

Implementation of Bayesian Inference in Dynamic Discrete Choice Models

Andriy Norets *

anorets@princeton.edu

Department of Economics, Princeton University

August 30, 2009

Abstract

This paper experimentally evaluates a recently developed methodology for Bayesian inference in dynamic discrete choice models. It shows how to construct a Markov Chain Monte Carlo estimation procedure that can handle high-dimensional integration in the likelihood function of these models. It also describes several strategies for improving the efficiency of an algorithm for solving the dynamic program, which is used in conjunction with the estimation procedure. The paper provides a sequence of steps for implementation of reliable and computationally efficient Bayesian inference in dynamic discrete choice models. The experiments are conducted on a model with serially correlated unobserved state variables.

Keywords: dynamic discrete choice models, Bayesian estimation, MCMC, random grids, nearest neighbors.

*I am grateful to John Geweke, my dissertation advisor, for guidance and encouragement through the work on this project. I thank Charles Whiteman for helpful suggestions. I acknowledge financial support from the Economics Department graduate fellowship and the Seashore Dissertation fellowship at the University of Iowa. All remaining errors are mine.

1 INTRODUCTION

A dynamic discrete choice model (DDCM) describes the behavior of a forward-looking economic agent who chooses between several available alternatives repeatedly over time. DDCMs are attractive for empirical research since they are grounded in economic theory. Empirical work employing these models appears in labor economics, industrial organization, international trade, health economics, and marketing literature; see [Aguirregabiria and Mira \(2007\)](#) for a recent survey. Estimation of DDCMs is difficult computationally because the likelihood function of these models generally contains high dimensional integrals and the agent's dynamic optimization program has to be solved on each iteration of an estimation procedure. [Norets \(2009\)](#) proposes a methodology based on advances in Bayesian computational methods that ameliorates these two problems. First, he shows that the Gibbs sampler, employing data augmentation and the Metropolis-Hastings algorithm, can be used to handle the problem of multidimensional integration in the likelihood of DDCMs. Second, extending the work of [Imai et al. \(2005\)](#) he develops an algorithm for solving the dynamic program suitable for use in conjunction with the Gibbs sampler estimation procedure. The algorithm iterates the Bellman equation only once for each parameter draw on a random grid over the state space. To approximate the expected value functions on the current Gibbs sampler iteration, the algorithm uses importance sampling over the value functions from the previous Gibbs sampler iterations that correspond to the nearest neighbors of the current parameter draw. In this paper, I experimentally evaluate the methodology of [Norets \(2009\)](#). I document implementation details

and propose modifications that improve algorithm's efficiency. I provide a sequence of steps that an applied researcher can follow to construct a reliable and computationally efficient Bayesian estimation procedure for a DDCM.

Rust (1987) model of optimal bus engine replacement is used in this paper for illustration purposes. Model simplicity and availability of the data (<http://gemini.econ.umd.edu/jrust/nfxp.html>) make Rust's model very attractive for computational experiments. Several papers used this model for testing new methodologies for estimation of DDCMs, see, for example, Aguirregabiria and Mira (2002) and Bajari et al. (2007). In the original Rust's dynamic logit formulation, the model employs extreme value iid unobservables. The integration over the unobservables in the likelihood and in the solution of the DP in this case can be performed analytically, which results in a logit like likelihood function and a simplified Bellman equation. If, however, the assumption of serial independence in unobservables is relaxed Rust's model becomes more computationally challenging. The dimension of integrals in the model's likelihood function can reach several hundreds. Thus, Rust's model extended to include serially correlated unobservables is suitable for evaluating the estimation method proposed in Norets (2009).

The approach explored in this paper is well suited for estimation of DDCMs with different forms of heterogeneity such as individual specific parameters or serially correlated unobservables. Although experiments on data from Rust (1987) confirm Rust's conclusion of weak evidence of the presence of serial correlation in the unobservables for his model and dataset, experiments on artificial data show that the estimated choice probabilities implied by a dynamic logit model and a model with serially cor-

related unobservables can behave quite differently. More generally, the experiments demonstrate that ignoring individual heterogeneity in estimable DDCMs can lead to serious misspecification errors and that the methodology presented in this paper is a valuable tool for DDCM estimation.

The theoretical framework in [Norets \(2009\)](#) is flexible and leaves room for experimentation. Experiments with the algorithm for solving the DP led to a discovery of modifications that provided increases in speed and precision beyond those anticipated directly by the theory. First, iterating the Bellman equation on several smaller random grids and combining the results turns out to be a very efficient alternative to iterating the Bellman equation on one larger random grid. Second, the approximation error for a difference of expected value functions is considerably smaller than the error for an expected value function by itself (this can be taken into account in the construction of the Gibbs sampler.) Finally, iterating the Bellman equation several times for each parameter draw, using the Gauss-Seidel method and a direct search procedure, also produces significant performance improvement. All these algorithm improvements are carefully documented in the paper.

A verification of the algorithm implementation is provided in the paper. To assess the accuracy of the DP solving algorithm I apply it to a dynamic multinomial logit model, in which the exact DP solution can be quickly computed. The design and implementation of the posterior, prior, and data simulators are checked by joint distribution tests (see [Geweke \(2004\)](#).) Multiple posterior simulator runs are used to check the convergence of the MCMC estimation procedure. The estimation algorithm can be applied to dynamic multinomial logit models, for which an exact algorithm is

also available. A comparison of the estimation results for the approximate algorithm and the exact algorithm suggests that the estimation accuracy is excellent.

The rest of the paper is organized as follows. Section 2 sets up a general DDCM. The presentation is illustrated by Rust (1987) model with added serially correlated unobservables. Bayesian MCMC estimation procedure is described in Section 3. The algorithm for approximation of the DP solution and its implementation for Rust (1987) model are presented in Section 4. This section also describes experiments and improvements for the DP solving algorithm. Tests checking the implementation of the prior, posterior, and data simulators are discussed in Section 5.1. In Section 5.2, the accuracy of the estimation algorithm is evaluated on the dynamic logit model formulation, for which an exact algorithm is available. The role of serial correlation in unobservables is explored in Section 5.3. Estimation results for Rust’s data are presented in Section 5.4. The last section concludes with a summary and directions for future work.

2 MODEL FORMULATION

2.1 General DDCM

A DDCM is a single agent dynamic optimization model with discrete control variables. The per-period utility of the agent $u(s_t, d_t; \theta)$ depends on current state variables $s_t \in S$, the chosen alternative $d_t \in D(s_t)$, and a vector of parameters $\theta \in \Theta$. The state variables are assumed to evolve according to a controlled first order Markov process with a transition law denoted by $f(s_{t+1}|s_t, d_t; \theta)$. Time is discounted with a

factor β . In the recursive formulation, the lifetime utility of the agent or the value function is given by the maximum of the alternative-specific value functions:

$$V(s_t; \theta) = \max_{d_t \in D(s_t)} \mathcal{V}(s_t, d_t; \theta) \quad (1)$$

$$\mathcal{V}(s_t, d_t; \theta) = u(s_t, d_t; \theta) + \beta E\{V(s_{t+1}; \theta) | s_t, d_t; \theta\} \quad (2)$$

This formulation embraces a finite horizon case if time t is included in the vector of the state variables.

2.2 Rust's model

[Rust \(1987\)](#) used a binary choice model of optimal bus engine replacement to demonstrate his dynamic logit model. In this model a maintenance superintendent of a bus transportation company decides every time period whether to replace a bus engine. The observed state variable is the bus mileage x_t since the last engine replacement. The control variable d_t takes on two values: 2 if the engine is replaced at t and 1 otherwise. The per-period utility function of the superintendent is the negative of per-period costs:

$$u(x_t, \epsilon_t, \nu_t, d_t; \alpha) = \begin{cases} \alpha_1 x_t + \epsilon_t & \text{if } d_t = 1 \\ \alpha_2 + \nu_t & \text{if } d_t = 2 \end{cases}$$

where ϵ_t and ν_t are the unobserved state variables ($s_t = (x_t, \epsilon_t, \nu_t)$), α_1 is the negative of per-period maintenance costs per unit of mileage, α_2 is the negative of the costs

of engine replacement. Rust assumes that ϵ_t and ν_t are extreme value iid. I assume ν_t is iid $N(0, h_\nu^{-1})$ truncated to $[-\bar{\nu}, \bar{\nu}]$, ϵ_t is $N(\rho\epsilon_{t-1}, h_\epsilon^{-1})$ truncated to $E = [-\bar{\epsilon}, \bar{\epsilon}]$, and $\epsilon_0 = 0$. A newly replaced engine is assumed to have no defects initially and thus ϵ_t is reset to zero if the engine is replaced at t . The bus mileage since the last replacement is discretized into M intervals $X = \{1, \dots, M\}$. The observed state x_t evolves according to

$$P(x_{t+1}|x_t, d_t; \eta) = \begin{cases} \pi(x_{t+1} - x_t; \eta) & \text{if } d_t = 1 \\ \pi(x_{t+1} - 1; \eta) & \text{if } d_t = 2 \end{cases} \quad (3)$$

and

$$\pi(\Delta x; \eta) = \begin{cases} \eta_1 & \text{if } \Delta x = 0 \\ \eta_2 & \text{if } \Delta x = 1 \\ \eta_3 & \text{if } \Delta x = 2 \\ 0 & \text{if } \Delta x \geq 3 \end{cases}$$

Rust assumes that if the mileage reaches the state M it stays in this state with probability 1. I instead assume that the engine is replaced at t if x_t exceeds $M - 1$, which slightly simplifies the DP solution. In the recursive formulation, the life-time utility for $x_t < M$ is given by

$$V(x_t, \epsilon_t, \nu_t; \theta) = \max\left\{ \begin{aligned} &\alpha_1 x_t + \epsilon_t + \beta \sum_{k=1}^3 \eta_k E[V(x_t + k - 1, \epsilon', \nu'; \theta) | \epsilon_t; \theta], \\ &\alpha_2 + \nu_t + \beta EV_2(\theta) \end{aligned} \right\} \quad (4)$$

where

$$EV_2(\theta) = \sum_{k=1}^3 \eta_k E[V(k, \epsilon', \nu'; \theta) | 0; \theta]$$

$$E[V(x_{t+1}, \epsilon', \nu'; \theta) | \epsilon_t; \theta] = \int V(x_{t+1}, \epsilon', \nu'; \theta) dP(\epsilon', \nu' | \epsilon_t; \theta)$$

For $x_t \geq M$:

$$V(x_t, \epsilon_t, \nu_t; \theta) = \alpha_2 + \nu_t + \beta EV_2(\theta)$$

3 BAYESIAN ESTIMATION

In an estimable dynamic discrete choice model it is usually assumed that some state variables are unobserved by econometricians. Let's denote the unobserved part of the state variables by y_t and the observed part by x_t . The set of the available alternatives $D(s_t)$ is assumed to depend only on the observed state variables. Hereafter, it will be denoted by D to simplify the notation. This is without loss of generality since we could set $D = \cup_{x_t \in X} D(x_t)$ and the alternatives unavailable at state x_t could be assigned a low per-period utility value.

A data set that is usually used for the estimation of a dynamic discrete choice model consists of a panel of I individuals. The observed part of the state and the decisions are known for each individual $i \in \{1, \dots, I\}$ for T_i periods: $\{x_{t,i}, d_{t,i}\}_{t=1}^{T_i}$. Assuming that the state variables are independent for the individuals in the sample,

the likelihood for the model can be written as

$$p(\{x_{t,i}, d_{t,i}\}_{t=1}^{T_i}, i \in \{1, \dots, I\} | \theta) = \prod_{i=1}^I p(x_{T_i,i}, d_{T_i,i}, \dots, x_{1,i}, d_{1,i} | \theta) = \quad (5)$$

$$\prod_{i=1}^I \int p(y_{T_i,i}, x_{T_i,i}, d_{T_i,i}, \dots, y_{1,i}, x_{1,i}, d_{1,i} | \theta) dy_{T_i,i}, \dots, dy_{1,i}$$

The joint density $p(y_{T_i,i}, x_{T_i,i}, d_{T_i,i}, \dots, y_{1,i}, x_{1,i}, d_{1,i} | \theta)$ could be decomposed as follows

$$p(y_{T_i,i}, x_{t,i}, d_{t,i}, \dots, y_{1,i}, x_{1,i}, d_{1,i} | \theta) \quad (6)$$

$$= \prod_{t=1}^{T_i} p(d_{t,i} | y_{t,i}, x_{t,i}; \theta) f(x_{t,i}, y_{t,i} | x_{t-1,i}, y_{t-1,i}, d_{t-1,i}; \theta)$$

where $f(\cdot | \cdot; \theta)$ is the state transition density, $\{x_{0,i}, y_{0,i}, d_{0,i}\} = \emptyset$, and $p(d_{t,i} | y_{t,i}, x_{t,i}; \theta)$ is a choice probability conditional on all state variables.

In general, evaluation of the likelihood function in (5) involves computing multi-dimensional integrals of an order equal to T_i times the number of components in y_t , which becomes very difficult for large T_i and/or multi-dimensional unobservables y_t . In a Bayesian framework, we do not need to perform the high dimensional integration over y_t at each iteration of the estimation procedure. Instead, we can explore the joint posterior distribution of the parameters and latent variables by a simulation method called Gibbs sampling. In models with latent variables, the Gibbs sampler simulates the parameters conditional on the data and the latent variables, and then simulates the latent variables conditional on the data and the parameters. The resulting sequence of the simulated parameters and latent variables is a Markov chain with the stationary distribution equal to the joint posterior distribution of the parameters

and the latent variables. The marginal posterior distribution of the parameters and parameter point estimates can be obtained from the output of this Markov chain.

Parameterizations of the Gibbs sampler that use all the state variables directly as the latent variables is very inefficient or even infeasible. In this case densities for all blocks of the Gibbs sampler are proportional to a product of expressions in (6) over i multiplied by a prior density for parameters. Expression (6) includes

$$p(d_{t,i}|x_{t,i}, y_{t,i}; \theta) = 1\{\mathcal{V}(y_{t,i}, x_{t,i}, d_{t,i}; \theta) \geq \mathcal{V}(y_{t,i}, x_{t,i}, d; \theta), \forall d \in D\}(y_{t,i}, x_{t,i}, d_{t,i}; \theta).$$

These indicator functions describe a set of inequalities that are nonlinear in parameters. Thus, densities of Gibbs sampler blocks for parameters will have support determined by $\sum_i T_i$ nonlinear inequalities. Simulating from such distributions is a challenging problem (in experiments, draws from an acceptance sampling algorithm were never accepted for $I \geq 100$). Moreover, the Gibbs sampler might not be ergodic in this case (see Section 3.2 below for a discussion of ergodicity).

The complicated truncated densities in the Gibbs sampler can be avoided if one uses $\mathcal{V}_{t,i} = \{\mathcal{V}_{t,d,i} = \mathcal{V}(s_{t,i}, d; \theta), d \in D\}$ or their differences as latent variables in the sampler instead of some of the state variables. Construction of the Gibbs sampler with such a parametrization for Rust's model is described below in detail. MCMC algorithms for other DDCMs can be constructed in a similar manner.

3.1 Gibbs sampler for Rust's model

The dataset includes observations on I buses. Each bus i is observed over T_i time periods: $\{x_{t,i}, d_{t,i}\}_{t=1}^{T_i}$ for $i = 1, \dots, I$. The parameters are $\theta = (\alpha, \eta, \rho, h_\epsilon)$; h_ν is fixed for normalization. It is convenient to parameterize the Gibbs sampler in terms of the alternative specific value functions rather than the unobserved state variables (a detailed discussion of Gibbs sampler parameterization for DDCMs can be found in [Norets \(2009\)](#), Section 2.) The latent variables used in the Gibbs sampler are $\{\Delta\mathcal{V}_{t,i}, \epsilon_{t,i}\}_{t=1}^{T_i}$, $i = 1, \dots, I$.

$$\Delta\mathcal{V}_{t,i} = x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} - \nu_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})$$

where $F_{t,i}(\theta, \epsilon) = \beta \sum_{j=1}^3 \eta_j (E[V(x_{t,i} + j - 1, \epsilon', \nu'; \theta) | \epsilon; \theta] - EV_2(\theta))$.

The compact space for parameters Θ is defined as follows: $\alpha_i \in [-\bar{\alpha}, \bar{\alpha}]$, $\rho \in [-\bar{\rho}, \bar{\rho}]$, $h_\epsilon \in [\bar{h}_\epsilon^l, \bar{h}_\epsilon^r]$, and η belongs to a three dimensional simplex. The joint distribution of the data, the parameters, and the latent variables is

$$p(\theta; \{x_{t,i}, d_{t,i}; \Delta\mathcal{V}_{t,i}, \epsilon_{t,i}\}_{t=1}^{T_i}; i = 1, \dots, I) = \\ p(\theta) \prod_{i=1}^I \prod_{t=1}^{T_i} [p(d_{t,i} | \Delta\mathcal{V}_{t,i}) p(\Delta\mathcal{V}_{t,i} | x_{t,i}, \epsilon_{t,i}; \theta) p(x_{t,i} | x_{t-1,i}; d_{t-1,i}; \eta) p(\epsilon_{t,i} | \epsilon_{t-1,i}, \rho, h_\epsilon)]$$

where $p(\theta)$ is a prior density for the parameters; $p(x_{t,i} | x_{t-1,i}; d_{t-1,i}; \eta)$ is given in [\(3\)](#)

and $p(x_{1,i}|x_{0,i}; d_{0,i}; \eta) = 1_{\{1\}}(x_{1,i})$ —all the buses start with a new engine;

$$p(d_{t,i}|\Delta\mathcal{V}_{t,i}) = \begin{cases} 1, & \text{if } d_{t,i} = 1, \Delta\mathcal{V}_{t,i} \geq 0 \text{ or } d_{t,i} = 2, \Delta\mathcal{V}_{t,i} \leq 0 \\ 0, & \text{if } d_{t,i} = 1, \Delta\mathcal{V}_{t,i} < 0 \text{ or } d_{t,i} = 2, \Delta\mathcal{V}_{t,i} > 0 \end{cases}$$

$$\begin{aligned} p(\Delta\mathcal{V}_{t,i}|x_{t,i}, \epsilon_{t,i}; \theta) &= \exp\{-0.5h_\nu(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])^2\} \quad (7) \\ &\cdot 1_{[-\bar{\nu}, \bar{\nu}]}(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \quad (8) \\ &\cdot \frac{h_\nu^{0.5}}{\sqrt{2\pi}[\Phi(\bar{\nu}h_\nu^{0.5}) - \Phi(-\bar{\nu}h_\nu^{0.5})]} \end{aligned}$$

$$p(\epsilon_{t,i}|\epsilon_{t-1,i}, \theta) = \frac{h_\epsilon^{1/2} \exp\{-0.5h_\epsilon(\epsilon_{t,i} - \rho\epsilon_{t-1,i})^2\}}{\sqrt{2\pi}[\Phi([\bar{\epsilon} - \rho\epsilon_{t-1,i}]h_\epsilon^{0.5}) - \Phi([- \bar{\epsilon} - \rho\epsilon_{t-1,i}]h_\epsilon^{0.5})]} 1_E(\epsilon_{t,i})$$

Gibbs sampler blocks

The Gibbs sampler block for $\Delta\mathcal{V}_{t,i}|\dots$ has a normal truncated distribution proportional to (7) and (8), and also truncated to R^+ if $d_{t,i} = 1$ or to R^- otherwise. An algorithm from Geweke (1991) is used to simulate efficiently from the normal distribution truncated to R^+ (or R^- .) Acceptance sampling handles the truncation in (8).

The density for $\epsilon_{t,i}|\dots$ is proportional to

$$\begin{aligned} p(\epsilon_{t,i}|\dots) &\propto \frac{\exp\{-0.5h_\nu(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])^2\}}{\Phi([\bar{\epsilon} - \rho\epsilon_{t-1,i}]h_\epsilon^{0.5}) - \Phi([- \bar{\epsilon} - \rho\epsilon_{t-1,i}]h_\epsilon^{0.5})} \\ &\cdot 1_{[-\bar{\nu}, \bar{\nu}]}(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \\ &\cdot \exp\{-0.5h_\epsilon(\epsilon_{t+1,i} - \rho\epsilon_{t,i})^2 - 0.5h_\epsilon(\epsilon_{t,i} - \rho\epsilon_{t-1,i})^2\} \cdot 1_E(\epsilon_{t,i}) \quad (9) \end{aligned}$$

This Gibbs sampler block uses a Metropolis step¹ with a normal truncated transition density proportional to (9). The blocks for $\epsilon_{t,i}$ with $t = 0$ and $t = T_i$ will be similar.

Assuming a normal prior $N(\underline{\rho}, \underline{h}_\rho^{-1})$ truncated to $[-\bar{\rho}, \bar{\rho}]$,

$$\begin{aligned}
p(\rho | \dots) &\propto \frac{\exp\{-0.5 h_\nu \sum_{i,t} (\Delta \mathcal{V}_{t,i} - [x_{t,i} \alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])^2\}}{\prod_{i,t} \Phi([\bar{\epsilon} - \rho \epsilon_{t-1,i}] h_\epsilon^{0.5}) - \Phi([- \bar{\epsilon} - \rho \epsilon_{t-1,i}] h_\epsilon^{0.5})} \\
&\cdot \prod_{i,t} 1_{[-\bar{\nu}, \bar{\nu}]}(\Delta \mathcal{V}_{t,i} - [x_{t,i} \alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \\
&\cdot \exp\{-0.5 \bar{h}_\rho (\rho - \bar{\rho})^2\} \cdot 1_{[-\bar{\rho}, \bar{\rho}]}(\rho)
\end{aligned} \tag{10}$$

where $\bar{h}_\rho = \underline{h}_\rho + \sum_i \sum_{t=2}^{T_i} \epsilon_{t-1,i}^2$ and $\bar{\rho} = \bar{h}_\rho^{-1} (\underline{h}_\rho \underline{\rho} + h_\epsilon \sum_i \sum_{t=2}^{T_i} \epsilon_{t,i} \epsilon_{t-1,i})$. A Metropolis step with a normal truncated transition density proportional to (10) is used for this block.

Assuming a gamma prior $\underline{s}^2 h_\epsilon \sim \chi^2(\underline{df})$, truncated to $[\bar{h}_\epsilon^l, \bar{h}_\epsilon^r]$,

$$\begin{aligned}
p(h_\epsilon | \dots) &\propto \frac{\exp\{-0.5 h_\nu \sum_{i,t} (\Delta \mathcal{V}_{t,i} - [x_{t,i} \alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])^2\}}{\prod_{i,t} \Phi([\bar{\epsilon} - \rho \epsilon_{t-1,i}] h_\epsilon^{0.5}) - \Phi([- \bar{\epsilon} - \rho \epsilon_{t-1,i}] h_\epsilon^{0.5})} \\
&\cdot \prod_{i,t} 1_{[-\bar{\nu}, \bar{\nu}]}(\Delta \mathcal{V}_{t,i} - [x_{t,i} \alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \\
&\cdot h_\epsilon^{(\bar{df}-2)/2} \exp\{-0.5 \bar{s}^2 h_\epsilon\} \cdot 1_{[\bar{h}_\epsilon^l, \bar{h}_\epsilon^r]}(h_\epsilon)
\end{aligned} \tag{11}$$

where $\bar{df} = \underline{df} + \sum_i T_i$ and $\bar{s}^2 = \underline{s}^2 + \sum_i \left(\sum_{t=2}^{T_i} (\epsilon_{t,i} - \rho \epsilon_{t-1,i})^2 + \epsilon_{1,i}^2 \right)$. For this block, I employ a Metropolis step with a truncated gamma transition density proportional to (11); draws from the transition density are obtained by acceptance sampling.

¹ To produce draws from some target distribution Metropolis or Metropolis-Hastings MCMC algorithm only needs values of a kernel of the target density. The draws are simulated from a transition density and they are accepted with probability that depends on the values of the target density kernel and the transition density. For more details, see, e.g., [Chib and Greenberg \(1995\)](#).

Assuming a Dirichlet prior with parameters (a_1, a_2, a_3) ,

$$\begin{aligned}
p(\eta|\dots) &\propto \exp\{-0.5h_\nu \sum_{i,t} (\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])^2\} \\
&\cdot \prod_{i,t} 1_{[-\bar{\nu}, \bar{\nu}]}(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \\
&\cdot \prod_{j=1}^3 \eta_j^{n_j + a_j - 1}
\end{aligned} \tag{12}$$

where $n_j = \sum_i \sum_{t=2}^{T_i} 1_{\{j-1\}}(x_{t,i} - x_{t-1,i})$. A Metropolis step with a Dirichlet transition density proportional to (12) is used in this block.

$$\begin{aligned}
p(\alpha|\dots) &\propto p(\alpha) \exp\{-0.5h_\nu \sum_{i,t} (\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])^2\} \\
&\cdot 1_{[-\bar{\alpha}, \bar{\alpha}] \times [-\bar{\alpha}, \bar{\alpha}]}(\alpha) \cdot \prod_{i,t} 1_{[-\bar{\nu}, \bar{\nu}]}(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])
\end{aligned}$$

A Metropolis-Hastings random walk algorithm is used for this block. The proposal density is normal truncated to $[-\bar{\alpha}, \bar{\alpha}] \times [-\bar{\alpha}, \bar{\alpha}]$ with a mean equal to the current parameter draw and a fixed variance. The variance matrix is usually chosen so that the acceptance probability would be between 0.2 – 0.6.

Although the truncation bounds $\bar{\alpha}$, $\bar{\rho}$, \bar{h}_ϵ^l , \bar{h}_ϵ^r , $\bar{\nu}$, $\bar{\epsilon}$ are required for theoretical results in the next section and [Norets \(2009\)](#), they are not used in experiments below. This is equivalent to choosing the truncation regions to be equal to the ranges of floating point numbers available on a computer used for calculations.

3.2 Uniform ergodicity of Gibbs sampler

The draws from the Gibbs sampler are often used for approximating posterior expectations by sample averages. Under certain conditions, the sample averages converge almost surely to the posterior expectations and a corresponding central limit theorem holds. Uniform ergodicity of the Gibbs sampler—a sufficient condition for these results (see [Tierney \(1994\)](#))—is established by the following theorem for the Gibbs sampler that uses the exact expected value functions. As shown in Section 4 of [Norets \(2009\)](#), this result is also a part of sufficient conditions for the good behavior of the estimation algorithm that uses DP solution approximations.

Theorem 1. *Consider the Gibbs sampler with the following order of blocks at each iteration:*

- 1) $(\Delta\tilde{\mathcal{V}}_{t,i}^{m+1}|\theta^m, \epsilon^m, d, x), \forall t, i;$
- 2) $(\rho^{m+1}|\theta^m, \epsilon^m, \Delta\tilde{\mathcal{V}}^{m+1}, d, x), (\alpha^{m+1}|\dots), (\eta^{m+1}|\dots), (h_\epsilon^{m+1}|\dots);$
- 3) $(\epsilon_{t,i}^{m+1}|\epsilon^m, \theta^{m+1}, \Delta\tilde{\mathcal{V}}^{m+1}), \forall t, i;$
- 4) $(\Delta\mathcal{V}_{t,i}^{m+1}|\theta^{m+1}, \epsilon^{m+1}, d, x), \forall t, i;$

where the blocks were described above. Block 4) is redundant but simplifies the proof.

Assume that the support of $\nu_{t,i}$ is sufficiently large relative to the support of ϵ and θ : $\Phi(-h_\nu^{0.5}\bar{\nu}) < 0.25$ and $\bar{\nu} > 2(\bar{u} + \bar{\epsilon} + \beta\overline{EV})$, where \bar{u} is an upper bound on the absolute value of the deterministic part of the per-period utility function, $\bar{\epsilon}$ is an upper bound on the absolute value of $\epsilon_{t,i}$, and $\overline{EV} = [\bar{u} + \bar{\epsilon} + 1 + 2h_\epsilon^{-1}]/(1 - \beta)$ is an upper bound on the absolute value of the expected value function (see the proof.)

Then, the Gibbs sampler is uniformly ergodic. Thus, by Theorems 3 and 5 in [Tierney \(1994\)](#), for any integrable (w.r.t. posterior) function $z(\Delta\mathcal{V}, \theta, \epsilon)$ the sample average $\bar{z}_n = 1/n \sum_m z(\Delta\mathcal{V}^m, \theta^m, \epsilon^m)$ converges a.s. to the posterior expectation $E(z|d, x)$. If

$E(z^2|d, x) < \infty$ then there exists a real number $\sigma^2(z)$ such that $\sqrt{n}(\bar{z}_n - E(z|d, x))$ converges in distribution to $N(0, \sigma^2(z))$.

The theorem is proved in Appendix A.

4 DP SOLUTION

4.1 Algorithm

In this section I describe a DP solution algorithm that is suitable for use in conjunction with a Bayesian estimation procedure. It was proposed in [Norets \(2009\)](#). The details of the algorithm are presented for Rust's model. The algorithm takes a sequence of parameter draws θ^m , $m = 1, 2, \dots$ as an input from the Gibbs sampler, where m denotes the Gibbs sampler iteration. For each θ^m , the algorithm generates random states $s^{m,j} \in S$, $j = 1, \dots, \hat{N}(m)$. At each random state, the approximations of the value functions $V^m(s^{m,j}; \theta^m)$ are computed by iterating the Bellman equation once. At this one iteration of the Bellman equation, the future expected value functions are computed by importance sampling over value functions from previous iterations $V^{k_1}(s^{k_1,j}; \theta^{k_1})$, where $k_1 \in \{m - N(m), \dots, m - 1\}$ is the index of the nearest neighbor for the current parameter vector θ^m :

$$k_1 = \arg \min_{i \in \{m - N(m), \dots, m - 1\}} \|\theta^m - \theta^i\|$$

and $N(m)$ is the size of the history the algorithm keeps track of. In contrast to conventional value function iteration, this algorithm iterates the Bellman equation

only once for each parameter draw and uses information from previous iterations of the estimation procedure to approximate the expectations in the Bellman equation and thus reduces the computational burden of the DP solution.

The random states $s^{m,j}$ are generated from a density $g(\cdot) > 0$ on S . This density $g(\cdot)$ is used as an importance sampling source density in approximating the expected value functions. The collection of the random states $\{s^{m,j}\}_{j=1}^{\hat{N}(m)}$ will be referred below as the random grid. The number of points in the random grid at iteration m is denoted by $\hat{N}(m)$ and it will be referred below as the size of the random grid (at iteration m .)

For Rust's model, the random grid $\{y^{m,j} = (\nu^{m,j}, \epsilon^{m,j})\}_{j=1}^{\hat{N}(m)}$ is generated from a normal distribution: $\nu_d^{m,j} \sim N(0, h_\nu^{-1})$ and $\epsilon^{m,j} \sim g(\cdot)$, where $g(\cdot)$ is a normal density. For each point on the random grid $y^{m,j}$, $j = 1, \dots, \hat{N}(m)$ and each x , the Bellman equation is iterated once:

$$V^m(x, y^{m,j}; \theta^m) = \max\left\{ \begin{aligned} &\alpha_1^m x + \epsilon^{m,j} + \beta \sum_{k=1}^3 \eta_k \hat{E}^m[V(x+k-1, y'; \theta^m) | y^{m,j}; \theta^m], \\ &\alpha_2^m + \nu^{m,j} + \beta E\hat{V}_2^m \end{aligned} \right\} \quad (13)$$

where the expected value functions are computed by importance sampling,

$$\hat{E}^m[V(x+k-1, y'; \theta^m) | y^{m,j}; \theta^m] = \sum_{r=1}^{\hat{N}(k_1)} V^{k_1}(x+k-1, y^{k_1,r}; \theta^{k_1}) W(\epsilon^{k_1,r}, \epsilon^{k_1,j}, \theta^m) \quad (14)$$

$$E\hat{V}_2^m = \sum_{x=1}^3 \eta_x \sum_{r=1}^{\hat{N}(m)} V^{k_1}(x, y^{k_1,r}; \theta^{k_1}) W(\epsilon^{k_1,r}, 0, \theta^m) \quad (15)$$

The importance sampling weights do not depend on $\nu^{m,j}$ because the target and the importance sampling source densities coincide for $\nu^{m,j}$:

$$W(\epsilon^{k_1,r}, \epsilon, \theta) = \frac{f(\epsilon^{k_1,r}|\epsilon; \theta)/g(\epsilon^{k_1,r})}{\sum_{q=1}^{\hat{N}(k_r)} f(\epsilon^{k_1,q}|\epsilon; \theta)/g(\epsilon^{k_1,q})} \quad (16)$$

The iterations in (13) should be performed for x in descending order. Then, for $k > 1$ the value functions $V^m(x + k - 1, y^{m,r}; \theta^m)$ would be available when $V^m(x, y^{m,j}; \theta^m)$ are being computed. Thus, for $k > 1$, k_1 should be set to m in (14). This procedure is somewhat analogous to the Gauss-Seidel method.

Norets (2009) shows that under weak regularity conditions on primitives of DDCM (essentially, the compactness of the state and parameter spaces and the continuity of the per-period utility function and transition density) and $N(m) \propto m^{\gamma_1}$, $\hat{N}(m) \propto m^{\gamma_2}$, $1 > \gamma_1 > \gamma_2 > 0$, the approximated expected value functions in (14) and (15) converge uniformly almost surely to the true values. He also shows that the output of the Gibbs sampler employing these DP solution approximations can be used to obtain estimates of parameters and other quantities of interest.

4.2 Algorithm improvements

The algorithm described above is flexible and leaves room for experimentation. Experiments led to a discovery of modifications that provided increases in speed and precision beyond those anticipated directly by the theory. In practice, iterating (13) several times in a row for one x before going to the next significantly improves the convergence speed. It happens because the expression for the value function at the

mileage x includes the expected value function at the same x . If the Bellman equation (13) is iterated several times the approximation error in $V^m(x, y^{m,j}; \theta^m)$ becomes smaller and affects the value functions for $\{1, \dots, x-1\}$ much less. Using only already updated $V^m(x, y^{m,j}; \theta^m)$ in computing the expectations further improves the performance. Thus, when (13) is iterated for the first time for a given x , the expectations for $k = 1$ are approximated as follows:

$$\hat{E}^m[V(x+k-1, y'; \theta^m) | y^{m,j}; \theta^m] = \sum_{r=1}^{j-1} V(x, y^{m,r}; \theta^m) W(y^{m,r}, y^{m,j}, \theta^m)$$

For $j = 1$, $\hat{E}^m[V(x+0, y'; \theta^m) | y^{m,1}; \theta^m]$ is a solution of the following equation:

$$\begin{aligned} \hat{E}^m[V(x+0, y'; \theta^m) | y^{m,1}; \theta^m] &= \max\{\alpha_2^m + \beta EV_2^{k_1}(\theta^{k_1}), \\ &\alpha_1^m x + \beta(\eta_1^m \hat{E}^m[V(x+0, y'; \theta^m) | y^{m,1}; \theta^m] \\ &+ \eta_2^m \hat{E}^m[V(x+1, y'; \theta^m) | y^{m,1}; \theta^m] + \eta_3^m \hat{E}^m[V(x+2, y'; \theta^m) | y^{m,1}; \theta^m])\} \end{aligned}$$

This equation is obtained by interchanging the places of the expectation and the max in the Bellman equation. After the first iteration on (13) for a given x , all the expectations are computed according to (14) on the subsequent iterations.

The value function iteration algorithm has linear convergence rates and convergence may slow down significantly near the fixed point. Employing a non-linear optimization procedure helps in obtaining a good approximation precision at reduced computational costs. An additional iteration of the Bellman equations in (13) for the fixed parameter vector θ^m , the random grid $\{y^{m,j}\}_{j=1}^{\hat{N}(m)}$, and all x could be seen as a

mapping that takes EV_2 as an input and updates it. Iterating this mapping produces a sequence of EV_2 's that converges monotonically. Taking this into account improves the performance. Appendix B presents a flowchart of a direct search procedure for finding the mapping fixed point EV_2^m . This procedure is used in the experiments presented below.

Since the Gibbs sampler changes only one or few components of the parameter vector at a time, the previous parameter draw θ^{m-1} turned out to be the nearest neighbor of the current parameter θ^m in most cases. Taking advantage of this observation and keeping track only of one previous iteration saves a significant amount of computer memory. In the estimation experiments presented below, the DP solving algorithm starting from $k_1 = m - 1$ required only 3-10 passages over (13) to find the fixed point EV_2^m or, equivalently, to solve the DP for θ^m on the random grid $\{y^{m,j}\}_{j=1}^{\hat{N}^{(m)}}$.

4.3 Experiments with DP solution

To implement the algorithm I wrote a program in C. The program uses BACC interface to libraries LAPACK, BLAS, and RANLIB for performing matrix operations and random variates generation (BACC is an open source software for Bayesian Analysis, Computation, and Communication available at www2.cirano.qc.ca/~bacc.) Higher level interpreted languages like Matlab would not provide necessary computation speed since the algorithm cannot be sufficiently vectorized. As a matter of future work, the algorithm could be easily parallelized with very significant gains in speed.

A simulation study was conducted to assess the quality of the DP solution ap-

proximations. The study explores how the randomness of the grid affects the approximations for fixed parameters and how these effects change with the random grid size. The parameter values for this experiment are $\theta = (\alpha, \rho, h_\nu, h_\epsilon, \eta) = (-.003, -10, .5, 6/\pi^2, 6/\pi^2, .34, .64, .02)$. First, I generated 1000 random grids $\{y^{m,j}\}_{j=1}^{\hat{N}}$, $m = 1, \dots, 1000$. Then, for each random grid m , I solved the DP as described in the previous section and computed the approximation of $EV_2^m(\theta)$ and the approximation of the difference in expectations $\hat{E}^m[V(80, y'; \theta)|0; \theta] - EV_2^m(\theta)$. These approximations are measurable functions of the random grid realization $\{y^{m,j}\}_{j=1}^{\hat{N}}$ and thus they are random variables. Using kernel smoothing, I estimated densities of those approximations. The estimated densities for $\hat{N} = 100, 500, 1000$ are presented in Figure 1.

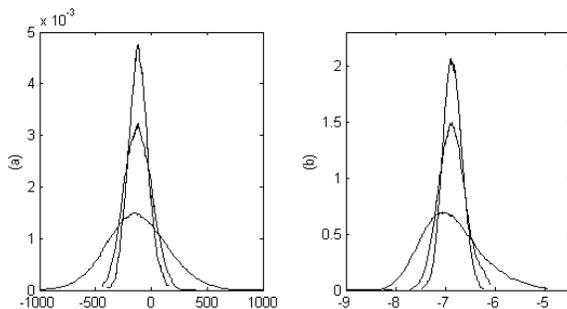


Figure 1: Estimated densities of $EV_2^m(\theta)$ (panel (a)) and $\hat{E}^m[V(80, y'; \theta)|0; \theta] - EV_2^m(\theta)$ (panel (b).) The tightest density corresponds to $\hat{N} = 1000$, the most widespread to $\hat{N} = 100$.

Visual inspection of the figure suggests that the approximations converge as the number of the points in the random grid increases. The mean of the distribution seems to be the same for $\hat{N} = 100, 500, 1000$. The variances are roughly proportional to \hat{N}^{-1} . The densities are also getting close to the fitted normal densities as the grid

size increases. All this hints that an analog to a CLT might hold for this problem. Comparison of the two figures shows that the maximal approximation error for the expected value function is larger by 2–3 orders of magnitude than the maximal approximation error for the difference in the expected value functions. This result seems to have a simple intuitive explanation. An approximate DP solution computed on a random grid could be far from the actual solution. However, the errors resulting from discretization and numerical integration very similarly affect the approximations of the future expected value functions for the same current state but different decisions. It probably happens because numerical integration over the future states is performed on the same random grid no matter which alternative is chosen in the current period. Thus, the approximations of the expected value functions have very high positive correlation and their variances are of similar magnitude. This results in a small variance for their difference. As I mentioned earlier, these findings motivate the choice of the Gibbs sampler parameterization, in which only the differences of the expected value functions are used.

To further verify that the method is implemented correctly I conducted a similar simulation study using the extreme value iid unobservables instead of the serially correlated unobservables. The results were analogous to the ones reported in the figure. The actual DP solution for the iid extreme value unobservables can be easily computed with a high precision as described in [Rust \(1994\)](#). As expected, the exact solutions were right at the means of the distributions obtained from the simulation study.

These experiments also suggest an improvement in the algorithm performance.

Solving the DP on several small random grids and combining the results seems to be a very efficient alternative to using one large grid. I separate the series of the approximations of EV_2 for $\hat{N} = 100$ into batches of size 10. For each batch I compute the mean and then use these means in kernel smoothing to obtain the estimated density for such approximations of EV_2 . The resulting density practically coincide with the density obtained for $\hat{N} = 1000$ and no batching. Thus, the approximation precision for these procedures is about the same. The time of iterating the Bellman equation on a grid of size \hat{N} is roughly proportional to \hat{N}^2 . Therefore, the time required for iterating the Bellman equation on a grid of size $\hat{N} = 100$ for ten different grids will be smaller by a factor of 10 than the time required for iterating the Bellman equation on one grid of size $\hat{N} = 1000$. These experimental results are intriguing. Investigating theoretical properties of this improved procedure, e.g. deriving complexity bounds, seems to be of interest and is a subject of future work. This improvement has not been incorporated into the estimation experiments in this paper. However, they are employed in [Norets \(2007\)](#) that uses artificial neural networks to approximate the expected value function as a function of the parameters θ and the state variables.

5 ESTIMATION EXPERIMENTS

5.1 Joint distribution tests

In order to produce error free reproducible results a researcher should subject an estimation procedure to a number of tests. One such test is to apply the estimation procedure to a large artificial dataset and check if parameter values used for data

generation can be recovered. This is done below, e.g., in Section 5.3. Although it is a useful procedure it often does not reveal subtle problems with simulators. Geweke (2004) developed so called joint distribution tests, which I found to be very useful for debugging and checking the simulator code. Geweke’s test (adopted to a model with latent variables) explores the joint prior distribution of the parameters, latent variables, and data by two different simulators. The first one, called marginal conditional simulator, produces direct draws from the joint prior distribution. In the context of the Rust’s model: $\theta^m \sim p(\theta)$ and $x^m, d^m, \Delta\mathcal{V}^m, \epsilon^m \sim p(x, d, \Delta\mathcal{V}, \epsilon|\theta^m)$. In the other simulator, called successive conditional, the posterior and data simulators are used successively. In the context of the Rust’s model, first, a parameters and latent variables draw $\theta^m, \Delta\mathcal{V}^m, \epsilon^m$ is produced by the posterior simulator described in Section 3.1 conditional on x^{m-1}, d^{m-1} . Second, the latent variables and the data are updated by the data simulator: $x^m, d^m, \Delta\mathcal{V}^m, \epsilon^m \sim p(x, d, \Delta\mathcal{V}, \epsilon|\theta^m)$. If the posterior simulator produces direct draws then the successive conditional simulator is a Gibbs sampler. If the posterior is an MCMC algorithm then the successive conditional simulator is a hybrid MCMC (see Geweke (2005) or Tierney (1994).) In either case the invariant stationary distribution of this Markov chain is the joint prior distribution of the parameters, latent variables, and data. If the prior, posterior and data simulators are derived and implemented correctly, then, for example, the sample mean of θ^m for both simulators should converge to the same values, which could be tested formally using a central limit theorem and an equality of means test. The estimated marginal densities for components of θ should also be the same. The results of the tests for Rust’s model are given below.

Table 1: Joint distribution test

Parameter	α_1	α_2	ρ	h_ϵ	η_1	η_2
Mean, succ. cond.	-0.003	-10.013	0.70086	0.599	0.3399	0.64043
SD, succ. cond.	0.00033	0.668	0.0497	0.0599	0.0473	0.0479
Mean, marg. cond.	-0.003	-10.002	0.6999	0.6004	0.3399	0.64006
SD, marg. cond.	0.00034	0.6694	0.0503	0.0601	0.0470	0.0477
p-value, param.	0.86	0.49	0.48	0.23	0.81	0.48
p-value, param. sqrd.	0.74	0.50	0.49	0.23	0.88	0.46

The test uses 10000 draws from the marginal conditional simulator and 100000 draws from the successive conditional simulator. Using smaller size of the artificial dataset results in better mixing of the chain in the successive conditional simulator. In this experiment $I = 2$. Tighter priors also increase the speed of convergence. The hypothesis of means equality was not rejected by the standard means equality test performed for the parameters and their squares. The means and standard deviations for both simulators and the test p-values are reported in Table 1. The numerical standard errors for the tests were computed by batching with a first order time series correction (the sequence of the draws is divided into batches, the means of the batches are assumed to follow an AR(1) process, and the corresponding standard error is computed.) The numerical standard errors computed by BACC function Expect1 were very similar. The results of the tests are also presented graphically in Figure 2.

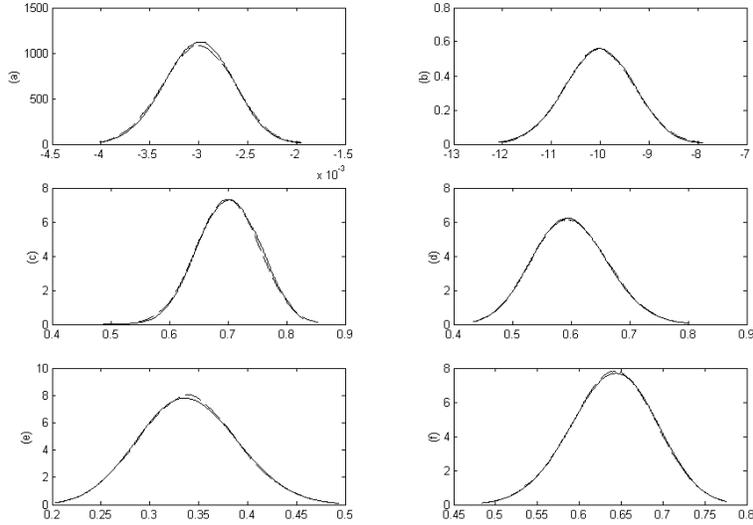


Figure 2: Densities estimated from the output of the marginal conditional and successive conditional simulators: (a) α_1 , (b) α_2 , (c) ρ , (d) h_ϵ , (e) η_1 , (f) η_2 .

The solid lines are the densities of the parameter draws from the successive conditional simulator estimated by kernel smoothing. These densities practically coincide with the dashed lines that show the prior densities. The joint distribution tests show no indication of errors in the simulators (on the debugging stage, a few errors were spotted by the tests.)

5.2 Exact and approximate estimation for dynamic logit

To evaluate the quality of the estimation results I conduct experiments on the model with extreme value unobservables—the dynamic logit model. For this model, the integration over the unobservables in solving the DP and in the likelihood function can be performed analytically. The estimation method that integrates the unobservables

analytically in the likelihood and in the DP solution will be referred below as the exact algorithm. The posterior simulator for this method also uses the Metropolis-Hastings algorithm since the logit-like choice probabilities comprising the likelihood function contain the expected value functions that can only be computed numerically.

The approximate algorithm will refer to the algorithm considered in this paper. The Gibbs sampler for the approximate algorithm is the same as the one for the Gaussian unobservables described in Section 3.1; except here the Gaussian probability densities are replaced by the densities for the extreme value distribution. Table 2 gives the estimation results for the exact and approximate algorithms. The experiments use an artificial dataset consisting of observations on $I = 70$ buses.

Table 2: Exact and approximate estimation results.

	Run	α_1	α_2	η_1	η_2	η_3
Post mean	1	-0.00228	-9.0721	0.34433	0.63394	0.021736
	2	-0.00247	-9.4999	0.34435	0.63392	0.021731
	3	-0.00203	-9.1815	0.3443	0.63397	0.021733
	4	-0.00207	-9.2569	0.34433	0.63394	0.021732
	5	-0.00229	-8.7955	0.34435	0.63393	0.021727
	6	-0.00241	-9.0610	0.34435	0.63392	0.021733
	7	-0.00229	-9.0519	0.34434	0.63392	0.02174
	8	-0.00231	-9.0797	0.34432	0.63395	0.021733
Post SD	1	0.00044	0.8538	0.006311	0.006399	0.001939
	2	0.00049	0.9795	0.006315	0.006403	0.001938
	3	0.00046	0.9681	0.006314	0.0064	0.00194
	4	0.00047	0.9655	0.00631	0.006394	0.001932
	5	0.00042	0.7790	0.006302	0.006396	0.001938
	6	0.00051	0.9789	0.006298	0.006395	0.001938
	7	0.00051	1.0028	0.006327	0.006412	0.001941
	8	0.00049	0.9680	0.006306	0.006396	0.001941
NSE for post mean	1	0.00015	0.3444	7.42E-06	7.31E-06	2.23E-06
	2	0.00019	0.4404	8.48E-06	8.12E-06	2.11E-06
	3	0.00007	0.1892	2.27E-05	2.10E-05	4.16E-06
	4	0.00007	0.2015	2.63E-05	2.36E-05	4.61E-06
	5	0.00005	0.1196	1.90E-05	1.74E-05	3.95E-06
	6	0.00007	0.1957	1.82E-05	1.71E-05	3.88E-06
	7	0.00007	0.1926	2.11E-05	1.97E-05	3.93E-06
	8	0.00006	0.1776	2.04E-05	1.88E-05	3.81E-06
Prior		N(-0.003, .0017)	N(-10, 5)	Dirichlet $a_1 = 34$	prior $a_2 = 64$	$a_3 = 2$
Actual	param	-0.003	-10	0.34	0.64	0.02

Runs 1–2 in the table are the runs of the exact posterior simulator started from different random initial values for the parameters. The length of runs 1–2 was 1000000. For the approximate algorithm three different realization of the random grid for solving the DP were used. Each grid realization corresponds to a pair of simulator runs: 3–4, 5–6, and 7–8. The random number generator was initialized differently for each

run. The length of runs 3–8 was about 500000.

Figure 3 shows the marginal posterior densities for the parameters obtained from the exact and approximate algorithms. The densities were obtained by kernel smoothing over all available simulator runs: 2 runs for the exact algorithm and 6 runs for the approximate algorithm. The results in the figure and table above suggest that the approximation quality of the proposed algorithm is very good.

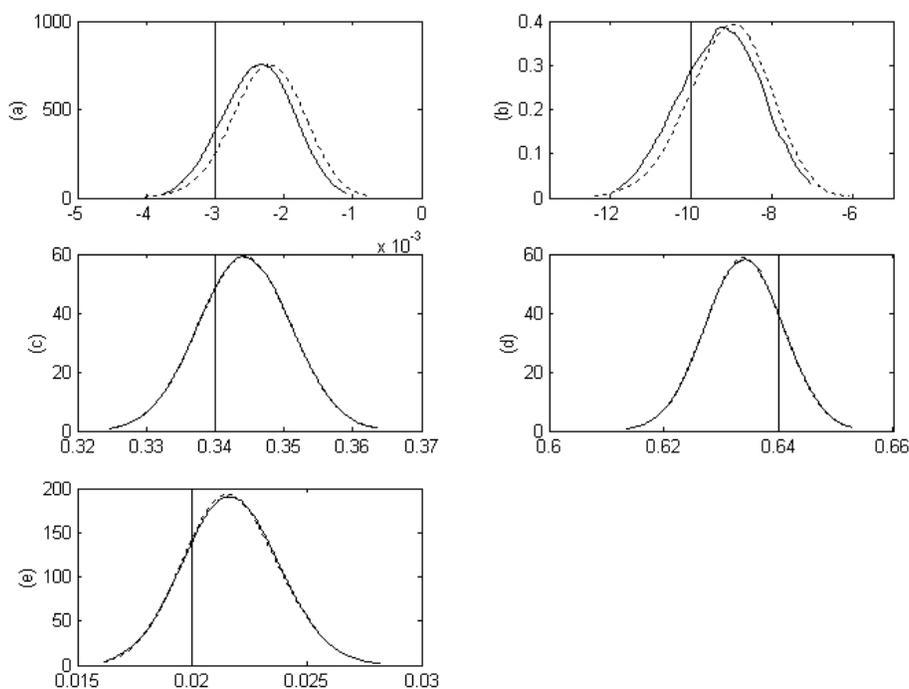


Figure 3: Comparison with exact and approximate estimation algorithms. Estimated posterior densities: (a) α_1 , (b) α_2 , (c) η_1 , (d) η_2 , (e) η_3 . The vertical lines show the actual parameter values. The solid line shows the posterior for exact estimation procedure, the dashed line – approximate estimation procedure.

5.3 The role of serial correlation in unobservables

In this section, I show how the presence of serial correlation in unobservables in the data generation process affects the estimation results for the dynamic logit model. For this purpose I use an artificial dataset simulated from the model with Gaussian serially correlated unobservables described in this paper (it will be referred in this section as the true model.) Then, I use this data in estimation of the dynamic logit model and the true model. The results are shown in the table below.

Table 3: Estimation results for the dynamic logit model and the model with Gaussian serially correlated unobservables.

	Run	α_1	α_2	ρ	η_1	η_2	η_3
Post mean	logit	-0.00091	-3.1431		0.35883	0.62733	0.0138
	true	-0.00276	-10.7342	0.8430	0.35887	0.6273	0.0138
Post SD	logit	0.00065	0.2275		0.012606	0.012703	0.003
	true	0.00098	1.3262	0.0610	0.012607	0.012719	0.003
NSE post mean	logit	0.00001	0.0050		1.38E-05	1.37E-05	3.2E-06
	true	0.00006	0.1848	0.0187	2.02E-05	2.0E-05	3.8E-06
Actual	param	-0.003	-10.0	0.8500	0.34	0.64	0.02
Prior		$N(-.003, .0017)$	$N(-10, 5)$	$N(0.5, 10^3)$ tr.[-.99,.99]	Dirichlet $a_1 = 34$	prior $a_2 = 64$	$a_3 = 2$

From this table, it might seem that the presence of serial correlation in unobservables produces the same effect as an increase in the variance of these unobservables would. The utility function parameters are almost proportional for both cases. To get more insight into the effects of serial correlation in unobservables, I compute the posterior means of the hazard function for each of the models.

As can be seen from panel (a) in Figure 4, the dynamic logit model would underestimate the probability of the engine replacement for low mileage and considerably

overestimate the probability for high mileage if serial correlation is present in the data generation process.

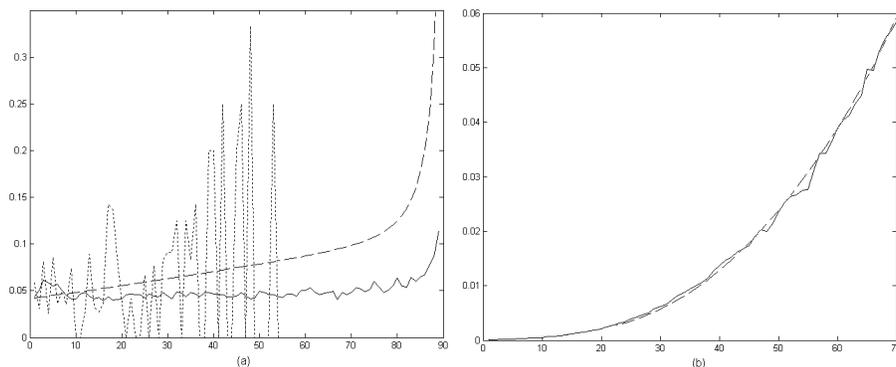


Figure 4: The posterior means of the hazard functions. Panel (a)—the data generated from the model with the serially correlated unobservables, panel (b)—the data generated from the dynamic logit model. The vertical axis is for the probability of engine replacement, the horizontal axis is for the mileage interval. The solid line is for the model with serially correlated unobservables. The dashed line—for dynamic multinomial logit, the dotted line—data hazard.

Moreover, the shape of the hazard function is also different. In the dynamic logit case, the hazard function is increasing, while for the true model it is decreasing at first. Although the estimated hazard is noisy, the decrease at the beginning was observed for several posterior simulator runs; thus it is not a result of the noise. For comparison, panel (b) shows the posterior means of the hazard functions estimated from the artificial data that were simulated from the dynamic logit model. In this case, the hazards for the dynamic logit model and for the model with Gaussian serially correlated unobservables seem to be very close and have the same shape. These

results support the claim that the disparities in the hazards observed in panel (a) are driven by the presence of serial correlation in the data but not by the different distributional assumptions on unobservables: Gaussian vs. extreme value. These experiment demonstrate that ignoring serial correlation in unobservables might lead to serious misspecification errors.

5.4 Estimation results for real data

The data set is group 4 from Rust’s paper. It contains observations on 37 buses that could be divided into $I = 70$ individual spells containing one engine replacement (or censored at the last observed x_t), which gives $\sum_i T_i = 4329$ monthly mileage/decision points. It takes about 50 seconds to produce 100 draws from the posterior on a 2002 vintage PC (the experiments were performed on Unix workstations.)

To start the Gibbs sampler I used the parameter estimates from Rust’s paper. The algorithm also works if the Gibbs sampler is started from a draw from the prior distribution or from the zero vector for the utility function parameters α and the data frequencies for the state transition probabilities η . The initial values for the latent variables are adjusted so that the observed decisions are optimal. In particular, given the parameter values, the serially correlated unobservables $\epsilon_{t,i}$ are simulated from the corresponding AR(1) process. Then, $\Delta\mathcal{V}_{t,i}$ are adjusted to satisfy the observed choice optimality constraints with a small margin. It is also possible to adjust $\epsilon_{t,i}$ together with $\Delta\mathcal{V}_{t,i}$. If the initial values for $\epsilon_{t,i}$ are not simulated from the AR(1) process or if the starting value for ρ is far in the tail of the posterior, a procedure similar to the simulated annealing might be helpful in starting the Gibbs sampler with high

acceptance rates: the acceptance probabilities for the parameters are multiplied by a decreasing quantity on the first hundred iterations.

Estimation results for 6 posterior simulator runs are presented in Table 4 and Figure 5. The number of draws for each simulator run was equal to 1000000.

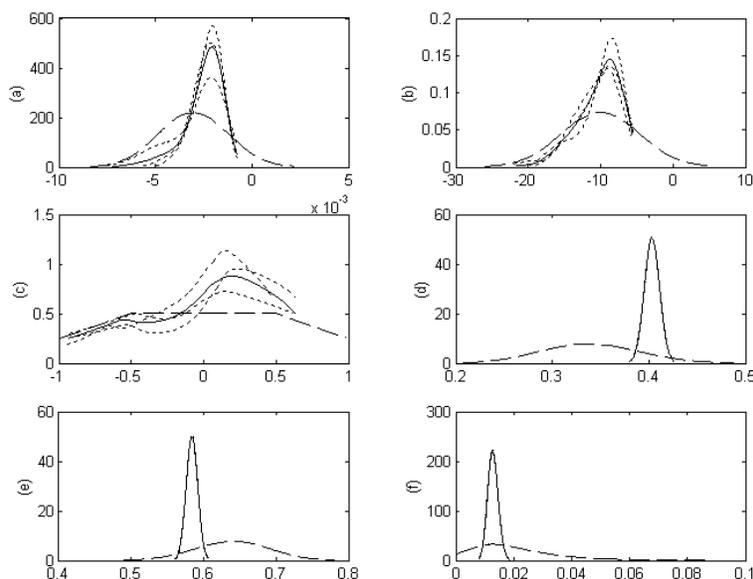


Figure 5: Estimated posterior densities for different grids: (a) α_1 , (b) α_2 , (c) ρ , (d) η_1 , (e) η_2 , (f) η_3 . The dashed lines are prior densities. The solid lines are posterior densities averaged over all simulator runs. The dotted lines show posterior densities averaged for runs 1–2, 3–4, and 5–6.

Three different random grids for solving the DP were used in these experiments (the random grid is generated before the estimation procedure starts and it stays fixed through the simulator run.) One grid was used for runs 1–2, another one for runs 3–4, and the last one for runs 5–6. The random number generator was initialized differently for each run.

Table 4: Estimation results for data from Rust’s paper, group 4.

	Run	α_1	α_2	ρ	η_1	η_2	η_3
Post mean	1	-0.00275	-9.5631	-0.1763	0.40325	0.58388	0.012874
	2	-0.00289	-10.0439	0.0945	0.40325	0.58388	0.012869
	3	-0.00248	-11.2845	0.1545	0.40329	0.58385	0.012868
	4	-0.00228	-9.8963	-0.1000	0.40328	0.58385	0.012868
	5	-0.00224	-10.1331	-0.0526	0.40318	0.58395	0.012875
	6	-0.00233	-10.3774	-0.0573	0.4032	0.58392	0.012872
Post SD	1	0.00123	2.4267	0.4229	0.00736	0.007394	0.001696
	2	0.00146	2.8896	0.2879	0.007381	0.007414	0.001694
	3	0.00084	3.2330	0.4170	0.0074	0.007434	0.001693
	4	0.00075	2.5904	0.4871	0.007388	0.007421	0.001694
	5	0.00066	2.9959	0.4805	0.007365	0.007403	0.001694
	6	0.00075	2.8908	0.4894	0.007379	0.007416	0.001694
NSE for post mean	1	0.00027	0.6762	0.0916	1.65E-05	1.59E-05	2.33E-06
	2	0.00039	0.9795	0.0451	1.71E-05	1.65E-05	2.34E-06
	3	0.00012	1.2031	0.1175	1.94E-05	1.86E-05	2.37E-06
	4	0.00013	0.9552	0.1463	2.23E-05	2.13E-05	2.94E-06
	5	0.00008	0.9910	0.1225	1.49E-05	1.44E-05	2.23E-06
	6	0.00010	0.9460	0.1258	1.41E-05	1.35E-05	2.37E-06
Prior		N(-.003, .0017)	N(-10, 5)	$N(0.5, 10^3)$ tr.[-.99,.99]	Dirichlet $a_1 = 34$	prior $a_2 = 64$	$a_3 = 2$

The figure and table above suggest that only the estimation results for ρ are significantly affected by the random grid realization. The results for η are not affected at all since the data on mileage transitions dominate the posterior for η . The qualitative results for ρ do not seem to depend on the grid realization. The posterior distribution for ρ is bimodal. The higher mode is positive and located at about 0.2, the lower mode is at about -0.6. The posterior mean is close to 0. The posterior probability of $\rho > 0$ is in 0.54–0.66 range. Overall, there seems to be no strong evidence that Rust’s assumption of no serial correlation in the unobservables is invalid.

A more objective criterion for studying the effects of the grid realization on the

estimation results would be to check how it affects conditional choice probabilities or results of some policy changes. If the effects are still present then there are several alternative ways to reduce them. The first one is to estimate the posterior densities from several posterior simulator runs corresponding to different grids. This was done for the experiment above and the resulting densities are shown by the solid lines in Figure 5.

Increasing the size of the grid seems to be a more attractive way to obtain better approximations for the posterior distribution. However, it would increase the computational burden of solving the DP and approximating the expectations in the Gibbs sampler. In both cases this burden can be ameliorated. As was described in Section 4.3, solving the DP on several small random grids and combining the results produces about the same approximation precision as solving the DP on one big random grid. However, using several smaller grids requires much less time. To speed up the approximation of the expectations in the Gibbs sampler a strategy proposed by [Keane and Wolpin \(1994\)](#) could be used. In solving the DP, the authors compute expectations using Monte Carlo integration only for a subset of states in the discretized state space. For the rest of the states the expectations are computed by interpolation. Such an interpolation function could be used for approximating the expectations in the Gibbs sampler. Alternatively, experiments in [Norets \(2007\)](#) suggest that employing artificial neural networks in approximating expected value functions is a promising approach.

6 CONCLUSION AND FUTURE WORK

In this paper, I experimentally evaluate a method for Bayesian inference in DDCMs and provide a sequence of steps for implementing a reliable and efficient estimation procedure. First, the parameterization of the Gibbs sampler should be carefully chosen. Parameterizations in which differences of expected value functions are used in the Gibbs sampler rather than expected value functions by themselves are preferred since experiments demonstrated that differences in expected value functions are much easier to approximate precisely. Second, the posterior, prior, and data simulators should be checked by joint distribution tests. Third, convergence of MCMC should be carefully evaluated. The simplest way to do this is to run multiple chains from randomly generated initial conditions. Fourth, the method for solving the DP should be evaluated on models for which exact solution is easy to obtain, e.g., dynamic logit models. Fifth, solving the DP on several small random grids and combining the results could be a very efficient alternative to using one large grid. Application of parallel computing to this task seems to be a fruitful way to achieve better performance. A number of other improvements in the DP solving algorithm are proposed in the paper. An important direction for future work is to develop strategies for decreasing the amount of correlation in the Gibbs sampler draws and thus reduce the length of the simulator runs required for attaining convergence. Overall, the paper demonstrates that Bayesian methods provide a promising approach to estimating DDCMs.

7 APPENDIX A. Proof of the Gibbs sampler uniform ergodicity

Proof. (Theorem 1)

Consider the following uniform probability density:

$$q(\Delta\mathcal{V}, \theta, \epsilon) = c \cdot 1_{\Theta}(\theta) \prod_{i,t} [1_E(\epsilon_{t,i}) \cdot p(d_{t,i} | \Delta\mathcal{V}_{t,i}) \cdot 1_{[-\bar{v}, \bar{v}]}(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})])] \quad (17)$$

where c is a normalization constant. The corresponding probability measure is denoted by $Q(\cdot)$.

Let's show that the transition probability measure for the Gibbs sampler satisfies the marginalization condition w.r.t. $Q(\cdot)$:

$$P((\mathcal{V}^{m+1}, \theta^{m+1}, \epsilon^{m+1}) \in A | \mathcal{V}^m, \theta^m, \epsilon^m, d, x) \geq bQ(A), \forall \mathcal{V}^m, \theta^m, \epsilon^m$$

where $b > 0$ is a constant. Then, the uniform ergodicity follows from Proposition 2 in [Tierney \(1994\)](#).

$$\begin{aligned}
& P((\mathcal{V}^{m+1}, \theta^{m+1}, \epsilon^{m+1}) \in A | \mathcal{V}^m, \theta^m, \epsilon^m, d, x) \\
&= \int_{R \times \dots \times R} \int_A \prod_{t,i} p(\Delta \tilde{\mathcal{V}}_{t,i}^{m+1} | \theta^m, \epsilon^m, d, x) \\
&\quad p(\rho^{m+1} | \theta^m, \epsilon^m, \Delta \tilde{\mathcal{V}}^{m+1}, d, x) \cdot p(\alpha^{m+1} | \rho^{m+1}, \eta^m, h_\epsilon^m, \alpha^m, \epsilon^m, \Delta \tilde{\mathcal{V}}^{m+1}, d, x) \\
&\quad p(\eta^{m+1} | \dots) \cdot p(h_\epsilon^{m+1} | \dots) \prod_{t,i} p(\epsilon_{t,i}^{m+1} | \dots) \\
&\quad \prod_{t,i} p(\Delta \mathcal{V}_{t,i}^{m+1} | \theta^{m+1}, \epsilon^{m+1}, d, x) d(\Delta \tilde{\mathcal{V}}^{m+1}, \theta^{m+1}, \epsilon^{m+1}, \Delta \mathcal{V}^{m+1})
\end{aligned} \tag{18}$$

where $p(\cdot|\cdot)$ are the densities for the Gibbs sampler blocks. The densities for the blocks with a Metropolis-Hastings step can be written in terms of the Dirac delta function (see, for example, chapter 4 in [Geweke \(2005\)](#).)

Given the assumptions on the support of $\nu_{t,i}$ let's show that there exist $\delta_1 > 0$ such that $\Delta \mathcal{V}_{t,i} \in [-\delta_1, \delta_1]$ implies $(\Delta \mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \in [-\bar{\nu}, \bar{\nu}]$, $\forall \theta, \epsilon$. It was stated in the formulation of the theorem that \overline{EV} is an upper bound on the absolute value of the expected value function. Note that an upper bound on the expected value function EV^{ub} exists. Let's show that it is no greater than \overline{EV} .

$$\begin{aligned}
E[|V(s'; \theta)|; |s, d; \theta] &= E[|\max\{\alpha_1 x + \epsilon + \beta E[V(s''; \theta) | s', d_1; \theta], \\
&\quad \alpha_2 + \nu + \beta E[V(s''; \theta) | s', d_2; \theta]\}|] \\
&\leq \bar{u} + \bar{\epsilon} + E[|\nu|] + \beta EV^{ub}
\end{aligned} \tag{19}$$

It was also assumed in the theorem that $\Phi(-\bar{\nu}) < 0.25$, which implies $E[|\nu|] \leq$

$1 + E[\nu^2] \leq 1 + 2h_\nu^{-1}$. Since (19) holds for any (s, d, θ) :

$$EV^{ub} \leq \frac{\bar{u} + \bar{\epsilon} + (1 + 2h_\epsilon^{-1})}{1 - \beta} = \overline{EV}$$

Therefore,

$$|[x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]| \leq 2(\bar{u} + \bar{\epsilon} + \beta\overline{EV})$$

Let $\delta_1 = \bar{\nu} - 2(\bar{u} + \bar{\epsilon} + \beta\overline{EV})$, which is positive by the assumption of the theorem.

Thus, for $|\Delta\mathcal{V}_{t,i}| \leq \delta_1$,

$$|\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]| \leq \delta_1 + 2(\bar{u} + \bar{\epsilon} + \beta\overline{EV}) = \bar{\nu}$$

To find a lower bound on the integral in (18), let's restrict the integration over $\Delta\tilde{\mathcal{V}}_{t,i}^{m+1}$ to $|\Delta\tilde{\mathcal{V}}_{t,i}^{m+1}| \leq \delta_1$ and use only the parts of the block densities corresponding to the accepted draws. The parts of the block densities for the accepted draws are equal to the MH transition densities multiplied by the acceptance probabilities. For $(\Delta\mathcal{V}_{t,i} - [x_{t,i}\alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \in [-\bar{\nu}, \bar{\nu}]$, these densities for the accepted draws are positive and continuous on Θ, E , and $[\Delta\mathcal{V}_{t,i} \geq 0]$ (or $[\Delta\mathcal{V}_{t,i} < 0]$ depending on $d_{t,i}$) for all blocks, and thus bounded away from zero. Let's denote the common

bound by $\delta > 0$. Then,

$$\begin{aligned}
P((\mathcal{V}^{m+1}, \theta^{m+1}, \epsilon^{m+1}) \in A | \mathcal{V}^m, \theta^m, \epsilon^m, d, x) &\geq \left(\prod_{t,i} \delta_1 \delta \right) \\
&\cdot \int_A 1_{\Theta}(\theta^{m+1}) \cdot \delta^4 \cdot \prod_{t,i} [\delta \cdot 1_E(\epsilon_{t,i})] \cdot \prod_{t,i} \delta \cdot p(d_{t,i} | \Delta \mathcal{V}_{t,i}) \\
&\cdot 1_{[-\bar{v}, \bar{v}]}(\Delta \mathcal{V}_{t,i} - [x_{t,i} \alpha_1 - \alpha_2 + \epsilon_{t,i} + F_{t,i}(\theta, \epsilon_{t,i})]) \\
d(\theta^{m+1}, \epsilon^{m+1}, \Delta \mathcal{V}^{m+1}) &= \frac{1}{c} \left(\prod_{t,i} \delta \right)^2 \cdot \delta^4 \cdot \prod_{t,i} \delta_1 \cdot Q(A)
\end{aligned}$$

Also, since $Q(\cdot)$ is absolutely continuous w.r.t. the posterior probability measure, the transition probability measure for the Gibbs sampler is irreducible w.r.t. the posterior probability measure. This completes the proof of the uniform ergodicity of the Gibbs sampler. \square

8 APPENDIX B. Direct search procedure

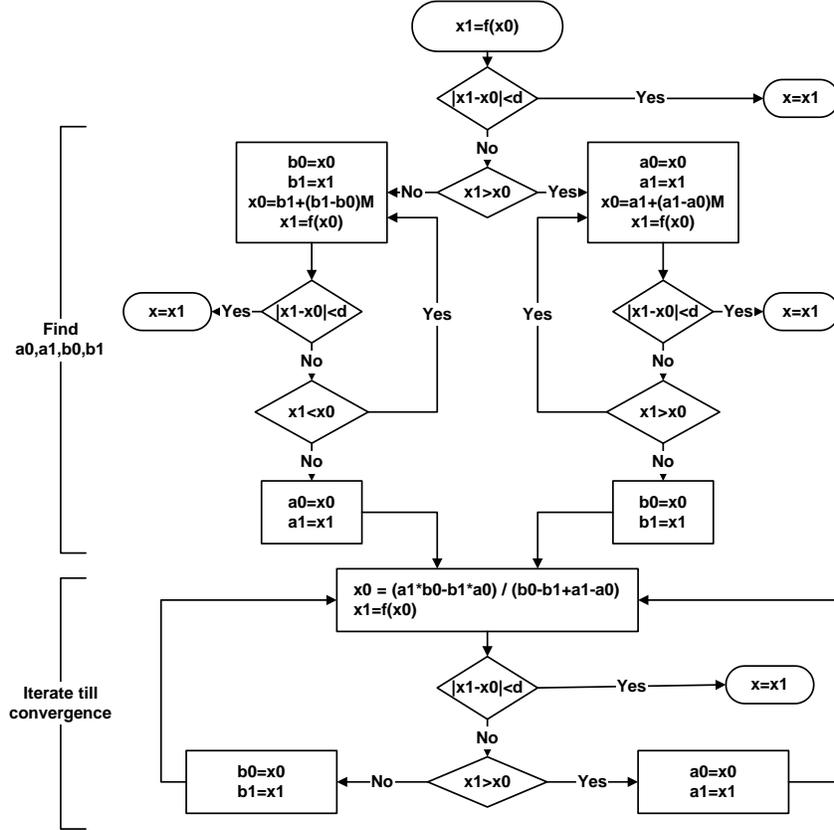


Figure 6: Flow chart of a direct search procedure for finding a fixed point.

$f(\cdot)$ in the flowchart denotes a mapping that takes EV_2 as an input and returns an updated value of EV_2 iterating the Bellman equations once. The algorithm searches for a fixed point $x = f(x)$. First, the algorithm finds bounds a_0, a_1, b_0, b_1 : $a_0 < a_1 = f(a_0) \leq x \leq b_1 = f(b_0) < b_0$ starting with x_0 . A scaling factor M is chosen experimentally. Updated x is obtained by cutting interval $[a_1, b_1]$ in proportions $(a_1 - a_0) : (b_0 - b_1)$. Note that this updated x is chosen to be the fixed point of

a linear approximation to $f(\cdot)$, as illustrated in Figure 7. After each iteration the difference $f(x_0) - x_0$ is compared to a tolerance parameter d . If the convergence has not been achieved a_0, a_1, b_0, b_1 are updated and the procedure is repeated.

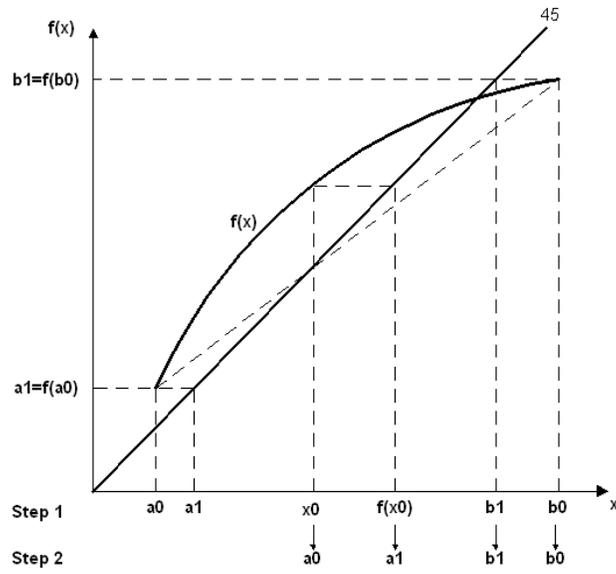


Figure 7: Geometric interpretation of the direct search procedure.

References

- Aguirregabiria, V. and Mira, P. Swapping the nested fixed point algorithm: A class of estimators for discrete markov decision models. *Econometrica*, 70:1519–1543, 2002.
- Aguirregabiria, V. and Mira, P. Dynamic discrete choice structural models: A survey, 2007.
- Bajari, P., Benkard, L., and Levin, J. Estimating dynamic models of imperfect competition. *Econometrica*, 75(5):1331–1370, 2007.
- Chib, S. and Greenberg, E. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.
- Geweke, J. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities, 1991.
- Geweke, J. Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Association*, 99:799–804, 2004.
- Geweke, J. *Contemporary Bayesian Econometrics and Statistics*. Wiley-Interscience, 2005.
- Imai, S., Jain, N., and Ching, A. Bayesian estimation of dynamic discrete choice models, 2005.

- Keane, M. and Wolpin, K. The solution and estimation of discrete choice dynamic programming models by simulation and interpolation: Monte carlo evidence. *Review of Economics and Statistics*, 76(4):648–672, 1994.
- Norets, A. Estimation of dynamic discrete choice models using artificial neural networks approximations, 2007.
- Norets, A. Inference in dynamic discrete choice models with serially correlated unobserved state variables. *Econometrica*, forthcoming, 2009.
- Rust, J. Optimal replacement of gmc bus engines: an empirical model of harold zurcher. *Econometrica*, 55(5):999–1033, 1987.
- Rust, J. Structural estimation of markov decision processes. In Engle, R. and McFadden, D., editors, *Handbook of Econometrics*. North Holland, 1994.
- Tierney, L. Markov chains for exploring posterior distributions. *The Annals of Statistics*, Vol. 22, No. 4, 1758-1762, 22(4):1758–1762, 1994.