CCPs and Sufficient Statistics: Hotz-Miller (1993)

C.Conlon - help from M. Shum and Paul Scott

Grad IO

Before we get started...

From NDP Notes: Policy Iteration (Howard 1960)

An alternative to value function iteration is policy function iteration.

- Make a guess for an initial policy, call it $a(x) = \arg\max_a U(a, x)$ that maps each state into an action
- Assume the guess is stationary compute the implied $V(a, x)$
- Improvement Step: improve on policy a_0 by solving

$$a' = \arg\max_a U(a, x) + \beta \sum_{x'} V(a, x') f(x'|x, a)$$

- Helpful to define $\tilde{f}(x'|x)$ as transition probability under optimal choice $a(x)$ post-decision transition rule.
- Determine if $\|a' - a\| < \epsilon$. If yes then we have found the optimal policy a^* otherwise we need to go back to step 2.

From NDP Notes: Policy Iteration (Howard 1960)

Policy Iteration is even easier if choices AND states are discrete.

- For Markov transition matrix $\sum_j f_{ij} = 1$,we want $\pi \, \mathbf{F} = \pi$
- $\lim_{t \to \infty} \mathbf{F}^t = \pi$ where the jth element of π represents the long run probability of state j.
- We want the eigenvalue for which $\lambda = 1$.

Now updating the value function is easy for kth iterate of PI

$$V^k(x) = Eu(a^k(x), x) + \beta \, \tilde{\mathbf{F}}^k \, V^k(x)$$
$$\Rightarrow V^k(x) = [1 - \beta \, \tilde{\mathbf{F}}^k]^{-1} \, Eu(a^k(x), x)$$

- Very fast when $\beta > 0.95$ and s is relatively small. (Rust says 500 more like 5000).
- Inverting a large matrix is tricky
- This trick is implicit in the HM/AM formulation.

Now for real

Motivation

- In Rust, we started with a guess of parameters θ, iterated on the Bellman operator to get $EV_\theta(x, j)$ and then constructed CCP's $Pr(a(x) = j|x, \theta) \equiv p(j|x, \theta)$.
- A disadvantage of Rust's approach is that it can be computationally intensive
  - With a richer state space, solving value function (inner fixed point) can take a very long time, which means estimation will take a very, very long time.
- Hotz and Miller's idea is to use observable data to form an estimate of (differences in) the value function from conditional choice probabilities (CCP's)
  - We observe $\hat{p}(j|x)$ directly in the data!
- The central challenge of dynamic estimation is computing continuation values. In Rust, they are computed by solving the dynamic problem. With Hotz-Miller (or the CCP approach more broadly), we "measure" continuation values using a function of CCP's.

Rust's Theorem 1: Values to CCP's

- In Rust (1987), CCPs can be derived from the value function:

$$p_j(x) = \frac{\partial}{\partial \pi_j(x)} W\left(\pi(x) + \beta E\left[V\left(x'\right)|x, j\right]\right)$$

where $W(u) = \int \max_j \{u_j + \varepsilon_j\} dG(\varepsilon)$ is the surplus function.

- For the logit case:

$$p_j(x) = \frac{\exp(v_j(x))}{\sum_{j' \in J} \exp\left(v_{j'}(x)\right)}$$

where the choice specific value function for action j in state x is

$$v_j(x) \equiv \pi_j(x) + \beta E\left[V\left(x'\right)|x, j\right]$$

HM's Proposition 1: CCP's to Values

- Notice that CCP's are unchanged by subtracting some constant from every conditional (choice-specific) value function. Thus, consider

$$D_{j,0}v\left(x\right) \equiv v_j\left(x\right) - v_0\left(x\right)$$

  where 0 denotes some reference action.

- Let $Q : \mathbb{R}^{|\mathbf{J}|-1} \rightarrow \Delta^{|\mathbf{J}|}$ be the mapping from the differences in conditional (choice-specific) values to CCP's.
- Note: we're taking for granted that the distribution of ε is identical across states, otherwise Q would be different for different x.

Hotz-Miller Inversion Theorem
Q is invertible.

HM inversion with logit errors

- Again, let's consider the case of where ε is i.i.d type I EV.
- Expression for CCP's:

$$p_j\left(x\right) = \frac{\exp\left(v_j\left(x\right)\right)}{\sum_{j' \in \mathbf{J}} \exp\left(v_j\left(x\right)\right)}.$$

- The HM inversion follows by taking logs and differencing across actions:

$$\ln p_j\left(x\right) - \ln p_0\left(x\right) = v_j\left(x\right) - v_0\left(x\right)$$

- Thus, in the logit case (this looks a lot like Berry (1994)):

$$Q_j^{-1}\left(\mathbf{p}\right) = \ln p_j - \ln p_0$$

- From now on, I will use $\phi(\mathbf{p})$ to denote $Q^{-1}(\mathbf{p})$.

Arcidiacono and Miller's Lemma

An equivalent result to the HM inversion was introduced by Arcidiacono and Miller (2011). It's worth introducing here because it makes everything from now on much simpler and more elegant.

Arcidiacono Miller Lemma: Statement
For any action-state pair (a, x), there exists a function ψ such that

$$V(x) = v_a(x) + \psi_a(\mathbf{p}(x))$$

Arcidiacono Miller Lemma: Proof

$$
\begin{aligned}
V(x) &= \int \max_j \{v_j(x) + \varepsilon_j\}\, dG(\varepsilon_j) \\
&= \int \max_j \{v_j(x) - v_a(x) + \varepsilon_j\}\, dG(\varepsilon_j) - v_a(x) \\
&= \int \max_j \{\phi_{ja}(\mathbf{p}(x)) + \varepsilon_j\}\, dG(\varepsilon_j) - v_a(x)
\end{aligned}
$$

Letting $\psi_a(\mathbf{p}(x)) = \int \max_j \{\phi_{ja}(\mathbf{p}(x)) + \varepsilon_j\}\, dG(\varepsilon_j)$ completes the proof

Important relationships

- The Hotz-Miller Inversion allows us to map from CCP's to differences in conditional (choice-specific) value functions:

$$\phi_{ja}\left(\mathbf{p}\left(x\right)\right) = v_j\left(x\right) - v_a\left(x\right)$$

- The Arcidiacono and Miller Lemma allows us to relate ex ante and conditional (choice specific) value functions:

$$V\left(x\right) = v_j\left(x\right) + \psi_j\left(\mathbf{p}\left(x\right)\right)$$

- For the logit case:

$$\phi_{ja}\left(\mathbf{p}\left(x\right)\right) = \ln\left(p_j\left(x\right)\right) - \ln\left(p_a\left(x\right)\right)$$
$$\psi_j\left(\mathbf{p}\left(x\right)\right) = -\ln\left(p_j\left(x\right)\right) + \gamma$$

where γ is Euler's gamma.

Estimation example: finite state space I

- Let's suppose that X is a finite state space. Furthermore, let's "normalize" the payoffs for a reference action $\pi_0(x) = 0$ for all x. (is this really a "normalization"?)
- Using vector notation (standard font, matrices bold) recall the definition of the choice-specific value function for the reference action:

$$v_0 = \underbrace{\pi_0}_{=0} + \beta\,\mathbf{F_0}\,V = \beta\mathbf{F_0}\,V$$

- Using the Arcidiacono-Miller Lemma:

$$\begin{aligned} V - \psi_0(p) &= \beta\,\mathbf{F_0}\,V \\ \Rightarrow V &= \left(\mathbf{I} - \beta\,\mathbf{F_0}\right)^{-1} \psi_0(p) \end{aligned}$$

Estimation example: finite state space II

- Now we have an expression for the ex ante value function that only depends on objects we can estimate in a first stage:

$$V = (\mathbf{I} - \beta\,\mathbf{F_0})^{-1}\,\psi_0\,(p)$$

- To estimate the utility function for the other actions,

$$\begin{aligned}
v_j &= \pi_j + \beta\,\mathbf{F_j}\,V \\
V - \psi_j\,(p) &= \pi_j + \beta\,\mathbf{F_j}\,V \\
\pi_j &= -\psi_j\,(p) + (\mathbf{I} - \beta\,\mathbf{F_j})\;V \\
\pi_j &= -\psi_j\,(p) + (\mathbf{I} - \beta\,\mathbf{F_j})\;(\mathbf{I} - \beta\,\mathbf{F_0})^{-1}\;\psi_0\,(p)
\end{aligned}$$

Identification of Models I

- If we run through the above argument with π_0 fixed to an arbitrary vector $\widetilde{\pi}_0$ rather than 0, we will arrive at the following:

$$\pi_j = \mathbf{A_j}\,\widetilde{\pi}_0 + b_j$$

  where A_j and b_j depend only on things we can estimate in a first stage:

$$\mathbf{A_j} = (1 - \beta\,\mathbf{F_j})\,(1 - \beta\,\mathbf{F_0})^{-1}$$
$$b_j = \mathbf{A_j}\,\psi_0\,(p) - \psi_j\,(p)$$

- We can plug in any value for $\widetilde{\pi}_0$, and each value will lead to a different utility function (different values for π_j). Each of those utility functions will be perfectly consistent with the CCP's we observe.

Identification of Models II

Another way to see that the utility function is under-identified:

- If there are $|X|$ states and $|J|$ actions, the utility function has $|X|\,|J|$ parameters.
- There are only $|X|\,(|J|-1)$ linearly independent choice probabilities in the data, so we have to restrict the utility function for identification.
- Magnac and Thesmar (2002) make this point as part of their broader characterization of identification of DDC models:
    - Specify a vector of utilities for the reference action $\widetilde{\pi}$, a distribution for the idiosycratic shocks G, and a discount factor, and we will be able to find a model rationalizing the CCPs that features $(\widetilde{\pi}, \beta, G)$.

Identification of Counterfactuals

Imposing a restriction like $\forall x : \pi_0(x) = 0$ is NOT a normalization:

- If we were talking about a static normalization, each x would represent a different utility function, and $\pi_0(x) = 0$ would simply be a level normalization. However, in a dynamic model, the payoffs in one state affect the incentives in other states, so this is a substantive restriction.

But do these restrictions affect counterfactuals?

- It turns out that some (but not all!) counterfactuals ARE identified, in spite of the under-identification of the utility function.
- Whatever value $\widetilde{\pi}_0$ we impose for the reference action, the model will not only rationalize the observed CCP's but also predict the same counterfactual CCP's.
- Kalouptsidi, Scott, and Souza-Rodrigues (2020) sort out when counterfactuals of DDC models are identified and when they are not.

Extensions

- We cheated a bit because we assumed that not only were actions discrete but so was the state space. This trick is often attributed to Pesendorfer and Schmidt-Dengler (ReStud 2008).
- If the state space is not discrete we need to do some forward-simulation [next slide]. (Hotz, Miller, Sanders, Smith ReStud 1994).
- Others have extended these ideas to dynamic games. See Aguirragabiria and Mira (Ecma 2002/2008) and Bajari Benkard and Levin (Ecma 2007).
- Srisuma and Linton (2009) [very hard] show how to use Friedholm integral equations of 2nd kind to extend to continuous case.

Continuous State Space

When state space is continuous instead of discrete:

Exact Problem

$$V(x) = \max_{a \in A(x)} \left[(1-\beta)\, u(x,a) + \beta \int V(x')\, f(dx'|x,a) \right]$$

Approximation to the problem:

$$\hat{V}(x) = \max_{a \in \hat{A}(x)} \left[(1-\beta)\, u(x,a) + \beta \sum_{k=1}^{N} \hat{V}(x')\, f(x'_k|x,a) \right]$$

- Now we need to do actual numerical integration instead of just summation.
- We cannot use the $[I - \beta\, \mathbf{F}]^{-1}$ to get the ergodic distribution.
- Usually requires interpolating between grid points to evaluate $EV(\cdot)$.

Forward Simulation

In practice, "truncate" the infinite sum at some period T:

$$\begin{aligned}
&\tilde{V}(x, d = 1; \theta) = \\
&u(x, d = 1; \theta) + \beta E_{x'|x,d=1} E_{d'|x'} E_{\epsilon''|d',x'} [u(x', d'; \theta) + \epsilon' \\
&+ \beta E_{x''|x',d''} E_{d''|x''} E_{\epsilon'|d'',x''} [u(x'', d''; \theta) + \epsilon'' + \cdots \\
&\beta E_{x^T|x^{T-1},d^{T-1}} E_{d^T|x^T} E_{\epsilon^T|d^T,x^T} [u(x^T, d^T; \theta) + \epsilon^T]]]
\end{aligned}$$

Also, the expectation $E_{\epsilon|d,x}$ denotes the expectation of the ϵ conditional choice d being taken, and current mileage x. For the logit case, there is a closed form:

$$E[\epsilon|d, x] = \gamma - \log(Pr(d|x))$$

where γ is Euler's constant $(0.577\cdots)$ and $Pr(d|x)$ is the choice probability of action d at state x.

Forward Simulation

Choice-specific value functions can be simulated by (for $d = 1, 2$):

$$\tilde{V}(x, d; \theta) \approx \frac{1}{S} \sum_s [u(x, d; \theta) + \beta[u(x'^s, d'^s; \theta) + \gamma - \log(\hat{P}(d'^s | x'^s)) + \beta[u(x''^s, d''^s; \theta) + \gamma - \log(\hat{P}(d''^s | x''^s)) + \beta \cdots]]]$$

- $x'^s \sim \hat{G}(\cdot | x, d)$ and $d'^s \sim \hat{p}(\cdot | x'^s)$ and $x''^s \sim \hat{G}(\cdot | x'^s, d'^s)$, etc.
- In short, you simulate $\tilde{V}(x, d; \theta)$ by drawing S sequences of (d_t, x_t) with a initial value of (d, x), and compute the present-discounted utility correspond to each sequence.
- Then the simulation estimate of $\tilde{V}(x, d; \theta)$ is obtained as the sample average.

Forward Simulation

Given an estimate of $\tilde{V}(\cdot, d; \theta)$, you can get the predicted choice probabilities:

$$\tilde{p}(d = 1|x; \theta) \equiv \frac{\exp\left(\tilde{V}(x, d = 1; \theta)\right)}{\exp\left(\tilde{V}(x, d = 1; \theta)\right) + \exp\left(\tilde{V}(x, d = 0; \theta)\right)} \tag{1}$$

and analogously for $\tilde{p}(d = 0|x; \theta)$.

- Note that the predicted choice probabilities are different from $\hat{p}(d|x)$, which are the actual choice probabilities computed from the actual data.
- The predicted choice probabilites depend on the parameters θ, whereas $\hat{p}(d|x)$ depend solely on the data.

An obvious estimator minimizes $\arg\min_\theta \|\tilde{p}(d|x; \theta) - \hat{p}(d|x)\|$

Rust and Hotz-Miller Comparison

Rust's NFXP Algorithm

$$
\begin{aligned}
V_\theta(x) &= f(V_\theta(x), x, \theta) \Rightarrow f^{-1}(x,\theta) \\
P(d|x,\theta) &= g(V_\theta(x), x, \theta) \\
P(d|x,\theta) &= g(f^{-1}(x,\theta))
\end{aligned}
$$

- At every guess of θ we solve the fixed point inverse
- Plug that in to get choice probabilities
- Evaluate the likelihood

Hotz-Miller (1993) to Aguirregabiria and Mira (2002)

- Choice probabilities conditional on any value of observed state variables are uniquely determined by the vector of normalized value functions
- HM show invertibility proposition (under some conditions).
- If mapping is one-to-one we can also express value function in terms of choice probabilities.

$$
\begin{aligned}
V_\theta(x) &= h(P(d|x,\theta), x, \theta) \\
P(d|x,\theta) &= g(V_\theta(x), x, \theta) \\
\Rightarrow P(d|x,\theta) &= g(h(P(d|x,\theta), x, \theta), x, \theta)
\end{aligned}
$$

- The above fixed point relation is used in Aguirregabiria and Mira (2002) in their NPL Estimation algorithm.

Hotz-Miller (1993) to Aguirregabiria and Mira (2002)

$$P^{k+1}(d|x,\theta) \;=\; g(h(\hat{P}^k(d|x,\theta),s,\theta),s,\theta)$$

- Key point here is that the functions $h(\cdot)$ and $g(\cdot)$ are quite easy to compute (compared to the inverse f^{-1}).
- We can substantially improve estimation speed by replacing P with $\hat{P}$ the Hotz-Miller simulated analogue.
- The idea is to reformulate the problem from value space to probability space.
- When initializing the algorithm with consistent nonparametric estimates of CCP, successive iterations return a sequence of estimators of the structural parameters
- Call this the K stage policy iteration (PI) estimator.

Hotz-Miller (1993) to Aguirregabiria and Mira (2002)

- This algorithm nests Hotz Miller $(K = 1)$ and Rust's NFXP $(K = \infty)$.
- Asymptotically everything has the same distribution, but finite sample performance may be increasing in K (at least in Monte Carlo).
- The Nested Pseudo Likelihood (NPL) estimator of AM $(K = 2)$ seems to have much of the gains.
- For games things are more complicated. Pesendorfer and Scmidt-Dengler describe some problems with AM2007.
- For a modern treatment see Blevins and Dearing (2020).