0.115)	(0.110)
θ_2	1.000	1.051	1.051	0.915	0.980	0.992	0.907	0.961	0.971	1.028
		(0.957)	(0.957)	(0.924)	(0.888)	(0.907)	(0.808)	(0.884)	(0.895)	(0.939)
β	0.900	0.634	0.634	0.338	0.623	0.617	0.562	0.587	0.591	0.612
		(0.417)	(0.417)	(0.305)	(0.406)	(0.419)	(0.423)	(0.432)	(0.430)	(0.424)
Discretization: $(\lvert X_1\rvert, \lvert X_3\rvert) = (201, 101)$										
	DGP	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
θ_0	2.000	2.179	2.179	2.111	2.124	2.120	2.212	2.129	2.126	2.166
		(0.647)	(0.647)	(0.629)	(0.609)	(0.607)	(0.636)	(0.616)	(0.609)	(0.642)
θ_1	-0.150	-0.222	-0.221	-0.276	-0.226	-0.221	-0.242	-0.222	-0.223	-0.222
		(0.099)	(0.099)	(0.103)	(0.101)	(0.104)	(0.107)	(0.107)	(0.107)	(0.100)
θ_2	1.000	1.035	1.035	0.884	0.986	1.002	0.897	0.953	0.959	1.012
		(0.907)	(0.907)	(0.821)	(0.824)	(0.848)	(0.755)	(0.837)	(0.843)	(0.889)
β	0.900	0.631	0.633	0.328	0.617	0.615	0.567	0.586	0.587	0.617
		(0.426)	(0.426)	(0.309)	(0.408)	(0.419)	(0.420)	(0.431)	(0.433)	(0.424)

Notes: Mean and standard deviations across 500 simulations.

Table 2: Mean and standard deviation for estimated parameters (N = 400)

Discretization: $(\lvert X_1\rvert, \lvert X_3\rvert) = (11, 6)$										
	DGP	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
θ_0	2.000	2.023	2.023	2.005	2.012	2.014	2.068	2.017	2.014	2.021
		(0.266)	(0.266)	(0.272)	(0.264)	(0.265)	(0.268)	(0.266)	(0.266)	(0.264)
θ_1	-0.150	-0.175	-0.175	-0.185	-0.176	-0.176	-0.210	-0.185	-0.181	-0.179
		(0.062)	(0.062)	(0.067)	(0.063)	(0.065)	(0.071)	(0.067)	(0.067)	(0.064)
θ_2	1.000	1.013	1.013	0.964	0.988	0.986	0.917	0.969	0.978	0.998
		(0.451)	(0.451)	(0.444)	(0.427)	(0.436)	(0.405)	(0.438)	(0.442)	(0.443)
β	0.900	0.802	0.802	0.739	0.801	0.790	0.601	0.749	0.762	0.776
		(0.249)	(0.249)	(0.260)	(0.251)	(0.256)	(0.276)	(0.268)	(0.264)	(0.254)
Discretization: $(\lvert X_1\rvert, \lvert X_3\rvert) = (51, 26)$										
	DGP	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
θ_0	2.000	2.042	2.042	1.999	2.021	2.023	2.110	2.021	2.025	2.039
		(0.295)	(0.295)	(0.288)	(0.292)	(0.295)	(0.305)	(0.295)	(0.296)	(0.294)
θ_1	-0.150	-0.176	-0.176	-0.197	-0.177	-0.175	-0.206	-0.176	-0.176	-0.179
		(0.060)	(0.060)	(0.067)	(0.061)	(0.062)	(0.065)	(0.063)	(0.063)	(0.062)
θ_2	1.000	0.966	0.966	0.898	0.936	0.939	0.868	0.923	0.932	0.957
		(0.434)	(0.434)	(0.434)	(0.417)	(0.419)	(0.383)	(0.416)	(0.420)	(0.434)
β	0.900	0.771	0.771	0.613	0.752	0.757	0.647	0.753	0.753	0.753
		(0.297)	(0.297)	(0.325)	(0.299)	(0.300)	(0.318)	(0.312)	(0.306)	(0.300)
Discretization: $(\lvert X_1\rvert, \lvert X_3\rvert) = (201, 101)$										
	DGP	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
θ_0	2.000	2.028	2.028	1.989	2.012	2.014	2.099	2.011	2.014	2.027
		(0.301)	(0.301)	(0.295)	(0.298)	(0.301)	(0.310)	(0.300)	(0.302)	(0.300)
θ_1	-0.150	-0.179	-0.179	-0.203	-0.183	-0.181	-0.209	-0.181	-0.181	-0.182
		(0.060)	(0.060)	(0.065)	(0.062)	(0.062)	(0.062)	(0.063)	(0.063)	(0.060)
θ_2	1.000	0.969	0.969	0.895	0.930	0.936	0.874	0.924	0.932	0.962
		(0.384)	(0.384)	(0.369)	(0.373)	(0.379)	(0.349)	(0.376)	(0.380)	(0.385)
β	0.900	0.749	0.749	0.573	0.716	0.723	0.631	0.724	0.721	0.733
		(0.309)	(0.309)	(0.316)	(0.317)	(0.318)	(0.321)	(0.324)	(0.320)	(0.310)

Notes: Mean and standard deviations across 500 simulations.

Table 3: Mean absolute bias for the estimated policy functions

Panel A: $(1/\lvert X\rvert)\sum_{x\in X}\lvert P_{\hat\theta}(d=1\mid x) - P_{\theta^*}(d=1\mid x)\rvert$									
	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
$N = 100$	0.073	0.073	0.087	0.074	0.073	0.088	0.077	0.078	0.073
$N = 400$	0.042	0.042	0.048	0.042	0.043	0.051	0.043	0.044	0.043
Panel B: $(1/\lvert X\rvert)\sum_{x\in X}\lvert (P^{cf}_{\hat\theta}(d=1\mid x) - P_{\hat\theta}(d=1\mid x)) - (P^{cf}_{\theta^*}(d=1\mid x) - P_{\theta^*}(d=1\mid x))\rvert$									
	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
$N = 100$	0.041	0.041	0.057	0.041	0.041	0.047	0.043	0.043	0.041
$N = 400$	0.025	0.025	0.034	0.026	0.026	0.034	0.027	0.027	0.027

Notes: Mean absolute bias computed as the mean of $(1/|X|)\sum_{x\in X}|P_{\hat\theta}(d=1|x) - P_{\theta^*}(d=1|x)|$ and $(1/|X|)\sum_{x\in X}|(P^{cf}_{\hat\theta}(d=1|x) - P_{\hat\theta}(d=1|x)) - (P^{cf}_{\theta^*}(d=1|x) - P_{\theta^*}(d=1|x))|$ across 500 simulations.

Table 4: CPU time to complete

					$N = 100$				
Discretization	FIML-1 (1)	FIML-2 (2)	SEQ-P1 (3)	SEQ-P3 (4)	SEQ-P5 (5)	SEQ-V1 (6)	SEQ-V3 (7)	SEQ-V5 (8)	SEQ-V6-2 (9)
$(\|X_1\|, \|X_3\|)$	11.207	1.075	0.084	0.135	0.187	0.068	0.159	0.250	0.630
$= (11, 6)$	(13.801)	(1.042)	(0.128)	(0.129)	(0.130)	(0.015)	(0.030)	(0.048)	(0.300)
$(\|X_1\|, \|X_3\|)$	76.810	5.444	0.068	0.135	0.198	0.067	0.158	0.250	1.131
$= (51, 26)$	(89.176)	(5.469)	(0.025)	(0.028)	(0.032)	(0.012)	(0.023)	(0.036)	(0.652)
$(\|X_1\|, \|X_3\|)$	1024.992	61.094	0.066	0.423	0.778	0.246	0.698	1.150	8.912
$= (201, 101)$	(1169.146)	(59.652)	(0.015)	(0.018)	(0.021)	(0.010)	(0.020)	(0.032)	(5.713)
					$N = 400$				
Discretization	FIML-1	FIML-2	SEQ-P1	SEQ-P3	SEQ-P5	SEQ-V1	SEQ-V3	SEQ-V5	SEQ-V6-2
$(\|X_1\|, \|X_3\|)$	7.004	1.063	0.074	0.126	0.177	0.071	0.179	0.280	0.976
$= (11, 6)$	(9.493)	(0.999)	(0.023)	(0.042)	(0.059)	(0.025)	(0.046)	(0.067)	(0.610)
$(\|X_1\|, \|X_3\|)$	64.602	4.557	0.072	0.134	0.195	0.071	0.179	0.286	1.593
$= (51, 26)$	(77.078)	(4.109)	(0.019)	(0.027)	(0.034)	(0.022)	(0.039)	(0.060)	(1.065)
$(\|X_1\|, \|X_3\|)$	919.040	52.682	0.097	0.353	0.607	0.186	0.517	0.844	9.079
$= (201, 101)$	(1132.197)	(48.819)	(0.055)	(0.072)	(0.089)	(0.027)	(0.057)	(0.086)	(6.267)

Notes: Mean and standard deviations across 500 simulations.

Table 5: The number of iterations in the inner loop for FIML-1 and FIML-2

	N=100		N=400	
Discretization	FIML-1	FIML-2	FIML-1	FIML-2
$(\|X_1\|, \|X_3\|)$	244.41	32.18	258.07	38.41
$= (11, 6)$	(226.04)	(21.78)	(200.08)	(13.17)
$(\|X_1\|, \|X_3\|)$	253.07	27.90	268.08	31.84
$= (51, 26)$	(231.59)	(18.20)	(212.07)	(12.92)
$(\|X_1\|, \|X_3\|)$	259.82	27.55	262.34	30.10
$= (201, 101)$	(233.63)	(18.29)	(213.48)	(12.75)

Notes: Mean and standard deviations across 500 simulations.

Aguirregabiria, V. and Mira, P. (2002), “Swapping the nested fixed point algorithm: a class of estimators for discrete Markov decision models,” *Econometrica*, 70, 1519–1543.

— (2007), “Sequential estimation of dynamic discrete games,” *Econometrica*, 75, 1331–1370.

— (2010), “Dynamic discrete choice structural models: A survey,” *Journal of Econometrics*, 156, 38–67.

Altuğ, S. and Miller, R. A. (1998), “The Effect of Work Experience on Female Wages and Labour Supply,” *Review of Economic Studies*, 65, 45–85.

Arcidiacono, P. and Ellickson, P. B. (2011), “Practical Methods for Estimation of Dynamic Discrete Choice Models,” *Annual Review of Economics*, 3, 363–394.

Arcidiacono, P. and Miller, R. A. (2011), “CCP estimation of dynamic discrete choice models with unobserved heterogeneity,” *Econometrica*, 79, 1823–1867.

Berkovec, J. and Stern, S. (1991), “Job Exit Behavior of Older Men,” *Econometrica*, 59, 189–210.

Bray, R. L. (2017), “Strong Convergence and the Estimation of Markov Decision Processes,” Northwestern University.

Ching, A. T., Imai, S., Ishihara, M., and Jain, N. (2012), “A Practitioner’s Guide to Bayesian Estimation of Discrete Choice Dynamic Programming Models,” *Quantitative Marketing and Economics*, 10, 151–196.

Das, M. (1992), “A Micro-econometric Model of Capital Utilization and Retirement: The Case of the Cement Industry,” *Review of Economic Studies*, 59, 277–297.

Das, S., Roberts, M. J., and Tybout, J. R. (2007), “Market Entry Costs, Producer Heterogeneity, and Export Dynamics,” *Econometrica*, 75, 837–873.

Eckstein, Z. and Wolpin, K. (1999), “Why youth drop out of high school: the impact of preferences, opportunities and abilities,” *Econometrica*, 67, 1295–1339.

Gayle, G.-L., Golan, L., and Soytas, M. A. (2014), “What Accounts for the Racial Gap in Time Allocation and Intergenerational Transmission of Human Capital?” .

Gilleskie, D. (66), “A dynamic stochastic model of medical care use and work absence,” *Econometrica*, 1–45.

Gotz, G. A. and McCall, J. J. (1980), “Estimation in sequential decisionmaking models: a methodological note,” *Economics Letters*, 6, 131–136.

Gowrisankaran, G. and Rysman, M. (2012), “Dynamics of consumer demand for new durable goods,” *Journal of Political Economy*, 120, 1173–1219.

Hotz, J. and Miller, R. A. (1993), “Conditional choice probabilities and the estimation of dynamic models,” *Review of Economic Studies*, 60, 497–529.

Imai, S., Jain, N., and Ching, A. (2009), “Bayesian Estimation of Dynamic Discrete Choice Models,” *Econometrica*, 77, 1865–1899.

Iskhakov, F., Lee, J., Rust, J., Schjerning, B., and Seo, K. (2016), “Comment on “Constrained Optimization Approaches to Estiamtion of Structural Models”,” *Econometrica*, 84, 365–370.

Kasahara, H. (2009), “Temporary Increases in Tariffs and Investment: The Chilean Experience,” *Journal of Business & Economic Statistics*, 27, 113–127.

Kasahara, H. and Lapham, B. (2012), “Productivity and the Decision to Import and Export: Theory and Evidence,” *Journal of International Economics*, 89, 297–316.

— (2013), “Productivity and the Decision to Import and Export: Theory and Evidence,” *Journal of International Economics*, 89, 297–316.

Kasahara, H. and Rodrigue, J. (2008), “Does the use of imported intermediates increase productivity? plant-level evidence,” *Journal of Development Economics*, 87, 106–118.

Kasahara, H. and Shimotsu, K. (2008), “Pseudo-likelihood Estimation and Bootstrap Inference for Structural Discrete Markov Decision Models,” *Journal of Econometrics*, 146, 92–106.

— (2009), “Nonparametric identification of finite mixture models of dynamic discrete choices,” *Econometrica*, 77, 135–175.

— (2011), “Sequential Estimation of Dynamic Programming Models,” University of British Columbia.

— (2012), “Sequential estimation of structural models with a fixed point constraint,” *Econometrica*, 80, 2303–2319.

Keane, M. P., Todd, P. E., and Wolpin, K. I. (2011), “The Structural Estimation of Behavioral Models: Discrete Choice Dynamic Programming Methods and Applications,” in *Handbook of Labor Economics*, eds. Ashenfelter, O. and Card, D., North-Holland, vol. 4, Part A, pp. 331 – 461.

Keane, M. P. and Wolpin, K. I. (1997), “The Career Decisions of Young Men,” *Journal of Political Economy*, 105, 473–522.

Kennan, J. and Walker, J. R. (2011), “The effect of expected income on individual migration decisions,” *Econometrica*, 79, 211–251.

Miller, R. (1984), “Job matching and occupational choice,” *Journal of Political Economy*, 92, 1086–1120.

Norets, A. (2009), "Inference in Dynamic Discrete Choice Models With Serially orrelated Unobserved State Variables," *Econometrica*, 77, 1665–1682.

Pakes, A. (1986), "Patents as options: some estimates of the value of holding European patent stocks," *Econometrica*, 54, 755–784.

— (1994), "Dynamic structural models, problems and prospects," in *Advances in Econometrics. Sixth World Congress*, ed. Sims, C., Cambridge University Press.

Puterman, M. L. (1994), *Markov Decision Processes*, John Wiley & Sons.

Rothwell, G. and Rust, J. (1997), "On the optimal lifetime of nuclear power plants," *Journal of Business and Economic Statistics*, 15, 195–208.

Rust, J. (1987), "Optimal replacement of GMC bus engines: an empirical model of Harold Zurcher," *Econometrica*, 55, 999–1033.

— (1994a), "Estimation of dynamic structural models, problems and prospects: discrete decision processes," in *Advances in Econometrics. Sixth World Congress.*, ed. Sims, C., Cambridge University Press.

— (1994b), "Structural estimation of Markov decision processes," in *Handbook of Econometrics*, eds. Engle, R. E. and McFadden, D. L., North-Holland, vol. 4.

— (2000), *Nested Fixed Point Algorithm Documentation Manual*, Yale University, https://editorialexpress.com/jrust/nfxp.pdf.

Rust, J. and Phelan, C. (1997), "How Social Security and Medicare Affect Retirement Behavior in a World of Incomplete Markets," *Econometrica*, 65, 781–832.

Srisuma, S. and Linton, O. (2012), "Semiparametric estimation of Markov decision processes with continuous state space," *Journal of Econometrics*, 166, 320–341.

Su, C.-L. and Judd, K. L. (2012), "Constrained optimization approaches to estimation of structural models," *Econometrica*, 80, 2213–2230.

Wolpin, K. (1984), "An Estimable Dynamic Stochastic Model of Fertility and Child Mortality," *Journal of Political Economy*, 92, 852–874.

Yamaguchi, S. (2016), "Effects of Parental Leave Policies on Female Career and Fertility Choices," McMaster University.

Figure 1: Estimated policy function: N = 100 and $(|X_1|, |X_3|) = (201, 101)$

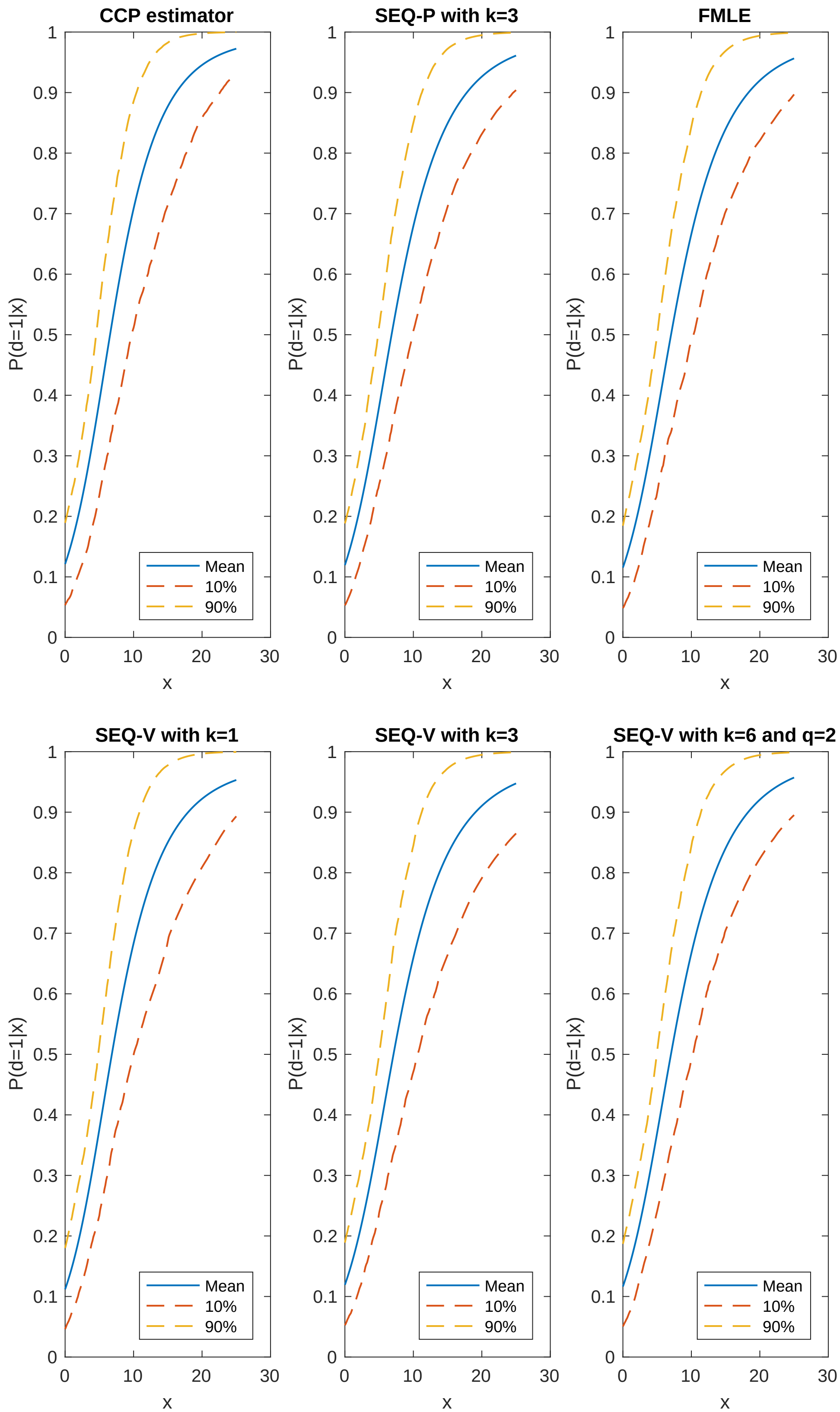

Figure 2: Estimated policy function: N = 400 and $(|X_1|, |X_3|) = (201, 101)$

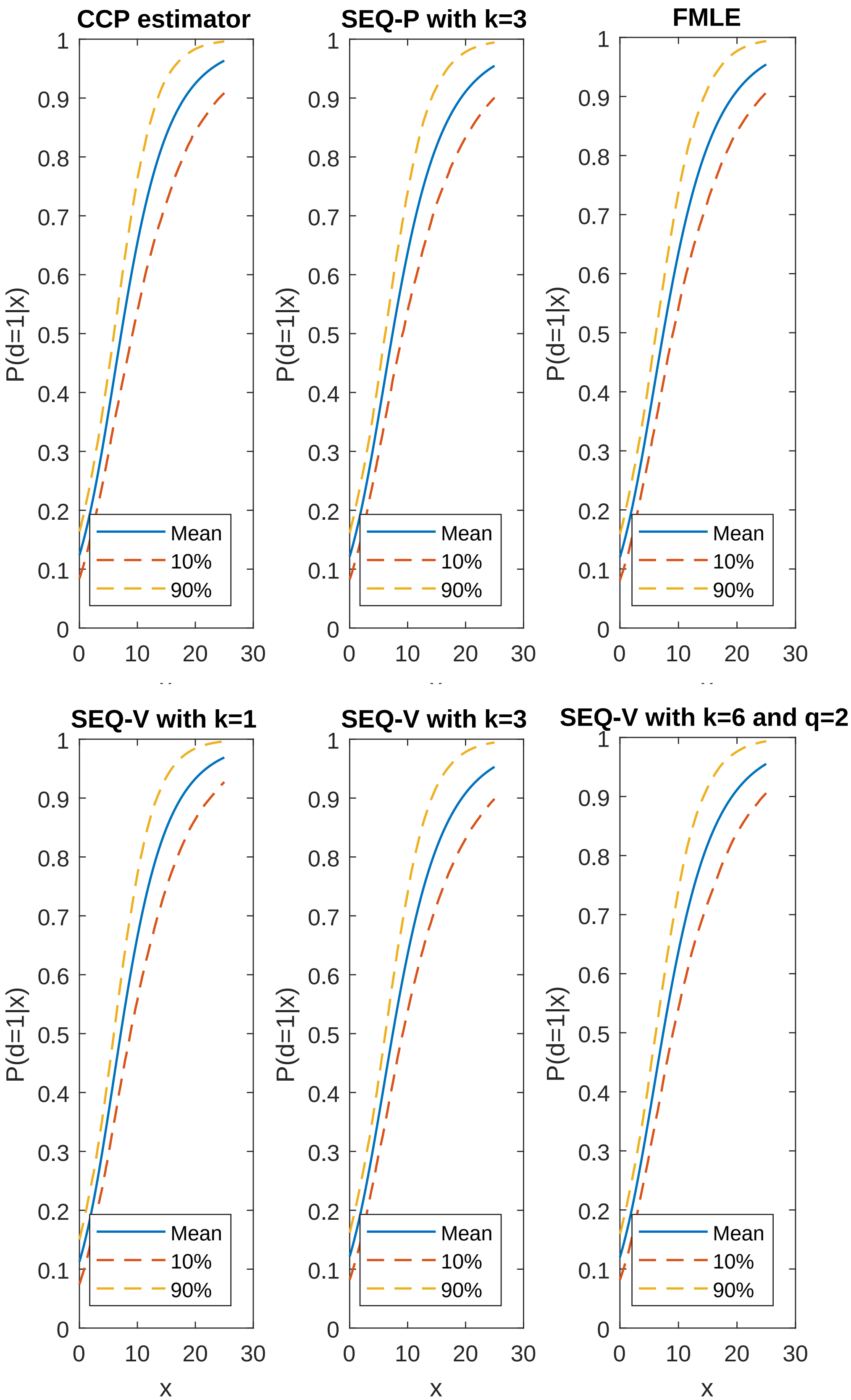

Figure 3: Estimated counterfactual effect: N = 100 and $(|X_1|, |X_3|) = (201, 101)$

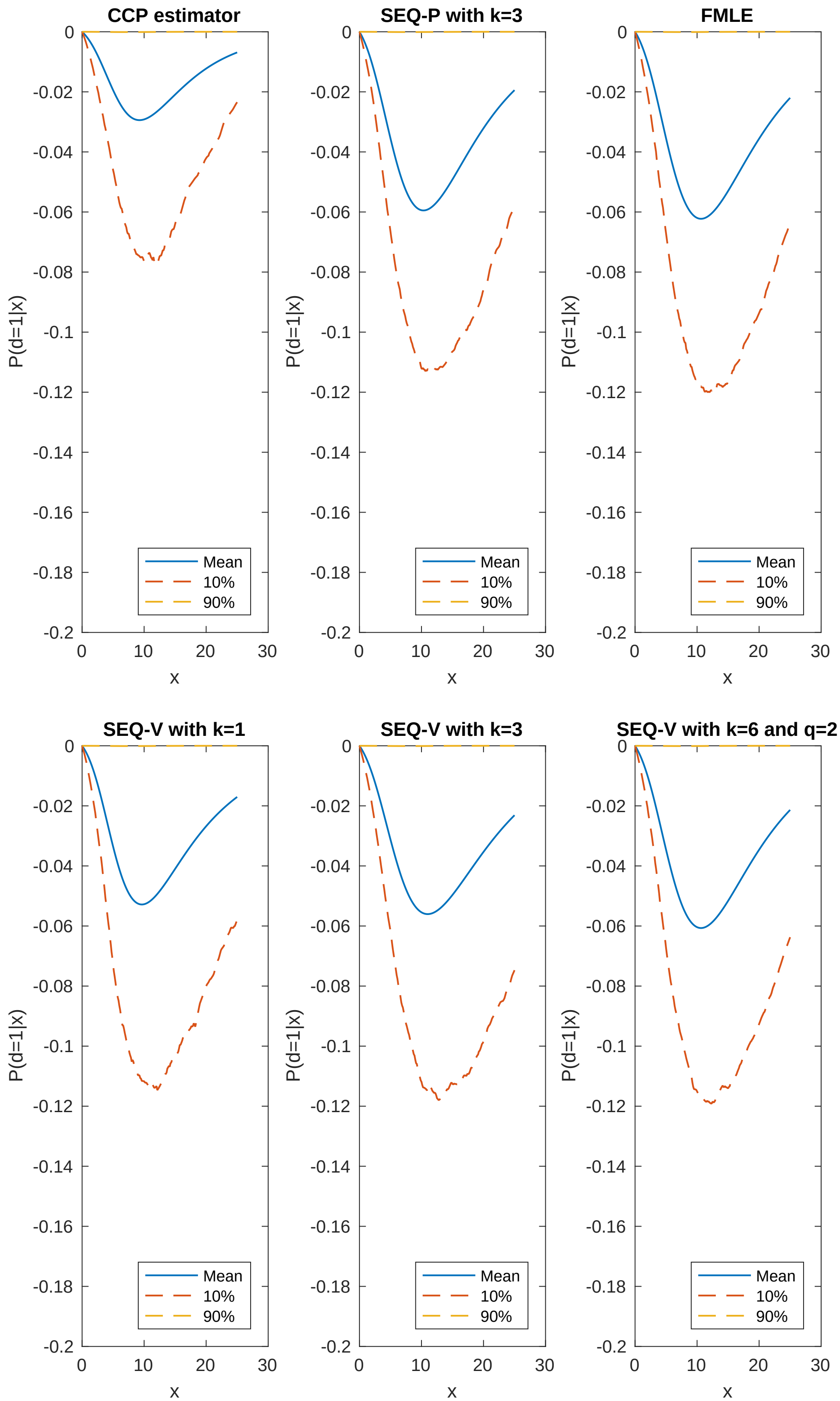

Figure 4: Estimated counterfactual effect: N = 400 and $(|X_1|, |X_3|) = (201, 101)$

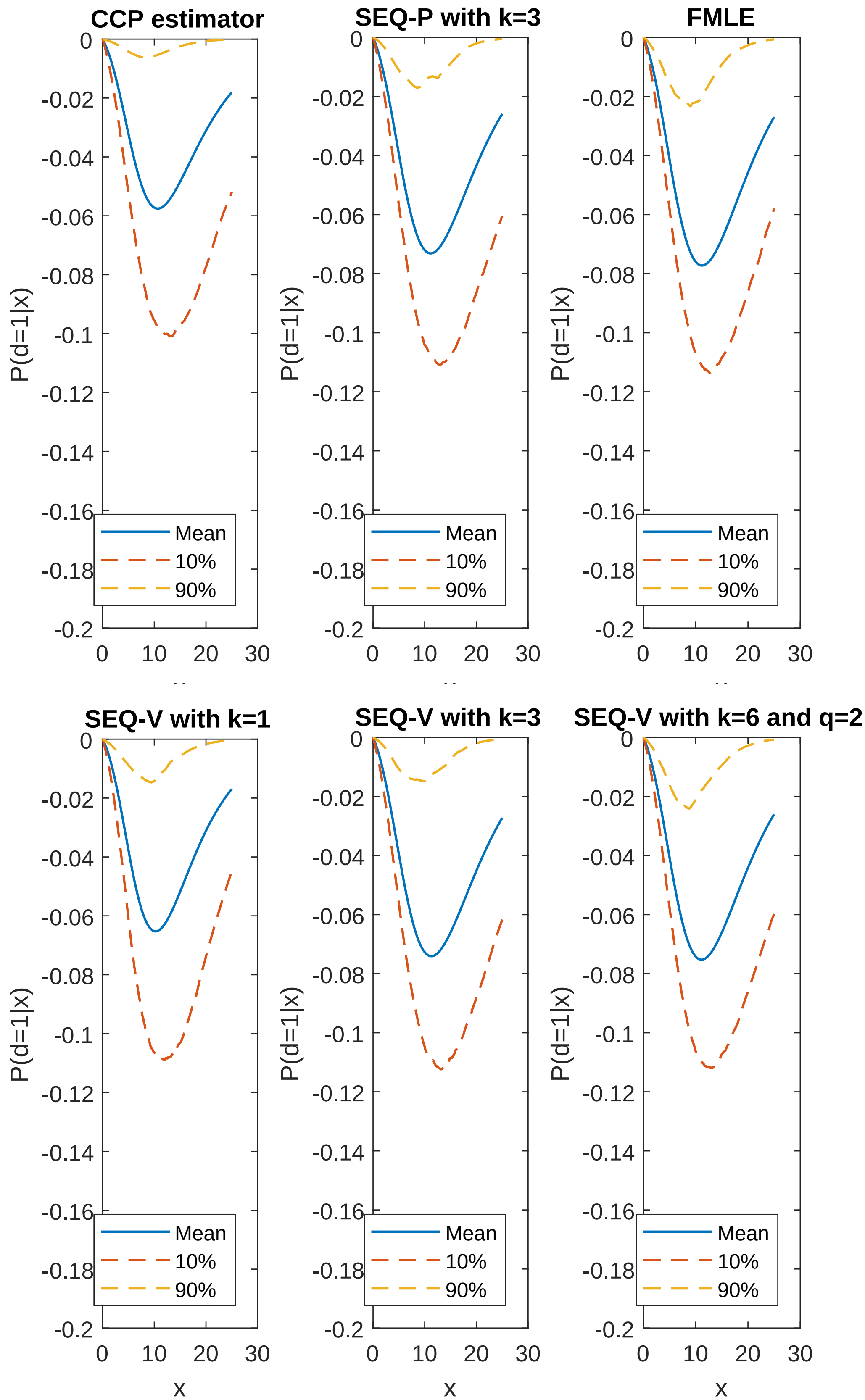

Figure 5: CPU time to complete and the size of state space

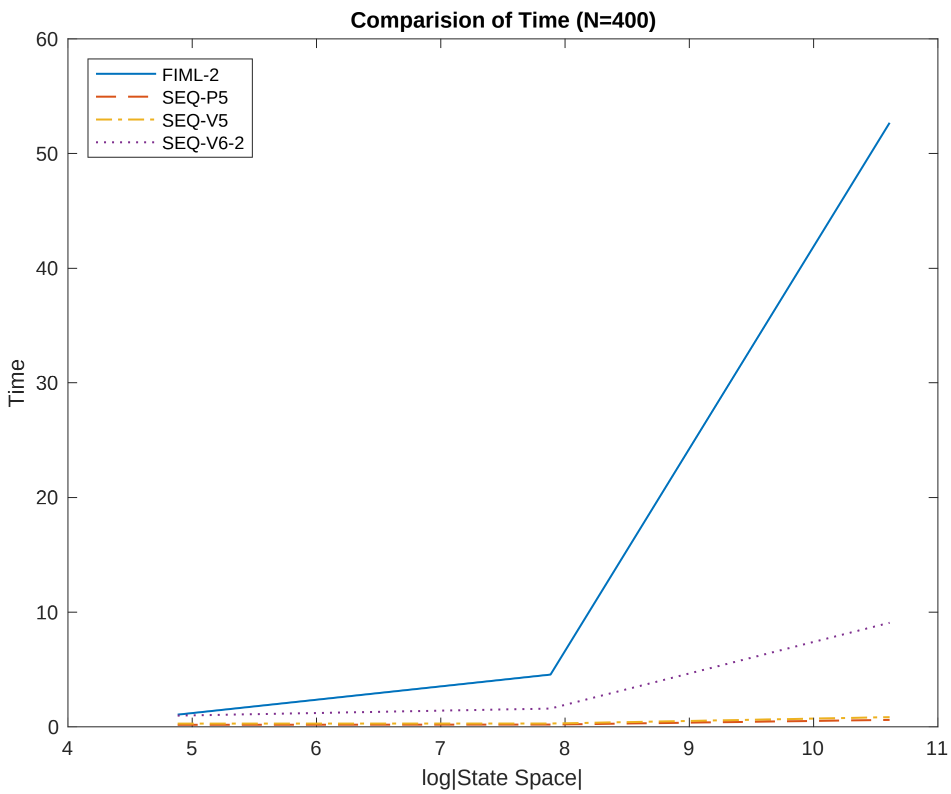

Figure 6: CPU time to complete and the size of state space