A *Gauss* program for the estimation of discrete choice dynamic programming models using a Nested Pseudo Likelihood algorithm

Victor Aguirregabiria
Boston University

September 2001

Abstract

This paper describes a *Gauss* program for the estimation of discrete choice dynamic programming models using the *Nested Pseudo Likelihood algorithm* in Aguirregabiria and Mira (2002). It discusses several econometric and computational issues in the implementation of the algorithm, and illustrates the use of the program with several examples using both actual and simulated data.

Keywords: discrete choice, dynamic programming, estimation, simulation.

1 Introduction

The estimation of discrete choice dynamic programming models has been one of the most active research areas in micro-econometrics during the last two decades.[1] A major contribution in Rust (1987) was to introduce the *Nested Fixed Point algorithm (NFXP)* to obtain maximum likelihood estimates of structural parameters in these models. Hotz and Miller (1993) proposed a simpler estimator than NFXP for this class of problems. Hotz-Miller estimator (or *Conditional Choice Probabilities,* CCP, estimator) is a two-stage procedure that provides consistent and asymptotically normal estimates of structural parameters without having to fully solve the dynamic programming problem. Previous conventional wisdom was that the CCP estimator

[1] For surveys of this literature see Eckstein and Wolpin (1989), Pakes (1994), Rust (1994a, 1994b) Wolpin (1996), and Miller (1997).

achieved a significant computational gain at the expense of efficiency, both in finite samples and asymptotically.

In Aguirregabiria and Mira (2002) we propose an estimation procedure, the *Nested Pseudo Likelihood (NPL)* algorithm, which bridges the gap between these two previous estimation strategies. When the NPL algorithm is initialized with consistent (non-parametric or semi-parametric) estimates of conditional choice probabilities, successive iterations return a sequence of estimators of the structural parameters which we call *$K-$stage Policy Iteration estimators (PIE).* This family includes as extreme cases a particular CCP estimator (for $K = 1$) and Rust's NFXP estimator (in the limit as the algorithm converges). Contrary to previous conventional wisdom, we show that the asymptotic distribution of all the estimators in the sequence is the same and equal to that of the maximum likelihood estimator. When the algorithm is initialized with arbitrary conditional choice probabilities (i.e., not necessarily consistent estimates) it also returns upon convergence the NFXP estimator. In a numerical example based on Rust's bus replacement model, we found that when using arbitrary initial probabilities NPL produces maximum likelihood estimates 5 to 15 times faster than NFXP. When the algorithm is initialized with nonparametric estimates of choice probabilities, the computational gains relative to NFXP are much larger. For this particular application, we find that the gains in terms of finite sample properties of using the 2-stage PI estimator instead of the 1-stage estimator (Hotz-Miller) are very significant, but additional gains by using the ML estimator instead of 2-stage PI are small.

This paper describes in more detail several computational and econometric issues associated with the NPL algorithm, as well as a *Gauss* program that implements this algorithm for a class of multinomial dynamic programming models.[2] Although NPL is an algorithm for a general class of dynamic programing models, this paper concentrates in the particular subclass of models for which the code has been written. This subclass is a dynamic programming version of McFadden conditional logit model (see McFadden, 1984, pp. 1411-1413).

The rest of the paper is organized as follows. Section 2 presents the model and

[2] The code, and future upgrades, can be found at http://people.bu.edu/vaguirre/programs/npl.html.

the different estimation methods, i.e., NFXP, CCP and NPL. Section 3 describes the *GAUSS* code and possible modifications and extensions of this code. Finally, I present two applications with actual data in section 4.

2 Model and estimation methods

2.1 Econometric model

Time is discrete and indexed by t. At each period t an agent observes the vector of state variables s_t and he chooses an action $a_t \in A = \{1, 2, ..., J\}$ in order to maximize the expected sum of current and future payoffs discounted by $\beta \in (0, 1)$:

$$E\left[\sum_{j=0}^{\infty} \beta^j \; u(a_{t+j}, s_{t+j}) \mid a_t, s_t\right] \tag{1}$$

where $u(a_t, s_t)$ represents payoff or utility at period t. From the point of view of the observing researcher there are two types of state variables, $s_t = (x_t, \varepsilon_t)$. The vector x_t is observable to the econometrician, and it has a discrete and finite support, $x_t \in X = \{x^1, ..., x^M\}$.[3] The vector of state variables ε_t is unobservable to the researcher, it has J components, $\{\varepsilon_{at} : a \in A\}$, and each of this components is a continuous random variable, with support the real line, and with continuous and twice differentiable distribution function.

The payoff function $u(a_t, x_t, \varepsilon_t)$ is additively separable in the observable and unobservable components, and multiplicatively separable in x_t and the structural parameters in preferences. More specifically,

$$u(a, x_t, \varepsilon_t) \quad = \quad z_a(x_t) \; \alpha + \varepsilon_{at} \tag{2}$$

where α is a $K \times 1$ vector of structural parameters, and $z_a(x_t)$ is a $1 \times K$ vector of functions of x_t. The unobservables $\{\varepsilon_{at} : a \in A\}$ are independently and identically distributed over time and over choice alternatives with a Extreme value distribution with mean zero and dispersion σ. Therefore, the specification of current utility is the one in McFadden's Conditional Logit model (see McFadden, 1984). The model in this paper is a dynamic programming version of McFadden's Conditional Logit.

[3] The vector of state variables x_t may contain lagged values of the decision variable, e.g., a_{t-1}.

When the agent makes his decision at period t he has uncertainty about future values of x and ε. His beliefs about uncertain future states can be represented by a Markov transition probability $p(x_{t+1}, \varepsilon_{t+1}|x_t, \varepsilon_t, a_t)$, which factors as (i.e., conditional independence assumption)[4]

$$p(x_{t+1}, \varepsilon_{t+1} \mid x_t, \varepsilon_t, a_t) = g_\sigma(\varepsilon_{t+1})\ f_\delta(x_{t+1}|x_t, a_t) \tag{3}$$

where g_σ is the density of ε_t, and f_δ is the conditional choice transition probability of x_t, that depends on the vector of parameters δ.

Given these assumptions, the model is a stationary Markov decision problem with state variables x_t and ε_t. Let $\theta \equiv \{\alpha, \sigma, \delta, \beta\}$ be the vector with the structural parameters of the model, and let $v_\theta(x_t, \varepsilon_t)$ be the value function. By Bellman's principle of optimality, v_θ is the unique fixed point of the following contraction mapping,

$$v_\theta(x_t, \varepsilon_t) = \max_{a\in A}\left\{ z_a(x_t)\ \alpha + \varepsilon_{at} + \beta \sum_{x_{t+1}}\ f_\delta(x_{t+1}|x_t, a)\ \int v_\theta(x_{t+1}, \varepsilon_{t+1})\ g_\sigma(d\varepsilon_{t+1}) \right\} \tag{4}$$

And the optimal decision rule $a_\theta^*(x_t, \varepsilon_t)$ can represented as the *arg max in a* of the term in brackets in previous equation.

For the description of the econometric model it is convenient to define versions of $v_\theta(x_t, \varepsilon_t)$ and $a_\theta^*(x_t, \varepsilon_t)$ which are integrated over the unobservables ε_t. Define the *integrated* value function $V_\theta(x_t) \equiv \int v_\theta(x_t, \varepsilon_t) g_\sigma(d\varepsilon_t)$. Taking into account this definition and the previous Bellman equation, it is clear that:

$$V_\theta(x_t) = \int \max_{a\in A}\left\{ z_a(x_t)\ \alpha + \varepsilon_{at} + \beta \sum_{x_{t+1}}\ f_\delta(x_{t+1}|x_t, a)\ V_\theta(x_{t+1}) \right\}\ g_\sigma(d\varepsilon_t), \tag{5}$$

The right-hand side of this equation is a contraction mapping in the integrated value function, and therefore V_θ is the unique fixed point of this mapping. Define also the integrated optimal decision rules, or *conditional choice probabilities (CCPs)*, $P_\theta^a(x_t) \equiv \int I\{a_\theta^*(x_t, \varepsilon_t) = a\} g_\sigma(d\varepsilon_t) = \Pr(a_t = a|x_t; \theta)$. These probabilities are the building blocks of any estimation procedure. Given that the unobservables are extreme value

[4] See Rust (1994a, 1994b) for discussions about the conditional independence assumption.

distributed, the CCPs have the following form:

$$P_\theta^a(x_t) = \frac{\exp\left\{ z_a(x_t)\dfrac{\alpha}{\sigma} + \dfrac{\beta}{\sigma}\sum_{x_{t+1}} f_\delta(x_{t+1}|x_t,a)\ V_\theta(x_{t+1})\right\}}{\sum_{j=1}^{J}\exp\left\{ z_j(x_t)\dfrac{\alpha}{\sigma} + \dfrac{\beta}{\sigma}\sum_{x_{t+1}} f_\delta(x_{t+1}|x_t,j)\ V_\theta(x_{t+1})\right\}} \tag{6}$$

Equations (5) and (6) describe the econometric model. For any vector of structural parameters θ the solution of Bellman equation (5) provides the vector of optimal values V_θ. Given these values it is possible to obtain choice probabilities using equation (6).

The NPL estimation algorithm exploits several properties of the *policy iteration operator* associated with the integrated Bellman equation (5). Let $P = \{P^a(x_t) : x_t \in X; a \in A\}$ be an arbitrary vector of conditional choice probabilities. The policy iteration operator is a fixed point mapping in the space of P, i.e., it maps vectors of CCPs into vectors of CCPs.[5] Let $\Psi_\theta(P) \equiv \{\Psi_\theta^a(x_t;P) : x_t \in X; a \in A\}$ be this *policy iteration operator.* When unobservables are extreme value distributed this operator has the following closed form:

$$\Psi_\theta^a(x_t;P) \quad = \quad \frac{\exp\left\{\ \tilde{z}_a(x_t;P,\delta)\ \dfrac{\alpha}{\sigma} + \tilde{e}_a(x_t;P,\delta)\ \right\}}{\sum_{j=1}^{J}\exp\left\{\ \tilde{z}_j(x_t;P,\delta)\ \dfrac{\alpha}{\sigma} + \tilde{e}_j(x_t;P,\delta)\ \right\}} \tag{7}$$

$$\begin{aligned} \textit{where}: \qquad \tilde{z}_a(x_t;P,\delta) &= z_a(x_t) + \beta\sum_{x_{t+1}} f_\delta(x_{t+1}|x_t,a)\ W_z(x_{t+1};P,\delta); \\ \tilde{e}_a(x_t;P,\delta) &= \beta\sum_{x_{t+1}} f_\delta(x_{t+1}|x_t,a)\ W_e(x_{t+1};P,\delta) \end{aligned}$$

$W_z(P,\delta) \equiv \{W_z(x;P,\delta) : x \in X\}$ is a $M \times K$ matrix, and $W_e(P,\delta) \equiv \{W_e(x;P,\delta) : x \in X\}$ is a $M \times 1$ vector with the following definitions:

$$\begin{aligned} W_z(P,\delta) &= \left[I - \beta\ \textstyle\sum_{a=1}^{J} P^a * F_\delta^a\right]^{-1}\left[\textstyle\sum_{a=1}^{J} P^a * Z_a\right]; \\ W_e(P,\delta) &= \left[I - \beta\ \textstyle\sum_{a=1}^{J} P^a * F_\delta^a\right]^{-1}\left[\textstyle\sum_{a=1}^{J} P^a * (Euler - \ln(P^a))\right]; \end{aligned} \tag{8}$$

[5] In general, a policy iteration operator is a fixed point mapping in the space of decision rules. In our case, conditional choice probabilities can be interpreted as the decision rules associated with the integrated Bellman equation (see Aguirregabiria and Mira, 2002).

where $*$ is the element-by-element (or Hadamard) product; $\{F_\delta^a : a \in A\}$ are the $M \times M$ matrices of transition probabilities with $f_\delta(x^{m'}|x^m, a)$ at position (m, m'); and $\{Z_a : a \in A\}$ are the $M \times K$ matrices with $z_a(x^m)$ at the $m-th$ row.

This mapping has a clear interpretation. First, $W_z(x; P, \delta)$ is the expected and discounted value of the stream of future $z's$ provided that the agent behaves according to choice probabilities in P. $W_e(x; P, \delta)$ has a similar interpretation: it is the expected and discounted value of the stream of future $\varepsilon's$ given that the agent behaves according to P. Therefore, $W_z(P, \delta) + W_e(P, \delta)$ is the vector of integrated values associated with following a decision rule with vector of probabilities P, i.e., it is a *policy valuation operator*. For the same reason,

$$\tilde{z}_{at}\ \alpha + \tilde{e}_{at} = z_a(x_t) + \sum_{x_{t+1}} f_\delta(x_{t+1}|x_t, a)\left[W_z(x_{t+1}; P, \delta) + W_e(x_{t+1}; P, \delta)\right]$$

is the value of choosing alternative a today given that in the future decisions will be made according to P. Given the conditional choice values $\{\tilde{z}_{at}\alpha + \tilde{e}_{at} : a \in A\}$, the optimal current decision (i.e., *policy improvement operator*) is to choose the alternative with the maximum value $\tilde{z}_{at}\alpha + \tilde{e}_{at}$. The choice probabilities associated with this decision rule are $\exp\{\tilde{z}_{at}\alpha + \tilde{e}_{at}\}\left[\sum_{j=1}^{J}\exp\{\tilde{z}_{jt}\alpha + \tilde{e}_{jt}\}\right]^{-1}$. Therefore, the mapping Ψ_θ is the composite of a policy valuation operator and a policy improvement operator. The following are some important properties of the mapping Ψ_θ.

PROPOSITIONS 1-2 (Aguirregabiria and Mira, 2002): Given additive separability of $u(a, x, \varepsilon)$ in x and ε, and the conditional independence assumption, we have that for any vector θ: (a) Ψ_θ is a contraction mapping; (b) P_θ is the unique fixed point of this mapping; (c) Ψ_θ is the policy iteration operator (or Newton operator) associated with Bellman equation (5); and (d) the Jacobian matrix $\partial\Psi_\theta/\partial P$ is zero at the fixed point P_θ.

The main computational cost in the evaluation of Ψ_θ comes from the computation of $W_z(P, \delta)$ and $W_e(P, \delta)$, and more specifically from the inversion of matrix $I - \beta \sum_{a=1}^{J} P^a * F_\delta^a$. The computational gains associated with NPL algorithm and Policy Iteration estimators result from avoiding repeated computation of $W_z(P, \delta)$ and $W_e(P, \delta)$ during the evaluation of the (pseudo) likelihood function.

2.2 Estimation

Consider that the data set consists of a panel of individuals with information on their actions and observable state variables at different periods of time: $\{x_{it}, a_{it} : i = 1, 2, ..., N; t = 1, 2, ..., T\}$. We are interested in the estimation of the vector of structural parameters θ. Consistent estimates of δ can be obtained maximizing the partial likelihood $\sum_{i=1}^{N}\sum_{t=1}^{T-1} \ln f_{\delta}(x_{i,t+1}|x_{it}, a_{it})$. This estimation of δ does not require one to solve the Markov decision model. I concentrate here in the estimation of the rest of the structural parameters taking as given a consistent estimate of δ. Furthermore, I adopt the normalization $\sigma = 1$, and take as known the discount factor β. Therefore, the vector of parameters to estimate is α. For notational simplicity I omit δ as an argument in P and Ψ.

The (partial) log-likelihood function for this model and data is,

$$l(\alpha) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{J} I\{a_{it} = j\} \ \ln P_{\alpha}^{j}(x_{it}) \tag{9}$$

where, for any vector α, $P_{\alpha} = \{P_{\alpha}^{j}(x) : j \in A; x \in X\}$ is the unique fixed point of the mapping Ψ_{α}. For an arbitrary vector of probabilities P, define the *pseudo log-likelihood function*,

$$\tilde{l}(\alpha; P) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{J} I\{a_{it} = j\} \ \ln \Psi_{\alpha}^{j}(x_t; P) \tag{10}$$

2.2.1 Nested pseudo-likelihood (NPL) algorithm

Start with an initial guess for the conditional choice probabilities, $P^0 \in [0, 1]^{MJ}$. At iteration $K \geq 1$, apply the following steps:

> *Step 1:* Obtain the matrix $W_z(P^{K-1})$ and the vector $W_e(P^{K-1})$. For each sample value of x_{it} obtain $\tilde{z}_{a,it}^{K-1} = z_a(x_{it}) + \beta \sum_{x'} f_{\delta}(x'|x_{it}, a) W_z(x'; P^{K-1})$, and $\tilde{e}_{a,it}^{K-1} = \beta \sum_{x'} f_{\delta}(x'|x_{it}, a) W_e(x'; P^{K-1})$.
>
> *Step 2:* Given $\{\tilde{z}_{a,it}^{K-1}\}$ and $\{\tilde{e}_{a,it}^{K-1}\}$ obtain a pseudo maximum likelihood estimate of α, by maximizing the *pseudo log-likelihood function* $\tilde{l}(\alpha; P^{K-1})$. That is, estimate a conditional logit model with explanatory variables

$\{\tilde{z}_{a,it}^{K-1}\}$ and $\{\tilde{e}_{a,it}^{K-1}\}$, where the parameter associated with $\{\tilde{e}_{a,it}^{K-1}\}$ is restricted to be one. Call α^K to this estimator.

Step 3: Update P using the estimated probabilities in Step 2: $P^K = \{P^{a,K}(x) : a \in A; x \in X\}$, where

$$P^{a,K}(x) \quad = \quad \frac{\exp\left\{ \ \tilde{z}_a(x; P^{K-1}) \ \alpha^K + \tilde{e}_a(x; P^{K-1}) \ \right\}}{\sum_{j=1}^{J} \exp\{ \ \tilde{z}_j(x_t; P^{K-1}) \ \alpha^K + \tilde{e}_j(x_t; P^{K-1}) \ \}}$$

Iterate in K until convergence in P (or α) is reached.

Notice that at Step 2 the values $\{\tilde{z}_{a,it}^{K-1}\}$ and $\{\tilde{e}_{a,it}^{K-1}\}$ are given, and therefore the evaluation of the pseudo log-likelihood function (and of its gradient and Hessian) does not require repeated computation of W_z and W_e. Furthermore, the pseudo log-likelihood function is globally concave in α, which in general it is not the case for the log-likelihood function.

PROPOSITION 3: (Aguirregabiria and Mira, 2002) If NPL converges it does so to a root of the likelihood equations.

2.2.2 Policy Iteration (PI) estimators

Given our sample and model, it is possible to identify nonparametrically the true (population) conditional choice probabilities $P_* = \{\Pr(a_{it} = a | x_{it} = x) : a \in A; x \in X\}$. Therefore, instead of using an arbitrary vector P^0, we can initialize the NPL algorithm using a consistent estimator of P_*. Performing one, two and in general K iterations of the NPL algorithm yields a sequence $\{\hat{\alpha}^1, \hat{\alpha}^2, ..., \hat{\alpha}^K\}$ of statistics which can be used as estimators of α. These are the *Policy Iteration (PI) estimators.*

PROPOSITION 4 (Aguirregabiria and Mira, 2002): Under the regularity conditions for the consistency and asymptotic normality of the maximum likelihood estimator, every PI estimator is root-n-consistent, asymptotically normal, and asymptotically equivalent to the (partial) maximum likelihood estimator.

The asymptotic distribution of all the PI estimators is the same and equal to the one of the maximum likelihood estimator. Therefore, as long as initial estimates of CCPs are consistent, the precision of these initial estimates does not affect the asymptotic precision of the PI estimators. However, it seems reasonable to believe that it will affect their finite sample properties.

In principle, we can use a nonparametric kernel estimator to obtain initial consistent estimates of CCPs. However, since the model is fully parametric, we know the distribution of the unobservables. That is, we know that the true choice probabilities have the following form:

$$P_*^a(x_{it}) \quad = \quad \frac{\exp\left\{g_a^*(x_{it})\right\}}{\sum_{j=1}^J \exp\left\{g_a^*(x_{it})\right\}}$$

where $g_a^*(x_{it}) = \tilde{z}_a^*(x_{it})\alpha^* + \tilde{e}_a^*(x_{it})$ are vectors of unknown functions of x_{it}. This is semiparametric model for a_{it} conditional on x_{it}, and it is possible to obtain consistent estimates of CCPs using a polynomial series semiparametric estimator. Define $P_h^a(x_{it}) \equiv \exp\left\{h(x_{it})'\gamma_a)\right\}\left[\sum_{j=1}^J \exp\left\{h(x_{it})'\gamma_a\right\}\right]^{-1}$, where $h(x_{it})$ is a vector of polynomial terms in x_{it} (i.e., linear terms, quadratic, etc), and γ_a is a vector of unknown parameters. This is a standard multinomial logit model with vector of explanatory variables $h(x_{it})$. Estimates of the parameters $\{\gamma_a\}$ can be obtained by maximum likelihood. Solving these estimates in $P_h^a(x_{it})$ provides a consistent semiparametric estimate of $P_*^a(x_{it})$.

3 Gauss program and procedures

The software package for the implementation of NPL and PI estimation consists of a main program KPIE.PRG and several procedures which are called by this program. I describe here the main program following the same order as in the code flow. A detailed description of the procedures can be found in the Appendix.

Step 1: User's description of decision variables:

1.1. Number of decision variables: In some applications the discrete decision variable a can be a discretized continuous variable, or the result of combining several discrete and/or discretized continuous variables. The program allows for this possibility.

1.2. Type of discretization: The user selects whether discretization of the decision variable(s) is needed or not. If discretization is needed, the user can choose between two types of discretizations: (1) a uniform grid in the space of the variable; or (2) a uniform grid in the space of the empirical probability distribution of the variable.

1.3. Number of cells in discretization

1.4. Minimum and maximum values in discretization: The user does not choose a minimum and maximum value in the space of the variable, but a minimum and maximum percentile (which can be 0 and 100, respectively). The empirical distribution of the variable will be used to obtain the minimum and maximum values in the space of the variable.

Step 2: User's description of state variables:

2.1. Number of state variables

2.2. Type of discretization (See point 1.2. above)

2.3. Number of cells in discretization

2.4. Minimum and maximum values in discretization (See point 1.4. above)

Step 3: Data:

3.1. Name and address of data and output files

3.2. Number of observations and individuals: If the data has a panel structure, the user should provide both the number of individuals and the total number of observations. The panel can be either balanced or unbalanced, but observations should be sorted first by individual and then over time.

Step 4: Reading data file

Step 5: Call to procedure for discretizations: DISCKPIE.SRC:

The program calls this procedure twice: one for the discretization of observable state variables, and the second to discretize decision variables. The procedure returns a matrix "*xval*" and a vector "*indobsx*". The matrix "*xval*" contains the values of the variables at each cell in the discretized space. The vector "*indobsx*" contains, for every observation in the sample, the label or indicator of the cell in which the observation lies. These matrix and vector (for state and decision variables) have all the sample information that will be used to estimate the model. If decision and/or state variables do not need to be discretized, *Step 6: Specification of utility function*

6.1. Matrix $Z = [Z_1, Z_2, ..., Z_J]$: The user defines the matrices $Z_a = \{z_a(x) : x \in X\}$ in his model. For doing so, he uses the matrix *"xval"* obtained from the discretization of state variables in Step 5.

6.2. Names of parameters: Define a vector with names of parameters in α.

6.3. Value of discount factor β

Step 7: Specification of transition probability function f_δ

The program allows for the following specification of the transition of the observable state variables. Let $x_{k,t}$ be the $k-th$ state variable in vector x_t:

$$x_{k,t+1} = \delta_{k0}(a_t) + \delta_{k1}(a_t)\ x_{k,t} + \omega_{k,t+1}$$

where $\{\delta_{k0}(a) : a \in A\}$ and $\{\delta_{k1}(a) : a \in A\}$ are parameters, and $\{\omega_{k,t+1}\}$ is a shock that is *iid* distributed with density $f_{\omega k}$. This structure encompasses many different specification used in applications.

Example 1: Lagged decision variable, $x_{k,t+1} = a_t$: $\delta_{k0}(a_t) = a_t$; $\delta_{k1}(a_t) = 1$ for every a_t; $\omega_{k,t+1} = 0$.

Example 2: Deterministic capital accumulation, $x_{k,t+1} = \delta_k x_{k,t}$: $\delta_{k0}(a_t) = 0$, for every a_t; $\delta_{k1}(a_t) = \delta_k$ for every a_t; and $\omega_{k,t+1} = 0$.

Example 3: Stochastic capital accumulation, $x_{k,t+1} = \delta_k\ x_{k,t} + \omega_{k,t+1}$: $\delta_{k0}(a_t) = 0$, for every a_t; and $\delta_{k1}(a_t) = \delta_k$ for every a_t.

Example 4: Replacement, $x_{k,t+1} = I(a_t = 0)x_{k,t} + \omega_{k,t+1}$: $\delta_{k0}(a_t) = 0$, for every a_t; and $\delta_{k1}(a_t) = I(a_t = 0)$.

The user can fix some or all the parameters δ_k and/or the density $f_{\omega k}$, or he can let the program to estimate these primitives from the data. If the transition probabilities in the user's model are not encompassed by this specification, he can provide his own transition matrices.

7.1. Parameters δ_k: If some parameters δ_k are fixed by the user, the should provide their values here.

7.2. Deterministic or stochastic transition: For every state variable, the user decides whether the transition is stochastic or deterministic (i.e., $\omega_{kt} = 0$).

7.4. Call to procedure for estimation of transition probabilities TRANPROB.SRC: This procedure returns the matrices of estimated conditional choice transition probabilities, $\{F^a_{\hat{\delta}} : a \in A\}$.

8. Initial choice probabilities:

Initial probabilities are estimated using the reduced form multinomial logit estimator described in section 2.2.2. The user can also provided its own

8.1. Vector of explanatory variables: The user defines the vector of explanatory variables for the reduced form multinomial logit, i.e., the vector $h(x_{it})$.

8.2. Call to procedure for estimation of initial probabilities MULTILOG.SRC.

9. Structural estimation:

9.1. Number of policy iterations for PI estimator: The parameter "*kstage*" determines the number of policy iterations to use. If the user wants the procedure to iterate until convergence (i.e., maximum likelihood estimator) he should fix $kstage \geq 5$.

9.2. Call to procedure for PI estimator: KPIE.SRC: This procedure returns the sequence of the first "*kstage*" PI estimators, their respective variance-covariance matrices, and their vectors of predicted conditional choice probabilities for every value x. It calls the procedure *CLOGIT.SRC* to obtain maximum likelihood estimates of a McFadden's conditional logit model. The matrix W_z and vector W_e are obtained solving a system of linear equations that uses a Crout (LU) decomposition of matrix $I - \beta \sum_{a=1}^{J} P^a * F^a_{\hat{\delta}}$.

4 Applications

4.1 Bus engine replacement model (Rust, 1987)

The first example comes from the bus engine replacement model in Rust (1987). The data consists of monthly observations on the odometer readings and maintenance records of 162 buses in the fleet of Madison Metropolitan Bus Company over the period December, 1974 to May, 1985. From this information, Rust constructs two variables: (1) a_{it} is the indicator of *engine replacement* (i.e., actual physical replacement or a major overhaul of the engine) for bus i at month t; and (2) x_{it} is the cumulative mileage of the bus since last engine replacement. See Rust (1987) for a description of the data.

The model assumes that when the maintenance manager decides whether to replace or not a bus engine, his objective is to minimize expected and discounted costs of maintenance and replacement.. Costs are assumed to be separable over buses and additively separable over time. Costs associated with bus i at period t are,

$$Costs_{it} = \begin{cases} mc(x_{it}) + \varepsilon_{0,it} & if \quad a_{it} = 1 \\ rc + \varepsilon_{1,it} & if \quad a_{it} = 2 \end{cases}$$

where $a_{it} = 1$ means no replacement, and $a_{it} = 2$ means replacement; $mc(x_{it})$ is the component of maintenance costs which is associated with cumulative mileage; rc represents average replacement cost; and $\varepsilon_{0,it}$ and $\varepsilon_{1,it}$ are components of maintenance and replacement costs, respectively, which are unobservable to the researcher. The transition rule for cumulative mileage has the following form:

$$x_{i,t+1} = \begin{cases} \delta_0 + \delta_1 \ x_{it} + \omega_{i,t+1} & if \quad a_{it} = 0 \\ \omega_{i,t+1} & if \quad a_{it} = 1 \end{cases}$$

$\delta_1 < 1$ implies that monthly mileage tends to decrease as cumulative mileage increases. I present estimates of two models, one linear maintenance costs and other with quadratic costs. For a quadratic specification of maintenance costs, $mc(x_{it}) = mc_1 x_{it} + mc_2 x_{it}^2$, the vector of parameters α is $(rc, mc_1, mc_2)'$ and the vectors of functions $\{z_a(x_{it})\}$ are $z_1(x_{it}) = (0, x_{it}, x_{it}^2)$ and $z_1(x_{it}) = (1, 0, 0)$.

Estimates of δ_0 and δ_1 are statistically different to 0 and 1, respectively: $\hat{\delta}_0 = 3564$ $miles$ ($s.e. = 36.77$), and $\hat{\delta}_1 = 0.9980$ ($s.e. = 0.0002$). I use these estimates to construct transition probabilities. I present here estimates of structural parameters using a discretization of the state variable with 400 cells and a uniform grid in the space of the variables. Results are very similar when using a discretization based on a uniform grid in the probability distribution of the state variable. I obtain PI estimates using two different initial estimates of conditional choice probabilities, P^0: (1) a constant probability, $P^0(a = 1|x) = (1/n) \sum_i \sum_t a_{it}$; and (2) P^0 from the estimation of a logit model where the explanatory variables are the terms of a cubic polynomial in x_{it}.

Table 1 shows that for this data and model the 1-stage PI estimator performs very well even when the initial P^0 is just a constant. This contrasts with the results from the Monte Carlo experiment in Aguirregabiria and Mira (2002) using the same model but with simulated data. In that experiment we show that the 2-stage PI estimator performs very well in finite samples, but that the 1-stage PI estimator performs badly. However, for this particular data set, the 1-stage PI estimator is statistically and almost numerically equivalent to MLE. In all the case 2-stages is enough to obtain the MLE, and for quadratic adjustment costs and a cubic P^0 the 1-stage estimator is numerically equivalent to MLE.

Table 1
Bus engine replacement model

	Model 1 : $\mathbf{mc}(\mathbf{x}) = \mathbf{mc}_1\mathbf{x}$; $\beta = 0.99$					
	P^0 constant			P^0 cubic		
	1-stage	2-stage	MLE (2 stages)	1-stage	2-stage	MLE (2 stages)
rc	6.354	6.356	6.356	6.356	6.356	6.356
	(0.267)	(0.267)	(0.267)	(0.266)	(0.267)	(0.267)
mc_1	0.0112	0.0112	0.0112	0.0112	0.0112	0.0112
	(0.0012)	(0.0012)	(0.0012)	(0.0012)	(0.0012)	(0.0012)
	Model 2: $\mathbf{mc}(\mathbf{x}) = \mathbf{mc}_1\mathbf{x} + \mathbf{mc}_2\mathbf{x}^2$; $\beta = 0.99$					
	P^0 constant			P^0 cubic		
	1-stage	2-stage	MLE (2 stages)	1-stages	2-stages	MLE (1 stage)
rc	8.987	8.979	8.979	8.979	8.979	8.979
	(0.906)	(0.903)	(0.903)	(0.903)	(0.903)	(0.903)
mc_1	0.0451	0.0450	0.0450	0.0450	0.0450	0.0450
	(0.0097)	(0.0097)	(0.0097)	(0.0097)	(0.0097)	(0.0097)
mc_2	$-8.77*10^{-5}$	$-8.76*10^{-5}$	$-8.76*10^{-5}$	$-8.76*10^{-5}$	$-8.76*10^{-5}$	$-8.76*10^{-5}$
	$(2.39*10^{-5})$	$(2.39*10^{-5})$	$(2.39*10^{-5})$	$(2.39*10^{-5})$	$(2.39*10^{-5})$	$(2.39*10^{-5})$

Data: 8260 bus-month observations from 104 buses (models 1,2,3 and 4).
Number of cells = 400; Uniform discretization in the space of the state variable.

4.2 Labor demand model (Aguirregabiria and Alonso, 1999)

Appendix: Description of procedures

DISCKPIE.SRC: It discretizes a vector of variables according to a criterion selected by the user. Format:

$$\{xval, indx0\} = disckpie(xobs, dtype, minpct, maxpct, numx)$$

$$\begin{array}{llcl}
Inputs: & xobs & = & (nobs \times kvar)\ matrix\ with\ observations\ of\ variables. \\
 & dtype & = & (kvar \times 1)\ with\ discretization\ criteria. \\
 & & & \text{“}dtype[j] = 1\text{”} \rightarrow Variable\ \text{“}j\text{”}\ is\ discrete\ and\ not\ more \\
 & & & discretization\ is\ needed; \\
 & & & \text{“}dtype[j] = 2\text{”} \rightarrow Variable\ \text{”}j\text{”}\ will\ be\ discretized\ using \\
 & & & \ a\ uniform\ grid\ in\ the\ space\ of\ the\ variable; \\
 & & & \text{“}dtypea[j] = 3\text{”} \rightarrow Variable\ \text{“}j\text{”}\ will\ be\ discretized\ using \\
 & & & \ a\ uniform\ grid\ in\ the\ space\ of\ its\ probability\ distribution. \\
 & minpct & = & (kvar \times 1)\ vector\ with\ percentiles\ for\ the\ minimum\ values \\
 & & & \ in\ the\ discretized\ spaces. \\
 & maxpct & = & (kvar \times 1)\ vector\ with\ percentiles\ for\ the\ \max imum\ values \\
 & & & \ in\ the\ discretized\ spaces. \\
 & numx & = & (kvar \times 1)\ vector\ with\ number\ of\ cells\ in\ the\ discretized\ spaces. \\
 & & & \\
Outputs: & xval & = & (totnumx \times kvar)\ matrix\ with\ the\ discretized\ support\ of\ the \\
 & & & \ variables.\ Column\ \text{“}j\text{”}\ corresponds\ to\ variable\ \text{“}j\text{”}. \\
 & & & Rows\ are\ sorted\ by\ variables. \\
 & indx0 & = & (nobs \times 1)\ vector\ with\ indexes\ of\ the\ discretized\ observations. \\
 & & & E.g.,\ if\ discretized\ value\ of\ xobs[i]\ is\ xval[j]\ then\ indx0[i] = j.
\end{array}$$

PCTILES.SRC: It obtains percentiles of a random variable given a sample. Format:

$$pcy = pctiles(y, p)$$

$$\begin{array}{llcl}
Inputs: & y & = & (nobs \times 1)\ vector\ of\ observations. \\
 & p & = & (k \times 1)\ vector\ of\ probabilities\ (in\ \%). \\
 & & & \\
Outputs: & pcy & = & (k \times 1)\ vector\ of\ percentiles.
\end{array}$$

DISCTHRE.SRC: It discretizes a variable using a vector of thresholds provided by the user. Format:

$$discy = discthre(y, thre)$$

$$\begin{array}{llcl}
Inputs: & y & = & (nobs \times 1)\ vector\ of\ observations. \\
 & p & = & (k \times 1)\ vector\ of\ thresholds. \\
 & & & \\
Outputs: & discy & = & (nobs \times 1)\ vector\ with\ codes\ of\ the\ discretized \\
 & & & observations.\ Example: If\ y[i] \in (thre[5], thre[6]], \\
 & & & then\ discy[i] = 6.
\end{array}$$

TRANPROB.SRC: It estimates Markov transition probabilities $f(x_{t+1}|x_t, a_t)$ where x_t is a vector of K variables, a_t is discrete, and the form of the transition rule for variable x_{kt} is: $x_{k,t+1} = \delta_{k0}(a_t) + \delta_{k1}(a_t)\ x_{k,t} + \omega_{k,t+1}$, where $\{\delta_{k0}(a), \delta_{k1}(a) : a = 1, 2, ...J\}$ are parameters, and $\omega_{k,t+1}$ is an *iid* random shock. Format:

$$fmat = tranprob(indx, inda, id, xval, fixdel, vdel0, vdel1, vomega)$$

$$\begin{array}{llcl}
Inputs: & xobs & = & (nobs \times kvar)\ vector\ with\ observations\ of\ x. \\
 & inda & = & (nobs \times 1)\ vector\ with\ observations\ of\ a. \\
 & id & = & (nobs \times 1)\ vector\ with\ the\ IDs\ of\ individuals. \\
 & & & If\ the\ data\ is\ not\ a\ panel\ this\ parameter\ should\ be\ zero. \\
 & xval & = & (numx \times kvar)\ vector\ with\ discretized\ support\ of\ x. \\
 & fixdel & = & (kvar \times J)\ matrix\ of\ zeros\ and\ ones. \\
 & & & fixdel[k, j] = 0 \rightarrow \delta_k(j)\ will\ be\ estimated\ from\ the\ data; \\
 & & & fixdel[k, j] = 1 \rightarrow \delta_k(j)\ is\ provided\ by\ the\ user. \\
 & vdel0 & = & (kvar \times J)\ matrix\ where\ vdel0[k, j]\ is\ the\ value\ of\ \delta_{k0}(j) \\
 & & & provided\ by\ the\ user.\ If\ \delta_{k0}(j)\ will\ be\ estimated, \\
 & & & vdel0[k, j]\ can\ be\ arbitrary. \\
 & vdel1 & = & (kvar \times J)\ matrix\ where\ vdel0[k, j]\ is\ the\ value\ of\ \delta_{k0}(j) \\
 & & & provided\ by\ the\ user.\ If\ \delta_{k0}(j)\ will\ be\ estimated, \\
 & & & vdel0[k, j]\ can\ be\ arbitrary. \\
 & vomega & = & (kvar \times J)\ matrix\ of\ zeros\ and\ ones. \\
 & & & vomega[k, j] = 0 \rightarrow deterministic\ transition\ for\ x_{kt} \\
 & & & vomega[k, j] = 0 \rightarrow stochastic\ transition\ for\ x_{kt} \\
 & & & \\
Outputs: & fmat & = & (numx \times numx * nchoice)\ matrix\ with\ conditional\ choice \\
 & & & transition\ probabilities:\ \ fmat1{}^{\sim}fmat2{}^{\sim}...{}^{\sim}fmatJ.
\end{array}$$

KERNEL1.SRC: Kernel estimation of a univariate density function using a Gaussian kernel. The bandwidth is equal to Silverman's rule of thumb divided by $(\pi n)^{1/9}$. This choice intentionally generates under-smoothing. Format:

$$pest = kernel1(xobs, xpred)$$

$$\begin{array}{llll} Inputs: & xobs & = & (n \times 1)\ vector\ of\ observations. \\ & xpred & = & (k \times 1)\ vector\ of\ of\ values\ where\ the\ pdf. \\ & & & will\ be\ estimated. \\ \\ Outputs: & pest & = & (k \times 1)\ vector\ of\ estimates. \end{array}$$

MULTILOG.SRC: Estimation of a multinomial logit model by maximum likelihood. The optimization algorithm is a Newton's method withwith analytic expressions for the gradient and Hessian.. Format:

$$\{best, varest\} = multilog(yobs, xobs)$$

$$\begin{array}{llll} Inputs: & yobs & = & (nobs \times 1)\ with\ observations\ of\ dependent\ variable. \\ & & & I.e.,\ indicator\ of\ discrete\ choice \in \{1, 2, ..., J\}. \\ & xobs & = & (nobs \times k)\ matrix\ with\ observations\ of\ explanatory\ variables. \\ \\ Outputs: & best & = & (k * (J-1) \times 1)\ vector\ with\ parameter\ estimates. \\ & & & Normalization: \beta_1 = 0. \\ & varest & = & (k * (J-1) \times (k * (J-1))\ matrix\ with\ estimated\ covariance\ matrix. \end{array}$$

KPIE.SRC: It estimates structural parameters of a discrete choice dynamic programming model using the a K-stage Policy iteration estimator. Format:

$$\{tetaest, varest, pest\} = kpie(inda, indx, zmat, pini, bdisc, fmat, kstage, names)$$

$$\begin{array}{lll}
Inputs: & inda & = (nobs \times 1)\ vector\ with\ observations\ of\ discrete\ decision\ variable. \\
 & indx & = (nobs \times 1)\ vector\ with\ observations\ of\ the\ index\ of\ state\ vector\ x. \\
 & zmat & = (zmat1\tilde{}zmat2\tilde{}...\tilde{}zmatJ)\ matrix\ with\ values\ of\ the\ functions\ z_a(x). \\
 & pini & = (numx \times J)\ vector\ with\ initial\ reduced\ form\ estimate\ of \\
 & & \quad conditional\ choice\ probabilities\ Pr(a = j|x). \\
 & bdisc & = Discount\ factor\ (between\ 0\ and\ 1). \\
 & fmat & = (fmat1\tilde{}fmat2\tilde{}...\tilde{}fmatJ)\ matrix\ with\ conditional\ choice \\
 & & \quad transition\ probabilities. \\
 & kstage & = Number\ of\ \text{``}outer\text{''}\ policy\ iterations.If\ kstage \geq 5\ the\ procedure \\
 & & \quad iterates\ until\ convergence\ and\ returns\ the\ ML\ estimator. \\
 & names & = (K \times 1)\ vector\ with\ names\ of\ parameters. \\
 & & \\
Outputs: & tetaest & = (K \times kstage)\ matrix\ with\ estimates\ of\ structural\ parameters \\
 & & \quad for\ each\ of\ the\ k\ stages.. \\
 & varest & = (K \times K * kstage)\ matrix\ with\ asymptotic\ covariance\ matrices \\
 & & \quad for\ each\ of\ the\ k\ stages. \\
 & pest & = (numx \times kstage)\ matrix\ with\ estimated\ choice\ probabilities\ \Pr(a = j|x) \\
 & & \quad for\ each\ of\ the\ k\ stages.
\end{array}$$

CLOGIT.SRC: Maximum Likelihood estimation of McFadden's Conditional Logit. Some parameters can be restricted. Optimization algorithm: Newton's method with analytical gradient and Hessian.. Format:

$$\{best, varest\} = clogit(ydum, xobs, restx, namesb)$$

$$\begin{array}{lll}
Inputs: & ydum & = (nobs \times 1)\ vector\ with\ observations\ of\ dependet\ variable. \\
 & & \quad Categorical\ variable\ with\ values\ \{1, 2, ..., J\}. \\
 & xobs & = (nobs \times k * J)\ matrix\ with\ observations\ of\ explanatory\ variables \\
 & & \quad associated\ with\ unrestricted\ parameters. \\
 & & \quad First\ k\ columns\ correspond\ to\ alternative\ 1, and\ so\ on. \\
 & restx & = (nobs \times J)\ matrix\ with\ observations\ of\ explanatory\ variables \\
 & & \quad with\ parameters\ rerestricted\ to\ be\ 1\ (without\ loss\ of\ generality). \\
 & & \quad First\ column\ corresponds\ to\ alternative\ 1, and\ so\ on. \\
 & namesb & = (k \times 1)\ vector\ with\ names\ of\ parameters. \\
 & & \\
Outputs: & best & = (k \times 1)\ vector\ with\ parameter\ estimates. \\
 & varest & = (k \times k)\ matrix\ with\ estimated\ covariance\ matrix.
\end{array}$$

REFERENCES:

Aguirregabiria, V. (1999): "The dynamics of markups and inventories in retailing firms," *The Review of Economic Studies*, 66, 275-308.

Aguirregabiria, V., and C. Alonso-Borrego (1999): "Labor Contracts and Flexibility: Evidence from a Labor Market Reform in Spain," manuscript.

Aguirregabiria and Mira (2002): "Swapping the nested fixed point algorithm: A class of estimators for Markov decision models", *Econometrica*, forthcoming.

Eckstein, Z., and K. Wolpin (1989): "The specification and estimation of dynamic stochastic discrete choice models," *Journal of Human Resources,* 24, 562-598.

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

McFadden, D. (1981): "Econometric models of probabilistic choice", in C. Manski and D. McFadden (eds.), *Structural Analysis of Discrete Data with Econometric Applications.* MIT Press.

McFadden, D. (1984): "Econometric Analysis of Qualitative Response Models" in Grilliches, Z., and M. Intriligator (eds.) Handbook of Econometrics, Vol 2.

Miller, R. (1997): "Estimating Models of Dynamic Optimization with Microeconomic Data," in H. Pesaran and P. Smidth (eds.) *Handbook of Applied Econometrics: Microeconomics (Vol 2).* Blackwell.

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

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

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

Rust, J. (1994b): "Structural estimation of Markov decision processes," in R. E. Engle and McFadden (eds.) *Handbook of Econometrics Volume 4*, North-Holland.

Wolpin, K. (1996): "Public-Policy Uses of Discrete-Choice Dynamic Programming Models," *American Economic Review*, 86, 427-32.