Title

A new use of importance sampling to reduce computational burden in simulation estimation

Journal

QME: Quantitative Marketing and Economics, 7(4)

ISSN

1573-711X

Author

Ackerberg, Daniel A.

Publication Date

2009-12-01

DOI

10.1007/s11129-009-9074-z

Peer reviewed

Quant Mark Econ (2009) 7:343–376
DOI 10.1007/s11129-009-9074-z

A new use of importance sampling to reduce computational burden in simulation estimation

Daniel A. Ackerberg

Received: 22 April 2008 / Accepted: 27 July 2009 / Published online: 25 August 2009

Abstract Simulation estimators (Lerman and Manski 1981; McFadden, Econometrica 57(5):995–1026, 1989; Pakes and Pollard, Econometrica 57:1027–1057, 1989) have been of great use to applied economists and marketers. They are simple and relatively easy to use, even for very complicated empirical models. That said, they can be computationally demanding, since these complicated models often need to be solved numerically, and these models need to be solved many times within an estimation procedure. This paper suggests methods that combine importance sampling techniques with changes-of-variables to address this caveat. These methods can dramatically reduce the number of times a particular model needs to be solved in an estimation procedure, significantly decreasing computational burden. The methods have other advantages as well, e.g. they can smooth otherwise non-smooth objective functions and can allow one to compute derivatives analytically. There are also caveats—if one is not careful, they can magnify simulation error. We illustrate with examples and a small Monte-Carlo study.

Keywords Simulation estimators · Importance sampling · Monte-Carlo study

JEL Classifications C13 · C16 · C63

1 Introduction

Simulation methods such as Simulated Maximum Likelihood (SML) (Lerman and Manski 1981) and the Method of Simulated Moments (MSM) (McFadden

D. A. Ackerberg (✉)
Dept. of Economics, University of California, Los Angeles, Los Angeles, CA, USA
e-mail: ackerber@econ.ucla.edu

1989; Pakes and Pollard 1989) have great value to applied economists and marketers estimating structural models. However, use of these methods is still limited by computational constraints. This is because one often needs to numerically solve the equilibria of these models many times within an estimation procedure. Often such numeric procedures are CPU intensive. Examples include 1) complicated static equilibrium problems, e.g. discrete games or complicated auction models, and 2) dynamic programming with large state spaces or significant amounts of heterogeneity. In these estimation procedures, straightforward simulation involves solving the equilibrium of such a model numerous times, typically once for every simulation draw; for every observation; for every parameter vector that is ever evaluated in a numeric optimization procedure. If one has N observations, performs S simulation draws per observation, and numeric optimization requires R function evaluations, estimation requires "solving" the model $N * S * R$ times. This can be unwieldy for these complicated problems.

We suggest using a change of variables and importance sampling to help alleviate this problem. Importance sampling (Kloek and Van Dijk 1978) is a very simple numerical integration technique that has been used for various purposes in the prior literature, e.g. for reducing levels of simulation error (e.g. Berry et al. 1995), to smooth (e.g. McFadden 1989, Gasmi et al. 1991) or to aid in Bayesian estimation (e.g. Geweke 1989). We show that importance sampling can be also be used to dramatically reduce the number of times a complicated model needs to be solved within SML and MSM estimation procedures. With this change of variables and importance sampling one only needs to solve the model $N * S$ times or S times, instead of $N * S * R$ times. Since R can be quite large, particularly as the dimension of the parameter vector becomes large, this can lead to very significant time savings.

There are also a number of additional benefits of our importance sampling procedure. As illustrated in a simple model by McFadden (1989), importance sampling procedures can *smooth* objective functions that would be non-smooth using straightforward simulation techniques. Second, the objective functions generated by our procedure will often have *analytic* first (and second) derivatives. This can also generate time savings and increase accuracy when one is using derivative-based optimization procedures, because one does not need to use numerical derivatives. Lastly, the importance sampling technique is easily parallelizable, which can generate increased time savings with modern multiprocessor computers.

There is another strand of literature on computational methods in dynamic structural models, starting with Hotz and Miller (1993), and continuing with Hotz et al. (1994), Aguirregabiria and Mira (2002, 2007), Jofre-Bonet and Pesendorfer (2003), Bajari et al. (2007a), Pakes et al. (2007), Pesendorfer and Schmidt-Dengler (2008), and Bajari et al. (2008). These are useful and popular techniques, and they can usually reduce computation time by *more* than our methods (since they often never require explicit solution of equilibrium strategies). But we believe that our suggested techniques can still be valuable, since they are very straightforward to use in models that allow for lots of unobserved

heterogeneity (e.g. consumer or firm-specific, time-invariant unobservables). In contrast, the HM-related techniques can become considerably harder to use when there is more unobserved heterogeneity. Our importance sampling approach is actually more related to the approaches suggested by Keane and Wolpin (1994) and Rust (1987). We discuss how they relate and how the two approaches can be complementary.[1]

Since a prior working paper version of this paper (Ackerberg 2001), a number of researchers have used our procedure successfully, e.g. Hartmann (2006), Pantano (2008), Bajari et al. (2009a), Goettler and Clay (2009). Also, Bajari et al. (2007b) (BFR) (and Bajari et al. 2009a) have suggested a related set of techniques. Their techniques can reduce computational burden of numerically challenging dynamic programming problems in a way similar to our importance sampling based techniques. Both their approach and our approach essentially presolve the dynamic programming model a fixed number of times, and input these pre-solved solutions into an optimization problem. Deciding which is preferable for a given situation might depend on whether one prefers to work with parameterized continuous distributions, or with more discrete non-parametric approximations. Our suggested techniques/models are more naturally parametric (of course, these parametric specifications can be chosen flexibly), while the BFR techniques are more naturally non-parametric. An advantage of the BFR technique is that in many cases, computation of the parameter estimates reduces to a linear programming problem. Numerically, these are easier and more reliably solved than the non-linear programming problems generally used in structural estimation (and in our suggested techniques). One the other hand, because it is naturally non-parametric, one may need considerable amounts of data to use the BFR approach effectively, especially with high dimensional unobserved heterogeneity.

While these ideas can potentially be very powerful in reducing computational burden, the use of importance sampling is not without an important caveat. Importance sampling can dramatically change variance properties of simulation based estimators (either increasing or decreasing simulation error). It has the potential to significantly increase levels of simulation error, particularly with high dimensional unobserved heterogeneity. This can be much more dangerous than straightforward simulators that don't involve importance sampling. Often, standard simulation estimators have a very useful property—there is a natural bound on the level of simulation error relative to the level of sampling error. Importance sampling simulators do not generally have such a bound. In some cases, the variance of the simulation error can (in theory)

[1]There are at least two other sets of work that address similar computationally burdensome equilibrium models. Imai et al. (2009) and Norets (2009) suggest computational techniques for Bayesian estimation of dynamic programming models. Judd and Su (2008) suggest treating computationally burdensome functions as the constraints of a numeric nonlinear programming approach. Computational burden can be decreased with this approach because the equilibrium constraints and the optimization problem are being solved simultaneously (in contrast to a "nested" fashion, which is typically computationally inefficient).

be infinite. Hence, one needs to be very careful in applying this methodology. We discuss these issues in depth through the paper, particularly in Section 5, and these issues are also illustrated in the Monte-Carlo experiments in Section 6.

2 The basic MSM estimator

Consider a parametric econometric model

$$y_i = f(x_i, \epsilon_i, \theta_0)$$

where x_i and ϵ_i are vectors of predetermined variables, observed and unobserved to the econometrician respectively. y_i is a vector of dependent variables determined within the model.[2] θ_0 is a finite-dimensional parameter vector that the econometrician is trying to estimate. Suppose that the conditional distribution of unobservables $p(\epsilon_i \mid x_i, \theta_0)$ is specified up to a subset of the parameters. $p(\epsilon_i \mid x_i, \theta_0)$ could be a fairly simple parametric distribution (e.g. normal, exponential, logistic), or a more flexible distribution within a parametric framework, e.g. a mixture of normals.

Given data $\{x_i, y_i\}_{i=1}^N$ generated at some true θ_0, a simple MSM estimator of θ_0 can be formed by examining the generic moment:

$$E\left[y_i - E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right] \quad | \quad x_i\right]$$

Since $y_i = f(x_i, \epsilon_i, \theta_0)$, this moment is identically zero at $\theta = \theta_0$. So is the expectation of any function $h(x_i)$ of the conditioning variables multiplied by the difference between y and its expectation, i.e.

$$E\left[\left(y_i - E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right]\right) \otimes h(x_i)\right] = 0 \qquad \text{at } \theta = \theta_0 \tag{1}$$

As a result, the value of θ, say $\widehat{\theta}$, that sets the sample analog of this moment

$$G_N(\theta) = \frac{1}{N} \sum_i \left[\left(y_i - E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right]\right) \otimes h(x_i)\right]$$

equal to zero or as close as possible to zero is a consistent estimator of θ_0 (we assume away possible identification problems, which this "computational" paper ignores). Appropriate regularity conditions ensure asymptotic normality of $\widehat{\theta}$ (Hansen 1982).

Simulation is often used when the function $E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right]$ is not easily computable. The straightforward way of simulating this expectation is by averaging $f(x_i, \epsilon_i, \theta)$ over a set of S random draws $(\epsilon_{i1}, \ldots\ldots, \epsilon_{iS})$ from the distribution $p(\epsilon_i \mid x_i, \theta)$, i.e.

$$\widehat{Ef_i}(\theta) = \frac{1}{S} \sum_s f(x_i, \epsilon_{is}, \theta) \tag{2}$$

[2] Note that the vector y_i can contain higher order moments of the outcome variables (e.g. y_i^2, $y_{1i} y_{2i}$, etc.).

$\widehat{Ef}(\theta)$ is trivially an unbiased simulator of the true expectation $E\big[f(x_i, \epsilon_i, \theta) \mid x_i\big]$. McFadden (1989) and Pakes and Pollard (1989) prove statistical properties of the MSM estimator that sets the simulated moment vector

$$\widehat{G_N}(\theta) = \frac{1}{N}\sum_i \big[(y_i - \widehat{Ef_i}(\theta)) \otimes h(x_i)\big]$$

as close as possible to zero. Perhaps most important of these statistical properties is the fact that these estimators are typically consistent for *finite* S (in contrast, SML estimators are typically not consistent unless $S \to \infty$ at a certain rate). The intuition behind this result is that the simulation error (i.e. the difference between the simulated expectation and the true expectation, $\widehat{Ef}_i(\theta) - E\big[f(x_i, \epsilon_i, \theta) \mid x_i\big]$) averages out over observations as $N \to \infty$.[3]

Note that this simulation procedure can be thought of as a data generating procedure. Each draw ϵ_{is} generates simulated dependent variables y_{is}. Moments of these simulated y_{is}'s are then compared to moments of the observed y_i's in the data. This illuminates how general this estimation procedure is. To estimate the model, one only needs to be able to simulate data according to the model.

3 Importance sampling and a change of variables to reduce computational burden

A significant caveat of the above simulation procedure is that $f(x_i, \epsilon_{is}, \theta)$ may be time-consuming to compute, often requiring numeric methods. The problem is that the arguments of $f(x_i, \epsilon_{is}, \theta)$ change across observations (x_i), across simulation draws (ϵ_{is}), and across different parameter vectors (θ). Hence, in an estimation procedure, $f(x_i, \epsilon_{is}, \theta)$ typically needs to be evaluated $N * S * R$ times—once for each observation (N), for each simulation draw (S), for each parameter vector ever evaluated by one's optimization routine (R denotes this total number of function evaluations needed for optimization). This is particularly problematic as the number of parameters increases since R can increase quickly in the number of parameters. This paper shows how *importance sampling* and a *change of variables* can be used to significantly reduce the number of times that $f(x_i, \epsilon_{is}, \theta)$ needs to be evaluated.

Importance sampling (Kloek and Van Dijk 1978) addresses the simulation of $E\big[f(x_i, \epsilon_i, \theta) \mid x_i\big]$. Assume $p(\epsilon_i \mid x_i, \theta)$ is a continuous distribution, and consider an arbitrary integrable p.d.f. $g(\epsilon_i \mid x_i)$ which 1) has non-zero density

[3] Another nice property of these estimators is that the extra variance imparted on the estimates due to the simulation is relatively small—asymptotically it is 1/S. This means, e.g., that if one uses just 10 simulation draws, simulation increases the asymptotic variance of the parameter estimates by just 10%. It is important to note that this property *will not* hold for the importance sampling procedure suggested here.

over the support of ϵ, and 2) does not depend on θ. Dividing and multiplying by $g(\epsilon_i \mid x_i)$ we have:

$$E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right] = \int f(x_i, \epsilon_i, \theta) p(\epsilon_i \mid x_i, \theta) d\epsilon_i$$
$$= \int f(x_i, \epsilon_i, \theta) \frac{p(\epsilon_i \mid x_i, \theta)}{g(\epsilon_i \mid x_i)} g(\epsilon_i \mid x_i) d\epsilon_i$$

Importance sampling notes that instead of drawing from $p(\epsilon_i \mid x_i, \theta)$ and forming Eq. 2, one can take random draws from $g(\epsilon_i \mid x_i)$ and form:

$$\overline{Ef}_i(\theta) = \frac{1}{S} \sum_s f(x_i, \epsilon_{is}, \theta) \frac{p(\epsilon_{is} \mid x_i, \theta)}{g(\epsilon_{is} \mid x_i)}$$

Like $\widehat{Ef}_i(\theta)$, $\overline{Ef}_i(\theta)$ is also an unbiased simulator of $E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right]$. Unfortunately, using $\overline{Ef}_i(\theta)$ in an estimation procedure does not solve the computational problem - it also requires computing f $N * S * R$ times. We need to combine this importance sampling with a *change of variables* to solve this computational issue.

Consider the following property of a parameterized econometric model:

Property (CS)—"Constant Support": The econometric model $y_i = f(x_i, \epsilon_i, \theta)$ can be expressed as $y_i = \widetilde{f}(u(x_i, \epsilon_i, \theta))$, where the "change-of-variables" vector-valued function $u(x_i, \epsilon_i, \theta)$ satisfies:

I) $\forall x_i, \theta$, the function $u(x_i, \epsilon_i, \theta)$ and the distribution $p(\epsilon_i \mid x_i, \theta)$ are such that one can analytically (or quickly) compute the change of variables density of the random vector $u_i = u(x_i, \epsilon_i, \theta)$

II) $\forall x_i$, the support of the random vector $u_i = u(x_i, \epsilon_i, \theta)$ *does not* depend on θ.

For a model to satisfy Property (CS) it needs to be able to be expressed in a form where the finite set of parameters enter the model only through functions u which have I) easily computable change-of-variable densities and II) supports that do not depend on θ. For I) to hold, the u functions will generally need to have fairly simple structure. In contrast, the $\widetilde{f}$ function will typically be complicated, since we are concerned with cases where the original function f is complicated or time consuming to evaluate. So in a sense, rewriting $y_i = f(x_i, \epsilon_i, \theta)$ as $y_i = \widetilde{f}(u(x_i, \epsilon_i, \theta))$ divides a complicated original model f into a model with two components, one simple (u) and one complicated ($\widetilde{f}$).

Note that for II) to hold, one needs a sufficient amount of heterogeneity in one's model. For example, one cannot have an element of θ enter u through an element (recall that u is a vector) that does not depend on some of the unobservables ϵ_i, e.g.. $u(x_i, \theta)$ and $u(\theta)$. The support of such a deterministic function would necessarily depend on θ, contradicting II).

Many commonly used econometric models satisfy Property (CS)—this is exhibited in examples later. We will also discuss cases where it may not be

satisfied and show how one can either 1) still benefit from computational savings using our technique, or 2) how an econometric model can be perturbed to satisfy Propert (CS).

Assuming (CS) is satisfied, let $p(u_i \mid x_i, \theta)$ denote the density of u_i obtained by the change of variables formula. We can then write

$$\begin{aligned} E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right] &= \int f(x_i, \epsilon_i, \theta) p(\epsilon_i \mid x_i, \theta) d\epsilon_i \\ &= \int \widetilde{f}(u(x_i, \epsilon_i, \theta)) p(\epsilon_i \mid x_i, \theta) d\epsilon_i \\ &= \int \widetilde{f}(u_i) p(u_i \mid x_i, \theta) du_i \end{aligned}$$

Next, combine this change of variables with an importance sampling density for u_i, $g(u_i \mid x_i)$, resulting in:

$$E\left[f(x_i, \epsilon_i, \theta) \mid x_i\right] = \int \widetilde{f}(u_i) \frac{p(u_i \mid x_i, \theta)}{g(u_i \mid x_i)} g(u_i \mid x_i) du_i$$

This integral can be simulated using:

$$\widetilde{Ef}_i(\theta) = \frac{1}{S} \sum_s \widetilde{f}(u_{is}) \frac{p(u_{is} \mid x_i, \theta)}{g(u_{is} \mid x_i)} \tag{3}$$

where the u_{is}'s are draws from g. One can then consider the importance sampling simulation estimator that sets the sample moment

$$\widetilde{G}_N(\theta) = \frac{1}{N} \sum_i \left[(y_i - \widetilde{Ef}_i(\theta)) \otimes h(x_i)\right]$$

as close as possible to zero.

The most important aspect of Eq. 3 for our purposes is that when θ changes, the u_{is}'s *can be held constant*. As a result, $\widetilde{f}$ *does not* need to be recomputed when θ changes. The only components of Eq. 3 that need to be reevaluated as θ changes are the numerators of the importance sampling weights, i.e. $p(u_{is} \mid x_i, \theta)$. Unlike recomputing the complicated f (or $\widetilde{f}$), recomputing this change of variables density is typically not computationally burdensome.[4] In summary, an estimation procedure using $\widetilde{Ef}_i(\theta)$ only needs to compute the complicated part of the model $N * S$ times, rather than $N * S * R$ times with the conventional simulator $\widehat{Ef}_i(\theta)$.

Under appropriate regularity conditions, the estimator using the simulator $\widetilde{Ef}_i(\theta)$ can be shown to be consistent and asymptotically normal using, e.g. Theorems 3.1 and 3.3 of Pakes and Pollard (1989). However, it is very

[4]For example suppose $u(x_i, \epsilon_i, \theta) = f(x_i, \theta) + \epsilon_i$ and that ϵ_i is multivariate normal. Then the distribution of u_i is also multivariate normal, and computation of p is trivial. Computation of p is also trivial for more flexible distributions, e.g. mixtures of normals.

important to note that these regularity conditions will tend to be considerably stronger than what is necessary when using the conventional simulator $\widehat{Ef}_i(\theta)$. The problem is that importance sampling can dramatically affect the precision of simulation (both positively and negatively). The variance of the simulator $\widetilde{Ef}_i(\theta)$ will depend on the variance of the importance sampling weights $\frac{p(u_{is}|x_i,\theta)}{g(u_{is}|x_i)}$. This variance depends on the behavior of p and g in the tails, and can be infinite even if $\widetilde{f}$ is bounded. So, for example, it is possible that to obtain the theoretical result, one may need to artificially bound the support of u_i (e.g. $u_i \in [-R, R]$ for some large R) in order to theoretically bound the simulation error. This is in stark contrast with the conventional simulator $\widehat{Ef}_i(\theta)$, for which the level of simulation error is naturally bounded (if f is bounded) because the importance sampling weights are implicitly always equal to 1.

These additional regularity conditions are not just a theoretical curiosity. They have important practical implications. Conventional pure frequency simulators like $\widehat{Ef}_i(\theta)$ often have the property that the simulation inflates the asymptotic variance of the parameter estimates by proportion $\frac{1}{S}$ (Pakes and Pollard 1989; McFadden 1989). Importance sampling simulators *do not* have this property—the additional variance imparted by the simulation depends on the choice of g and can be significantly greater than $\frac{1}{S}$. As a result, one needs to pay particular attention to the issue of simulation error, much more so than with conventional simulators.[5] We discuss this issue in more detail in Section 5.3, including ideas on how to choose g. For more on the general problems and issues that can arise using importance sampling simulators, see Geweke (1989).

Additional computational savings are possible if one chooses an importance sampling density $g(u_i \mid x_i)$ that is the same across observations, i.e. $g(u_i)$. In this case, the draws u_{is}'s also can be held constant across observations i. Hence, in this case h only needs to be computed S times. One caveat here is that using the same simulation draws across observations may limit the extent to which simulation error averages out across observations, and may change its asymptotic properties.

Note that $\widetilde{Ef}_i(\theta)$ is not much harder to program than $\widehat{Ef}_i(\theta)$—one only needs some additional density calculations. In addition, for a similar reason as noted by Bajari et al. (2007b), the S necessary calculations of $\widetilde{f}$ are done before the search procedure. Hence, the calculations can easily be parallelized to take advantage of modern multiprocessor computers.

Lastly note that there is some intuition behind our alternative simulator $\widetilde{Ef}_i(\theta)$. As θ changes, rather than holding each of the ϵ_{is} and their implicit

[5] Obviously, one cannot simply standard errors for the importance sampler by simply multiplying the normal GMM variance formulas (i.e. ignoring simulation error) by $1 + \frac{1}{S}$. Instead, to adjust the standard errors, one would need to formally estimate the variance in $\widetilde{Ef}_i(\theta)$. This can be done using the variation in $\widetilde{f}(u_{is}) \frac{p(u_{is}|x_i,\theta)}{g(u_{is}|x_i)}$ across simulation draws.

weights ($\frac{1}{S}$) constant, this procedure holds the u_{is} constant and varies the "weights" $\left(\frac{1}{S}\frac{p(u_{is}|x,\theta)}{g(u_{is})}\right)$ on each of the draws. Put another way, rather than changing our "simulated observations" when we change θ, we change the weight which we put on each "simulated observation". This avoids needing to recompute the complicated $\widetilde{f}$ for new simulated observations. Another way to think about this estimator is that the estimator uses the simulation draws to in a sense "span" parameter space. The simulator is approximating the objective function at different parameter values by reweighting these simulation draws in different ways.

3.1 Application to simulated maximum likelihood problems

Our importance sampling methodology can also be applied to Simulated Maximum Likelihood (SML) estimation procedures. SML estimators are not quite as general as MSM estimators. One reason is that straightforward SML is often a bit complicated with continuous outcome variables (although measurement error methods suggested by Keane and Wolpin 2000 can address this issue). However, SML is typically more efficient and often easier to use than MSM in panel settings, and thus has frequently been applied to models with discrete outcome variables, e.g. dynamic programming discrete choice models (e.g. Erdem and Keane 1996; Ackerberg 2003; Crawford and Shum 2005; Hendel and Nevo 2006; and Hartmann 2006).

In these types of models, the likelihood function for observation i typically has the form of an integral:

$$L_i = \int f(y_i \mid x_i, \epsilon_i, \theta) p(\epsilon_i \mid x_i, \theta) d\epsilon_i \tag{4}$$

where ϵ_i represents unobserved heterogeneity and $f(y_i \mid x_i, \epsilon_i, \theta)$ is the distribution of the observed outcome variables conditional on this unobserved heterogeneity. In panel situations, y_i is typically a vector of observations for i over time. Often these models include unobservables in the form of analytically integrable i.i.d. choice specific logit errors (following Rust 1987), in which case $f(y_i \mid x_i, \epsilon_i, \theta)$ is the probability (implied by the logit errors) of the observed outcomes y_i, conditional on both x_i and the unobserved heterogeneity ϵ_i. Computing $f(y_i \mid x_i, \epsilon_i, \theta)$ often requires solving a dynamic discrete choice optimization problem.

Straightforward simulation involves taking S draws from $p(\epsilon_i \mid x_i, \theta)$ and forming the simulated likelihood:

$$\widehat{L}_i = \frac{1}{S}\sum f(y_i \mid x_i, \epsilon_{is}, \theta)$$

As above, this will generally require resolving the dynamic programming problems as one searches over θ in the optimization routine.

Suppose we can find a vector-valued, change-of-variables function $u_i = u(x_i, \epsilon_i, \theta)$ such that $f(y_i \mid x_i, \epsilon_i, \theta) = \widetilde{f}(y_i \mid u(x_i, \epsilon_i, \theta))$ and where $u(x_i, \epsilon_i, \theta)$ satisfies the 2 conditions of Property (CS). Then we can write:

$$\begin{aligned} L_i &= \int \widetilde{f}(y_i \mid u(x_i, \epsilon_i, \theta)) p(\epsilon_i \mid x_i, \theta) d\epsilon_i \\ &= \int \widetilde{f}(y_i \mid u_i) p(u_i \mid x_i, \theta) du_i \\ &= \int \widetilde{f}(y_i \mid u_i) \frac{p(u_i \mid x_i, \theta)}{g(u_i \mid x_i)} g(u_i \mid x_i) du_i \end{aligned}$$

and use the importance sampling simulator:

$$\widetilde{L}_i = \frac{1}{S} \sum \widetilde{f}(y_i \mid u_{is}) \frac{p(u_{is} \mid x_i, \theta)}{g(u_{is} \mid x_i)}$$

where the u_{is}'s are S draws from the distribution g. Again, θ only enters this simulated likelihood through the density function p, so as θ changes, one does not need to recompute $\widetilde{f}$.[6] With appropriate regularity conditions, standard results such as those in Gourieroux and Monfort (1991) or Train (2003) apply, and one obtains consistency, asymptotic normality and efficiency as long as S increases at a rate faster than $\sqrt{N}$.[7,8] However, analogous to the discussion regarding MSM, these regularity conditions have important practical implications. Specifically, one needs to be much more careful about simulation error when using importance sampling simulators (as discussed further in Section 5.3). Example 1 below provides a more concrete example of applying our technique to SML estimation.

4 Examples

We next provide 3 simple examples of applications of our importance sampling simulator. This is useful in that it shows the wide range of problems to which it can be applied. They also aid in interpreting Property (CS) and help illustrate important caveats of the simulator described in the next section.

[6] As with the MSM version, if one uses the same g for all observations, one can use the same simulation draws for all observations. The only caveat is that in the SML case, f depends on y_i. As a result, one would still need to compute $S * N$ different f's. However, it is often the case that computing $f(y_i|u_{is})$ for different y_i (holding u_{is} constant) is relatively easy. In a dynamic discrete choice problem, the solution to the dynamic programming problem only depends on u_{is}, not the realization of y_i. Thus, computing $f(y_i|u_{is})$ for different y_i (holding u_{is} constant) does not require resolving the dynamic programming problem.

[7] At this "faster than $\sqrt{N}$" rate, the asymptotic variance of the estimates is the same as the asymptotic variance of estimates if one could compute the integrals analytically (i.e. without simulating). Hence, standard MLE variance formulas can be used.

[8] Gourieroux and Monfort (1991) show that if one uses the same simulation draws across observation, one needs the S to increase at a faster rate (faster than N).

4.1 Example 1: a dynamic programming problem

This example is similar to Hartmann (2006), who applies the importance sampling simulator in his empirical work. We also use this example in our Monte-Carlo experiments below.

Consider a panel of consumers across time $t = 1, ..., T$ who are choosing between $j = 1, ..., J$ discrete products (and an outside alternative ($j = 0$)) in each period. We allow for a simple form of state dependence in the utility function, which makes optimal consumer behavior the solution to a dynamic programming problem. Suppose the single period utility consumer i obtains from making choice j in period t is given by:

$$U_{ijt} = \alpha_{ij} + \beta_i I(c_{it-1} = j) - \gamma_i p_{ijt} + \nu_{ijt}$$

where p_{ijt} is the (assumed exogenous) price of product j faced by consumer i in period t, and $c_{it-1} \in \{0, ..., J\}$ is i's choice in period $t-1$. The indicator function $I(c_{it-1} = j)$ term captures state dependence—i.e., consumers recieve a positive (or negative) utility value β_i from consuming the same product in consecutive periods. α_{ij} represents consumer i's time-invariant preferences for product j, and γ_i captures consumer i's disutility from price. The ν_{ijt} are assumed to be i.i.d. logit errors and the utility from the outside alternative is normalized to zero in every period, i.e. $U_{i0t} = 0$.

If the vector of prices $p_{it} = (p_{i1t}, ..., p_{iJt})$ follows a first order markov process, the state space of this problem can be thought of as $(p_{it}, c_{it-1}, \nu_{it})$—i.e. current prices, last period's choice, and the vector of logit errors. Even though the dimension of this state space may be large, it is not prohibitively hard to numerically solve for consumer i's value function $V_i(p_{it}, c_{it-1}, \nu_{it})$ and optimal policy (choice) function $C_i(p_{it}, c_{it-1}, \nu_{it})$.[9] Note, however, that these value and policy functions are indexed by i. This is because consumers differ in their unobserved heterogeneity, i.e. the vector $(\{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i)$. Consumers with different $(\{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i)$ will have different value and policy functions. To more explicitly illustrate the dependence of the value and policy functions on the unobserved heterogeneity, we can incorporate this heterogeneity as direct arguments in the value and policy functions, i.e.

$$V_i(p_{it}, c_{it-1}, \nu_{it}) = V\big(p_{it}, c_{it-1}, \nu_{it}; \{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i\big)$$

$$C_i(p_{it}, c_{it-1}, \nu_{it}) = C\big(p_{it}, c_{it-1}, \nu_{it}; \{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i\big)$$

For estimation, one needs to specify the distribution of the unobserved heterogeneity $\big(\{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i\big)$ as a function of data and parameters. It is typical

[9] Using the "alternative specific" value function methodology of Rust (1987), this is made considerably easier by the i.i.d. logit assumption on the ν_{ijt}. It becomes even easier if one assumes consumers believe prices follow an i.i.d. process over time.

to specify the α_i's, β_i, and γ_i as functions of observed consumer characteristics x_i (e.g. income, family size) plus unobservable terms, e.g.

$$\begin{aligned}\left\{\alpha_{ij} = x_{ij}'\theta_j + \epsilon_{ij}\right\}_{j=1}^{J} \\ \beta_i = x_{iJ+1}'\theta_{J+1} + \epsilon_{iJ+1} \\ \gamma_i = x_{iJ+2}'\theta_{J+2} + \epsilon_{iJ+2}\end{aligned}$$

where ϵ_i is multivariate normal and independent of x_i and p_{it},[10] and where each of $x_{i1}, ..., x_{iJ+2}$ are $\subseteq x_i$.[11]

Given this specification, we can write the joint likelihood of consumer i 's *sequence* of choices $(c_{i1}, ..., c_{iT})$ as:

$$\begin{aligned}L_i = \int\int I\big(C\big(p_{it}, c_{it-1}, v_{it}; \left\{x_{ij}'\theta_j + \epsilon_{ij}\right\}_{j=1}^{J}, x_{iJ+1}'\theta_{J+1} + \epsilon_{iJ+1}, x_{iJ+2}'\theta_{J+2} + \epsilon_{iJ+2}\big) \\ = c_{it} \ \forall t\big) p(v_i)dv_i\, p(\epsilon_i|\theta)d\epsilon_i\end{aligned}$$

where the inner integral is over the logit errors and the outer integral is over the unobserved components of the heterogeneity, i.e. ϵ_i.[12]

Following Rust (1987), the i.i.d. logit assumption implies that the integral over v_i is analytically computable, resulting in:

$$L_i = \int f\big(c_i\big|p_i, \left\{x_{ij}'\theta_j + \epsilon_{ij}\right\}_{j=1}^{J}, x_{iJ+1}'\theta_{J+1} + \epsilon_{iJ+1}, x_{iJ+2}'\theta_{J+2} + \epsilon_{iJ+2}\big) p(\epsilon_i|\theta)$$

where f is the likelihood of individual i's observed sequence of choices c_i *conditional* on the unobserved heterogeneity $\left(\left\{\alpha_{ij}\right\}_{j=1}^{J}, \beta_i, \gamma_i\right)$ (and prices faced $p_i = (p_{i1}, ..., p_{iT})$).

Straightforward simulated maximum likelihood estimation would proceed by taking S sets of simulation draws $\left(\left\{\epsilon_{ijs}\right\}_{j=1}^{J}, \epsilon_{iJ+1s}, \epsilon_{iJ+2s}\right)$ from $p(\epsilon_i \mid \theta)$ and forming:

$$\widehat{L}_i = \frac{1}{S}\sum_s f\big(c_i\big|p_i, \left\{x_{ij}'\theta_j + \epsilon_{ijs}\right\}_{j=1}^{J}, x_{iJ+1}'\theta_{J+1} + \epsilon_{iJ+1s}, x_{iJ+2}'\theta_{J+2} + \epsilon_{iJ+2s}\big) \quad (5)$$

Clearly, as the parameters θ change, f needs to be recomputed for every consumer for every simulation draw. Recomputing f requires resolving the dynamic programming problem. In other words, to estimate this model based

[10] This is just a simple example. One can easily use non-linear index functions, or more flexible distributions, e.g. mixtures of normals.

[11] As discussed at the end of Section 5.1, our technique has the benefit that one can run many alternative specifications (more specifically, alternative specifications with different sets of x_{ij}'s) without having to newly resolve dynamic programming problems.

[12] Note that this likelihood ignores potential initial conditions problems (i.e. c_{i0} is assumed constant). Pantano (2008) suggests a clever way to model initial conditions problems while using this importance sampling simulator.

on $\widehat{L}_i$, one would need to solve for $V_i(p_{it}, v_{it}, c_{it-1})$ and $C_i(p_{it}, v_{it}, c_{it-1})$ $N * S * R$ times within the estimation procedure.

The importance sampling simulator can reduce this computational burden. Consider the change of variables function:

$$u_i = u(x_i, \epsilon_i, \theta) = \begin{pmatrix} \{\alpha_{ij}\}_{j=1}^{J} \\ \beta_i \\ \gamma_i \end{pmatrix} = \begin{pmatrix} \{x'_{ij}\theta_j + \epsilon_{ij}\}_{j=1}^{J} \\ x'_{iJ+1}\theta_{J+1} + \epsilon_{iJ+1} \\ x'_{iJ+2}\theta_{J+2} + \epsilon_{iJ+2} \end{pmatrix}$$

noting that the multivariate normal assumption on ϵ_i implies that the support of u does not depend on θ (as long as the variances of the ϵ_{ij}'s are bounded away from 0). Thus, the change of variables function u satisfies Property (CS). The likelihood can then be rewritten as:

$$\begin{aligned} L_i &= \int \widetilde{f}(c_i \mid p_i, u(x_i, \epsilon_i, \theta)) p(\epsilon_i \mid x_i, \theta) d\epsilon_i \\ &= \int \widetilde{f}(c_i \mid p_i, u_i) p(u_i \mid x_i, \theta) du_i \\ &= \int \widetilde{f}(c_i \mid p_i, u_i) \frac{p(u_i \mid x_i, \theta)}{g(u_i \mid x_i)} g(u_i \mid x_i) du_i \end{aligned}$$

where $p(u_i \mid x_i, \theta)$ is the density of u_i from the standard change-of-variables formula, and $g(u_i \mid x_i)$ is an importance sampling density with the same support as $p(u_i \mid x_i, \theta)$. This can be simulated using:

$$\widetilde{L}_i = \frac{1}{S} \sum_s \widetilde{f}(c_i \mid p_i, u_{is}) \frac{p(u_{is} \mid x_i, \theta)}{g(u_{is} \mid x_i)} \tag{6}$$

where the u_{is}'s are draws from $g(u_i \mid x_i)$.

As parameters change, the u_{is}'s need not change. As such, the conditional likelihood function $\widetilde{f}(c_i \mid p_i, u_{is})$ (and thus the dynamic programming problem) only needs to be computed $N * S$ times—once for each simulation draw for each individual. As described above, one could reduce the number of needed dynamic programming solutions to S by choosing a $g(u_{is} \mid x_i)$ that is the same across observations (i.e. does not depend on x_i) and using the same simulation draws u_{is} for each individual (see Section 6 for an example of this).

As discussed further at the end of Section 5.1, our techniques have an important side benefit in these sorts of problems. Specifically, one can run many alternative specifications (more specifically, alternative specifications with different sets of x_{ij}'s entering the $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$) *without having to resolve new dynamic programming problems*. As long as the $g(u_{is} \mid x_i)$ and the simulation draws are held constant, one can try alternative sets of x_{ij}'s without having to resolve the $\widetilde{f}$'s. The only thing that needs to change across the specifications is the $p(u_{is} \mid x_i, \theta)$'s. This can allow researchers to try alternative specifications or investigate robustness in a very computationally cheap way.

4.2 Example 2: a discrete game

Next consider a model of discrete quantity competition similar to that in Davis (2006), who examines supermarket chains' decisions of how many stores to open in a particular market. Suppose one observes a cross section of markets $i = 1, ..., N$, each with J firms. Each firm $j = 1, ..., J$ in each market chooses an integer number of retail stores $q_{ij} \in \mathbb{Z}$ to operate. The total costs of firm j in market i operating q_{ij} stores is given by

$$c(q_{ij}) = (\beta x_{ij} + \epsilon_{ij} + (\alpha x_{ij} + \eta_{ij})q_{ij})q_{ij}$$

where x_{ij} are observables and ϵ_{ij} and η_{ij} are unobservables. Note that this formulation allows there to be increasing or decreasing returns to scale in q_{ij}, allows these returns to scale to change in q_{ij}, and also allows heterogeneity across firms and markets in these effects.

Because of competition, the total revenue of firm j in market i depends not only on the number of stores j operates, but also the total number of stores operated by all competitors in the market, $Q_i = \sum_j q_{ij}$. Assume that this total revenue is given by:

$$r(q_{ij}, Q_i) = (\delta_0 - \delta_1 Q_i + \delta_2 z_i + \mu_i)\, q_{ij}$$

where z_i are observables that shift overall demand in market i and μ_i is an unobserved market demand shifter. This leads to the profit function:

$$\pi(q_{ij}, Q_i) = (\delta_0 - \delta_1 Q_i + \delta_2 z_i + \mu_i)\, q_{ij} - (\beta x_{ij} + \epsilon_{ij} + (\alpha x_{ij} + \eta_{ij})q_{ij})q_{ij}$$

Assume that the unobservables ϵ_{ij}, η_{ij}, and μ_i are joint normal and independent of the observables x_{ij} and z_i.[13]

Davis (2006) considers a simultaneous move, perfect information, Nash equilibrium of this game.[14] While there are multiple equilibrium in this game, he shows conditions under which *all* equilibrium consist of the same total number of stores Q_i. Thus, to avoid explicitly dealing with multiple equilibria in estimation, he only considers the model's (unique) prediction of Q_i, not the individual q_{ij}'s (this strategy for addressing multiple equilibrium problems is similar to that of Berry 1992). One can then think of the model as generating the outcome variable

$$Q_i = f\left(\{x_{ij}\}_{j=1}^{J_i}, \{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, z_i, \mu_i; \theta\right)$$

[13] If one wanted to ensure that costs are positive, one could use an alternative specification such as $c(q_{ij}) = (h_1(\beta x_{ij} + \epsilon_{ij}) + h_2(\alpha x_{ij} + \eta_{ij})q_{ij})q_{ij}$ where the h functions have only positive (but constant) support, e.g. exponential functions.

[14] It is a perfect information game in the sense that all firms observe the unobservables of all other firms (as well as the market specific shock).

where the function f takes the primitives of the model as inputs and computes the equilibrium total number of stores. The expected number of stores (conditional on observables) is thus:

$$EQ_i(\theta) = \int f\left(\{x_{ij}\}_{j=1}^{J_i}, \{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, z_i, \mu_i; \theta\right) p\left(\{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, \mu_i; \theta\right)$$

where $p(\{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, \mu_i; \theta)$ is the parameterized distribution of the unobservables.

Suppose one has data $(Q_i, \{x_{ij}\}_{j=1}^{J_i}, z_i)$ for a cross section of markets. Since underlying profits are not observed, a normalization of profit units is necessary. We choose the normalization $\delta_1 = 1$.[15] Assuming identification conditions hold, straightforward MSM estimation could proceed using the sample analog of a moment condition such as:

$$E\left[\left(Q_i - \widehat{E}Q_i(\theta)\right) \otimes \begin{pmatrix} \{x_{ij}\}_{j=1}^{J_i} \\ z_i \end{pmatrix}\right] = 0 \quad \text{at } \theta = \theta_0$$

where

$$\widehat{E}Q_i(\theta) = \frac{1}{S} \sum_s f\left(\{x_{ij}\}_{j=1}^{J_i}, \{\epsilon_{ijs}\}_{j=1}^{J_i}, \{\eta_{ijs}\}_{j=1}^{J_i}, z_i, \mu_{is}; \theta\right)$$

and where $\{\epsilon_{ijs}\}_{j=1}^{J_i}, \{\eta_{ijs}\}_{j=1}^{J_i}, \mu_{is}$ are draws from $p(\{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, \mu_i; \theta)$. However, this can be time-consuming, as an interative tatonnment procedure is required to solve the function f, and estimation would require computing this function $N * S * R$ times.

To apply the importance sampling simulator, consider the change of variables function:

$$u_i = u\left(\{x_{ij}\}_{j=1}^{J_i}, \{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, z_i, \mu_i; \theta\right) = \begin{pmatrix} \{\beta x_{ij} + \epsilon_i\}_{j=1}^{J_i} \\ \{\alpha x_{ij} + \eta_i\}_{j=1}^{J_i} \\ \delta_0 + \delta_2 z_i + \mu_i \end{pmatrix}$$

and note that the model can be rexpressed as

$$Q_i = \widetilde{f}\left(u\left(\{x_{ij}\}_{j=1}^{J_i}, \{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, z_i, \mu_i; \theta\right)\right)$$

This change of variables function u will satisfy Property (SC) if the unobservables have full support $(-\infty, \infty)$.

[15] This normalization is different than what might typically be used (e.g. that the variance of one of the unobservables equals one) but is an identical model given that own profits depend negatively on other firms' number of stores. Interestingly, this alternative normalization helps the model satisfy Property (CS). Bajari et al. (2009b) use a similar normalization in their application of the importance sampling simulator. This illustrates that when using the importance sampling simulator, it may be beneficial to carefully consider choice of normalization.

To apply the importance sampling technique, note that we can now write:

$$\begin{aligned} EQ_i(\theta) &= \int \widetilde{f}\left(u\left(\{x_{ij}\}_{j=1}^{J_i}, \{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, z_i, \mu_i; \theta\right)\right) p\left(\{\epsilon_{ij}\}_{j=1}^{J_i}, \{\eta_{ij}\}_{j=1}^{J_i}, \mu_i; \theta\right) \\ &= \int \widetilde{f}(u_i)\, p\left(u_i \middle| \{x_{ij}\}_{j=1}^{J_i}, z_i; \theta\right) \\ &= \int \widetilde{f}(u_i) \frac{p\left(u_i \middle| \{x_{ij}\}_{j=1}^{J_i}, z_i; \theta\right)}{g\left(u_i \middle| \{x_{ij}\}_{j=1}^{J_i}, z_i\right)} g\left(u_i \middle| \{x_{ij}\}_{j=1}^{J_i}, z_i\right) \end{aligned}$$

and consider the importance sampling simulator:

$$\widetilde{EQ}_i(\theta) = \frac{1}{S} \sum_s \widetilde{f}(u_{is}) \frac{p\left(u_{is} \middle| \{x_{ij}\}_{j=1}^{J_i}, z_i, \theta\right)}{g\left(u_{is} \middle| \{x_{ij}\}_{j=1}^{J_i}, z_i\right)}$$

where the u_{is} are draws from the importance sampling density $g(u_i \mid \{x_{ij}\}_{j=1}^{J_i}, z_i)$. As the parameters change, the u_{is} draws can be held constant – as a result the $\widetilde{f}$ functions need not be recomputed as θ changes. With this simulator, the complicated equilibrium only needs to be computed $N * S$ times instead of $N * S * R$ times. If one uses a $g\left(u_i \middle| \{x_{ij}\}_{j=1}^{J_i}, z_i\right)$ that is the same across markets (i.e. does not depend on $\{x_{ij}\}_{j=1}^{J_i}$ and z_i), $\widetilde{f}$ would need to be computed only S times. As in the prior example, one can again estimate some different specifications (i.e. models with different elements in $\{x_{ij}\}_{j=1}^{J_i}, z_i$) *without* needing to resolve the $\widetilde{f}$ functions.

4.3 Example 3: an asymmetric auction model

Consider a first-price private values auction model with asymmetric bidders, similar to that considered in Bajari (1998a). In auction i, bidder j's reservation value is given by

$$V_{ij} = X_{ij}\beta + \eta_{ij} + \lambda_{ij}$$

X_{ij} are auction-bidder specific factors that are common knowledge to all bidders and observed by the econometrician, η_{ij} is an auction-bidder specific factor that is common knowledge to all firms but unobserved by the econometrician, and λ_{ij} is bidder j's private value in auction i, observed only by bidder j. Suppose that $\eta_{ij} \sim iid\ N(0, \sigma_\eta^2)$, $\lambda_{ij} \sim iid\ N(0, \sigma_{\lambda i}^2)$, and $\sigma_{\lambda i}^2 \sim iid\ \ln N(\mu_\lambda, \sigma_\lambda^2)$. Note that the across-bidder variance of the private value component, $\sigma_{\lambda i}^2$, is allowed to vary across different auctions i. A more parsimonious specification might restrict this variance to be identical across auctions. The additional heterogeneity has been added to the model to help satisfy Property (CS) (see Section 5.1 for further discussion).

Most of the empirical auction literature prior to Bajari (1998a) assumes symmetric, i.e. ex ante identical, bidders. The reason is that when bidders are heterogeneous, the optimal bidding function

$$b_{ij} = f(X_{ij}\beta + \eta_{ij} + \lambda_{ij}, \{X_{ik}\beta + \eta_{ik}\}_{k=1}^{J}, \sigma_{\lambda i}^2)$$

is the solution to a system of non-linear differential equations that can be time consuming to solve (see Maskin and Riley 1996 and Bajari 1998b). Note that the bidding function depends the bidder's own reservation value, the bidder's expectations of his/her competitors reservation values, and $\sigma^2_{\lambda i}$. It depends on $\sigma^2_{\lambda i}$ because $\sigma^2_{\lambda i}$ affects bidder j's perceptions about other bidders' reservation values.

Straightforward MSM estimation of this model (using moments in the difference between observed bids and expected bids) requires repeatedly solving this system of differential equations at different parameter values.. To apply the importance sampling/change-of-variables technique, consider the change of variables function:[16]

$$u_i = u(\{X_{ij}, \eta_{ij}, \lambda_{ij}\}_{j=1}^J, \sigma^2_{\lambda,i}, \theta) = \begin{pmatrix} u_i^A \\ u_i^B \\ u_i^C \end{pmatrix} = \begin{pmatrix} \{X_{ij}\beta + \eta_{ij} + \lambda_{ij}\}_{j=1}^J \\ \{X_{ij}\beta + \eta_{ij}\}_{j=1}^J \\ \sigma^2_{\lambda i} \end{pmatrix}$$

and note that the optimal bidding function of bidder j in market i can be reexpressed as:

$$b_{ij} = \widetilde{f}\left(u_{ij}^A, \{u_{ij}^B\}_{j=1}^J, u_i^C\right)$$

i.e. bidder j's bid depends on his own valuation, what bidder j knows about the other bidders' valuations (and what the other bidders know about j's valuation), and $\sigma^2_{\lambda i}$

Then an importance sampling simulator of the expected bid vector in market i can be formed with

$$\widehat{EB}_i(\theta) = \frac{1}{S}\sum_s \begin{pmatrix} \widetilde{f}(u_{i1s}^A, \{u_{ijs}^B\}_{j=1}^J, u_{is}^C) \\ \widetilde{f}(u_{i2s}^A, \{u_{ijs}^B\}_{j=1}^J, u_{is}^C) \\ . \\ \widetilde{f}(u_{iJs}^A, \{u_{ijs}^B\}_{j=1}^J, u_{is}^C) \end{pmatrix} \frac{p\left(\{u_{ijs}^A\}_{j=1}^J, \{u_{ijs}^B\}_{j=1}^J, u_{is}^C \mid X_i, \theta\right)}{g\left(\{u_{ijs}^A\}_{j=1}^J, \{u_{ijs}^B\}_{j=1}^J, u_{is}^C \mid X_i\right)}$$

where the u's are draws from the importance sampling density g. As in the prior examples, as the parameters change, the u_{is}'s do not change and the optimal bid functions do not need to be recomputed as the parameters change. Note that in this example, the elements of u_i are by construction correlated—u_i^A and u_i^B are correlated through η_{ij} and the distribution of u_i^B depends on the variance term $u_i^C = \sigma^2_{\lambda i}$. However, it is easy to construct p and g (as well as take simulation draws from g) using conditional distributions, i.e.[17]

$$\begin{aligned} &p\left(\{u_{ij}^A\}_{j=1}^J, \{u_{ij}^B\}_{j=1}^J, u_i^C \mid X_i, \theta\right) \\ &\quad = p_C\left(u_i^C \mid X_i, \theta\right) p_B\left(\{u_{ij}^B\}_{j=1}^J \mid X_i, \theta\right) p_A\left(\{u_{ij}^A\}_{j=1}^J \mid \{u_{ij}^B\}_{j=1}^J, u_i^C, X_i, \theta\right) \end{aligned}$$

[16] Note that u satisfies Property (CS). The support of the first two sets of elements is the real line, the support of the last element is the positive real line. One could also easily restrict the reservation values to be positive if one was so inclined.

[17] p_C is a log normal distribution and p_A and p_B are multivariate normal distributions.

5 Discussion

5.1 Satisfying or partially satisfying property (CS)

In our three examples, we were able to find change of variables functions u that satisfied Property (CS). However, in some models this may not be the case. One common example of this is when there are parameters in one's model that 1) do not vary unobservably across the population *and* 2) do not enter into "index" functions that have at least one unobservable component that varies across the population. In Example 1, we did not consider estimation of the discount factor (it was implicitly assumed to be known). Suppose that one did want to estimate the discount factor δ, and furthermore wanted to assume that all consumers have the same discount factor $\delta_i = \delta$. Such a model would not easily satisfy Property (CS). The problem is that in this case, it will be hard to find a change of variables function u that summarizes the impact of the discount factor parameter on the model, that has a constant support, and that has a easily computable change of variables density. From a more intuitive perspective, the problem here is that since there is no heterogeneity in δ across the population, one cannot easily "span" δ space using the simulation draws (so we cannot learn anything about the likelihood function when, e.g., $\delta = 0.8$ from solutions to the model when, e.g., $\delta = 0.9$).[18]

A first approach is to apply the importance sampling approach to only a subset of parameters. Suppose that the parameter vector can be divided into components θ_1 and θ_2. Suppose that the model can be expressed in a form where the θ_2 parameters enter through change of variables functions u that have constant support, but that this cannot be done with the θ_1 parameters. Then $\widetilde{f}$ will need to be recomputed as θ_1 changes (though not as θ_2 changes). In Example 1 with a homogeneous discount factor parameter that needs to be estimated, the discount factor would be in θ_1, the rest of the parameters in θ_2.[19]

In these situations where Property (CS) is partially satisfied, a first option is to use derivative based optimization methods. In computing first derivatives, $\widetilde{f}$ needs to be recomputed only when elements of θ_1 are perturbed. This will reduce the computational time of computing these derivatives by approximately $\frac{\dim(\theta_1)}{\dim(\theta)}$ relative to straightforward simulation using numeric derivatives. A second alternative is to use a nested search algorithm. On the outside, one searches over θ_1; on the inside, over θ_2. During the inside search algorithm,

[18] A similar situation would arise in Example 2 if, e.g., $\sigma_\eta^2 = 0$, or in Example 3 if the variance of the private values were the same across auctions (i.e. $\sigma_\lambda^2 = 0$).

[19] The simulator in this case would be $\widetilde{L}_i = \frac{1}{S}\sum_s \widetilde{f}(c_i|p_i, u_{is}, \theta_1)\frac{p(u_{is}|x_i,\theta_2)}{g(u_{is}|x_i)}$, so changes in θ_2 are adjusted for with importance sampling weights, changes in θ_1 adjusted for with changes in $\widetilde{f}$.

one needs not recompute f's. As these nested search algorithms are generally inefficient, this approach may only be reasonable if the dimension of θ_1 is small.

A second general approach to satisfying Property (CS) is to note that if a coefficient is heterogeneous across the population (and has a constant support, e.g. normals, log-normals, or functions of such variables), it will automatically satisfy Property (CS). Therefore, if one allows heterogeneity in the discount factor across the population, e.g. $\delta_i = \frac{\exp(\delta+\sigma_\delta \mu_i)}{1+\exp(\delta+\sigma_\delta \mu_i)}$ where μ_i has support $(-\infty, \infty)$ (as in an early version of Hartmann 2006),[20] Property (CS) can be satisfied by including $u_i = \frac{\exp(\delta+\sigma_\delta \mu_i)}{1+\exp(\delta+\sigma_\delta \mu_i)}$ as an element of the change of variables function. Strictly speaking, this is not a generalization of the model with $\delta_i = \delta$, since the variance of μ_i needs to bounded away from 0 to apply the importance sampling simulator.[21] It is interesting, however, that the importance sampling method in a sense works better with *more* unobserved heterogeneity in one's model.[22] This contrasts with methods for estimating dynamic programming problems related to HM, which tend to be harder to apply when there is unobserved heterogeneity (though these methods do have other advantages, e.g. being able to estimate structural parameters without even solving a single dynamic programming problem).

Importantly, note that while having *all* coefficients in one's model be heterogeneous across the population (as in the dynamic models considered by Bajari et al. 2007b) is often is a *sufficient* condition for Property (CS) to hold, it is *not a necessary condition*. In our method, *coefficients need not be heterogeneous* across the population as long as they enter the model through change-of-variables functions that include sufficient heterogeneity. This turns out to be very useful for introducing new individual level covariates into a model in a parsimonious way. Moreover, it also allows a researcher to estimate many alternative specifications (e.g. models with different sets of individual level covariates) *without needing to compute* new $\widetilde{f}$*'s.*

Example 1 is again illustrative of this. Note that the consumer characteristics x_i enter the model through index functions, $x_{ij}'\theta_j + \epsilon_{ij}$, where the parameter vectors θ_j are *homogeneous* across observations i (i.e. the effect of a change in x_{ij} on the mean of the taste distribution is the same across i). What is crucial is that these fixed parameters enter the model through index functions that contain at least some unobserved heterogeneity (and that admit a simple change of variables density).

[20] Obviously, the functional form is chosen to restrict the discount factor between 0 and 1.

[21] And as noted previously, there may be large amounts of simulation error if the variance approaches 0. In practice, one should be careful to watch for these variances (e.g. σ_δ) approaching zero during estimation. If they do, it may be best to switch to the alternative approach suggested next, i.e. applying the importance sampling approach to only a subset of the parameters.

[22] This statement ignores two important caveats. First, additional unobserved heterogeneity might create identification problems (we ignore these in this paper by simply assuming identification). Second, increased dimensionality of the unobserved heterogeneity may generate higher levels of simulation error.

This property of our estimator makes it very convenient to add new x_{ij} 's to the model, e.g. when an empirical researcher is trying alternative specifications. First, one can add a new x_{ij} to the model by just introducing a single new parameter, not an entire new distribution. Second, one can add (or subtract) x_{ij}'s to the model and re-estimate the new model *without having to recompute the* $\widetilde{f}$ *'s* (as long as one continues to use the same importance sampling density and same simulation draws). We feel that this is a very important benefit, as it may allow researchers to experiment with more alternative specifications than they otherwise would.

5.2 Smoothness and analytic derivatives

As noted in the introduction, there are additional benefits of the importance sampling estimator. As McFadden (1989) originally noted in the case of a multinomial probit model, importance sampling can be used to smooth simulated objective functions. In many cases, $f(x_i, \epsilon_{is}, \theta)$ is discontinuous in θ due to discreteness in one's model. As a result, standard simulated objective functions have flats (areas with zero derivatives w.r.t. θ) and discontinuous jumps. In contrast, for distributions that are commonly used, $p(u_{is} \mid x_i, \theta)$ is typically *smooth* in θ. The change of variables and importance sampling technique essentially moves θ from inside f to inside p. Thus, it can convert an objective function that is discontinuous in θ to one that is continuous in θ.[23] The discrete quantity game discussed above is an example one of these cases—note that $\widehat{EQ}_i(\theta)$ is discontinuous in θ, while $\widetilde{EQ}_i(\theta)$ is continuous in θ.[24] Smoothness can be a big advantage in estimation. First, it allows one to use derivative based search algorithms, which are often faster than non-derivative based routines. Second, even non-derivative based routines can have significant problems trying to optimize discontinuous functions with flats and jumps.

A related advantage of importance sampling objective functions concerns derivatives with respect to θ. When f is complicated, the objective functions of most standard simulation estimators often do not have analytic derivatives. If one is using a derivative based optimization routine, this lack of analytic derivatives necessitates use of numerical methods to obtain derivative information. This can be time-consuming and is potentially imprecise. In contrast, our importance sampling objective functions often *do* have analytic derivatives.

To see this point, compare the straightforward simulator of $Ef_i(\theta)$,

$$\widehat{Ef_i}(\theta) = \frac{1}{S} \sum_s f(x_i, \epsilon_{is}, \theta) \tag{7}$$

[23]The 1999 working paper version of this work contained a number of more elaborate examples of how importance sampling can be used to smooth even very complicated economic models. For a copy please consult the author.

[24]Note that Example 1 is not a good example of this smoothing property because the likelihood function there is already smooth due to the analytically integrated logit errors.

to the importance sampling simulator:

$$\widetilde{Ef}_i(\theta) = \frac{1}{S}\sum_s \widetilde{f}(u_{is})\frac{p(u_{is} \mid x_i, \theta)}{g(u_{is} \mid x_i)}$$

Clearly, $\frac{\partial \widehat{Ef}_i(\theta)}{\partial \theta}$ depends on $\frac{\partial f(x_i, \epsilon_{is}, \theta)}{\partial \theta}$. As $f(x_i, \epsilon_{is}, \theta)$ typically cannot be computed analytically, $\frac{\partial f(x_i, \epsilon_{is}, \theta)}{\partial \theta}$ will typically also not be analytically computable. In contrast,

$$\frac{\partial \widetilde{Ef}_i(\theta)}{\partial \theta} = \frac{1}{S}\sum_s \frac{\widetilde{f}(u_{is})}{g(u_{is} \mid x_i)}\frac{\partial p(u_{is} \mid x_i, \theta)}{\partial \theta}$$

depends on $\frac{\partial p(u_{is}|x_i, \theta)}{\partial \theta}$, which is analytically computable in many cases, e.g. when p is multivariate normal and linear in x_i. These analytic derivatives can generate increased time-savings and precision in estimation.

Note that these properties suggest that importance sampling can also be helpful in computing the derivatives necessary to estimate standard errors. With discontinuous objective functions (especially those with flats), calculating these derivatives can be unreliable. This is not a problem using an importance sampled simulator

5.3 Choice of g and simulation error

As mentioned earlier, one traditional use of importance sampling is to reduce the variance of simulation estimators. An appropriate choice of g can accomplish this goal. Unfortunately, if one is not careful, importance sampling can also dramatically increase the variance of simulation estimators. When performing the above change of variables and importance sampling procedure, one needs to be very aware of this issue. Unlike standard pure frequency simulators, where the simulation error is naturally bounded, this is not necessarily the case with importance sampling simulators. This can result in large amounts of simulation error if one is not careful.

Perhaps the most obvious choice for g is p itself at some arbitrary initial parameter vector θ^{init}, i.e. $g(u_i \mid x_i) = p(u_i \mid x_i, \theta^{init})$. This importance sampling simulator is then identical to the pure frequency simulator when evaluated at $\theta = \theta^{init}$. Of course, θ^{init} is generally not going to equal the true parameter vector θ_0. And with g based on θ^{init}, the effect of simulation error on estimates can be quite large if θ_0 is far away from θ^{init}. This can be a significant issue in practice, both for efficiency, and for the reliability of the non-linear search over θ. We have a few informal suggestions for minimizing these problems, but a general point to remember is that one needs to be much more careful with importance sampling simulators than with standard simulators because of these issues.

A first suggestion is to iterate the entire importance sampling estimation procedure multiple times. In other words, set $g_1(u_i \mid x_i) = p(u_i \mid x_i, \theta^{init})$ at some exogenously chosen θ^{init}, use the resulting simulator to form a simulated

objective function, and optimize to obtain an estimate $\widehat{\theta}^1$. Then iterate the entire estimation procedure by creating a new importance sampling distribution, $g_2(u_i \mid x_i) = p(u_i \mid x_i, \widehat{\theta}^1)$, taking new simulation draws, and re-estimating to obtain a new estimate $\widehat{\theta}^2$.[25] This iterating can obviously be continued. Each of $\widehat{\theta}^1$, $\widehat{\theta}^2$, $\widehat{\theta}^3$,... is a consistent estimator of θ_0. But the hope is that after one (or multiple) iterations, the g distribution will be closer to $p(u_i \mid x_i, \theta_0)$, and the simulation error in the estimates will be smaller. Of course, there is a computational cost to iterating, as the "complicated" functions $\widetilde{f}$ need to be recomputed each time g is "reinitialized" and new draws are taken. But as we show in our Monte-Carlo experiments, computational burden is still far less than that with a straightforward simulator.

There are some unanswered questions regarding such a iteration process that are beyond the scope of this paper, e.g. Is this iteration process guaranteed to converge starting at any θ^{init}?; Does it converge to the same $\widehat{\theta}^{\infty}$ regardless of θ^{init}?;[26] What are the asymptotic properties of $\widehat{\theta}^{\infty}$? Another question is how to appropriately compute standard errors of such an estimator. Note that an estimated variance of $\widehat{\theta}^n$ based on standard SML or MSM formulas is not exactly right, since it ignores the variation in the importance sampling density g_n due to the estimation of $\widehat{\theta}^{n-1}$ from the prior iteration.

One may want to update the g density quicker than the above. In other words, if an optimization procedure moves θ far from θ^{init}, one may want to update g even if the procedure has not converged (due to concern about increased simulation error). One normalized statistic that might be useful for this is

$$IS_{stat} = \frac{\frac{1}{S}\left(\frac{1}{S}\sum_s \left(\widetilde{f}(u_s)\frac{p(u_s|x_i;\theta)}{g(u_s|x_i)} - \frac{1}{S}\sum_s \widetilde{f}(u_s)\frac{p(u_s|x_i,\theta)}{g(u_s|x_i)}\right)^2\right)}{\frac{1}{S}\left(\frac{1}{S}\sum_s \left(\widetilde{f}(u_s) - \frac{1}{S}\sum_s \widetilde{f}(u_s)\right)^2\right)}$$

This statistic is related to Geweke's (1989) "Relative Numeric Efficiency" (RNE) statistic. The numerator is an estimate, for observation i, of the simulation variance in the importance sampled $\widetilde{Ef}_i(\theta)$ at θ. The denominator is an estimate of the simulation variance in the standard simulator $\widehat{Ef}_i(\theta)$ at θ^{init}. This statistic is simple to compute (it does not require resolving $\widetilde{f}$), and gives a somewhat standardized measure of the amount of simulation error in $\widetilde{Ef}_i(\theta)$ relative to straightforward simulators.[27] If the average value of this statistic

[25] To make full use of past solutions of $\widetilde{f}$, one could actually use *both* the old draws (from $g_1(u_i \mid x_i)$) *and* the new draws (from $g_2(u_i \mid x_i)$) when the estimation procedure is iterated. In this case, the g for the full set of draws would be a mixture of g_1 and g_2. More generally, at the tth, iteration, one could use draws (and $\widetilde{f}$ solutions) from all past iterations.

[26] In our Monte-Carlo experiments, we (very) casually investigated this and did find that the iterations always converged to the same parameter vector for a wide range of θ^{init}. But this is obviously far from a proof, this is only one example, and Monte-Carlo generated data may be better behaved than actual data.

[27] Of course, one would prefer to evaluate the denominator at θ rather than θ^{init}. But doing this would require resolving $\widetilde{f}$.

(across observations) gets high, it suggests high levels of simulation error. So if an optimization procedure moves θ into an area where this is high, one may want to change g (probably to p at the current θ) and take new simulation draws. In any case, this is easy to compute and probably a useful statistic to monitor if one is using the importance sampling approach.

A second approach is to use derivative based search procedures, and only use the importance sampling method for derivative calculations. Consider the objective function based on the pure frequency simulator, i.e.

$$\widehat{Ef_i}(\theta) = \frac{1}{S}\sum_s f(x_i, \epsilon_{i,s}, \theta) \tag{8}$$

Derivatives of this objective function generally require using numeric differentiation, and repeatedly recomputing the complicated f function as one perturbs θ.

Alternatively, one could use the importance sampling objective function (with $g(u_i \mid x_i) = p(u_i \mid x_i, \theta)$) to compute these derivatives. As noted in the prior section, we have

$$\frac{\partial \widetilde{Ef}_i(\theta)}{\partial \theta} = \frac{1}{S}\sum_s \frac{\widetilde{f}(u_{is})}{g(u_{is} \mid x_i)} \frac{\partial p(u_{is} \mid x_i, \theta)}{\partial \theta}$$

which can always be done without recomputing $\widetilde{f}$, and can usually be done analytically.

Suppose one starts their non-linear search at θ^{init}. One could use importance sampling (with importance sampling density $g(u_i \mid x_i) = p(u_i \mid x_i, \theta^{init})$) to compute the derivatives of the objective function without resolving $\widetilde{f}$. Then for the "step" of the derivative based procedure, one could use standard simulators (this would require resolving $\widetilde{f}$). After the step, at the new θ', importance sampling could again be used to compute derivatives (using $g(u_i \mid x_i) = p(u_i \mid x_i, \theta')$). This could then be repeated. This is probably the most conservative way to use the importance sampling idea, since at every θ, one is using an importance sampling density g that is identical to p at that θ. Of course, there is also higher computation burden, as the time savings only applies to the derivative calculations.[28] There is also an important caveat that such a search is not guaranteed to converge. The reason is that the derivative information is not exactly right. More precisely, in this procedure, one is optimizing the objective function based on $\widehat{Ef_i}(\theta)$, but derivative information is coming from the objective function based on $\widetilde{Ef}_i(\theta)$. These derivatives will be similar (they both converge to the true derivative of $Ef_i(\theta)$ as $S \to \infty$), but are not numerically equivalent.

A third set of possibilities comes from the importance sampling literature. As noted by Geweke (1989), among others, it can help for an importance

[28]This method is also convenient when a subset of the parameters do not satisfy Property (CS). One simply needs to recompute the f functions for numeric perturbations of the parameters in the subset.

sampling density to have thick tails. This can prevent high levels of simulation error. Intuitively, a wide g means that the initial set of u^s points are spread out - thus they can be weighted to approximate behavior at a wider range of θ. One way to pick a wide g is to base g on a θ^{init} where the variance parameters are set relatively large. In the iterative procedures above, one might want to artificially inflate the variance related parameters when choosing g's

Another possibility suggested by the importance sampling literature is to normalize the importance sampling weights to sum to one. Formally, this involves the alternative importance sampling simulator

$$\widetilde{\widetilde{Ef}}_i(\theta) = \frac{1}{S}\sum_s \widetilde{f}(u_{is})\frac{p(u_{is} \mid x_i, \theta)}{g(u_{is} \mid x_i)}\left(\sum_s \frac{p(u_{is} \mid x_i, \theta)}{g(u_{is} \mid x_i)}\right)^{-1}$$

This restricts the importance sampling weights to be between 0 and 1, and thus could be more numerically stable in practice. One caveat is that $\widetilde{\widetilde{Ef}}_i(\theta)$ is no longer an unbiased simulator. Hence, the MSM result of consistency for finite S will not hold.

Lastly, note that if one is particularly concerned with these issues, one can alternatively use "importance sampled objective functions" simply as a numeric tool to get "close" to the parameter estimates. The idea is to start by optimizing an objective function based on $\widetilde{Ef}_i(\theta)$ that is easy to compute (perhaps updating g occasionally as suggested above), but eventually switch to an objective function based on a more standard simulator, e.g. $\widehat{Ef}_i(\theta)$. Note that there are no implications of doing this on estimated standard errors, since the results using $\widetilde{Ef}_i(\theta)$ are only used as starting values for optimizing based on $\widehat{Ef}_i(\theta)$.

5.4 Comparison to discretation/randomization approaches

An alternative strategy for estimating the dynamic programming problem of Example 1 would be to explicitly solve for the value and policy functions as depending on the individual specific heterogeneity, i.e. consider

$$V(p_{it}, v_{it}, c_{it-1}, \{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i) \text{ and } C(p_{it}, v_{it}, c_{it-1}, \{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i)$$

If one could solve for these functions at all possible values of their arguments, one would only need to solve them *once*. Then, when simulating a particular individual at a particular parameter vector, one could just plug the resulting $(\{\alpha_{ijs}\}_{j=1}^J, \beta_{is}, \gamma_{is})$ into V or C to compute the simulated likelihood function. However, the time required to do this to a given degree of accuracy will generally increase exponentially in the dimension of the unobserved heterogeneity, i.e. there is a "curse of dimensionality". Moreover, since $(\{\alpha_{ij}\}_{j=1}^J, \beta_i, \gamma_i)$ are often continuous variables, such a procedure would also require some discretization and approximation, as V can only be numerically solved at a finite number of points and because the simulation draws $(\{\alpha_{ijs}\}_{j=1}^J, \beta_{is}, \gamma_{is})$

encountered in simulating the likelihood will generally not equal the finite set of points at which V has been computed.

Keane and Wolpin (1994) and Rust (1997) (KW/R) suggest using randomization techniques to approximate $V(p_{it}, v_{it}, c_{it-1}, \{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$. Instead of discretizing the arguments of V in a deterministic way, one *randomly* chooses K points at which to approximate the value function (or alternative-specific value functions). After using such an approach to approximate V, simulation estimation can proceed by taking simulation draws $(\{\alpha_{ijs}\}_{j=1}^{J}, \beta_{is}, \gamma_{is})$ (conditional on θ) to simulate the likelihood. Again, since these simulation draws will generally not equal the random points at which the value function has been approximated, one needs to use additional approximation (e.g. interpolation, polynomial approximation) in $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$ space to calculate the simulated likelihood.

Note that there are two sources of simulation error in the KW/R approach. One source comes from the random draws of the K points at which to solve V, and one source comes from the random draws of $(\{\alpha_{ijs}\}_{j=1}^{J}, \beta_{is}, \gamma_{is})$ in computing the simulated likelihood function. One can think about the importance sampling approach as a modification of the KW/R approach that 1) only has one source of simulation error, and 2) doesn't require additional approximation in $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$ space. The importance sampling approach does this because it allows one to hold the draws $(\{\alpha_{ijs}\}_{j=1}^{J}, \beta_{is}, \gamma_{is})$ *constant* regardless of the value of θ. Hence, the importance sampling approach can use the *same* set of simulation draws $\left\{(\{\alpha_{ijs}\}_{j=1}^{J}, \beta_{is}, \gamma_{is})\right\}_{s=1}^{S}$ both to solve for V and to simulate the likelihood. Thus, there is only one source of simulation error. This also implies there is no additional approximation necessary in $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$ space, since the points used in simulating the likelihood are exactly the points where V has been solved.[29] Lastly, note that with the importance sampling approach, standard results on Monte-Carlo integration imply that the importance sampling approach breaks the "curse of dimensionality" in the dimension of $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$.[30]

[29] If the other state variables (e.g. p_{it}, v_{it}, and c_{it-1}) are continuous, either approach would generally need approximations in these dimensions. Of course, if the v_{it}'s (or p_{it}) are i.i.d., they can be removed from the "effective" state space by working with alternative specific value functions (Rust 1987).

[30] In contrast, it is not clear that the standard KW/R approach breaks the curse of dimensionality in $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$ space. Because $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$ are constant over time, their transition density does not satisfy Rust's (1997) Assumption (A4). Of course, as proven by Rust, the KW/R approach can break the curse of dimensionality in state variables that evolve stochastically (and smoothly) over time. So, for example, if the KW/R approach is used for the state variable p_{it}, and if the importance sampling simulator is used for $(\{\alpha_{ij}\}_{j=1}^{J}, \beta_i, \gamma_i)$; then Example 1 has no curse of dimensionality as the number of products J increases (since c_{it-1} takes a finite set of values and v_{ijt}'s are i.i.d. logit errors).

5.5 Relation to Keane and Wolpin (2000, 2001)

Independently, in two empirical papers, Keane and Wolpin use an importance sampling procedure that is related to ours in order to solve problems of unobserved state variables. These papers analyze dynamic programming problems of educational choice (Keane and Wolpin 2001) and fertility/marriage choice (Keane and Wolpin 2000). In the first paper, where individuals schooling, work, and savings decisions are analyzed over a lifetime, a significant problem is that assets (a state variable) are not observed in some years of the data (there are other state variables, choice variables, and initial conditions, e.g. schooling and hours worked, that are also occasionally unobserved). To estimate this using standard methods would be exceedingly complex, as one would need to integrate out over very complicated conditional distributions of the missing data.

Their approach starts by simulating S unconditional (i.e. there are no predetermined variables) outcome paths—these are what they call their "simulated paths". To create each of these paths, one needs to solve the simulated agent's dynamic programming problem. If all outcome variables were discrete, one could in theory compute the likelihood for observation i by the proportion of "simulated paths" that match observation i's path. Practically, since there are so many possible paths (and since some of the outcome variables are continuous), this would result in likelihood zero events. To mitigate this problem, Keane and Wolpin add measurement error to all outcome variables. This gives any observed path a positive likelihood and allows for estimation using SML.

What is similar to our paper is the fact that Keane and Wolpin use importance sampling while searching over θ. This means that as they change θ, there is no need to draw new simulated paths. Instead, one only needs to compute the likelihood of the original simulated paths at the new θ. This likelihood is much simpler that the original problem since the simulated paths have no missing data. The importance sampling also smooths the likelihood function in θ. However, unlike our procedure, it generally *does* require re-solving S dynamic programming problems when θ changes.

Formally, and in our notation, Keane and Wolpin are computing $L(f(\epsilon_i, \theta) + \eta_i = y_i)$, the likelihood of the observed data y_i , where η_i is measurement error and $f(\epsilon_i, \theta)$ are outcomes of the dynamic programming problem. Integrating out over the density of $f(\epsilon_i, \theta)$ gives:

$$L(f(\epsilon_i, \theta) + \eta_i = y_i) = \int L(f_i + \eta_i = y_i \mid f_i) p(f_i \mid \theta)$$

The inner likelihood is over the measurement error process conditional on the dynamic programming outcomes, while $p(f_i \mid \theta)$ is the distribution of dynamic

programming outcomes (without measurement error). Importance sampling these dynamic programming outcomes with distribution g gives:

$$L(f(\epsilon_i, \theta) + \eta_i = y_i) = \int L(f_i + \eta_i = y_i \mid f_i) \frac{p(f_i \mid \theta)}{g(f_i)} g(f_i)$$

Keane and Wolpin use $g = p(f_i \mid \theta')$ at some initial θ' and form the importance sampling simulator:

$$\frac{1}{S} \sum_s L(f_s + \eta_i = y_i \mid f_s) \frac{p(f_s \mid \theta)}{g(f_s)}$$

where the f_s's are simulated paths generated at θ'. As θ changes, only $p(f_s \mid \theta)$ needs to be recomputed. This is analogous to the likelihood of a standard dynamic programming problem where there is no missing state variable data. However, unlike our procedure, it does generally require resolving the dynamic programming problems of the simulated agents (there are some parameters of the Keane and Wolpin model, e.g. those determining the proportion of each simulated "type" in the population, where the DP problem does not need resolving as these parameters change.)

6 Monte-Carlo results

To informally investigate how this importance sampling procedure might work in practice, we ran some monte-carlo experiments. These are inspired by the model in of Hartmann (2006), as detailed in Example 1 of the current paper. Given concerns about how well these methods might work with more than a few dimensions of unobserved heterogeneity, we chose the number of products (J) to equal 8. This means that time-invariant heterogeneity across consumers is 10 dimensional—i.e. consumers are characterized by mean preferences for the 8 products ($\alpha_{i1}, ..., \alpha_{iJ}$), a state dependence parameter (β_i), and marginal utility of price (γ_i). As detailed in Example 1, these consumer specific tastes are modelled as:

$$\begin{gathered} \left\{\alpha_{ij} = x_{ij}'\theta_j + \sigma_j \epsilon_{ij}\right\}_{j=1}^{J} \\ \beta_i = x_{iJ+1}'\theta_{J+1} + \sigma_{J+1}\epsilon_{iJ+1} \\ \gamma_i = x_{iJ+2}'\theta_{J+2} + \sigma_{J+2}\epsilon_{iJ+2} \end{gathered}$$

where the x_{ij}'s each contain a constant term and one observed exogenous variable (distributed *iid* $N(0, 1)$). The $(\epsilon_{i1}, ..., \epsilon_{i10})$ are joint normal, independent of x's and p's, and have an identity covariance matrix. The σ_j parameters are all set to 1. As for the θ parameters, the constant terms (i.e. $\theta_1^0, ..., \theta_{J+2}^0$) are all set to 0, and the slope terms (i.e. $\theta_1^1, ..., \theta_{J+2}^1$) are all set to 1. The prices p_{ijt} are distributed *iid* $N(0, 1)$.

We chose $N = 500$ and $T = 20$. We consider the discount factor to be known. In fact, for computational reasons, we set the discount factor equal to 0, i.e. we assume consumers are myopic. Because this implies that there is no dynamic programming problem, it allows us to do more Monte-Carlo repetitions than would otherwise be possible. Of course, this implies that actual measures of computational time are not particularly relevant. So the way we compare computational time across different procedures is by a simple count of the number of times the dynamic programming problem *would need to have been solved* within an estimation procedure (if in fact there was one to be solved). Given we are using this metric, we see no obvious reason why the relative performance of the various estimation algorithms would differ if the discount factor was non-zero, but it is certainly possible.

For our importance sampling densities g, we use p's evaluated at some initial $\theta^{init} = \left(\left\{\theta_j^0\right\}_{j=1}^{J+2}, \left\{\theta_j^1\right\}_{j=1}^{J+2}, \left\{\sigma_j\right\}_{j=1}^{J+2}\right)$. As discussed above, choice of this θ^{init} may be quite important. We do two separate experiments. In the first experiment, we use "good" starting values. Across monte-carlo replications, starting values are randomly drawn from $U(0, 1)$ distributions centered at the true parameters. In this experiment, we only run the importance sampling estimation routine once for each replication.

In the second experiment, we use "bad" starting values. The θ^0 and θ^1 parameters are drawn from $U(-5, 5)$, and the σ parameters are drawn from $U(0.5, 10.5)$[31] In this experiment, we iterate the importance sampling estimation routine as described in Section 5.3. At the end of each estimation routine iteration, we construct a new importance sampling density g, based on p at the current estimations, for use in the next estimation routine. Thus at each restart, we need to resolve the "dynamic programming problems". We iterated the estimation routine in this way until it "converged" up to a numeric tolerance. Interestingly, it always converged, and the average number of estimation iterations was 17.24.

We compare our importance sampling routine to SML simulation based on a standard simulated likelihood Eq. 5. In the standard SML routine, we set $S = 30$, i.e. 30 simulation draws per observation i. This means that there are a total of $N * S = 15000$ simulation draws and that the "dynamic programming problem" needs to be solved 15000 times for each likelihood function evaluation. In the importance sampling routine, we also use a total of 15000 draws. However, recall that with the importance sampling routine, if one chooses the same importance sampling density g across observations, the *same* simulation draws can be used for all observations. Since this seems to be the most efficient use of the draws, we do this—i.e. we use the same $g(u_i)$ and

[31]The choice for the σ parameters was governed by 1) the fact that σ must be ≥ 0, and 2) the discussion above suggesting that in practice, it probably better to choose larger values for the importance sampling densities of variance related parameters.

the same 15000 draws for all observations.[32] This also requires 15000 "dynamic programming problem" solutions, but unlike the standard SML routine, these solutions do not have to be recalculated as θ changes.

Table 1 contains the results. 100 monte-carlo replications were run for each specification. In the first column are results from the standard SML procedure. On average, estimation here required 45.6 million "dynamic programming solutions". Each function evaluation required 15000 evaluations, and on average 3040 function evaluations were necessary to find the maximum in 30-dimensional space.[33] One can see that there are some very clear biases in the estimates, presumably due to the low number of simulation draws. Particular strong biases are evident in $\sigma_1, ..., \sigma_8$, which are estimated to be about half of the true parameters.

In the second column are results from the importance sampling routine using the "good" starting values. In this case we do not iterate the estimation routine, so each estimation only requires 15000 "dynamic programming solutions" (which compared to 45.6 million is a speed improvement of over 3000 times). The parameter results also illustrate some small sample biases, though in most cases not nearly as big as in Column 1. Interestingly, the comparison of standard deviations of the estimated parameters across monte-carlo repetitions is ambiguous between Columns 1 and 2. There is no clear winner—in some cases the Column 1 estimates are less variable, while in other cases the Column 2 estimates are.

The last row of the table computes the average (across monte-carlo repetitions) "true" likelihood, evaluated at the point estimates from the two procedures. To compute the "true" likelihood, we simply used a standard simulator with far more simulation draws, i.e. 10000 per observation. Less negative values of this imply that the estimation procedure has found a "better" parameter vector according to the "true" likelihood. On average, the standard SML procedure generates a point estimate with a "true" likelihood of −15303.5, while the importance sampling procedure on average generates a point estimate with a "true" likelihood of −15120.62. So the importance

[32]The way we construct a reasonable g that is the *same across observations* is to define $g(u_i)$ to be a mixture (with equal probabilities) of the 500 $p(u_i|x_i, \theta^{init})$ distributions (i.e. each of these 500 distributions depends on one of the x_i's in the sample). More precisely we use the mixture distribution:

$$g(u_i) = \frac{1}{500} \sum_i p(u_i|x_i, \theta^{init})$$

A simple way to take 15000 draws from this mixture distribution is to simply take 30 draws from $p(u_i|x_i, \theta^{init})$ for each of the 500 x_i's observed in the dataset.

[33]The 3040 includes those function evaluations necessary for derivative calculations. We used derivative based methods for all the optimization. Starting values for optimization with the standard SML routine were the "good" starting values—this number of function evaluations necessary would be slightly higher than 3040 using the "bad" starting values.

Table 1 Monte-Carlo results

		1		2		3		4		5	
		Standard SML		Importance Sampling "Good" Start Values		Importance Sampling "Bad" Start Values					
						1st Iteration		3rd Iteration		Convergence	
Parameters	Truth	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Intercepts											
θ_1^0	0	0.2366	0.0889	0.0014	0.1284	−0.4293	1.1052	−0.0685	0.3438	0.0930	0.1252
θ_2^0	0	0.2492	0.0847	0.0028	0.1270	−0.2865	1.2800	−0.0593	0.3745	0.0909	0.1308
θ_3^0	0	0.2452	0.0820	−0.0001	0.1245	−0.2833	1.3319	−0.0560	0.4202	0.0769	0.1355
θ_4^0	0	0.2219	0.0767	−0.0194	0.1204	−0.3684	1.3072	−0.0675	0.3535	0.0801	0.1304
θ_5^0	0	0.2520	0.0895	0.0106	0.1243	−0.5726	1.1853	−0.0825	0.3836	0.0927	0.1322
θ_6^0	0	0.2313	0.0833	−0.0233	0.1245	−0.5034	1.2058	−0.0695	0.3405	0.0974	0.1218
θ_7^0	0	0.2470	0.0780	0.0134	0.1332	−0.4626	1.0521	−0.0738	0.3234	0.0980	0.1295
θ_8^0	0	0.2308	0.0953	−0.0148	0.1120	−0.3787	1.1953	−0.0743	0.3517	0.0924	0.1235
θ_9^0	0	−0.0133	0.0628	−0.0050	0.0516	−0.0043	0.2904	−0.0056	0.0492	−0.0032	0.0474
θ_{10}^0	0	0.3942	0.0672	0.0743	0.0732	−0.4345	0.4803	−0.0169	0.0793	0.1068	0.0645
Slopes											
θ_1^1	1	0.8559	0.0662	0.8391	0.1066	1.4697	0.5612	0.9664	0.1083	0.8498	0.0924
θ_2^1	1	0.8414	0.0641	0.8250	0.0976	1.3624	0.5364	0.9327	0.1302	0.8247	0.0896
θ_3^1	1	0.8473	0.0702	0.8296	0.0992	1.3767	0.5923	0.9284	0.1461	0.8388	0.0800
θ_4^1	1	0.8569	0.0700	0.8546	0.0899	1.3811	0.5569	0.9466	0.1115	0.8489	0.0845

θ_5^1	1	0.8521	0.0673	0.8270	0.1024	1.4750	0.5597	0.9477	0.1275	0.8314	0.0825
θ_6^1	1	0.8525	0.0670	0.8506	0.0921	1.4863	0.6070	0.9572	0.1229	0.8389	0.0870
θ_7^1	1	0.8460	0.0717	0.8117	0.0920	1.4566	0.5093	0.9626	0.1017	0.8338	0.0853
θ_8^1	1	0.8483	0.0566	0.8444	0.0892	1.4197	0.5169	0.9401	0.1124	0.8339	0.0829
θ_9^1	1	0.8165	0.0669	0.9372	0.0524	1.5056	0.2489	1.0168	0.0658	0.9482	0.0522
θ_{10}^1	1	0.8971	0.0648	0.9017	0.0835	1.4227	0.3361	0.9539	0.0818	0.9115	0.0735
σ Terms											
σ_1	1	0.5312	0.2243	0.9845	0.1162	3.4664	1.1010	1.2305	0.1508	0.9411	0.1033
σ_2	1	0.5227	0.2303	0.9694	0.1179	3.2377	1.1721	1.1783	0.2837	0.9118	0.2174
σ_3	1	0.4877	0.2281	0.9727	0.1174	3.2143	1.1595	1.1630	0.3458	0.9064	0.2840
σ_4	1	0.5411	0.2094	0.9893	0.1042	3.2114	1.2027	1.1904	0.2809	0.9269	0.1889
σ_5	1	0.5340	0.1895	0.9743	0.1163	3.4441	1.0813	1.2049	0.1562	0.9284	0.0962
σ_6	1	0.5277	0.1960	0.9953	0.1055	3.4007	1.2647	1.1941	0.2721	0.9150	0.2138
σ_7	1	0.5026	0.2106	0.9537	0.1238	3.3815	0.9892	1.2218	0.1340	0.9326	0.0900
σ_8	1	0.5294	0.1869	0.9898	0.1072	3.3104	1.0056	1.2174	0.1304	0.9322	0.0929
σ_9	1	0.8938	0.0502	0.9404	0.0434	1.6888	0.5982	0.9739	0.3635	0.8938	0.3268
σ_{10}	1	0.8478	0.0875	0.9607	0.0916	2.4352	0.6084	1.0619	0.2229	0.9426	0.2027
# of Dynamic Program Solutions		45.6 million		15000		15000		45000		258600	
Average "True" LnLikelihood at Starting Values		−15609.24		−15609.24		−28551.32		–		–	
Average "True" LnLikelihood at Estimates		−15303.15		−15120.62		−16862.99		−15133.09		−15081.79	

sampling procedure is not only far quicker, but by using far more draws also seems to be producing better estimates.

Columns 3 through 5 run the importance sampling estimator using the "bad" starting values. Here we iterated the estimation routine until convergence. In Column 3 we present results after the first iteration, in Column 4 we present results after the third iteration, and in Column 5 we present results after convergence (which took an average of 17.24 iterations). Since the "dynamic programming solutions" need to be resolved at each iteration, the number of "dynamic programming solutions" necessary for each column is just the number of estimation iterations times 15000.

Column 3 indicates how important starting values are if one is only going to run the importance sampling routine only once. The estimates are very imprecise and highly biased in comparison to even Column 1. The average "true" likelihood at the estimates is also much worse, −16862.99. One needs to keep this in mind when using this procedure.

However, iterating seems to make these biases disappear quite quickly. In Column 4, after the 3rd iteration, the biases seem approximately the same level as the biases in Column 1, and the average "true" likelihood is considerable better, −15133.09. At convergence in Column 5, the biases seem quite low. The average "true" likelihood at convergence is −15081.79, better than all the other results, including the importance sampled estimates using the "good" starting values (though it interesting that in some dimensions, Column 1 or Column 2 perform better). Compared to the standard SML approach in Column 1, the converged results in Column 5 require approximately 0.006 times the number of the "dynamic programming solutions". So even at the bad starting values, the iterated importance sampling method is both quicker and produces what are arguably better estimates.

7 Conclusion

This paper suggests a new use of importance sampling to reduce computational burden in simulation estimation of complicated models in economics and marketing. We show that combining a *change of variables* with *importance sampling* can reduce computational time by dramatically reducing the number of times that a complicated model needs to be solved or simulated in an estimation procedure. The technique is applicable to a wide range of models, including single or multiple agent dynamic programming problems, and complicated equilibrium problems such as discrete games or auction models. The technique is particularly amenable to allowing considerable amounts of unobserved heterogeneity in one's model. We hope that this technique allows researchers to estimate models that allow for more unobserved heterogeneity, and, more generally, more realistic models. The technique is not without caveats though. In particular, special care must be taken, since misuse of importance sampling can potentially generate high levels of simulation error.

Acknowledgements Thanks to Pat Bajari, Peter Davis, Gautam Gowrisankaran, Wes Hartmann, Mike Keane, Whitney Newey, Ariel Pakes, Juan Pantano, and particularly to Steve Berry for suggestions at an early stage. I would also like to thank 3 anonymous referees, the editor Peter Rossi, and participants at the 2000 Cowles Conference on Strategy and Decision Making, the MIT Econometrics Lunch, UCLA, and the 2000 SITE Conference on Structural Econometric Methods for helpful discussions. A 1999 version of this paper circulated under the title "Importance Sampling and the Method of Simulated Moments", though the 2001 NBER working paper version has the current title. All errors are my own.

References

Ackerberg, D. (2001). *A new use of importance sampling to reduce computational burden in simulation estimation*. NBER working paper #T273

Ackerberg, D. (2003). Advertising, learning, and consumer choice in experience. Good markets: A structural empirical examination. *International Economic Review, 44*, 1007–1040.

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

Aguirregabiria, V., & Mira, P. (2007). Sequential estimation of dynamic discrete games. *Econometrica 75*, 1–53.

Bajari, P. (1998a). *Econometrics of the first price auction with asymmetric bidders*. Mimeo, Stanford Univ.

Bajari, P. (1998b). Econometrics of sealed bid auctions. In *1998 Proceedings of the Business and Economic Statistics Section of the American Statistical Association* (pp. 41–49).

Bajari, P., Benkard, C. L., & Levin, J. (2007a). Estimating dynamic models of imperfect competition. *Econometrica 75*, 1331–1370.

Bajari, P., Fox, J., & Ryan, S. (2007b). Linear regression estimation of discrete choice models with nonparametric random coefficient distributions. *American Economic Review, 97*, 459–463.

Bajari, P., Chernozhukov, V., Hong, H., & Nekipelov, D. (2008). Nonparametric and semiparametric analysis of a dynamic game model. Unpublished working paper.

Bajari, P., Fox, J., Kim, K., & Ryan, S. (2009a). *Discrete choice models with a nonparametric distribution of random coefficient*. Mimeo, U. of MN.

Bajari, P., Hong, H., & Ryan, S. (2009b). Identification and estimation of discrete games of complete information. *Econometrica*, in press.

Berry, S. T. (1992). Estimation of a model of entry in the airline industry. *Econometrica, 60*, 889–917.

Berry, S., Levinsohn, J., & Pakes, A. (1995). Automobile prices in market equilibrium. *Econometrica, 63*, 841–890.

Crawford, G., & Shum, M. (2005). Uncertainty and learning in pharmaceutical demand. *Econometrica, 73*, 1135–1174.

Davis, P. (2006). Estimation of quantity games in the presence of indivisibilities and heterogeous firms. *Journal of Econometrics, 134*, 187–214.

Erdem, T., & Keane, M. (1996). Decision making under uncertainty: Capturing dynamic brand choice processes in turbulent consumer goods markets. *Marketing Science, 15*, 1–20.

Gasmi, F., Laffont, J-J, & Vuong, Q. (1991). Econometric analysis of collusive behavior in a soft drink industry. *Journal of Economics and Management Strategy, 1*, 277–311.

Geweke, J. (1989). Efficient simulation from the multivariate normal distribution subject to linear inequality constraints and the evaluation of constraint probabilities. *Econometrica, 57*, 1317–1339.

Goettler, R., & Clay, K. (2009). *Tariff choice with consumer learning and switching costs*. Mimeo, Chicago GSB.

Gourieroux, C., & Monfort, A. (1991). Simulation based inference in models with heterogeneity. *Annales d'Economie et de Statistique, 20/21*, 69–107.
Hansen, L. (1982). Large sample properties of generalized method of moments estimators. *Econometrica, 50*, 1029–1064.
Hartmann, W. (2006). Intertemporal effects of consumption and their implications for demand elasticity estimates. *Quantitative Marketing and Economics, 4*, 325–349.
Hendel, I., & Nevo, A. (2006). Measuring the implications of sales and consumer inventory behavior. *Econometrica, 74*, 1137–1173.
Hotz, V. J., & Miller, R. A. (1993). Conditional choice probabilities and the estimation of dynamic models. *Review of Economic Studies 60*, 497–529.
Hotz, V. J., Miller, R. A., Sanders, S., & Smith, J. (1994). A simulation estimator for dynamic models of discrete choice. *Review of Economic Studies 61*, 265–289.
Imai, S., Jain, N., & Ching, A. (2009). Bayesian estimation of dynamic discrete choice models. *Econometrica*, in press.
Jofre-Bonet, M., & Pesendorfer, M. (2003). Estimation of a dynamic auction game. *Econometrica 71*, 1443–1489.
Judd, K., & Su, C. (2008). *Constrained optimization approaches to estimation of structural models*. Mimeo, Stanford.
Keane, M., & Wolpin, K. (1994). The solution and estimation of discrete choice dynamic programming models by simulation and interpolation. *Review of Economics and Statistics, 76*, 648–72.
Keane, M., & Wolpin, K. (2000). *Estimating the effect of welfare on the education, employment, fertility and marraige decisions of women*. Mimeo, UPenn.
Keane, M., & Wolpin, K. (2001). The effect of parental transfers and borrowing constraints on educational attainment. *International Economic Review, 42*, 1051–1103.
Lerman, S., & Manski, C. (1981). On the use of simulated frequencies to approximate choice probabilities. In C. Manski & D. McFadden (Eds.) *Structural analysis of discrete data with econometric applications.* (pp. 305–319). Cambridge: MIT.
Maskin, E., & Riley, J. (1996). *Existence of equilibrium in sealed, high bid auctions*. Mimeo.
McFadden, D. (1989) A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica, 57*(5), 995–1026.
Kloek, T., & Van Dijk, H. (1978): Bayesian estimation of equation system parameters: An application of integration by Monte-Carlo. *Econometrica, 46*, 1–20.
Norets, A. (2009) Inference in dynamic discrete choice models with serially correlated unobserved state variables. *Econometrica*, in press.
Pakes, A., Ostrovsky, M., & Berry, S. (2007). Simple estimators for the parameters of discrete dynamic games, with entry/exit examples. *RAND Journal of Economics 38*, 373–399.
Pakes, A., & Pollard, D. (1989). Simulation and the asymptotics of optimization estimators. *Econometrica, 57*, 1027–1057.
Pantano, J. (2008). *Labor market stigma in a forward looking model of criminal behavior*. Mimeo, Washington U, St. Louis.
Pesendorfer, M., & Schmidt-Dengler, P. (2008). Asymptotic least squares estimators For dynamic games. *Review of Economic Studies, 75*, 901–928.
Rust, J. (1987). Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica, 55*, 999–1033.
Rust, J. (1997). Using randomization to break the curse of dimensionality. *Econometrica, 65*, 487–516.
Train, K. (2003). *Discrete choice methods with simulation*. Cambridge: Cambridge University Press.