Estimating Single-Agent Dynamic Models

Paul T. Scott

Empirical IO
Fall, 2013

Why are dynamics important?

- The motivation for using dynamics is usually external validity: we want to simulate counterfactuals involving changes not observed in the data.

Why are dynamics important?

- The motivation for using dynamics is usually external validity: we want to simulate counterfactuals involving changes not observed in the data.
- For example, often the data feature short-run price variation, but we are interested in responses to long-run changes.

Why are dynamics important?

- The motivation for using dynamics is usually external validity: we want to simulate counterfactuals involving changes not observed in the data.
- For example, often the data feature short-run price variation, but we are interested in responses to long-run changes.
  - Thinking back to Hendel and Nevo (2006), consumers may purchase extra laundry detergent during sales and then store it. Static elasticities are biased away from zero vs. long-run elasticities.

Why are dynamics important?

- The motivation for using dynamics is usually external validity: we want to simulate counterfactuals involving changes not observed in the data.
- For example, often the data feature short-run price variation, but we are interested in responses to long-run changes.
  - Thinking back to Hendel and Nevo (2006), consumers may purchase extra laundry detergent during sales and then store it. Static elasticities are biased away from zero vs. long-run elasticities.
  - Looking ahead to Scott (2013), farmers may respond little to year-to-year variation in prices when clearing land for crops, but might respond more to a long-run price increase. Static elasticities are biased toward zero vs. long-run elasticities.

Why are dynamics difficult?

- The computational burden of solving dynamic problems blows up with the state space. Consequently, much of the literature has been motivated by avoiding or alleviating the burden of having to solve the dynamic model.
- Other issues:
  - serially correlated unobservables
  - unobserved heterogeneity
  - solving for equilibria, multiplicity (when we get to dynamic games)

Why not two-stage models?

- Two-stage models are big simplifications which are only defensible for stable markets. They don't make sense for empirical applications where the identifying variation comes from changes over time.

Outline

Rust (1987)

Hotz and Miller (1993)

Su and Judd (2012)

Magnac and Thesmar (2002)

"Optimal Replacement of GMC Bus Engines:
An Empirical Model of Harold Zurcher"
John Rust (1987)

The "application"

- The decision maker decides whether replace bus engines or not, minimizing the expected discounted cost
- The trade-off: engine replacement is costly, but with increased use, the probability of a very costly breakdown increases
- Single agent setting: prices are exogenous, not worried about externalities across buses

Model, part I

- state variable: x_t is the bus engine's mileage
  - For computational purposes, Rust discretizes the state space into 90 intervals.
- Action $i_t \in \{0, 1\}$, where
  - $i_t = 1$ - replace the engine,
  - $i_t = 0$ - keep the engine and perform normal maintenance.

Model, part II

- per-period profit function:

$$u\left(i_t, x_t, \theta_1\right) = \begin{cases} -c\left(x_t, \theta_1\right) + \varepsilon_t\left(0\right) & \text{if } i_t = 0 \\ -\left(RC - c\left(0, \theta_1\right)\right) + \varepsilon_t\left(1\right) & \text{if } i_t = 1 \end{cases}$$

  where

  - $c\left(x_t, \theta_1\right)$ - regular maintenance costs (including expected breakdown costs),
  - RC - the net costs of replacing an engine,
  - ε - payoff shocks.

- x_t is observable to both agent and econometrician, but ε is only observable to the agent.
- ε is necessary for a coherent model, for sometimes we observe the agent making different decisions for the same value of x.

Model, part III

- Can define value function using Bellman equation:

$$V_\theta\left(x_t, \varepsilon_t\right) = \max_i \left[u\left(i, x_t, \theta\right) + \beta EV_\theta\left(x_t, \varepsilon_t, i_t\right)\right]$$

where

$$EV_\theta\left(x_t, \varepsilon_t, i_t\right) = \int V_\theta\left(y, \eta\right) p\left(dy, d\eta | x_t, \varepsilon_t, i_t, \theta_2, \theta_3\right)$$

Parameters

- θ_1 - parameters of cost function
- θ_2 - parameters of distribution of ε (these will be normalized away)
- θ_3 - parameters of x-state transition function
- RC - replacement cost
- discount factor β will be imputed

Conditional Independence

Conditional Independence Assumption

The transition density of the controlled process $\{x_t, \varepsilon_t\}$ factors as:

$$p\left(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, i_t, \theta_2, \theta_3\right) = q\left(\varepsilon_{t+1} | x_{t+1}, \theta_2\right) p\left(x_{t+1} | x_t, i_t, \theta_3\right)$$

- CI assumption is very powerful: it means we don't have to treat ε_t as a state variable, which would be very difficult since it's unobserved.
- While it is possible to allow the distribution of ε_{t+1} to depend on x_{t+1}, authors (including Rust) typically assume that any conditionally independent error terms are also identically distributed over time.

Theorem 1 preview

- Assumption CI has two powerful implications:
  - We can write $EV_\theta(x_t, i_t)$ instead of $EV_\theta(x_t, \varepsilon_t, i_t)$,
  - We can consider a Bellman equation for $EV_\theta(x_t, i_t)$, which is computationally simpler than the Bellman equation for $V_\theta(x_t, \varepsilon_t)$.

Theorem 1

Theorem 1

Given CI,

$$P\left(i|x,\theta\right) = \frac{\partial}{\partial u\left(x,i,\theta_1\right)} G\left(u\left(x,\theta_1\right) + \beta EV_\theta\left(x\right)|x,\theta_2\right)$$

and EV_θ is the unique fixed point of the contraction mapping:

$$EV_\theta\left(x,i\right) = \int_y G\left(u\left(y,\theta_1\right) + \beta EV_\theta\left(y\right)|y,\theta_2\right) p\left(dy|x,i,\theta_3\right)$$

where

- $P\left(i|x,\theta\right)$ is the probability of action i conditional on state x
- $G\left(\cdot|,x,\theta_2\right)$ is the surplus function:

$$G\left(v|,x,\theta_2\right) \equiv \int_\varepsilon \max_i \left[v\left(i\right) + \varepsilon\left(i\right)\right] q\left(d\varepsilon|x,\theta_2\right)$$

Theorem 1, example

- Let $v_\theta(x,i) \equiv u(x,i,\theta_1) + \beta EV_\theta(x,i)$. This is often called the *conditional value function*.
- Suppose that $\varepsilon(i)$ is distributed independently across i with $Pr(\varepsilon(i) \le \varepsilon_0) = e^{-e^{-\varepsilon_0}}$. Then,

$$\begin{aligned} G(v(x)) &= \int \max_i [v(x,i) + \varepsilon(i)] \prod_i e^{-\varepsilon(i)} e^{-e^{-\varepsilon(i)}} d\varepsilon \\ &= \ln\left(\sum_i \exp(v(x,i))\right) + \gamma \end{aligned}$$

  where $\gamma \approx .577216$ is Euler's gamma.
- It is then easy to derive expressions for conditional choice probabilities:

$$P(i|x,\theta) = \frac{\exp(v_\theta(x,i))}{\sum_{i'} \exp(v_\theta(x,i'))}$$

Some details

- He assumes ε is i.i.d with an extreme value type 1 distribution, and normalizes its mean to 0 and variance to $\pi^2/6$ (this is the case on the previous slide).
- Transitions on observable state:

$$\begin{aligned} p\left(x_{t+1} - x_t = 0 | , x_t, i_t, \theta_3\right) &= \theta_{30} \\ p\left(x_{t+1} - x_t = 1 | , x_t, i_t, \theta_3\right) &= \theta_{31} \\ p\left(x_{t+1} - x_t = 2 | , x_t, i_t, \theta_3\right) &= 1 - \theta_{30} - \theta_{31} \end{aligned}$$

- He tries several different specifications for the cost function and favors a linear form:

$$c\left(x, \theta_1\right) = \theta_{11} x.$$

Nested Fixed Point Estimation

- Rust first considers a case with a closed-form expression for the value function, but this calls for restrictive assumptions on how mileage evolves. His nested fixed point estimation approach, however, is applicable quite generally.
- Basic idea: to evaluate objective function (likelihood) at a given θ, we should solve the value function for that θ

Nested Fixed Point Estimation

Steps:

1. Impute a value of the discount factor β
2. Estimate θ_3 – the transition function for x – which can be done without the behavioral model
3. Inner loop: search over (θ_1, RC) to maximize likelihood function. When evaluating the likelihood function for each candidate value of (θ_1, RC):
   3.1 Find the fixed point of the the Bellman equation for $(\beta, \theta_1, \theta_3, RC)$. Iteration would work, but Rust uses a faster approach.
   3.2 Using expression for conditional choice probabilities, evaluate likelihood:

$$\prod_{t=1}^{T} P\left(i_t | x_t, \theta\right) p\left(x_t | x_{t-1}, i_{t-1}, \theta_3\right)$$

Estimates

TABLE IX

STRUCTURAL ESTIMATES FOR COST FUNCTION $c(x, \theta_1) = .001\theta_{11}x$

FIXED POINT DIMENSION = 90

(Standard errors in parentheses)

Parameter		Data Sample			Heterogeneity Test	
Discount Factor	Estimates/ Log-Likelihood	Groups 1, 2, 3 3864 Observations	Group 4 4292 Observations	Groups 1, 2, 3, 4 8156 Observations	LR Statistic ($df = 4$)	Marginal Significance Level
$\beta = .9999$	RC	11.7270 (2.602)	10.0750 (1.582)	9.7558 (1.227)	85.46	1.2E − 17
	θ_{11}	4.8259 (1.792)	2.2930 (0.639)	2.6275 (0.618)		
	θ_{30}	.3010 (.0074)	.3919 (.0075)	.3489 (.0052)		
	θ_{31}	.6884 (.0075)	.5953 (.0075)	.6394 (.0053)		
	LL	−2708.366	−3304.155	−6055.250		
$\beta = 0$	RC	8.2985 (1.0417)	7.6358 (0.7197)	7.3055 (0.5067)	89.73	1.5E − 18
	θ_{11}	109.9031 (26.163)	71.5133 (13.778)	70.2769 (10.750)		
	θ_{30}	.3010 (.0074)	.3919 (.0075)	.3488 (.0052)		
	θ_{31}	.6884 (.0075)	.5953 (.0075)	.6394 (.0053)		
	LL	−2710.746	−3306.028	−6061.641		
Myopia test:	LR Statistic ($df = 1$)	4.760	3.746	12.782		
$\beta = 0$ vs. $\beta = .9999$	Marginal Significance Level	0.0292	0.0529	0.0035		

Discount factor

- While Rust finds a better fit for $\beta = .9999$ than $\beta = 0$, he finds that high levels of β basically lead to the same level of the likelihood function.
- Furthermore, the discount factor is non-parametrically non-identified. Note: He loses ability to reject $\beta = 0$ for more flexible cost function specifications.

Discount factor

TABLE VIII

SUMMARY OF SPECIFICATION SEARCH[a]

Cost Function	Bus Group 1, 2, 3	Bus Group 4	Bus Group 1, 2, 3, 4
Cubic $c(x, \theta_1) = \theta_{11}x + \theta_{12}x^2 + \theta_{13}x^3$	Model 1 −131.063 −131.177	Model 9 −162.885 −162.988	Model 17 −296.515 −296.411
quadratic $c(x, \theta_1) = \theta_{11}x + \theta_{12}x^2$	Model 2 −131.326 −131.534	Model 10 −163.402 −163.771	Model 18 −297.939 −299.328
linear $c(x, \theta_1) = \theta_{11}x$	Model 3 −132.389 −134.747	Model 11 −163.584 −165.458	Model 19 −300.250 −306.641
square root $c(x, \theta_1) = \theta_{11}\sqrt{x}$	Model 4 −132.104 −133.472	Model 12 −163.395 −164.143	Model 20 −299.314 −302.703
power $c(x, \theta_1) = \theta_{11}x^{\theta_{12}}$	Model 5[b] N.C. N.C.	Model 13[b] N.C. N.C.	Model 21[b] N.C. N.C.
hyperbolic $c(x, \theta_1) = \theta_{11}/(91 - x)$	Model 6 −133.408 −138.894	Model 14 −165.423 −174.023	Model 22 −305.605 −325.700
mixed $c(x, \theta_1) = \theta_{11}/(91 - x) + \theta_{12}\sqrt{x}$	Model 7 −131.418 −131.612	Model 15 −163.375 −164.048	Model 23 −298.866 −301.064
nonparametric $c(x, \theta_1)$ any function	Model 8 −110.832 −110.832	Model 16 −138.556 −138.556	Model 24 −261.641 −261.641

[a] First entry in each box is (partial) log likelihood value ℓ^2 in equation (5.2)) at $\beta = .9999$. Second entry is partial log likelihood value at $\beta = 0$.

[b] No convergence. Optimization algorithm attempted to drive $\theta_{12} \to 0$ and $\theta_{11} \to +\infty$.

Application

Expected Replacement Demand Function
Annual Replacement Demand for Model 11

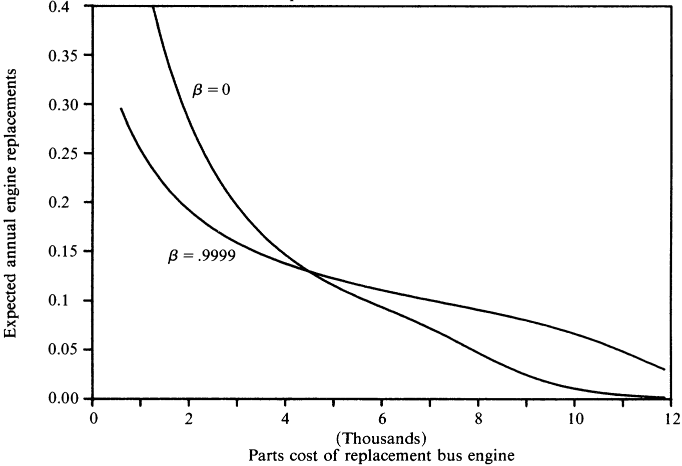

FIGURE 7

"Conditional Choice Probabilities and the Estimation of Dynamic Models"
Hotz and Miller (1993)

Motivation

- 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)
- Rather than following the details of Hotz and Miller (1993) exactly, these slides aim to emphasize what you can do with the Hotz-Miller inversion and how it differs from Rust.

Rust's Theorem 1: Values to CCP's

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

$$P(i|x,\theta) = \frac{\partial}{\partial u(x,i,\theta)} G\left(u(x,\theta) + \beta EV(x)\,|x,\theta_2,\theta\right)$$

- For the logit case:

$$P(i|x,\theta) = \frac{\exp\left(v(x,i,\theta)\right)}{\sum_{i'\in\mathbf{I}} \exp\left(v(x,i',\theta)\right)}$$

where $\mathbf{I}$ is some finite choice set, and

$$v(x,i,\theta) \equiv u(x,i,\theta) + \beta EV(x,i,\theta)$$

HM's Proposition 1: CCP's to Values

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

$$dv(x,i) \equiv v(x,i) - v(x,0)$$

where $i = 0$ is some reference action.

- Let $Q : \mathbb{R}^{|\mathbf{I}|-1} \to \Delta^{|\mathbf{I}|}$ be the mapping from the differences in conditional values to CCP's.

- Note: we're taking for granted that the distribution of ε identical across states, otherwise Q would be different for different x.

Proposition 1

Q is invertible.

HM inversion with logit errors

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

$$P\left(i|x,\theta\right)=\frac{\exp\left(v\left(x,i,\theta\right)\right)}{\sum_{i'\in\mathbf{I}}\exp\left(v\left(x,i',\theta\right)\right)}.$$

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

$$\ln P\left(i|x,\theta\right)-\ln P\left(0|x,\theta\right)=v\left(x,i,\theta\right)-v\left(x,0,\theta\right)$$

- Thus, in the logit case

$$Q_i^{-1}\left(p\right)=\ln p_i-\ln p_0$$

HM estimation overview

1. Estimate process governing evolution of x
2. Estimate conditional choice probabilities
   - For a discrete state space, in principle we can just obtain frequency estimates for each CCP
   - For continuous state spaces, common to use some non-parametric estimator (e.g., using kernels or sieves)
3. Recover value functions from estimated CCP's using HM inversion.
4. Estimate θ based on estimated value functions.

How we do step 3 depends on the setting, and there are several possibilities for objective functions in step 4.

Example 1: estimation with terminal action

- Suppose $i = 0$ is a terminal action, i.e., $EV(x, 0, \theta) = 0$, and

$$v(x, 0, \theta) = u(x, 0, \theta)$$

- Then, sticking with the logit case,

$$\begin{aligned} EV(x_t, i_t, \theta) &= \int_{x_{t+1}} \ln\left(\sum_{i'} \exp\left(v\left(x_{t+1}, i', \theta\right)\right)\right) p\left(dx_{t+1}|x_t, i_t\right) + \gamma \\ &= \int_{x_{t+1}} \ln\left(\sum_{i'} \exp\left(dv\left(x_{t+1}, i', \theta\right)\right.\right. \\ &\quad + u\left(x_{t+1}, 0, \theta\right)\left.\left.\right)\right) p\left(dx_{t+1}|x_t, i_t\right) + \gamma \\ &= \int_{x_{t+1}} \ln\left(\sum_{i'} \exp\left(dv\left(x_{t+1}, i', \theta\right)\right)\right) p\left(dx_{t+1}|x_t, i_t\right) \\ &\quad + \int_{x_{t+1}} u\left(x_{t+1}, 0, \theta\right) p\left(dx_{t+1}|x_t, i_t\right) + \gamma \end{aligned}$$

Example 1: estimation with terminal action

Next, plug in the estimate from the Hotz-Miller inversion,

$$\tilde{dv}\left(x,i\right)=\ln\hat{P}\left(x,i\right)-\ln\hat{P}\left(x,0\right),$$

to construct

$$\begin{aligned}\tilde{EV}\left(x_t,i_t,\theta\right)= \quad & \int_{x_{t+1}}\ln\left(\sum_i\exp\left(\tilde{dv}\left(x,i\right)\right)\right)p\left(dx_{t+1}|x_t,i_t\right) \\ & +\int_{x_{t+1}}u\left(x_{t+1},0,\theta\right)p\left(dx_{t+1}|x_t,i_t\right).\end{aligned}$$

Example 1: estimation with terminal action

The expression for $\tilde{dv}$ can be fed into the expression for continuation values:

$$\tilde{v}(x, i, \theta) = u(x, i, \theta) + \beta \tilde{EV}(x, i, \theta),$$

which can be used to form new expressions for CCP's:

$$\tilde{P}(x, i, \theta) = \frac{\exp(\tilde{v}(x, i, \theta))}{\sum_{i'} \exp(\tilde{v}(x, i', \theta))}.$$

Example 1: estimation with terminal action

- Note that, unlike Rust's predicted choice probabilities, $\tilde{P}$ can be computed without solving a value function.
- Finally, reconstructed CCP's can be used to create a pseudo-log-likelihood function:

$$\hat{\theta}^{NPL} = \arg\max_{\theta} \sum_{t=1}^{T} \ln\left(\tilde{P}\left(x_t, i_t, \theta\right)\right).$$

- Another possibility is to minimize the distance between predicted and estimated CCP's:

$$\hat{\theta} = \arg\min_{\theta} \left\| \tilde{P}\left(\theta\right) - \hat{P} \right\|.$$

Aguirregabiria and Mira (2002)

Hotz and Miller's approach may be computationally less demanding than Rust's nested fixed point approach, but it is also less efficient (in the asymptotic sense). Aguirregabiria and Mira (2002) develop a class of estimators that bridges the gap between the two. (left as reading)

Example 2: finite state space

Let's consider another way of computing the surplus:

$$E\left(\max_i \left\{v\left(x,i,\theta\right)+\varepsilon\left(i\right)\right\}\right)$$

$$=\sum_i P\left(x,i,\theta\right) E\left[v\left(x,i,\theta\right)+\varepsilon\left(i\right)|\forall i' : v\left(x,i,\theta\right)+\varepsilon\left(i\right)\geq v\left(x,i',\theta\right)+\varepsilon\left(i'\right)\right]$$

$$=\sum_i P\left(x,i,\theta\right)\left(v\left(x,i,\theta\right)+\psi\left(x,i,\theta\right)\right)$$

where

$$\psi\left(x,i,\theta\right)=E\left[\varepsilon\left(i\right)|\forall i' : v\left(x,i,\theta\right)+\varepsilon\left(i\right)\geq v\left(x,i',\theta\right)+\varepsilon\left(i'\right)\right]$$

Example 2: finite state space

- In the case of logit errors, we have the simple expression $\psi(x, i, \theta) = \gamma - \ln P(x, i, \theta)$.
- Define $F(i)$ as the $|X| \times |X|$ matrix of state transitions for action i. Then,

$$EV(\theta) = \left(I_{|X|} - \beta \sum_i P(i) * F(i) \right) -1 \left(\sum_i P(i) * (u(i, \theta) + \psi(i, \theta)) \right)$$

where * denotes elementwise multiplication.

- Again, we can construct $\tilde{EV}(\theta)$ using first-stage estimates of conditional choice probabilities.
- Then, as before, we can plug $\tilde{EV}(\theta)$ into our expressions for conditional values and conditional choice probabilities.

"Constrained Optimization Approaches to Estimation of Structural Models"
Su and Judd (2012)

MPEC approach

- Rust's approach was based on writing a likelihood function like so:

$$\max_{\theta} \mathcal{L}\left(\theta, EV\left(\theta\right), X\right)$$

where $V\left(\theta\right)$ is the value function, and X is the data.

- $EV\left(\theta\right)$ is defined as the unique solution to $EV = T\left(EV, \theta\right)$
- Su and Judd suggest formulating the following constrained optimization problem instead:

$$\max_{\theta} \mathcal{L}\left(\theta, EV, X\right)$$

subject to

$$EV = T\left(EV, \theta\right)$$

MPEC approach: computational advantages

- Rather than solving the value function for each candidate θ, the Bellman equation is a constraint.
- The result is that the solver need not impose $EV = T(EV, \theta)$ every time the objective function is evaluated, but the Bellman equation must hold at the solution.
- the result is that the Bellman equation is evaluated many fewer times during the optimization routine, and there can be substantial speed gains.
- Note: this estimator is equivalent to Rust's, it's just a different algorithm. In contrast, estimators based on the Hotz-Miller inversion are typically different estimators.

"Identifying Dynamic Discrete Decision Processes"
Magnac and Thesmar (2002)

Setup

- $x \in X$ - state variables
- $p_i(x)$ - choice probabilities (data)
- $u_i(x)$ - per-period utility from action i in state x
- $v_i(x)$ - conditional value function of action i in state x
- K - the reference action
- G - distribution of conditionally independent shocks
- q - the Hotz-Miller inversion function. i.e., $q_i(p(x)) = v_i(x) - v_K(x)$
- R - the surplus function, $R(v; G) = E_G(\max_i \{v_i + \varepsilon_i\})$

Lemma 1

Lemma 1 is basically a convenient restatement of the Hotz Miller inversion.

Lemma 1

For any action j and state x,

$$\begin{aligned} u_j(x) &= u_K(x) + q_j(p(x); G) \\ &\quad -\beta\left(E\left[v_K(x') \mid x, j\right] - E\left[v_K(x') \mid x, K\right]\right) \\ &\quad -\beta\left(E\left[R(q(p(x'); G)) \mid x, j\right] - E\left[R(q(p(x'); G)) \mid x, K\right]\right) \end{aligned}$$

Lemma 1, example

Let's derive Lemma 1 for the case of logit errors.

$$
\begin{aligned}
\ln\left(\frac{p_j(x)}{p_K(x)}\right) &= v_j(x) - v_K(x) \\
&\Leftrightarrow \\
u_j(x) &= u_K(x) + \ln\left(\frac{p_j(x)}{p_K(x)}\right) \\
&\quad -\beta\left(E\left[\bar{V}(x')\,|x,j\right] - E\left[\bar{V}(x')\,|x,K\right]\right) \\
&= u_K(x) + \ln\left(\frac{p_j(x)}{p_K(x)}\right) \\
&\quad -\beta E\left[\ln\sum_i \exp\left(\frac{p_i(x')}{p_K(x')} + v_K(x')\right)|x,j\right] \\
&\quad +\beta E\left[\ln\sum_i \exp\left(\frac{p_i(x')}{p_K(x')} + v_K(x')\right)|x,K\right]
\end{aligned}
$$

Lemma 1, example

Let's derive Lemma 1 for the case of logit errors.

$$\begin{aligned}
\ln\left(\frac{p_j(x)}{p_K(x)}\right) &= v_j(x) - v_K(x) \\
&\Leftrightarrow \\
u_j(x) &= u_K(x) + \ln\left(\frac{p_j(x)}{p_K(x)}\right) \\
&\quad -\beta\left(E\left[\bar{V}(x')\,|x,j\right] - E\left[\bar{V}(x')\,|x,K\right]\right) \\
&= u_K(x) + \ln\left(\frac{p_j(x)}{p_K(x)}\right) \\
&\quad -\beta\left(E\left[v_K(x')\,|x,j\right] - E\left[v_K(x')\,|x,K\right]\right) \\
&\quad -\beta\left(E\left[\ln\sum_i \exp\left(\frac{p_i(x')}{p_K(x')}\right)|x,j\right]\right. \\
&\quad \left.-E\left[\ln\sum_i \exp\left(\frac{p_i(x')}{p_K(x')}\right)|x,K\right]\right)
\end{aligned}$$

Lemma 1

Proposition 2

Let $C = \{c | c = (\beta, G, u_K(\cdot), v_K(\cdot))\}$.
(i) For a given $c \in C$, there exists one vector $(u_1(\cdot), \ldots, u_{K-1}(\cdot))$ compatible with $p(X)$.
(ii) Let $\mathbf{u}$ be the $(u_1(\cdot), \ldots, u_{K-1}(\cdot))$ vector associated with c and let $\mathbf{u}'$ be associated with $c' \neq c$. Then, $(\mathbf{u}, c)$ and $(\mathbf{u}', c')$ are observationally equivalent.

Thus, there is an observationally equivalent model associated with each element of $C = \{c | c = (\beta, G, u_K(\cdot), v_K(\cdot))\}$

Restrictions for identification

- It's common to do something like assuming $u_K(x) = 0$ for all x
- This is not innocent. Note that restricting $u(x) = 0$ for a single x is an innocent normalization, but restricting payoffs to be flat across states is a substantive assumption.
  - Sometimes such restrictions are very natural. Think about what the restriction is in Rust (1987).
- In some (limited) cases, counterfactuals are identified even though the utility function isn't fully identified:
  - Norets and Tang (2013), "Semiparametric Inference in Dynamic Binary Choice Models"
  - Aguirregabiria and Suzuki (2013), "Identification and Counterfactuals in Dynamic Models of Market Entry and Exit"
- Suggestion: one might avoid restrictive assumptions with data about the value function (e.g., rental rates or land values)

Identifying the discount factor

Idea behind Magnac and Thesmar's Proposition 4:

- Suppose you have different values of state variables which give the same current profits, but have different expectations for future values of the state variables.
  - e.g., perhaps we observe both current and futures prices
- In this case, one can identify the discount factor because we have something that shifts continuation values rather than shifting both current profits and continuation values.