A New Use of Importance Sampling to Reduce Computational Burden in Simulation Estimation

Daniel A. Ackerberg[*]

June 18, 2001

Abstract

Method of Simulated Moments (MSM) estimators introduced by McFadden (1989) and Pakes and Pollard (1989) are of great use to applied economists. They are relatively easy to use even for estimating very complicated economic models. One simply needs to generate simulated data according to the model and choose parameters that make moments of this simulated data as close as possible to moments of the true data. This paper uses importance sampling techniques to address a significant computational caveat regarding these MSM estimators – that often one's economic model is hard to solve. Examples include complicated equilibrium models and dynamic programming problems. We show that importance sampling can reduce the number of times a particular model needs to be solved in an estimation procedure, significantly decreasing computational burden.

[*] Dept. of Economics, UCLA, Los Angeles, CA 90095 and NBER (ackerber@econ.ucla.edu). Thanks to Steve Berry, Ariel Pakes, Whitney Newey, Peter Davis and participants at the 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 prior version of this paper circulated under the title "Importance Sampling and the Method of Simulated Moments" (1999). All errors are my own.

1 Introduction

Method of Simulated Moments (MSM) estimators (MacFadden (1989), Pakes and Pollard (1989)) have great value to applied economists estimating structural models due to their simple and intuitive nature. Regardless of the degree of complication of the econometric model, one only needs the ability to generate simulated data according to that model. Moments of these simulated data can then be matched to moments of the true data in an estimation procedure. The value of the parameters that sets the moments of the simulated data ”closest” to the moments of the actual data is an MSM estimate. Such estimates typically have nice properties such as consistency and asymptotic normality, even for a finite amount of simulation draws.

This paper addresses a caveat of such procedures that occurs when it is time consuming to solve and generate data from one’s model. Examples include 1) complicated equilibrium problems, e.g. discrete games or complicated auction models, and 2) dynamic programming problems with large state spaces or significant amounts of heterogeneity. In the above estimation procedure, one usually needs to solve such a model numerous times, typically once for every simulation draw, for every observation, for every parameter vector that is ever evaluated in an optimization procedure. If one has N observations, performs NS simulation draws per observation, and optimization requires R function evaluations, estimation requires solving the model $NS * N * R$ times. This can be unwieldy for these complicated problems.

We suggest using a change of variables and importance sampling to alleviate or remove this problem. Importance sampling is a technique most noted for its ability to reduce levels of simulation error. We show that importance sampling can be also be used to dramatically reduce the number of times a complicated economic model needs to be solved within an estimation procedure. Instead of solving the model $NS * N * R$ times, with importance sampling one only needs to solve the model $NS * N$ times or NS times. Since R can be quite large (e.g. when the number of parameters is around 8 and the function is well behaved, at a minimum R might $= 500$ — and R tends to increase exponentially in the number of parameters), this can lead to very significant time savings.

2 The Simple Data Generation MSM Estimator

Consider an 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. θ_0 is a parameter vector that one is trying to estimate.

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:

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

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 $g(x_i)$ of the conditioning variables multiplied by the difference between y and its expectation, i.e.

$$E\left[(y_i - E\left[f(x_i, \epsilon_i, \theta)|x_i\right]) * g(x_i) \quad \right] = 0 \qquad \text{at } \theta = \theta_0 \tag{1}$$

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

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

equal to zero or as close as possible to zero is a consistent estimator of θ_0. Under appropriate regularity conditions, one obtains asymptotic normality of $\widehat{\theta}$ (Hansen (1982)).[1]

Simulation enters the picture when the function $E\left[f(x_i, \epsilon_i, \theta)|x_i\right]$ is not easily computable. The straightforward way of simulating this expectation is by averaging $f(x_i, \epsilon_i, \theta)$ over a set of NS random draws $(\epsilon_{i,1}, \ldots\ldots., \epsilon_{i,NS})$ from the distribution of ϵ_i, i.e.

[1] Note that the vector y can contain higher order moments of the dependent variable (e.g. y, y^2, etc.). As the number of moments used increases, one can approach asymptotic efficiency by the right choice of instruments (i.e. the g function).

$$\widehat{Ef_i}(\theta) = \frac{1}{NS}\sum_{ns} f(x_i, \epsilon_{i,ns}, \theta) \qquad (2)$$

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

$$\widehat{G_N}(\theta) = \frac{1}{N}\sum_i \left[(y_i - \widehat{Ef_i}(\theta)) * g(x_i)\right]$$

as close as possible to zero. Perhaps most important of these statistical properties is the fact that these estimators are typically consistent for *finite* NS. The intuition behind this is that the simulation error (i.e. the difference between the simulated expectation and the true expectation, $\widehat{Ef}_i(\theta) - E\left[f(x_i, \epsilon_i, \theta)|x_i\right]$) averages out over observations as $N \to \infty$.[2] This consistency property gives the estimator an advantage over alternative estimation approaches such as simulated maximum likelihood, which typically is not consistent for a finite number of simulation draws[3,4].

Note that this simulation procedure can be thought of as a data generating procedure. Each draw $\epsilon_{i,ns}$ generates new dependent variables $y_{i,ns}$. Moments of these generated $y_{i,ns}$'s are then matched to the observed y_i's. This illuminates how general this estimation procedure is. One only needs to be able to generate data according to the model.

[2] 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/NS. This means, e.g., that if one uses just 10 simulation draws, simulation increases the variances of the parameter estimates by just 10%.

[3] The difference between consisitency or inconsistency for fixed simulation draws can often be seen dramatically in degree of small sample bias (see, e.g., Ackerberg (1999)).

[4] Both McFadden and Pakes and Pollard note that it is essential to hold the draws $\epsilon_{i,ns}$ contant over different function evaluations (i.e. different θ). Otherwise the likelihood function is infinitely jumpy. It is also usually helpful to use different simulation draws for different observations, as this will tend to make the simulation error average out faster as N increases.

3 Importance Sampling and a Change of Variables to Reduce Computational Burden

A significant caveat of the above simulation procedure is that $f(x_i, \epsilon_{i,ns}, \theta)$ may be hard to compute. Often numerical methods to are needed to evaluate f. The problem is that performing such operations NS times for *each* observation *each* time the function is evaluated within an optimization procedure can be time consuming. This is particularly problematic as the number of parameters increases since the number of function evaluations needed for convergence tends to increase exponentially 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 $f(x_i, \epsilon_{i,ns}, \theta)$ needs to be computed.

Importance sampling addresses the simulation of $E[f(x_i, \epsilon_i, \theta)|x_i]$. Consider an arbitrary integrable p.d.f. g whose density is non-zero over the support of ϵ. Dividing and multiplying by g we have:

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

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

$$\overline{Ef}_i(\theta) = \frac{1}{NS} \sum_{ns} f(x_i, \epsilon_{i,ns}, \theta) \frac{p(\epsilon_{i,ns} \mid x_i, \theta)}{g(\epsilon_{i,ns} \mid x_i)}$$

This is trivially an unbiased simulator of $E[f(x_i, \epsilon_i, \theta)|x_i]$. Unfortunately, using $\overline{Ef}_i(\theta)$ in an estimation procedure still requires computing f $NS * N * R$ times. We combine this importance sampling with a change of variables to solve this computational issue.

Assumption (A1): There exists a function $u(x_i, \epsilon_i, \theta)$ such that $f(x_i, \epsilon_i, \theta) = f(u(x_i, \epsilon_i, \theta))$ and:

.

I) given any $x_i \in \Re^J$, $u(x_i, \epsilon_i, \theta)$ is a random vector whose support *does not* depend on θ.

.

II) given x_i and θ, one can analytically (or quickly) compute the change of variables density of $u(x_i, \epsilon_i, \theta)$ induced by the distribution of ϵ_i.

.

Note the slight abuse of notation as $f(x_i, \epsilon_i, \theta)$ has different arguments than $f(u(x_i, \epsilon_i, \theta))$. One important case where (A1) is violated is when an element of u must contain a parameter by itself. In this case, the support of u clearly does depend on θ. However, many economic models satisfy (A1) – this is exhibited in examples later. We also discuss cases where it is not satisfied and show how one can either 1) still benefit from computational savings using our technique, or 2) how economic models can be perturbed to satisfy it.

Let $p(u_i \mid x_i, \theta)$ be the density of u_i obtained by the change of variables formula. Combining this change of variables with an importance sampling density for u_i, $g(u_i \mid x_i)$, we have:

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

Now consider the unbiased importance sampling simulator of $E\left[f(x_i, \epsilon_i, \theta)|x_i\right]$:

$$\widetilde{Ef}_i(\theta) = \frac{1}{NS}\sum_{ns} f(u_{i,ns})\frac{p(u_{i,ns} \mid x_i, \theta)}{g(u_{i,ns} \mid x_i)}$$

where the $u_{i,ns}$'s are draws from g. Now when θ changes, the $u_{i,ns}$'s *do not* change. As a result, f needs not be recomputed when θ changes. The only components that need to be reevaluated are the numerators of the importance sampling weights, $p(u_{i,ns} \mid x_i, \theta)$, which are typically not computationally burdensome.[5] As a result, in an estimation procedure using $\widetilde{Ef}_i(\theta)$ one only needs to compute f $NS * N$ times. Additionally, if one uses the same g function for every observation, f only needs to be computed NS times. The caveat here

[5] For example suppose $f(x_i, \epsilon_i, \theta) = f(x_i\theta + \epsilon_i)$ and that ϵ_i is multivariate normal. Then with the change of variables $u_i = x_i\theta + \epsilon_i$, the distribution of u_i is also mltivariate normal

is that using the same g function may limit the extent to which simulation error averages out over observations, as simulation error is correlated across observations.

Note the intuition behind this procedure. As we change θ, rather than holding each of the $\epsilon_{i,ns}$ and their implicit weights ($\frac{1}{NS}$) constant, this procedure holds the $u_{i,ns}$ constant and varies the "weights" $\left(\frac{p(u_{i,ns}|x,\theta)}{NS*g(u_{i,ns})}\right)$ on each of the draws. Put another way, rather than changing our simulated "people" when we change θ, we change the weight which we put on each simulated person. As such, f does not need to be recomputed for new simulated people. An additional benefit of this procedure is that while the pure frequency simulator (2) is often discontinuous (if there is any discreteness in one's economic model), these importance sampling simulators are typically smooth[6].

3.0.1 Example 1: A Discrete Game

We consider a model similar to that in Davis (1999). Firm j chooses the number of stores $q_j \in (0, ..., S)$ to operate in a given market m. The cost of operating q_j stores is given by

$$c(q_j) = (\beta x_j + \epsilon_j + (\alpha x_j + \eta_j)q_j)q_j$$

where x_j are observables and ϵ_j and η_j are unobservables. $(\beta x_j + \epsilon_j)$ measures firm j's level of costs, $(\alpha x_j + \eta_j)$ measures its returns to scale. Market inverse demand in market m is a function of the total number of stores $Q_m = \sum_j q_j$ and equal to

$$P(Q_m) = \delta_0 - \delta_1 Q_m + \delta_2 z_m + \mu_m$$

where z_m are observables that shift overall demand and ϵ_m is an unobserved market demand shifter. As there is only actual data on equilibrium Q, and not P, a units normalization is necessary. We normalize $\delta_1 = 1$,[7]

[6] The use of importance sampling as a smoother is briefly discussed in McFadden (1989). An earlier version of the current paper contained a number of interesting examples of how importance sampling can be used to smooth even very complicated economic model. For a copy please consult the author.

[7] This normalization is different than that used by Davis (who normalized $\sigma_\mu = 1$), but is an identical model given that demand is downward sloping. Interestingly, this alternative normalization is what helps satisfy (A1).

implying a profit function:

$$\pi(s_j, Q_m) = (\delta_0 + \delta_2 z_m + \mu_m + \beta x_j + \epsilon_j)s_j + (\alpha x_j + \eta_j)s_j^2 - Q_m s_j$$

While there are multiple equilibrium in this game, Davis shows conditions under which *all* equilibrium consist of the same total number of stores Q_m. Thus he uses an estimation strategy similar to Berry (1992) by estimating the equation

$$y_m = Q_m = \sum_{j_m} q_j = f(x_1, ..., x_{J_m}, \epsilon_1, ..., \epsilon_{J_m}, \eta_1, ..., \eta_{J_m}, z_m, \mu_m, \theta)$$

with the simulated moment

$$E\left[y_m - \frac{1}{NS} \sum_{ns} f(\{x_j\}_{j=1}^{J_m}, \{\epsilon_{j,ns}\}_{j=1}^{J_m}, \{\eta_{j,ns}\}_{j=1}^{J_m}, z_m, \mu_{m,ns}, \theta) \quad | \quad x_m, z_m \right]$$

In this case, not only is the expectation of f not analytic, but the function f itself is very complicated. Given simulated primitives $(\{x_j\}_{j=1}^{J_m}, \{\epsilon_{j,ns}\}_{j=1}^{J_m}, \{\eta_{j,ns}\}_{j=1}^{J_m}, z_m, \mu_{m,ns}, \theta)$, an interative tatonnment procedure is required to solve for Q_m. This estimation algorithm requires computation of this f $NS * N * R$ times (where N is the number of markets).

Consider the change of variables function:

$$u_m = u(\{x_j\}_{j=1}^{J_m}, \{\epsilon_j\}_{j=1}^{J_m}, \{\eta_j\}_{j=1}^{J_m}, z_m, \mu_m, \theta) = \begin{pmatrix} \{\beta x_j + \epsilon_j\}_{j=1}^{J_m} \\ \{\alpha x_j + \eta_j\}_{j=1}^{J_m} \\ \delta_0 + \delta_2 z_m + \mu_m \end{pmatrix}$$

The elements of u_m are sufficient to compute the equilibrium Q_m (this is clear from the profit function), and under Davis' joint normality assumption on (ϵ, η, μ), the function satisfies (A1). The distribution of u_m, $p(u_m \mid x_m, z_m, \theta)$, is simply multivariate normal.[8]

Now consider the importance sampling simulator:

$$\widetilde{Ef}_m(\theta) = \frac{1}{NS}\sum_{ns} f(u_{m,ns})\frac{p(u_{m,ns} \mid x_m, z_m, \theta)}{g(u_{m,ns} \mid x_m, z_m)}$$

where the $u_{m,ns}$ are draws from some distribution $g(u_m)$. As the parameters change, the importance sampling holds the $u_{m,ns}$ constant – as a result the f functions need not be recomputed as θ changes. With this simulator, f only needs to be computed $NS * N$ times instead of $NS * N * R$ times. If one uses the same g function for all markets, f would need to be computed NS times. Note that this importance sampling also smooths the objective function – this is in contrast to the pure frequency simulator, which has flats and jumps.

3.0.2 Example 2: A Dynamic Programming Problem

Consider a dynamic model of automobile choice. Suppose that in a given year the utility consumer i obtains from using car j with characteristics X_j and age a_j is given by $U_{ij} = \beta_i X_j - \gamma_i a_j$ where β_i is a vector of consumer i's idiosyncratic tastes for the characteristics and γ_i measures consumer i's distaste for older cars. In each period the consumer has the option of keeping their old car or purchasing a new one from some set of J cars. The single period utility from purchasing or not purchasing, respectively are

$$\begin{aligned} U_p &= \max_j \left\{\beta_i X_j - \alpha_i p_j\right\} \\ U_{np} &= \beta_i X_{c_i} - \gamma_i a_{c_i} \end{aligned}$$

[8] If one wanted to ensure that the marginal cost of an additional store was positive, one could, for example use $\exp(\beta x_j + \epsilon_j)$ instead of $(\beta x_j + \epsilon_j)$. The first set of elements of the u function then become $\left\{\exp(\beta x_j + \epsilon_j)\right\}_{j=1}^{J_m}$. Note that this function also satisfies (A1) as the support of these elements of u is always $(0,\infty)$ regardless of θ.

where X_{c_i} are characteristics of i's current car, and a_{c_i} is the age of the current car. α_i is consumer i's distaste for price. a_{c_i} does not enter the utility from purchasing a new car because new cars are age 0.

The formal state space of this problem is (c_i, a_{c_i}), i.e. the individual's current car type and its age[9]. This is of fairly small dimension, so it would be possible to numerically solve for i's value function $V_i(c_i, a_{c_i})$ and optimal policy (choice) function $P_i(c_i, a_{c_i})$. Note that the value and policy functions are indexed by i because they depend on consumer i's characteristics, i.e. the vector $(\beta_{i1}, ..., \beta_{iK}, \alpha_i, \gamma_i)$.

Econometrically, one might specify β_i's, α_i, and γ_i as linear functions of consumer characteristics z_i (e.g. income, family size) plus unobservable terms, i.e.

$$\{\beta_{ik} = z_i\beta_k + \epsilon_{ik}\}_{k=1}^{K}$$

$$\alpha_i = z_i\alpha + \epsilon_{iK+1}$$

$$\gamma_i = z_i\gamma + \epsilon_{iK+2}$$

and specify the joint distribution of ϵ_i. Estimation could proceed by simulating from the distribution of ϵ_i, solving the dynamic programming problem for each simulated individual (characterized by $(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins})$) and matching simulated choices to actual choices, i.e.

$$G_N(\theta) = \frac{1}{N}\sum_i \left[(P_i - \widehat{EP}_i(\theta)) \otimes g(X, z_i)\right]$$

where $\widehat{EP}_i(\theta)$ is the average of the simulated choices (policies)[10], i.e.

$$\widehat{EP}_i(\theta) = \frac{1}{NS}\sum_{ns} P(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins}, c_i, a_{c_i})$$

and P_i is the observed choice.

[9] This assumes prices and characteristics are not changing over time. Because of the large number of products, it would likely not be feasible to include a complicated stochastic path of prices. On the other hand, an iid price process could likely be incorporated using alternative specific value functions similar to Rust (1988). We also ignore initial conditions problem regarding correlation between current car and tastes. This might be valid, e.g. if the sample was a panel of first car buyers.

[10] One can think of P as a vector of 0-1 choices (i.e. which car is bought).

The problem with the above estimation procedure is that as θ changes, the simulated $(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins})$'s change. Thus, the dynamic programming problem needs to be solved $NS * N * R$ times. Again importance sampling can help reduce computational burden. Consider changes of variables given by:

$$u_i = u(z_i, \epsilon_1, ..., \epsilon_{K+2}, \theta) = \begin{pmatrix} \{z_i\beta_k + \epsilon_{ik}\}_{k=1}^{K} \\ z_i\alpha + \epsilon_{iK+1} \\ z_i\gamma + \epsilon_{iK+2} \end{pmatrix}$$

and the importance sampling simulator

$$\widetilde{EP}_i(\theta) = \frac{1}{NS}\sum_{ns} f(u_{i,ns})\frac{p(u_{i,ns} \mid z_i, \theta)}{g(u_{i,ns} \mid z_i)}$$

As parameters change, the $u_{i,ns}$'s do not change. As such, the dynamic programming problem $V_i(c_i, a_{c_i})$ only needs to be computed $NS * N$ times – once for each simulation draw for each individual. As with the previous model, one could reduce the number of computations to NS by using the same simulation draws for each individual.

4 Discussion

4.0.3 Satisfying or Partially Satisfying Assumption 1

The two examples above satisfy (A1), but for some models one might not be able to find a u that does. The most common case is when there are parameters in one's model that do not vary unobservably across the population and do not enter into an index function that has some unobservable component that varies across the population. In Example 2, for instance, one might be interested in estimating (rather than fixing) a discount factor that is constant across the population. As the parameter has a degenerate distribution, its support *does* change with θ. It would also be very hard to find some random function of the discount factor which both 1) summarizes its impact on the model and 2) has an analytic density. In Example 1, one might consider an alternative model where returns to scale are the same across firms, i.e. $\alpha x_j + \eta_j = \alpha$. In this case it would again be hard to find a u to satisfy (A1). In these examples, the f functions would need to be recomputed if the discount factor or the returns to scale parameter changed.

While it might be hard to find a u that fully satisfies (A1), it is often possible to find u's that partially satisfy it. By partially satisfying it, we mean that we can find a u that has an analytic density and whose support only depends on a *subset* of the parameters. Denote by θ_1 the set of parameters which affect the support of u — θ_2 is the set of parameters that do not change the support. As θ_1 changes, f needs to be recomputed, but as θ_2 changes, it does not. Clearly, u should be chosen to minimize the number of parameters in θ_1. In the dynamic programming model with the discount factor, for example, the discount factor would be in θ_1, the rest of the parameters in θ_2[11].

If (A1) is partially satisfied, a first option is to use derivative based optimization methods. In computing numeric first derivatives, f needs to be recomputed only when elements of θ_1 are perturbed. This reduces computational time by $\frac{\dim(\theta_1)}{\dim(\theta)}$ relative to a standard procedure. A second alternative is to use a nested search algorithm. On the outside, one searches over θ_1, on the inside, over θ_2. For the inside search algorithm, one needs not recompute f's. As these nested search algorithms are generally inefficient, this approach is reasonable only if the dimension of θ_1 is small, e.g. 1 or 2.

An alternative to the above approach is to slightly perturb one's model to satisfy (A1). Typically this involves adding unobserved heterogeneity to the model. When estimating the discount factor, one might be willing to assume that agents are heterogenous in their discount factors. This model would satisfy (A1)[12]. In example 2, one might allow heterogeneity in returns to scale (as in the text). Interestingly, our technique works better when there is more heterogeneity in the population. The intuition behind this is that the heterogeneity

[11] The simulator in this case would be $\widetilde{EP}_i(\theta) = \frac{1}{NS}\sum_{ns} f(u_{i,ns},\theta_1)\frac{p(u_{i,ns}|x_i,\theta_2)}{g(u_{i,ns}|x_i)}$, so changes in θ_2 are adjusted for with importance sampling weights, changes in θ_1 adjusted for with changes in f.

[12] As one needs the discount factor between 0 and 1, one could use, e.g. $\frac{\exp(\beta+\omega_i)}{1+\exp(\beta+\omega_i)}$ where ω_i is a normal.

allows the econometrician to “span” parameter space with the initial draws. If the parameter space can be spanned, then the moment condition can be evaluated at alternative θ’s by weighting the initial draws. One caveat of this procedure is that the variance of the unobserved heterogeneity must be bounded away from zero. If this variance is zero, then (A1) is no longer satisfied. In practice, one should be careful to watch for these variances approaching zero during estimation. If they do, it is probably best to switch to the first approach.

4.0.4 Choice of g

As mentioned, the 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 increase the variance of simulation estimators. When performing the above change of variables and importance sampling, one needs to be aware of this issue.

Perhaps the obvious choice for g is p itself at some initial parameter vector θ'. This importance sampling simulator is identical to the pure frequency simulator at $\theta = \theta'$. What is attractive about the pure frequency simulator is that asymptotically its variance is 1/NS times the variance due to the data generating process. Thus, with g defined as p at some θ', simulation error in our procedure also has this property at $\theta = \theta'$.

Unfortunately, with this choice of g simulation error can get quite large as θ gets far away from θ'. While theoretically this is not a problem if the parameter space is bounded, this can be an issue in practice. One needs to be careful that θ does not stray too far from θ'. There are a number of ways to do this that we have found to work well in some simple experiments. First, one might repeat the estimation process several times, updating θ' at each repetition. Second, if one is using a (first) derivative based search algorithm, one could at least begin the algorithm by changing θ' at every iteration. Since numeric derivatives are taken in a region where $\theta \simeq \theta'$, the simulation error in these derivative should be of order $1/NS$. Even though the f functions need to be recomputed at each iteration in this case, they do not need to be recomputed at each parameter perturbation when computing derivatives. Thus the time savings (relative to the standard procedure) will be

$1/(K+1)$, where K is the number of parameters[13]. After one is relatively confident that the parameters are in the neighborhood of the extremum, θ' can be held constant over iterations. This ensures that the procedure converges. Third, one might pay close attention to the search procedure. If parameters stray too far from θ', it can be updated.[14]

Lastly, note that one might be able to use the importance sampling to one's advantage in reducing simulation error. This would involve using an initial guess at θ and oversampling parts of the p distribution that are most informative about the integral (typically those that lead to a high value of the integrand).

4.0.5 Comparison to Discretation/Randomization Approaches

Note that an alternative strategy for the dynamic programming problem of example 2 would be to explicitly solve for the value and policy functions as depending on the individual specific parameters, i.e.

$$V(\beta_{i1}, ..., \beta_{iK}, \alpha_i, \gamma_i, c_i, a_{c_i}) \text{ and } P(\beta_{i1}, ..., \beta_{iK}, \alpha_i, \gamma_i, c_i, a_{c_i})$$

If one could solve for these functions, one would only need to solve it *once*. Then when simulating a particular individual at a particular parameter vector, one can just plug the resulting $(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins})$ into P to compute the simulated policy. However, the time required to solve a dynamic programming problem typically increases exponentially in this "state" space. Thus, if the dimension of heterogeneity (i.e. K) is large, this will generally not be feasible. Since the $(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins})$ are continuous, this would also require some discretation, as V can only be solved for at a finite number of points. Even so, if each dimension of heterogeneity is discretized into 10 points, this procedure would implicitly require solving for $V(c_i, a_{c_i})$ 10^{K+2} times, considerably more than the $NS * N$ or NS times above. This discretation also adds error to the problem and likely destroys econometric consistency.

[13] This is if one uses one-sided numeric derivatives. The time savings would be $1/2K$ if using two sided derivatives.

[14] Something else we have found to help is to use a g function where the variance of the heterogeneity in the model is larger than that at the initial set of parameters. This helps span parameter space better and appears to reduce simulation error at θ far from θ' (although it tends to increase simulation error nearby θ')

In recent work, Keane and Wolpin (1994) and Rust (1997) suggest using randomization to approximate $V(\beta_{i1},...,\beta_{iK},\alpha_i,\gamma_i,c,a_c)$. The procedure is that instead of discretizing the state space, one *randomly* chooses state space points at which to approximate the value function. Rust shows that such randomization can often break the curse of dimensionality in the dimension of the state vector, though computational time still increases polynomially in order to achieve a given degree of approximation error.

After using such an approach to approximate V, simulation estimation would proceed by drawing sets of $(\beta_{i1ns},...,\beta_{iKns},\alpha_{ins},\gamma_{ins})$, computing simulated choices $P(\beta_{i1ns},...,\beta_{iKns},\alpha_{ins},\gamma_{ins},c,a_c)$, and matching these simulated choices to observed choices. Since one's simulation draws will generally not equal the points at which the value function is approximated, one needs additional approximation (e.g. interpolation) to compute $P(\beta_{i1ns},...,\beta_{iKns},\alpha_{ins},\gamma_{ins},c_i,a_{c_i})$.

Our methodology is related to this in that the value function is also being computed at a random set of points. However, in our procedure, the points for which we solve the value function are *exactly* the points that are chosen by the simulation process in the estimation routine. As a result, there is no approximation error in computation of value and policy functions – the functions we solve for are exact[15]. While there is only one source of simulation error in our estimator (that in the estimation process), the Rust method has two (the estimation process and that in the value function approximation).

While the Rust methodology addresses the curse of dimensionality by brute force (directly going at the value function) our methodology in some sense avoids it. The key is that with our estimation method, one never needs to solve for the entire value function – one only need to solve it for the simulation draws used in the estimation procedure. As such, the standard results on breaking the curse of dimensionality through Monte-Carlo integration apply. There are caveats with both procedures however. Our procedure only breaks the curse of dimensionality in the unobserved individual heterogeneity "state variables", i.e. $(\beta_{i1},...,\beta_{iK},\alpha_i,\gamma_i)$. If the dimension of the observed state variables that evolve over time, i.e. (c,a_c), increases (e.g. consumers are

[15] This relies on c_i and a_C being in discrete space. Even if they were not, we would still expect considerably less approximation error in our procedure, as our procedure would only need to discretize a subset of the state space rather than the entire state space.

allowed to own multiple cars), computational time will go up exponentially. Interestingly, Rust's randomization method does the reverse. It tends to breaks the curse of dimensionality in the true state variables, but doesn't break the curse in unobserved individual heterogeneity[16]. The reason is that because this heterogeneity is constant over time, the value function doesn't have the ability to self approximate itself. It should be noted that this is more of a technical issue than a practical one – one can still solve for the value function at a random set of points and use approximation for points in between. One thing this discussion suggests is the possibility of combining the two methods to break the curse of dimensionality in *all* variables. To do this, one would follow our procedure and use the randomization technique to compute $V(c_i, a_{c_i})$ for each of the NS simulated individuals. This has the potential to break the curse of dimensionality in *all* the state variables.

4.0.6 Relation to Keane and Wolpin (2000)

Independently, in two empirical papers, Keane and Wolpin use a procedure that is related to ours in order to solve problems of unobserved state variables. These papers analyze dynamic programming problems of educational choice (forthcoming) and fertility/marriage choice (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 a number of 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

[16] In our example, it actually doesn't break the curse of dimensionality in a either. The reason is that a evolves deterministically. If a evolved stochastically with constant support, the Rust method would break the curse of dimensionality.

observation i's path. Practically, since there are so many possible paths (and since some of the outcome variables are continuous), this results in likelihood zero events. To mitigate this problem, Keane and Wolpin add measurement error to *all* outcome variables.[17] This gives any observed path a positive likelihood and allows for estimation using Simulated Maximum Likelihood.

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 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 *does* require re-solving NS 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, $p(f_i\mid\theta)$ is the distribution of dynamic programming outcomes (without measurement error). Importance sampling these dynamic programming outcomes with some distribution g over outcomes gives:

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

[17] Note that our simulation procedure is also prone to generating likelihood zero events, and thus is more appropriate for MSM (rather than MSL) estimation. If one wanted to use MSL with our technique, one could use Keane and Wolpin's measurement error methodolgy (or, e.g. kernel smoothing) to solve this issue.

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

$$\frac{1}{NS}\sum_{ns} L(f_{ns} + \eta_i = y_i \mid f_{ns}) \frac{p(f_{ns} \mid \theta)}{g(f_{ns})}$$

where the f_{ns}'s are simulated paths generated at θ'. As θ changes, only $p(f_{ns} \mid \theta)$ needs to be recomputed. This 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[18]

5 Conclusion

This paper suggests a new use of importance sampling to reduce computational burden in simulation of complicated economic models. We show that combining a change of variables with importance sampling can reduce estimation time by dramatically reducing the number of times that a complicated economic 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 or complicated equilibrium problems such as discrete games or auction models. Use of this technique allows economists to estimate models that were previously intractable to estimate.

References

[1] Ackerberg, D (1999) ”A Smooth Simulated Moment Estimator for Discrete Choice Panel Data with Arbitrary Error Structure” mimeo, Boston Univ.

[2] Berkovec, James; Stern, Steven. 1991. “Job Exit Behavior of Older Men”, *Econometrica*, 59(1), January 1991, pages 189-210.

[18] There are a few parameters of the Keane and Wolpin model, i.e. the proportion of each simulated “type” in the population, where the DP problem does not need resolving as these parameters change.

[3] Berry, Steven T. 1992 ”Estimation of a Model of Entry in the Airline Industry”, *Econometrica*, 60.

[4] Bőrsch Supan, A., and Hajivassiliou, V. 1993. “Smooth Unbiased Multivariate Probability Simulators for Maximum Likelihood Estimation of Limited Dependent Variable Models”, *Journal of Econometrics*, 58(3), 347-368.

[5] Davis, P. 1999 “Estimation of Cournot Models in the Presence of Indivisibilities and Heterogeous Firms” mimeo, MIT

[6] Elrod and Keane. 1995, “A Factor-Analytic Probit Model for Representing the Market Structure in Panel Data”, *Journal of Marketing Research*, Feb. 1995, Vol. XXXII, 1-16.

[7] Geweke, J. 1989, “Efficient Simulation from the Multivariate Normal Distribution Subject to Linear Inequality Constraints and the Evaluation of Constraint Probabilities”

[8] Geweke, John F.; Keane, Michael P.; Runkle, David E. 1997, “Statistical Inference in the Multinomial Multiperiod Probit Model”, *Journal of Econometrics*, 80(1), pages 125-65.

[9] Hajivassiliou, V. 1993, “Simulation of multivariate normal rectangle probabilities and their derivatives: the effects of vectorization”, *International Journal of Supercomputer Applications*, Fall, 231-253.

[10] Hajivassiliou, V. 1994, “A Simulation Estimation Analysis of External Repayments Problems of Developing Countries”, *Journal of Applied Econometrics*, 9(2), 109-132.

[11] Hajivassiliou, V. 1996. “A Monte Carlo Comparison of Leading Simulation Estimators for LDV Models”, Mimeo, Department of Economics, London School of Economics.

[12] Hajivassiliou, V. 1997, “Simulation-Based Inference and Diagnostic Tests: Some Practical Issues”, Cambridge University Press

[13] Hajivassiliou, V. and Ruud, P. 1994, “Classical Estimation Methods Using Simulation” Pages 2383-2441 of: Engle, R., and McFadden, D. (eds), *Handbook of Econometrics*, Vol. 4. North Holland.

[14] Hajivassiliou, Vassilis A.; McFadden, Daniel L. 1998, “The Method of Simulated Scores for the Estimation of LDV Models”, *Econometrica*, 66(4), July 1998, pages 863-96.

[15] Hajivassiliou, V., McFadden, D., and Ruud, P. 1996, “Simulation of Multivariate Normal Rectangle Probabilities and Their Derivatives: Theoretical and Computational Results”, *Journal of Econom7etrics*, 72(1&2), 85-134.

[16] Hansen, Lars (1982) “Large Sample Properties of Generalized Method of Moments Estimators” *Econometrica*, 50

[17] Keane, M. 1994. “A Computationally Efficient Practical Simulation Estimator for Panel Data”, *Econometrica*, 62(1), 95-116.

[18] Keane, Michael P.; Wolpin, Kenneth I. 1994, “The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation”, *Review of Economics and Statistics*, 76(4), November 1994, pages 648-72.

[19] Keane, Michael P.; Wolpin, Kenneth I. Forthcoming, “The Effect of Parental Transfers and Borrowing Constraints on Educational Attainment”.

[20] Keane, Michael P.; Wolpin, Kenneth I. 2000, “Estimating the Effect of Welfare on the Education, Employment, Fertility and Marraige Decisions of Women”, mimeo, NYU and UPenn.

[21] Lee, Lung Fei. 1995, “Asymptotic Bias in Simulated Maximum Likelihood Estimation of Discrete Choice Models”, *Econometric Theory*, 11(3), August 1995, pages 437-83.

[22] Lee, Lung Fei. 1998, “Simulated Maximum Likelihood Estimation of Dynamic Discrete Choice Statistical Models: Some Monte Carlo Results”, *Journal of Econometrics* 82(1), January 1998, pages 1-35.

[23] Lerman, S. and Manski, C. 1981. “On the Use of Simulated Frequencies to Approximate Choice Probabilities”, Pages 305-319 of: Manski, C., and McFadden, D. (eds), *Structural Analysis of Discrete Data with Econometric Applications. MIT Press*.

[24] McCulloch, R., and Rossi, P. 1994, “An Exact Likelihood Analysis of the Multinomial Probit Model”, *Journal of Econometrics*, 64.

[25] McFadden, D. 1989, “A Method of Simulated Moments for Estimation of Discrete Response Models without Numerical Integration”, *Econometrica*, 57(5), 995-1026.

[26] McFadden, Daniel; Ruud, Paul A. 1994, “Estimation by Simulation”, *Review of Economics and Statistics*, 76(4), November 1994, pages 591-608.

[27] Pakes, A., and Pollard, D. 1989, “Simulation and the Asymptotics of Optimization Estimators”, *Econometrica*, 57, 1027-1057.

[28] Rust, J. 1987 ”An Empirical Model of Harold Zurcher”, *Econometrica*

[29] Rust, J. 1997. ”Using Randomization to Break the Curse of Dimensionality”, *Econometrica, 66*

[30] Stern, S. 1992, “A Method for Smoothing Simulated Moments of Discrete Probabilities in Mutinomial Probit Models”, *Econometrica*, 60, 943-952.

[31] Stern, Steven 1994,“Two Dynamic Discrete Choice Estimation Problems and Simulation Method Solution”, *Review of Economics and Statistics*, 76(4), November 1994, pages 695-702.