Importance Sampling and the Method of Simulated Moments

Daniel A. Ackerberg[*]

First Version: August 1, 1999

This Version: April 20, 2000

Abstract

Method of Simulated Moments (MSM) estimators introduced by McFadden (1989) and Pakes and Pollard (1989) are of great use to applied economists because of their ease of use even for estimating extremely 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 two caveats regarding these MSM estimators. First, if there are discrete parts of one's model, MSM objective functions are typically discontinuous in the parameter vector, making them hard to miminize or mimimize correctly. McFadden (1989) briefly suggests the use of importance sampling to smooth simulated moments – we elucidate and expand on this technique. Second, 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, Boston University and NBER. Thanks to Steve Berry for helpful discussions. All errors are my own.

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 estimators typically have nice properties such as consistency and asymptotic normality, even for a finite amount of simulation draws.

This paper addresses two computational problems that can arise with such estimators. The first occurs when there is any discreteness in one's econometric model. In this case, the above simulation process typically results in an objective function that is not continuous in the parameter vector. This can be extremely problematic in optimization, particular when one is searching over many parameters. Not only can this make estimation take longer, but likely increases the probability of erroneously finding local extremum or non extremum.

The second problem occurs when one's economic model is computationally time consuming to solve. Examples include dynamic programming problems with large state spaces and complicated equilibrium problems. 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 I observations, performs NS simulation draws, and optimization requires R function evaluations, estimation requires solving the model $NS * I * R$ times. This can be unwieldly for complicated problems.

This paper suggests using importance sampling to alleviate or remove these problems. Importance sampling is a technique most noted for its ability to reduce levels of simulation error. McFadden (1989) briefly notes that importance sampling has an alternative use - that of smoothing simulated moments, i.e. addressing our first computational problem. The technique is quite simple for a simple multinomial choice model. This paper expands and develops this technique, noting that it can be applied to much more complex models. The key step in its application is finding the right change of variables to do the importance sampling over. We exhibit this smoothing technique with a number of examples.

We next exhibit that importance sampling can be used to alleviate our second problem. What we show is that importance sampling can be used to dramatically reduce the number of times a complicated economic model needs to be solved within an estimation procedure. Instead of naively solving the model $NS * I * R$ times, with importance sampling one only needs to solve the model $NS * I$ 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. This technique is again illustrated with examples.

1. 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 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, y) 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.

$$E\left[(y_i - E\left[f(x_i, \epsilon_i, \theta)|x_i\right]) * g(x_i) \quad \right] = 0 \qquad \text{at } \theta = \theta_0 \tag{1.1}$$

As such, 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 $E\left[f(x_i, \epsilon_i, \theta)\right]$ 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 $(\epsilon_1, \ldots\ldots, \epsilon_{NS})$ from the distribution of ϵ_i, i.e.

$$\widehat{Ef}(\theta) = \frac{1}{NS}\sum_{ns} f(x_i, \epsilon_{ns}, \theta)$$

[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).

$\widehat{Ef}(\theta)$ 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:

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

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 simulation error (i.e. the difference between the simulated expectation and the true expectation $\widehat{Ef}(\theta) - E\left[f(x_i, \epsilon_i, \theta)|x_i\right]$) 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]. Both McFadden and Pakes and Pollard note that it is essential to hold the draws ϵ_{ns} contant over different function evaluations (i.e. different θ). Otherwise the likelihood function is infinitely jumpy[4].

Note that this simulation procedure can be thought of as a data generating procedure. Each draw ϵ_{ns} generates a new dependent variable y_{ns}. The averages of these generated y_{ns}'s are then matched to the observed y's. This also illuminates how general this estimation procedure is. One simply needs to be able to generate data according to the model.

1.1. Caveats and Solutions

An important caveat of this estimation procedure is when the function $f(x_i, \epsilon_i, \theta)$ has some discreteness in it, i.e. when f is not continuous in its arguments. The simplest example of such discreteness is a binary discrete choice model. Other examples may have both continuous and discrete parts or have multiple discrete parts.

In such models, $E\left[f(x_i, \epsilon_i, \theta)|x_i\right]$, the true expectation, *is* typically continuous in the parameter vector θ. However, the simulated expectation above, $\widehat{Ef}(\theta)$, will tend *not to be* continuous in θ, typically having both flats and jumps. This can be very problematic in the numeric minimization of $\widehat{G_N}(\theta)$. Derivative based methods are useless, and in our experiences, non-derivative based methods (e.g. the Nelder-Mead simplex algorithm) work very poorly, especially as the number of parameters one is searching over increases.

[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] 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.

A second caveat is that $f(x_i, \epsilon_i, \theta)$ may be hard to compute. Examples include dynamic optimization problems by agents or complicated equilibrium problems. Both may require numerical methods to evaluate. Performing such operations NS times for *each* observation *each* time the function is evaluated within an optimization procedure can be time consuming. Again, this gets particularly problematic when the number of parameters to be estimated increases because the number of function evaluations needed for convergence tends to increase exponentially in the number of parameters.

Importance sampling is most noted for its ability to reduce simulation error. We suggest using importance sampling techniques for an alternative purpose - to overcome both non-smoothness problems and computational problems. McFadden (1989) noted the ability to use importance sampling to smooth simulations. We illuminate and expand this technique - the trick is to get the right change of variable to importance sampling over. We then show how importance sampling can help our second caveat by reducing the number of times that $f(x_i, \epsilon_i, \theta)$ needs to be computed.

The way we proceed is through use of examples. We start with a simple model, the binary probit, which actually doesn't require simulation, but makes for a simple example. We then illustrate 5 more examples of smoothing: an ordered model, a panel data discrete choice model, a discrete duopoly game model (similar to that in Berry (1992)), and a stochastic stopping time model (similar to that in Ackerberg, Machado, and Riordan (1999)). We end by examining two examples of how importance sampling can reduce computational burden. The first is a oligopolistic discrete quantity setting game (similar to that in Davis (1999)), and the second is a dynamic programming problem.

2. Smoothing - The Probit Model

For the probit case, we have the model

$$y_i = I(\theta x_i + \epsilon_i > 0)$$

Note that in this case $E\left[f(x_i, \epsilon_i, \theta)|x_i\right]$ is simply $\mathit{prob}(x_i, \theta)$, the probability that choice 1 is chosen given x_i. Straightforward application of the previous section results in a sample simulated moment[5]

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

where each ϵ_{ns} is a random draw from $p(\epsilon)$ (a normal distribution).

[5] Again, simulation isn't necessary here - this example is for illustrative purposes.

The problem here is that

$$\widehat{Ef}(\theta) = \frac{1}{NS}\sum_{ns} I(\theta x_i + \epsilon_{ns} > 0)$$

is not continuous in θ. Essentially this simulated probability is just a count - it is the proportion of draws where $\theta x_i + \epsilon_{ns} > 0$. As θ changes, this proportion will either not change or jump as the number of draws crossing the discrete threshold either doesn't change or changes discretely.

McFadden (1989) suggested that a way of smoothing $G_N(\theta)$ is importance sampling. Illuminating on this procedure, note that a change of variables gives:

$$E\left[f(x_i, \epsilon_i, \theta)\right] = E\left[I(\beta x_i + \epsilon_i > 0)\right] = \int I(\theta x + \epsilon > 0)p(\epsilon)d\epsilon = \int I(u > 0)p(u \mid x, \theta)du$$

where $u = \theta x + \epsilon$ and $p(u \mid x, \theta)$ is the distribution of u given x, θ, and $p(\epsilon)$. This

$$= \int \frac{I(u > 0)p(u \mid x, \theta)}{g(u)} g(u)du$$

for arbitrary integrable functions $g(u)$ that are non-zero over the entire support of u.

Suppose $g(u)$ is a p.d.f., and that we can draw random variables $u_1, ..., u_{NS}$ from this p.d.f.. Construct

$$\widetilde{Ef}(\theta) = \frac{1}{NS}\sum_{ns} \frac{I(u_{ns} > 0)p(u_{ns} \mid x, \theta)}{g(u_{ns})}$$

Note that

$$E\left[\widetilde{Ef}(\theta)\right] = E\left[\frac{I(u_{ns} > 0)p(u_{ns} \mid x, \theta)}{g(u_{ns})}\right] = \int \frac{I(u > 0)p(u \mid x, \theta)}{g(u)} g(u)du = E\left[f(x_i, \epsilon_i, \theta)\right]$$

so this importance sampling simulator is also an unbiased simulator of the true expectation[6].

For our purposes, what is most important is that the simulator $\widetilde{Ef}(\theta)$ will generally be continuous in θ and have non-zero derivative w.r.t θ. The reason is that $\widetilde{Ef}(\theta)$ only depends on θ through $p(u \mid x, \theta)$, which is continuous in θ given that $p(\epsilon)$ is continuous and non-zero over its support.

Note the intuition here. As we change θ, rather than holding each of the ϵ_{ns} and their implicit weights ($\frac{1}{NS}$) constant, this procedure holds the u_{ns} constant and varies the "weights" $\left(\frac{p(u_{ns}|x,\theta)}{NS*g(u_{ns})}\right)$ on each of the draws. Put

[6] This unbiased property is *not* the case for a Kernel smoothed simulator, an alternative smooth simulator suggested by McFadden. As such, estimators based on kernel smoothed simulators are generally not consistent unless the bandwidth approaches zero. Of course, as the bandwidth approaches zero, one approaches the step original functions. "Close to step" functions are likely just as hard to optimize over.

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 the indicator functions do not change when θ changes and the resultant simulator is smooth.

In enacting this simulator, one natural choice for $g(u)$ is $p(u \mid x, \theta^*)$ where θ^* is some guess or preliminary estimate of θ. This choice results in an importance sampling simulator that is exactly the straightforward simulator at $\theta = \theta^*$ (the difference arises away from $\theta = \theta^*$). In computation, the $I(u_{ns} > 0)$'s and $g(u_{ns})$'s should be stored as they do not vary as θ changes in the estimation procedure. Then as θ changes, one only needs to re-compute the density $p(u_{ns} \mid x, \theta)$.

This is not the only method for smoothing. A commonly used method for smoothing complicated problems is kernel smoothing. Kernel smoothing effectively adds some extra randomness to the model that smooths

3. More Complicated Examples of Smoothing

3.1. Ordered Model

We can express the ordered model as

$$y = f(x_i, \epsilon_i, \theta_0) = \begin{pmatrix} I\left(-\infty < X_i\beta + \epsilon_i < K_1\right) \\ I\left(K_1 < X_i\beta + \epsilon_i < K_2\right) \\ . \\ . \\ I\left(K_{J-1} < X_i\beta + \epsilon_i < \infty\right) \end{pmatrix}$$

Note that both the cutoffs $K_1, ..., K_J$ and β are part of the parameter vector θ. By simulating $Ef(x_i, \epsilon_i, \theta)$, we can use (1.1) as an MSM estimator of θ[7].

We focus on simulating one element of the dependent variable vector, $E[I\left(K_1 < X_i\beta + \epsilon_i < K_2\right)]$ - the other elements are similar. Again, straightforward simulation of $Ef(x_i, \epsilon_i, \theta)$, i.e.

$$\widehat{Ef}(\theta) = \frac{1}{NS}\sum_{ns} I(K_1 < X_i\beta + \epsilon_{ns} < K_2)$$

will not be continous in θ, as changing either K_1, K_2, or β will either not change or discretely change $\widehat{Ef}(\theta)$.

Note that the change of variables used in the previous section for the probit model will not work here –

[7] Again, simulation may not be necessary for this model, e.g. the ordered *probit*.

while that would smooth the problem with respect to β, $\widehat{Ef}(\theta)$ would still be discontinuous in the parameters K_1 and K_2. The solution here is to use a slightly different change of variables,

$$u_{ns} = \frac{x_i\beta + \epsilon_{ns} - K_1}{K_2 - K_1}$$

resulting in the smooth importance sampling simulator

$$\widetilde{Ef}(\theta) = \frac{1}{NS}\sum_{ns}\frac{I(0 < u_{ns} < 1)p(u_{ns} \mid x, \theta)}{g(u_{ns})}$$

Note that in this case, there is a non-unitary Jacobian in the transformation from ϵ_{ns} to u_{ns}. If ϵ_{ns} was $\mathrm{N}(0,1)$ for example, $p(u_{ns} \mid x, \theta)$ would equal $(K_2 - K_1)\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(K_2-K_1)u_{ns}-X\beta+K_1}{2\sigma^2}\right]$. Again a natural choice of $g\left(u_{ns}\right)$ is $p\left(u_{ns} \mid x, \theta\right)$ at some preliminary θ[8].

3.2. Panel Data Discrete Choice Model

As McFadden (1989), we express a panel data discrete choice model as:

$$y = f(x_i, \epsilon_i, \theta_0) = \begin{pmatrix} I_1 \\ I_2 \\ . \\ . \\ I_S \end{pmatrix} = \begin{pmatrix} I\left(X_1\theta + \epsilon_1 > 0\right) \cap I\left(X_2\theta + \epsilon_2 > 0\right) \cap ..... \cap I\left(X_T\theta + \epsilon_T > 0\right) \\ I\left(X_1\theta + \epsilon_1 < 0\right) \cap I\left(X_2\theta + \epsilon_2 > 0\right) \cap ..... \cap I\left(X_T\theta + \epsilon_T > 0\right) \\ I\left(X_1\theta + \epsilon_1 < 0\right) \cap I\left(X_2\theta + \epsilon_2 < 0\right) \cap ..... \cap I\left(X_T\theta + \epsilon_T > 0\right) \\ . \\ I\left(X_1\theta + \epsilon_1 < 0\right) \cap I\left(X_2\theta + \epsilon_2 < 0\right) \cap ..... \cap I\left(X_T\theta + \epsilon_T < 0\right) \end{pmatrix}$$

so each element of y is an indicator for a particular *sequence* of choices through time. Note that the number of elements of y is $S = J^T$ where J is the number of possible choices in each period and T is the number of time periods (the above equation is where there is a binary choice in each period). The multivariate distribution $p(\epsilon_1, ..., \epsilon_T; \theta)$ is specified and may depend on theta, e.g. if the ϵ's are serially correlated over time or if there is a random effect. Note that the expectation of f, $Ef(x_i, \epsilon_i, \theta)$, is a vector of the probabilities of observing each possible choice sequence. Unlike the above two examples, these sequence probabilities are typically at least T dimensional integrals that are generally not possible to compute analytically.

Again, straightforward simulation of $Ef(x_i, \epsilon_i, \theta)$ is not continuous in θ, but importance sampling can help.

[8] One can easily simulate all the elements of y jointly.

Return to the change of variables

$$u_{tns} = X_t\theta + \epsilon_{tns} \qquad \forall t$$

and the smooth importance sampling simulator

$$\widetilde{Ef}(\theta) = \frac{1}{NS}\sum_{ns} \frac{\left(\begin{array}{c} I\left(u_{1ns} > 0\right) \cap I\left(u_{2ns} > 0\right) \cap \ldots.. \cap I\left(u_{Tns} > 0\right) \\ . \\ I\left(u_{1ns} < 0\right) \cap I\left(u_{2ns} < 0\right) \cap \ldots.. \cap I\left(u_{Tns} < 0\right) \end{array} \right) p(u_{ns} \mid x, \theta)}{g(u_{ns})}$$

If ϵ is multivariate normal, $p(u_{ns} \mid x, \theta)$ is also multivariate normal.

This simulator has very useful properties. Of the S choice sequences, this simulator will be non-zero for at most NS of the sequences. For these NS sequences, the simulated probabilities vary smoothly as θ changes. The $(S - NS)$ sequences that get zero probability will have zero probability regardless of what θ is. This is actually a very good characteristic for estimation purposes. One of the problems with panel data discrete choice models is that when the length of the panel gets long, S gets extremely large (e.g. a binary model for 30 periods, $S = 2^{30} = 1$ billion). There do exist other smooth simulators for the panel probit model (e.g. the powerful GHK simulator). However, these alternative simulators put positive probability on *every* choice sequence. As such, the generic moment (1.1) has just too many non-zero elements to ever calculate (see Keane (1994)). Our smooth simulator does not have this problem – a maximum of $NS + 1$ elements of the moment are non-zero, for all θ.

3.3. Game Theoretic Models

This section presents a two firm version of the model Berry (1994). Consider a market with two firms who are simultaneously deciding whether to enter. Profits of firm i conditional on entering are given by

$$\pi_i = X_i\beta - \delta \ln(N + 1) + \varepsilon_i$$

where X_i are some firm specific variables and N is the total number of firms in the market (=0,1, or 2 in this case). One reason profits might depend on N is through oligopolistic intereration between the firms, e.g. a Cournot model. We allow arbitrary correlation between the unobservables ϵ_1 and ϵ_2.

This is a tough model to estimate because of the possiblity of multiple equilibrium. For any parameter vector θ, there are regions of ϵ space where either firm 1 would find it profitably to enter seperately *or* firm 2 would find it profitable to enter seperately, but it is not profitable for both firms to enter. What this means in the context of our model is that there is no *function* mapping $(X_1, X_2, \epsilon_1, \epsilon_2, \theta)$ into a exact market structure. This renders likelihood functions and moments of the exact market structure not well-defined - creating serious problems for likelihood or method of moment estimation. A popular approach to such multiple equilibrium models (Bresnahan and Reiss (1987)) is to look at functions of the exact market structure that are unique across the multiple equilibria. Berry shows that in his model there is a function f mapping $(X_1, X_2, \epsilon_1, \epsilon_2, \theta)$ into the total number of firms in the market,

$$y = \begin{pmatrix} I(\text{no firms enter}) \\ I(\text{one firm enters}) \\ I(\text{both firms enter}) \end{pmatrix} = f(X_1, X_2, \epsilon_1, \epsilon_2, \theta)$$

Since this is a function, it can be used for moments based estimation.

The expectation of f is not generally analytic, so simulation is necessary. For expostition we focus on simulating the 2nd element of y. We can write this out explicitly as as:

$$y = I\left(\text{one firm enters}\right) = I\begin{pmatrix} (X_1\beta - \delta\ln 2 + \varepsilon_1 > 0 & \cap & X_2\beta - \delta\ln 3 + \varepsilon_2 < 0) \\ & \cup & \\ (X_1\beta - \delta\ln 3 + \varepsilon_1 < 0 & \cap & X_2\beta - \delta\ln 2 + \varepsilon_2 > 0) \end{pmatrix}$$

The straightforward simulator

$$\widehat{Ef}(\theta) = \frac{1}{NS}\sum_{ns} I\begin{pmatrix} (X_1\beta - \delta\ln 2 + \varepsilon_{1ns} > 0 & \cap & X_2\beta - \delta\ln 3 + \varepsilon_{2ns} < 0) \\ & \cup & \\ (X_1\beta - \delta\ln 3 + \varepsilon_{1ns} < 0 & \cap & X_2\beta - \delta\ln 2 + \varepsilon_{2ns} > 0) \end{pmatrix}$$

is again not continuous in θ.

A change of variables to $u_{1ns} = X_1\beta - \delta\ln 2 + \varepsilon_{1ns}$ or $u_{1ns} = X_1\beta - \delta\ln 3 + \varepsilon_{1ns}$ will not result in a smooth simulator (as it doesn't remove a δ from inside the indicator function). The neccesary change of variable for a smooth importance sampling simulator is:

$$\begin{pmatrix} u_{1ns} \\ \\ u_{2ns} \end{pmatrix} = \begin{pmatrix} \frac{X_1\beta-\delta\ln 2+\varepsilon_{1ns}}{\delta} \\ \\ \frac{X_2\beta-\delta\ln 2+\varepsilon_{2ns}}{\delta} \end{pmatrix} = \begin{pmatrix} \frac{X_1\beta+\varepsilon_{1ns}}{\delta} - \ln 2 \\ \\ \frac{X_2\beta+\varepsilon_{2ns}}{\delta} - \ln 2 \end{pmatrix}$$

resulting in

$$\begin{aligned} \widetilde{Ef}(\theta) &= \frac{1}{NS}\sum_{ns} \frac{I\begin{pmatrix} (\frac{u_{1ns}}{\delta} > 0 \quad \cap \quad \delta(u_{2ns}+\ln 2-\ln 3) < 0) \\ \cup \\ \delta(u_{1ns}+\ln 2-\ln 3) < 0) \cap \quad \frac{u_{2ns}}{\delta} > 0) \end{pmatrix} p(u_{ns} \mid x, \theta)}{g(u_{ns})} \\ &= \frac{1}{NS}\sum_{ns} \frac{I\begin{pmatrix} (u_{1ns} > 0 \quad \cap \quad (u_{2ns}+\ln 2-\ln 3) < 0) \\ \cup \\ (u_{1ns}+\ln 2-\ln 3) < 0) \quad \cap \quad u_{2ns} > 0) \end{pmatrix} p(u_{ns} \mid x, \theta)}{g(u_{ns})} \end{aligned}$$

given the assumption that δ is positive (that a firm's profits fall in the number of its competitors). This simulator *is* smooth in the parameter vector.

3.4. Stochastic Stopping Time

Consider the following model adapted from Ackerberg, Machado, and Riordan (1999). Patients enter a health care treatment program at time 0 with some initial health status $h_{i0} = X_i\beta_0 + \epsilon_0$. This health status evolves according to a Markov process such that health status at time t is

$$h_t = X_i\beta_t + \alpha_t h_{t-1} + \epsilon_t$$

If at any t health status reaches an upper limit $h_U = X_i\gamma_t$ the patient is deemed cured and is discharged at t. Similarly, if health status drops below a level $h_L = X_i\delta_t$ the patient drops out or its kicked out of the program due to failure or non-compliance. Lastly, we will allow for a probability that a patient leaves treatment for other exogenous, non-health related reasons - suppose that in period t, one drops out with probability $p(X_i\pi_t)$, i.e. the patient drops out if a uniform random variable μ_t is less than $p(X_i\pi_t)$. One might allow correlation in the ϵ's or μ's across time or allow the two processes to be correlated..

There are three possible outcomes in this model – success, failure, or exogenous dropout. These outcome can occur in any period from 1 to T. We can think of our y vector in this model as a 3*T vector of dummies indicating a particular outcome in a particular time period.

Straightforward simulation of Ey in this model would involve sequentially drawing ϵ's and μ's to simulate an outcome/time-period. We again focus on one particular element of y, e.g. success at period t.

$$\begin{aligned} Ey &= E[\text{success at period } t] \\ &= \frac{1}{NS}\sum_{ns}\left[\prod_{\tau=1}^{t-1} I\left[X_i\delta_\tau < X_i\beta_\tau + \alpha_\tau h_{\tau-1} + \epsilon_{\tau ns} < X_i\gamma_\tau\right] I\left[\mu_{\tau ns} < p(X_i\pi_t)\right]\right] I\left[X_i\beta_t + \alpha_t h_{t-1} + \epsilon_t \geq X_i\gamma_t\right] \end{aligned}$$

In words, the draws must be such that h_t is between the boundaries up to t, that the patient doesn't drop out before t, and that h_t crosses over the upper boundary exactly at t. This will be discontinous in the parameters $(\alpha, \beta, \gamma, \delta, \pi)$ for a number of reasons. For example, as π changes, particular $\mu_{ns\tau}$ draws will jump $I\left[X_i\pi_\tau + \mu_{\tau ns} < 0\right]$ from 1 to 0 or from 0 to 1. The indicators including the h's will also change discretely as parameters $\alpha, \beta, \gamma, \delta$ change.

To make this continuous, let

$$\begin{aligned} z_\tau &= \frac{X_i\beta_\tau + \alpha_\tau h_{\tau-1} + \epsilon_{\tau ns} - X_i\delta_\tau}{X_i\gamma_\tau - X_i\delta_\tau} \\ w_\tau &= \mu_{ns\tau} - \Phi^{-1}(p(X_i\pi_t)) \end{aligned}$$

where Φ^{-1} is an arbitrary inverse CDF. Now,

$$Ey = \int \left[\prod_{\tau=1}^{t-1} I\left[0 < z_\tau < 1\right] I\left[w_\tau > 0\right]\right] I\left[z_t \geq 1\right] p(z^T, w^T \mid x, \theta) dz^T dw^T$$

where $p(z^T, w^T \mid x, \theta)$ is the joint distribution generated by $p(\epsilon^T, \mu^T)$ and the definitions of z and w. Now multiply and divide by arbitrary PDF $g(z^T, w^T \mid x)$ to get

$$Ey = \int \frac{\left[\prod_{\tau=1}^{t-1} I\left[0 < z_\tau < 1\right] I\left[w_\tau > 0\right]\right] I\left[z_t \geq 1\right] p(z^T, w^T \mid x, \theta)}{g(z^T, w^T \mid x)} g(z^T, w^T \mid x)\, dz^T dw^T$$

and the smooth importance sampling simulator:

$$\widetilde{Ef}(\theta) = \frac{1}{NS}\sum_{ns} \frac{\left[\prod_{\tau=1}^{t-1} I\left[0 < z_{ns\tau} < 1\right] I\left[w_{ns\tau} > 0\right]\right] I\left[z_{nst} \geq 1\right] p(z_{ns}^T, w_{ns}^T \mid x, \theta)}{g(z_{ns}^T, w_{ns}^T \mid x)}$$

Note also that in drawing from $g(z^T, w^T \mid x)$ and in computing $g(z^T_{ns}, w^T_{ns} \mid x)$ and $p(z^T_{ns}, w^T_{ns} \mid x, \theta)$, it is easiest to divide these distributions into products of conditional distributions, i.e. if the μ_τ process is independent of the ϵ_τ process, we have:

$$p(z^T, w^T \mid x, \theta) = \prod_{t=1}^{T} p(z_t, w_t \mid z_{t-1}, w_{t-1}, x, \theta) = \prod_{t=1}^{T} p(z_t \mid z_{t-1}, x, \theta) p(w_t \mid w_{t-1}, x, \theta)$$

4. Importance Sampling to Reduce Computational Burden

We next turn to the situation where the function $f(x_i, \epsilon_i, \theta)$ is hard to compute. Examples include dynamic optimization problems by agents or complicated equilibrium problems. Both may require numerical methods to evaluate. If one has I observations, performs NS simulation draws, and optimization requires R function evaluations, estimation requires solving $f(x_i, \epsilon_i, \theta)$ $NS * I * R$ times. This can be prohibitively burdensome for realistic models one might like to estimate. This section shows how one can use importance sampling to significantly reduce this computation burden. One can reduce the number of times $f(x_i, \epsilon_i, \theta)$ needs to be evaluated from $NS * I * R$ times to $NS * I$ times or even NS times. Our procedure is again illustrated with examples. The first is a complicated discrete game, the second is a dynamic programming problem.

4.1. Discrete Games

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

$$c(s_j) = (\beta x_j + \alpha s_j + \epsilon_j) s_j$$

where x_j are firm specific cost observables and u_j are firm specific cost unobservables. Market inverse demand in market i is a function of the total number of stores $Q_i = \sum_j s_j$ and equal to

$$P(Q) = \delta_0 - \delta_1 Q + \delta_3 z_i$$

where z_i are market specific variables that shift overall demand and ϵ_i 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 $\delta_1 = 1$, i.e.[9]

[9] This normalization is different than that used by Davis (who normalized $\sigma_u = 1$), but is an identical model given that demand is downward sloping. This alternative normalization makes our expostition easier.

$$P(Q) = \delta_0 - Q + \delta_3 z_i$$

These imply an underlying profit function of the model

$$\begin{aligned}\pi(s_j, Q) &= p(Q)s_j - c(s_j) \\ &= (\delta_0 + \delta_3 z_i + \beta x_j + \epsilon_j)s_j + \alpha s_j^2 - Qs_j\end{aligned}$$

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

$$y = Q_i = f(x_1, ..., x_{N_i}, \epsilon_1, ..., \epsilon_{N_i}, z_i, \theta)$$

with the generic moment

$$E\left[y_i - E\left[f(x_1, ..., x_{N_i}, \epsilon_1, ..., \epsilon_{N_i}, z_i, \theta)|x_i, z_i\right] \quad | \quad x_i, z_i\right]$$

In this case, not only is the expectation of f not analytic, but the function f itself is very complicated. Given all primitives $(x_1, ..., x_{N_i}, \epsilon_1, ..., \epsilon_{N_i}, z_i, \theta)$, an interative tatonnment procedure is required to solve for Q_i. The pure frequency simulator that Davis uses:

$$\begin{aligned}\widehat{Ef}(\theta) &= \frac{1}{NS}\sum_{ns} f(x_1, ..., x_{N_i}, \epsilon_{1ns}, ..., \epsilon_{N_i ns}, z_i, \theta) \\ &= \frac{1}{NS}\sum_{ns} f(\{\delta_0 + \delta_3 z_i + \beta x_j + \epsilon_{jns}\}_{j=1}^{N_i}, \alpha)\end{aligned} \tag{4.1}$$

requires computation of f $NS * I * R$ times, where I is the number of observations (markets), and R is the number of function evaluations neccesary to minimize the moment (R might be on the order of 1000 if there are 10 parameters to estimate). Like the previous examples, $\widehat{Ef}(\theta)$ also will have flats and jumps in θ.

Note the equality in the second line of equation (4.1). Equilibrium in this model is a function of just $\{\delta_0 + \delta_3 z_i + \beta x_j + \epsilon_{jns}\}_{j=1}^{N_i}$ and α, not the individual components. This follows from the profit function. As such, consider the change of variables

$$u_{jns} = \delta_0 + \delta_3 z_i + \beta x_j + \epsilon_{jns}$$

and the importance sampling simulator

$$\widetilde{Ef}(\theta) = \frac{1}{NS} \sum_{ns} \frac{f(\{u_{jns}\}_{j=1}^{N_i}, \alpha) p(u_{ns} \mid x, \theta)}{g(u_{ns})}$$

where the u_{ns} are draws from the some distribution $g(u_{ns})$ (again, $p(u_{ns} \mid x, \theta)$ at some initial guess of θ is a good candidate). For the moment ignore the parameter α. As the other parameters change, the importance sampling holds the $\{u_{jns}\}_{j=1}^{N_i}$ constant, and thus the function f need not be recomputed for each parameter vector. As a result, f only need be computed $NS * I$ times rather than $NS * I * R$ times. Note that this importance sampling also smooths the function.

The caveat here is the parameter α. Unfortunately, when α changes, the equilibrium does need to be resolved. This is not an issue, e.g. if one is willing to assume constant marginal costs (i.e. $\alpha = 0$), but there are a couple of other alternatives. First is to do an outside search algorithm over α and an inside search algorithm over the rest of the parameters. Equilibria need to be re-solved only when α changes, which will generally be about 40 times since it is a one dimensional search.

The second, perhaps more interesting, alternative is to slightly expand the model. Suppose we allow some heterogeneity across firms in their returns to scale, i.e.

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

where α is the average scale parameter and η_j is firm j's deviation from that mean (one might also allow ϵ and η be correlated). Now straightforward simulation requires drawing both a set of ϵ_{ns}'s and a set of η_{ns}'s . Consider the changes of variables

$$\begin{aligned} u_{jns} &= \delta_0 + \delta_3 z_i + \beta x_j + \epsilon_{jns} \\ z_{jns} &= \alpha + \eta_{jns} \end{aligned}$$

and the simulator

$$\widetilde{Ef}(\theta) = \frac{1}{NS}\sum_{ns}\frac{f(\{u_{jns}\}_{j=1}^{N_i}, \{\alpha_{jns}\}_{j=1}^{N_i})p(u_{ns}, \alpha_{ns} \mid x, \theta)}{g(u_{ns}, \alpha_{ns})}$$

This simulator is both smooth in all parameters and the equilibria do not need to be recomputed as the parameters change[10]. The intuition here is similar to that in the smoothing case. We start with a bunch of simulated equilibrium outcomes, then when we change the parameter vector, we dont change these simulated outcomes, but we do change the weight that each outcome gets.

Lastly, note that one can reduce computational burden even further by using the same $g(.)$ function (and same simulation draws) for different observations. In other words, we use the same $\{u_{jns}\}_{j=1}^{N_i}, \{\alpha_{jns}\}_{j=1}^{N_i}$ draws for each observation. In this case, one only needs to solve the f function NS times. Since the x's vary across observations, note that one still needs to compute $p(u_{ns}, \alpha_{ns} \mid x, \theta)$ seperately for each observation.

There are a few caveats to this additional procedure. First, because firms differ in x, there is no obvious choice of g. One alternative would be to use the p function (at some initial θ) with the means of x. Another alternative would be to use I different g functions, one for each observation's x. Secondly, note that the supports of u and α need to be the same across observations to do this. Third, this procedure creates correlation in the simulation error across observations. This means it can take longer for simulation error to average out as the number of observations increases. This correlation also destroys the nice $(1/NS)$ result regarding additional variance due to simulation. Of course, if one is able to increase the number of simulation draws because of the computational time savings, this might be compensated for.

4.2. A Dynamic Programming Problem

Consider a dynamic model of automobile choice. Suppose that in a given year the utility consumer i obtains from using a car 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. Therefore, the single period utility from purchasing or not purchasing, respectively

[10] Issue with σ_α needing to be bounded away from 0. Note - how to do this if want to restrict α_i to be positive, or restricted between 0 and 1.

are

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

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[11]. 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, α_i, and γ_i as linear functions of consumer characteristics y_i plus unobservable terms, i.e.

$$
\begin{aligned}
\beta_{i1} &= y_i \beta_1 + \epsilon_{i1} \\
& \quad . \\
& \quad . \\
\beta_{iK} &= y_i \beta_K + \epsilon_{iK} \\
\alpha_i &= y_i \alpha + \epsilon_{iK+1} \\
\gamma_i &= y_i \gamma + \epsilon_{iK+2}
\end{aligned}
$$

specifying 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.

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

[11] 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).

where $\widehat{EP}(\theta)$ is the average of the simulated choices (policies)[12],

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

and P_i is the observed choice.

The problem with the above straightforward simulation is that as θ changes (while the simulated ϵ's are held constant), the simulated $(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins})$'s change. Thus, the dynamic programming problem needs to be solved $NS * I * R$ times – once for each simulation draw for each observation for every parameter vector evaluated. Again importance sampling can help reduce computational burden. Consider changes of variables given by:

$$\begin{aligned}
\beta_{i1} &= y_i\beta_1 + \epsilon_{i1} \\
& . \\
& . \\
\beta_{iK} &= y_i\beta_K + \epsilon_{iK} \\
\alpha_i &= y_i\alpha + \epsilon_{iK+1} \\
\gamma_i &= y_i\gamma + \epsilon_{iK+2}
\end{aligned}$$

and the importance sampling simulator

$$\widetilde{EP}(\theta) = \frac{1}{NS}\sum_{ns} \frac{P(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins}, c_i, a_{c_i})p(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins} \mid y_i, \theta)}{g(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins})}$$

where $(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma_{ins})$ are draws from $g()$. Now when the parameters θ change, the vector $(\beta_{i1ns}, ..., \beta_{iKns}, \alpha_{ins}, \gamma$ does not change. As such, the dynamic programming problem $V_i(c_i, a_{c_i})$ only needs to be computed $NS * I$ times – once for each simulation draw for each individual. This can be a big difference when the number of parameters is large and the number of necessary function evaluations R is large. Again the intuition is that instead of changing our simulated individuals when we change θ, we change the weights we put on these simulated individuals. As with the previous model, one could reduce the number of computations to NS times by using the same simulation draws for each individual.

[12] Perhaps a vector of 0-1 choices (i.e. which car is bought).

4.2.1. Comparison to Alternative Approaches

Lastly, note that an alternative strategy for this problem 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 this function (and the associated policy function), one would only need to solve it once - 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 the V and P. 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 $NS*I$ or NS times above. The discretation also adds error to the problem and likely destroys econometric consistency.

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 points at which to approximate the value function. Rust proves that such randomization breaks 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[13].

After using such an approach to approximate $V(\beta_{i1},...,\beta_{iK},\alpha_i,\gamma_i,c,a_c)$ and $P(\beta_{i1},...,\beta_{iK},\alpha_i,\gamma_i,c,a_c)$, 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 would need additional interpolation or approximation to compute $V(\beta_{i1ns},...,\beta_{iKns},\alpha_{ins},\gamma_{ins},c_i,a_{c_i})$.

Our methodology is related to Rust's 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[14]. While there is only one source

[13] Does this solution to the curse of dimensionality hold even though $(\beta_{i1},...,\beta_{iK},\alpha_i,\gamma_i)$ do not change over time? This implies densities in Rust are degenerate and we end up with a computational problem that is more like multivariate function approximation.

[14] This relies on c_i and a_c being in discrete space. If they were not, we would still expect considerably less approximation error

of simulation error in our estimates (that in the estimation process), the Rust method has two (the estimation process and the value function approximation).

While the Rust methodology solves the curse of dimensionality by brute force (directly going at the value function) our methodology implicitly breaks the curse of dimensionality problem. 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[15].

5. Additional Points and Caveats

Monte-Carlo Experiments

Application to complicated auction models

Use in ML procedures or Indirect Estimation.

Problems with discrete distributions/distributions where support changes with θ.

Necessity to bound parameter space, e.g. variance of unobserved heterogeneity being bounded away from 0.

Having multiple unobservables in the same function.

6. Conclusion

References

[1] Ackerberg, D., Machado, M.and Riordan, M. 1999 “” mimeo, Boston University

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

[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 “” mimeo, MIT

in our procedure.

[15] Note that our methodology breaks the curse of dimensionality only in the dimension of the heterogeneity $(\beta_{i1ns},...,\beta_{iKns},\alpha_{ins},\gamma_{ins})$, not in the size of the “true” state space (c_i,a).

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

[20] 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.

[21] 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.*

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

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

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

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

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

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

[28] 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.