Lecture 8: Implementation of structural estimation methods for discrete decision problems (NFXP, MPEC, CCP + NPL, BBL, MSM)

2019 Econometric Society Summer School in Dynamic Structural Econometrics

Fedor Iskhakov, Australian National University
John Rust, Georgetown University
Bertel Schjerning, University of Copenhagen

Booth School of Business and Becker Friedman Institute for Economics
University of Chicago
July 8th - 14th, 2019

Road Map: Implementation of structural estimation methods

PART I: Single agent settings

1. Summary of Rust model + NFXP, MPEC
2. Nested Pseudo Likelihood (NPL) and CCP
3. Bajari, Benkard, Levin (2007) estimator (BBL)
4. Method of Simulated Moments (MSM)

PART II: Multiple decision makers setting: simple static entry game

1. NFXP
2. MPEC
3. CCP
4. NPL

Zurcher's Bus Engine Replacement Problem

- Choice set: Each bus comes in for repair once a month and Zurcher chooses between ordinary maintenance ($d_t = 0$) and overhaul/engine replacement ($d_t = 1$)
- State variables: Harold Zurcher observes:
  - x_t: mileage at time t since last engine overhaul
  - $\varepsilon_t = [\varepsilon_t(d_t = 0), \varepsilon_t(d_t = 1)]$: other state variable
- Utility function:

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

- State variables process x_t (mileage since last replacement)

$$p(x_{t+1}|x_t, d_t, \theta_2) = \begin{cases} g(x_{t+1} - 0, \theta_2) & \text{if } d_t = 1 \\ g(x_{t+1} - x_t, \theta_2) & \text{if } d_t = 0 \end{cases} \qquad (2)$$

- If engine is replaced, state of bus regenerates to $x_t = 0$.

Structural Estimation

Data: $(d_{i,t}, x_{i,t})$, $t = 1, ..., T_i$ and $i = 1, ..., n$

Likelihood function

$$\ell_i^f(\theta) = \sum_{t=2}^{T_i} log(P(d_{i,t}|x_{i,t},\theta)) + \sum_{t=2}^{T_i} log\left(p(x_{i,t}|x_{i,t-1}, d_{i,t-1}, \theta_2)\right)$$

where

$$P(d|x,\theta) = \frac{\exp\{u(x,d,\theta_1) + \beta EV_\theta(x,d)\}}{\sum_{d' \in \{0,1\}}\{u(x,d',\theta_1) + \beta EV_\theta(x,d')\}}$$

and

$$\begin{aligned} EV_\theta(x,d) &= \Gamma_\theta(EV_\theta)(x,d) \\ &= \int_y \ln\left[\sum_{d' \in \{0,1\}} \exp[u(y,d';\theta_1) + \beta EV_\theta(y,d')]\right] p(dy|x,d,\theta_2) \end{aligned}$$

The Nested Fixed Point Algorithm

NFXP solves the *unconstrained* optimization problem

$$\max_{\theta} L(\theta, EV_{\theta})$$

Outer loop (Hill-climbing algorithm):

- Likelihood function $L(\theta, EV_{\theta})$ is maximized w.r.t. θ
- Quasi-Newton algorithm: Usually BHHH, BFGS or a combination.
- Each evaluation of $L(\theta, EV_{\theta})$ requires solution of EV_{θ}

Inner loop (fixed point algorithm):
The implicit function EV_{θ} defined by $EV_{\theta} = \Gamma(EV_{\theta})$ is solved by:

- Successive Approximations (SA)
- Newton-Kantorovich (NK) Iterations

Mathematical Programming with Equilibrium Constraints

MPEC solves the *constrained* optimization problem

$$\max_{\theta, EV} L(\theta, EV) \text{ subject to } EV = \Gamma_{\theta}(EV)$$

using general-purpose constrained optimization solvers such as KNITRO

Su and Judd (Ecta 2012) considers two such implementations:

MPEC/AMPL:

- AMPL formulates problems and pass it to KNITRO.
- Automatic differentiation (Jacobian and Hessian)
- Sparsity patterns for Jacobian and Hessian

MPEC/MATLAB:

- User need to supply Jacobians, Hessian, and Sparsity Patterns
- Su and Judd do not supply analytical derivatives.
- ktrlink provides link between MATLAB and KNITRO solvers.

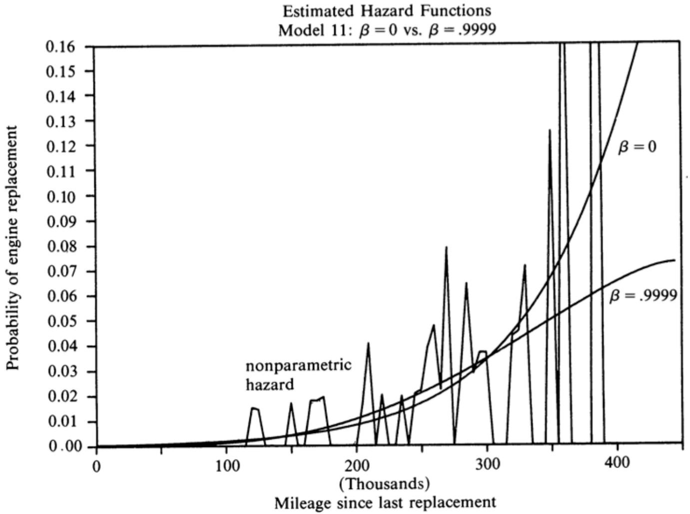
Matlab Implementation

NFXP (bus data):
run_busdata.m

MPEC and NFXP
(Monte Carlo):
run_nfxp_vs_mpec.m

Small sample problems

Sometimes it can be hard to get a precise nonparametric estimate of CCP



Main Contributions of Aguirregabiria and Mira (2002)

Nested Pseudo Likelihood (NPL) algorithm

- Solution of the DP problem in choice probability space (rather than value functions space)

Statistical and computational properties of the estimator.

- When NPL is initialized with consistent nonparametric estimates of conditional choice probabilities, successive iterations return a sequence of estimators of the structural parameters callled K-stage policy iteration estimators.
- The sequence includes as extreme cases a Hotz-Miller estimator (for K = 1) and Rust's nested fixed point MLE estimator (in the limit when $K \to \infty$).

Monte Carlo experiments

- Monte Carlo based on Rust's bus replacement model.
- Reveal a trade-off between finite sample precision and computational cost in the sequence of policy iteration estimators.

The General Problem

Bellman equation

$$V(s;\theta) = \max_{a \in \mathcal{A}(s)} \{u(s,a;\theta_u) + \beta \int V(s';\theta)p(s'|s,a;\theta_g,\theta_f)ds'\}$$

u and p: known up to a set of parameters, $\theta = (\theta_u, \theta_g, \theta_f)$

- **The agent's problem:** Maximize expected sum of current and future discounted utilities
  - a : Discrete control variable, $a \in \mathcal{A}(s) = \{1, 2, .., J\}$.
  - s : Current state, fully observed by agent
  - $s\prime$: Future state; possibly continuous and subject to uncertainty
- **The agents beliefs about s′:**
  - Obeys a (controlled) Markov transition probability $p(s_{t+1}|s_t, a_t; \theta_g, \theta_f)$
- **Model solution,** $V(s;\theta)$
  - Find the fixed point for the Bellman equation

A&M maintain Rust's Assumptions

Assumption (Conditional Independence (CI))

State variables, $s_t = (x_t, \varepsilon_t)$ obeys a (conditional independent) controlled Markov process with probability density

$$p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, a, \theta_g, \theta_f) = g(\varepsilon_{t+1} | x_{t+1}, \theta_g) f(x_{t+1} | x_t, a, \theta_f)$$

Assumption (Additive Separability (AS))

$$U(s_t, a) = u(x_t, a; \theta_u) + \varepsilon_t(a)$$

Assumption (Finite Domain of Observable State Variables)

$$x \in X = \{x^1, x^1, ..., x^m\}$$

Assumption (XV)

The unobserved state variables, ε_t are assumed to be multivariate iid. extreme value distributed

Bellman equation and choice probabilities

Define the smoothed value function $V_\sigma(x) = \int V(x,\epsilon)g(\epsilon|x)d\epsilon$ where σ represents parameters that index the distribution of the $\epsilon's$.
($\sigma = \theta_2$ in Rust notation)

Under assumptions CI, AS and finite domain of x, we can summarize the solution by the smoothed Bellman operator, $\Gamma_\sigma(V_\sigma)$

$$V_\sigma(x) = \int \max_{a\in\mathcal{A}(x)} \left\{ u(x,a) + \epsilon(a) + \beta \sum_{x'} V_\sigma(x')f(x'|x,a) \right\} g(\epsilon|x)d\epsilon$$

The conditional choice probability (CCP)

$$P(a|x) = \int I\{a = \arg\max_{j\in\mathcal{A}(x)} \{v(j,x) + \epsilon(j)\}\, g(\epsilon|x)d\epsilon$$

where $v(x,a) = u(x,a) + \beta \sum_{x'} V_\sigma(x')f(x'|x,a)$ is the choice-specific value function

From Conditional Choice Probabilities to Value Functions

- $P(a|x)$ is uniquely determined by the vector of normalized value function differences $\tilde{v}(x,a) = v(x,a) - v(x,1)$
- That is, there exists a vector mapping $Q_x(.)$ such that $\{P(a|x) : a > 1\} = Q_x(\tilde{v}(x,a) : a > 1)$, where, without loss of generality, we exclude the probability of alternative one.
- In general

$$Q_x^j(\tilde{v}(x,a) : a > 1) = \partial S([0, \{\tilde{v}(x,a) : a > 1\}], x) / \partial \tilde{v}(x,j)$$

where

$$S(\{v(x,a) : a \in A\}, x) = \int \max[v(x,a) + \epsilon(a)] g(d\epsilon|x)$$

is McFadden's social surplus function.

From Conditional Choice Probabilities to Value Functions

- Under assumption (XV), social surplus function is the well known "log-sum" formula

$$S(\{v(x,a) : a \in A\}, x) = \int \max_{a \in A}[v(x,a) + \epsilon(a)]g(d\epsilon|x)$$
$$= \sigma \log \sum_{j \in A} exp(v(x,j)/\sigma)$$

the j'th component Q_x takes the well known logistic form

$$Q_x^j(\tilde{v}(x,a)) = \frac{\exp(\tilde{v}(x,a)/\sigma)}{1 + \sum_{j=2}^{A} \exp(\tilde{v}(x,j)/\sigma)}$$

- NOTE, it's not hard to invert Q_x in this case

From Conditional Choice Probabilities to Value Functions

The Smooth Bellman equation can be re-written as

$$V_\sigma(x) = \sum_{a \in A} P(a|x) \left\{ u(x,a) + E[\epsilon(a)|x,a] + \beta \sum_{x'} V_\sigma(x') f(x'|x,a) \right\}$$

where $E[\epsilon(a)|x,a]$ is the expectation of the unobservable ϵ conditional on the optimal choice of alternative a:

$$\begin{aligned} E[\epsilon(a)|x,a] = [P(a|x)]^{-1} \int & \epsilon(a) I\{\tilde{v}(x,a) + \epsilon(a) \\ & > \tilde{v}(x,k) + \epsilon(j), j \in A(x)\} g(d\epsilon|x) \end{aligned}$$

$E[\epsilon(a)|x,a]$ clearly depends on $\tilde{v}(x,a)$, but due to the invertibility of Q_x we can express it probability space

$$E[\epsilon(a)|x,a] = e_x(a, \{P(j|x)\}_{j \in A}).$$

Under XV we have $E[\epsilon(a)|x,a] = \gamma - \ln P(a|x)$ where $\gamma = 0.5772156649...$ is Euler's constant

From Conditional Choice Probabilities to Value Functions

In compact matrix notation we can write

$$V_\sigma = \sum_{a \in A} P(a) * \{u(a) + e(a, P) + \beta F(a) V_\sigma\}$$

where $*$ is the element by element product and $P(a)$, $u(a)$, $e(a, P)$ and V_σ are all $M \times 1$ vectors and $F(a)$ is the $M \times M$ matrix of conditional transition probabilities $f(x'|x, a)$

This system of fixed point equations can be solved for the value function to obtain V_σ as a function of P:

$$V_\sigma = \psi(P) = [I - \beta F^U(P)]^{-1} \sum_{a \in A} \{P(a) * [u(a) + e(a, P)]\}$$

where $F^U(P) = \sum_{a \in A} P(a) F(a)$ is the $M \times M$ matrix of unconditional transition probabilities induced by P.

The Fixed Point Problem in Probability Space

Recall that

$$V_\sigma = \psi(P) = [I - \beta F^U(P)]^{-1} \sum_{a \in A} \{P(a) * [u(a) + e(a, P)]\} \qquad (3)$$

and

$$P(a|x) = \int I\{a = \arg \max_{j \in \mathcal{A}(x)} \{v(j, x) + \epsilon(j)\}\, g(\epsilon|x) d\epsilon \qquad (4)$$

where $v(x, a) = u(x, a) + \beta \sum_{x'} V_\sigma(x') f(x'|x, a)$

The policy iteration operator, Ψ

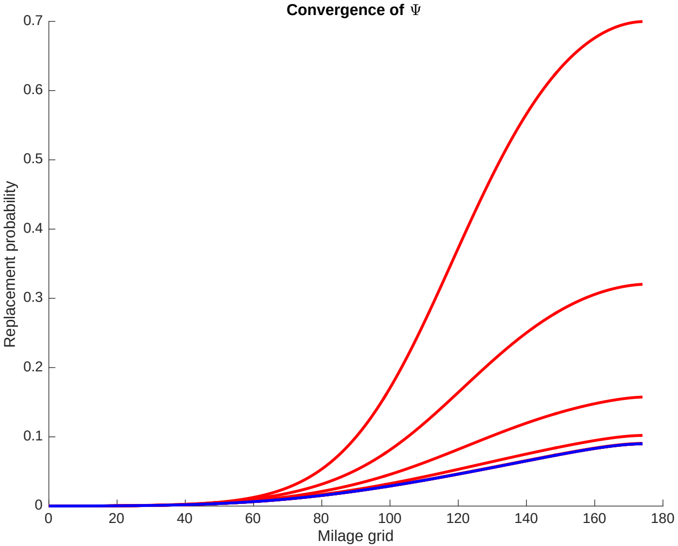
Substituting the *policy valuation operator*, $\psi(P)$ defined by (3) into the formula for CCP's in (4) we obtain the cornerstone of NPL algorithm algorithm:

$$P = \Psi(P) = \Lambda(\psi(P))$$

- Hence, the optimal choice probabilities P is a fixed point of Ψ.
- Thus the original fixed point problem in "value space" can be reformulated as a fixed point problem in "probability space"

Finding fixed point, $P = \Psi(P)$

Fast convergence of successive approximations, $P_{k+1} = \Psi(P_k)$



Likelihood function

Data: $(a_{i,t}, x_{i,t})$, $t = 1, ..., T_i$ and $i = 1, ..., n$

Likelihood function

$$\ell_i^f(\theta) = \ell_i^1(\theta) + \ell_i^2(\theta_f) = \sum_{t=2}^{T_i} log(P_\theta(a_{i,t}|x_{i,t})) + \sum_{t=2}^{T_i} log\left(f_{\theta_f}(x_{i,t}|x_{i,t-1}, a_{i,t-1})\right)$$

Two Step-Estimator

- Consistent estimates of the conditional transition probability parameters θ_f can be obtained from transition data without having to solve the Markov decision model.
- We focus on the estimation of $\alpha = (\theta_u, \theta_g)$ given initial consistent estimates of θ_f obtained from maximizing the partial log-likelihood $\ell^2(\theta_f) = \sum_i \ell_i^2(\theta_f)$
- Originally suggested in Rust(1987)

Nested Pseudo Likelihood Algorithm

Initialization

- Let $\hat{\theta}_f$ be an estimate of θ_f .
- 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 a new pseudo-likelihood estimate of α, α^K, as

$$\alpha^K = \arg\max_{\alpha \in \Theta} \sum_{i=1}^{n} \ln \Psi_{\alpha,\hat{\theta}_f}(P^{K-1})(a_i|x_i) \tag{5}$$

  where $\Psi_\theta(P)(a|x)$ is the (a,x)'s element of $\Psi_\theta(P)$.
- Step 2: Update P using the arg max from step 1, i.e.

$$P^K = \Psi_{(\alpha^K,\hat{\theta}_f)}(P^{K-1}) \tag{6}$$

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

Sequential Policy Iteration Estimators

- Performing one, two, and in general K iterations of the NPL algorithm yields a sequence $\{\hat{\alpha}_1, \hat{\alpha}_1, ..., \hat{\alpha}_K\}$ of statistics that can be used as estimators of the true value of α, α^*
- A&M call them sequential policy iteration (PI) estimators.

The K-stage PI estimator is defined as:

$$\hat{\alpha}^K = \arg\max_{\alpha \in \Theta} \sum_{i=1}^{n} \ln \Psi(P^{K-1})(a_i|x_i)$$

where $P^K = \Psi_{(\hat{\alpha}^K, \hat{\theta}_f)}(P^{K-1})$ and P^0 is a consistent, non-parametric estimate of the true conditional choice probabilities

Hotz-Miller's two step estimator

- The CCP estimators were defined as the values of α that solve systems of equations of the form

$$\arg\min_{\alpha \in \Theta} \sum_{i=1}^{N} \sum_{j=1}^{J} Z_i^j \left[I(a_i = j) - \tilde{P}_{(\alpha, \hat{\theta}_f)}(P^0)(j|x) \right]$$

where Z_i is are vectors of instrumental variables (e.g.) functions of x_i

- Easy to show that the 1-stage PI estimator is a CCP estimator with $Z_i = \partial \Psi(P^0)(a_i|x_i)/\partial\alpha$ is used as instrument.

The Precision of PI Estimators: A Monte Carlo Evidence

TABLE I

MONTE CARLO EXPERIMENT

	Experiment design
Model:	Bus engine replacement model (Rust)
Parameters:	$\theta_0 = 10.47$; $\theta_1 = 0.58$; $\beta = 0.9999$
State space:	201 cells
Number observations:	1000
Number replications:	1000
Initial probabilities:	Kernel estimates

Monte Carlo distribution of MLE
(In parenthesis, percentages over the true value of the parameter)

	θ_0	θ_1
Mean Absolute Error:	2.08 (19.9%)	0.17 (29.0%)
Median Absolute Error:	1.56 (14.9%)	0.13 (22.7%)
Std. dev. estimator:	2.24 (21.4%)	0.16 (26.9%)
Policy iterations (avg.):	6.2	

Monte Carlo distribution of PI estimators (relative to MLE)
(All entries are 100* (K-PI statistic-MLE statistic)/MLE statistic)

		Estimators		
Parameter	Statistics	1-PI	2-PI	3-PI
θ_0	Mean AE	4.7%	1.6%	0.3%
	Median AE	14.2%	0.2%	−0.3%
	Std. dev.	6.8%	1.2%	0.3%
θ_1	Mean AE	18.7%	1.5%	0.2%
	Median AE	25.1%	0.7%	0.6%
	Std. dev.	11.0%	1.3%	0.2%

Statistical properties of K-PI estimator

For any K

- $\hat{\alpha}^K$ is asymptotically equivalent to MLE
- $\hat{\alpha}^K$ is $\sqrt{n}$ consistent
- $\hat{\alpha}^K$ is asymptotic normal with known variance-covariance matrix (A&M has an expression that accounts for first step estimation error)

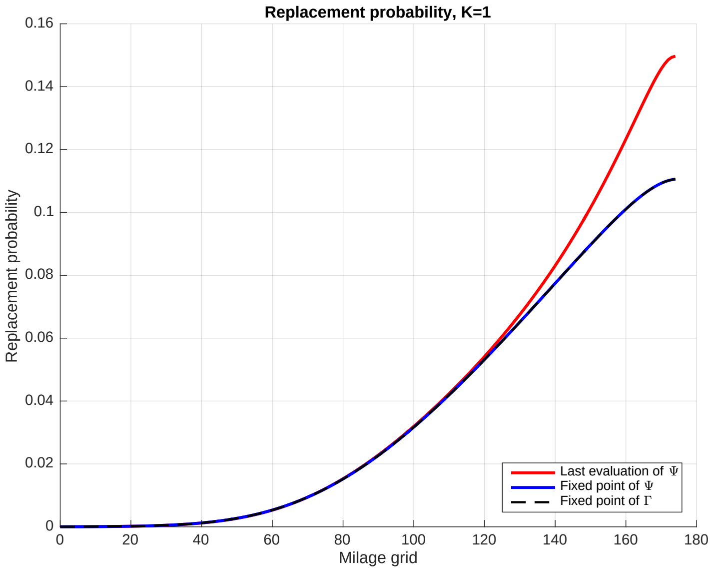
For $K = 1$

- $\hat{\alpha}^K$ encompasses Hotz-Miller (1993) estimator

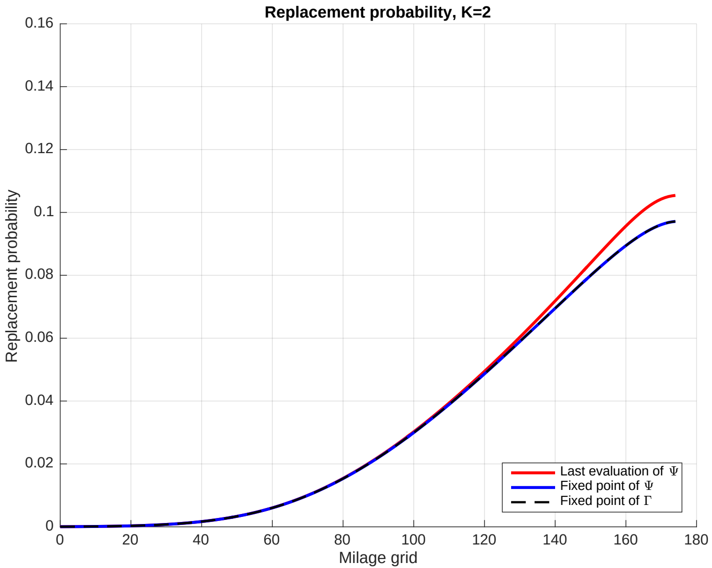
As $K \to \infty$

- $\hat{\alpha}^K$ converges to the MLE estimator obtained by NFXP
- Standard inference.

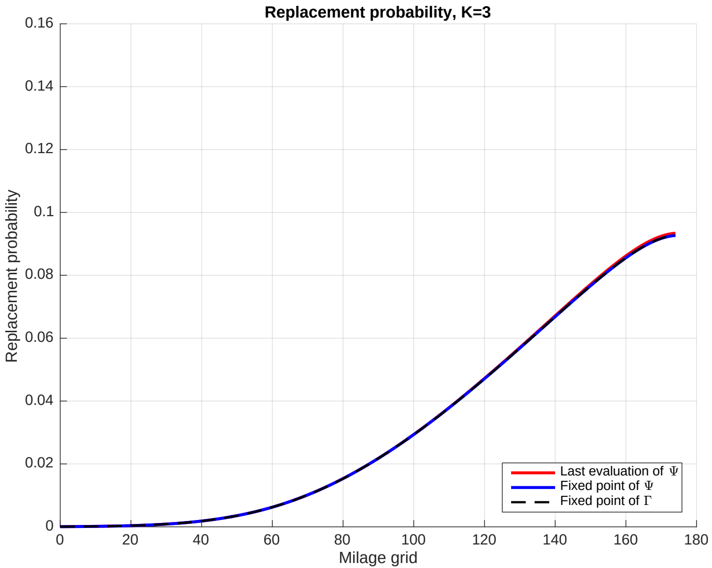
Replacement probability, K=1



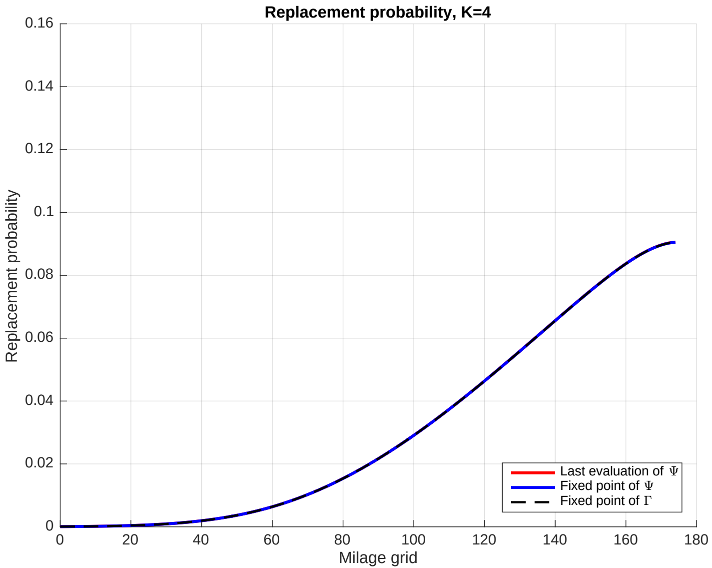
Replacement probability, K=2



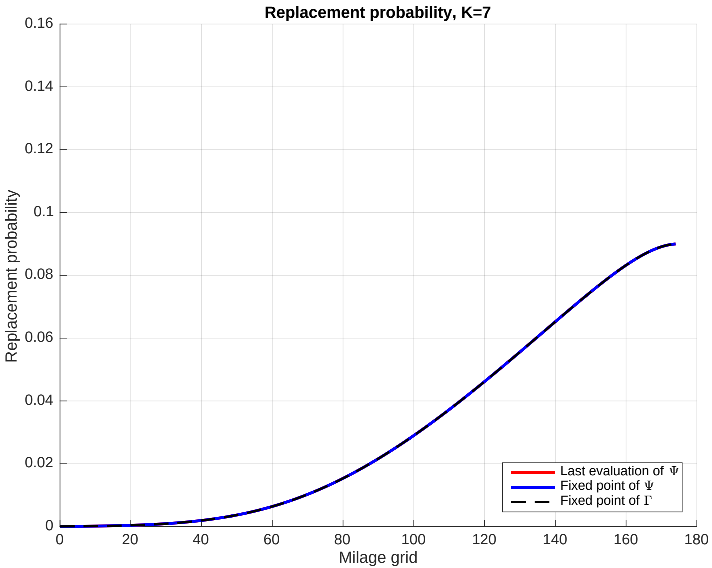
Replacement probability, K=3



Replacement probability, K=4



Replacement probability, MLE



Hotz-Miller's two step estimator

- The CCP estimators were defined as the values of α that solve systems of equations of the form

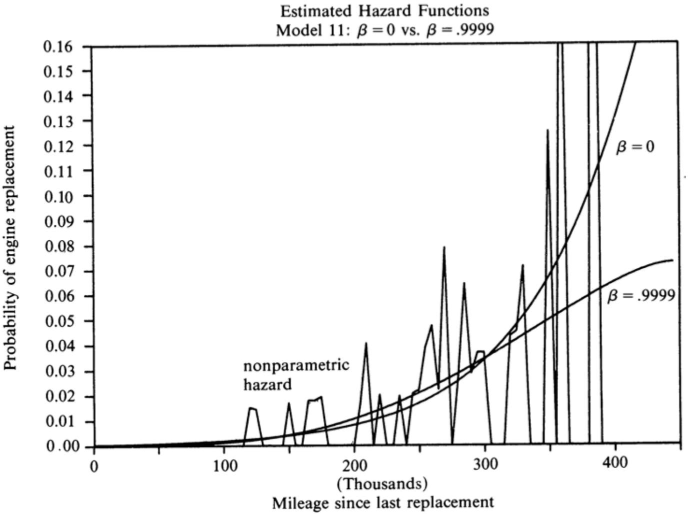
$$\arg\min_{\alpha \in \Theta} \sum_{i=1}^{N} \sum_{j=1}^{J} Z_i^j \left[I(a_i = j) - \tilde{P}_{(\alpha, \hat{\theta}_f)}(P^0)(j|x) \right]$$

where Z_i is are vectors of instrumental variables (e.g.) functions of x_i

- Easy to show that the 1-stage PI estimator is a CCP estimator with $Z_i = \partial \Psi(P^0)(a_i|x_i)/\partial\alpha$ is used as instrument.

Small sample problems

Sometimes it can be hard to get a precise nonparametric estimate of CCP



The Precision of PI Estimators: A Monte Carlo Evidence

TABLE I

MONTE CARLO EXPERIMENT

	Experiment design
Model:	Bus engine replacement model (Rust)
Parameters:	$\theta_0 = 10.47$; $\theta_1 = 0.58$; $\beta = 0.9999$
State space:	201 cells
Number observations:	1000
Number replications:	1000
Initial probabilities:	Kernel estimates

Monte Carlo distribution of MLE
(In parenthesis, percentages over the true value of the parameter)

	θ_0	θ_1
Mean Absolute Error:	2.08 (19.9%)	0.17 (29.0%)
Median Absolute Error:	1.56 (14.9%)	0.13 (22.7%)
Std. dev. estimator:	2.24 (21.4%)	0.16 (26.9%)
Policy iterations (avg.):	6.2	

Monte Carlo distribution of PI estimators (relative to MLE)
(All entries are 100* (K-PI statistic-MLE statistic)/MLE statistic)

		Estimators		
Parameter	Statistics	1-PI	2-PI	3-PI
θ_0	Mean AE	4.7%	1.6%	0.3%
	Median AE	14.2%	0.2%	−0.3%
	Std. dev.	6.8%	1.2%	0.3%
θ_1	Mean AE	18.7%	1.5%	0.2%
	Median AE	25.1%	0.7%	0.6%
	Std. dev.	11.0%	1.3%	0.2%

The Precision of PI Estimators: A Monte Carlo Evidence

TABLE II

RATIO BETWEEN ESTIMATED STANDARD ERRORS AND STANDARD DEVIATION OF MONTE CARLO DISTRIBUTION

Parameters	Statistics	Estimators			
		1-PI	2-PI	3-PI	MLE
θ_0	Ratio	0.801	1.008	1.027	1.022
θ_1	Ratio	0.666	1.043	1.076	1.065

Relative Speed of NPL and NFXP

- For most problems the fixed point iterations (i.e., policy iterations) are much more expensive than likelihood and pseudo-likelihood "hill" climbing iterations.
- The size of the state space does not affect the number of policy iterations in any of the two algorithms.
- Both algorithms were initialized with Hotz-Miller Estimates.
- A&M found that NPL around 5 and 10 times faster than NFXP

Matlab Implementation

NPL (bus data):
run_npl.m

Main Contributions of Bajari, Benkard, Levin (2007)

Two-step algorithm for estimating dynamic games

- In the first step, the policy functions and the law of motion for the state variables are estimated.
- In the second step, the remaining structural parameters are estimated using the optimality conditions for equilibrium.
- The second step estimator is a simple simulated minimum distance estimator.
- The algorithm applies to a broad class of models, including industry competition models with both discrete and continuous controls such as the Ericson and Pakes (1995) model.

Monte Carlo experiments BBL test the algorithm on

- a dynamic discrete choice model with normally distributed errors
- a dynamic oligopoly model similar to that of Pakes and McGuire (1994).

Application of BBL approach to the bus engine problem

- Suppose we knew the choice-specific value functions $v(x, 1, \theta^*)$ (value of replacing a bus with x miles since last engine replacement) and $v(x, 0, \theta^*)$ (value of not replacing the bus engine)
- Optimal decision rule, incorporating unobserved cost shocks $\epsilon = (\epsilon(0), \epsilon(1))$ is $\delta(x, \epsilon, \theta^*)$ given by

$$\delta(x, \epsilon, \theta^*) = \begin{cases} 1 & \text{if } v(x, 1, \theta^*) + \epsilon(1) \leq v(x, 0, \theta^*) + \epsilon(0) \\ 0 & \text{otherwise} \end{cases}$$

- Let $\delta^c(x, \epsilon, \theta^*) = 1 - \delta(x, \epsilon, \theta^*)$. It is another decision rule, but by definition it is a *suboptimal decision rule.* Hence we have for any (x, ϵ)

$$v(x, \delta(x, \epsilon, \theta^*), \theta^*) + \epsilon(\delta(x, \epsilon, \theta^*)) \leq v(x, \delta^c(x, \epsilon, \theta^*), \theta^*) + \epsilon(\delta^c(x, \epsilon, \theta^*))$$

i.e. the value of taking the optimal action is always less than the value of taking a suboptimal action prescribed by $\delta^c(x, \epsilon, \theta^*)$.

Application of BBL approach to the bus engine problem

- Define the function $g(x, \epsilon, \theta)$ by

$$\begin{aligned} g(x, \epsilon, \theta) &= v(x, \delta^c(x, \epsilon, \theta^*), \theta) + \epsilon(\delta^c(x, \epsilon, \theta^*)) \\ &- v(x, \delta(x, \epsilon, \theta^*), \theta) + \epsilon(\delta(x, \epsilon, \theta^*)) \end{aligned}$$

- Notice that at $\theta = \theta^*$ we have $g(x, \epsilon, \theta^*) \geq 0$. If $H(x, \epsilon)$ is some CDF over (x, ϵ) then we have

$$\theta^* \in \underset{\theta}{argmin} \int_x \int_\epsilon (\min[g(x, \epsilon, \theta), 0])^2 \, dH(x, \epsilon) \quad \text{MIE}$$

- Thus, if we can somehow be able to compute/simulate $v(x, 1, \theta)$ and $v(x, 0, \theta)$ for any θ and uncover/estimate and simulate the true optimal decision rule $\delta(x, \epsilon, \theta^*)$ for any value of (x, ϵ) then we can compute $g(x, \epsilon, \theta)$ and then search for the value θ^* that minimizes the criterion in the moment inequality estimator, MIE.

Application of BBL approach to the bus engine problem

- How do we uncover $\delta(x, \epsilon, \theta^*)$ if we do not know θ^*?
- Recall from the Hotz-Miller inversion of the CCPs, if we knew $P(1|x, \theta^*)$ and $P(0|x, \theta^*) = 1 - P(1|x, \theta^*)$ then we have

$$\log(P(1|x, \theta^*)/P(0|x, \theta^*)) = v(x, 1, \theta^*) - v(x, 0, \theta^*)$$

- Then we have

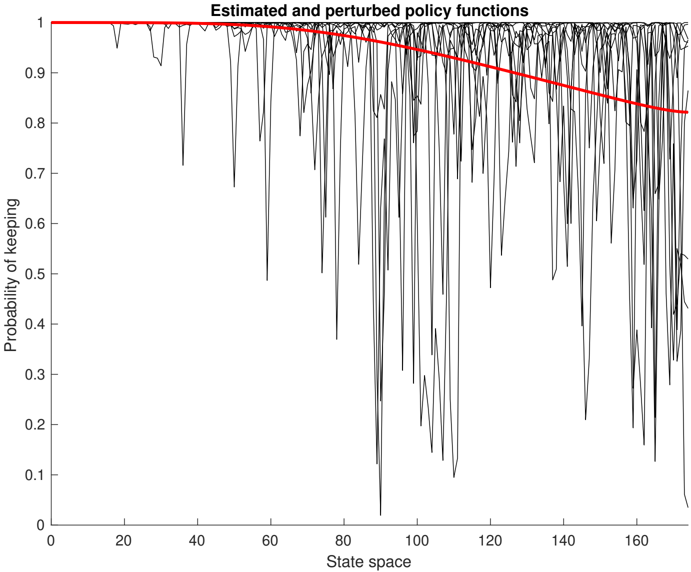
$$\begin{aligned} \delta(x, \epsilon, \theta^*) &= \begin{cases} 1 & \text{if } v(x, 1, \theta^*) - v(x, 0, \theta^*) \leq \epsilon(0) - \epsilon(1) \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } \log(P(1|x, \theta^*)/P(0|x, \theta^*)) \leq \epsilon(0) - \epsilon(1) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

- Thus if we can non-parametrically estimate $P(1|x, \theta^*)$ and simulate draws of $\epsilon = (\epsilon(0), \epsilon(1))$, then we can simulate values of $\delta(x, \epsilon, \theta^*)$ for any (x, ϵ) even though we do not know θ^*.

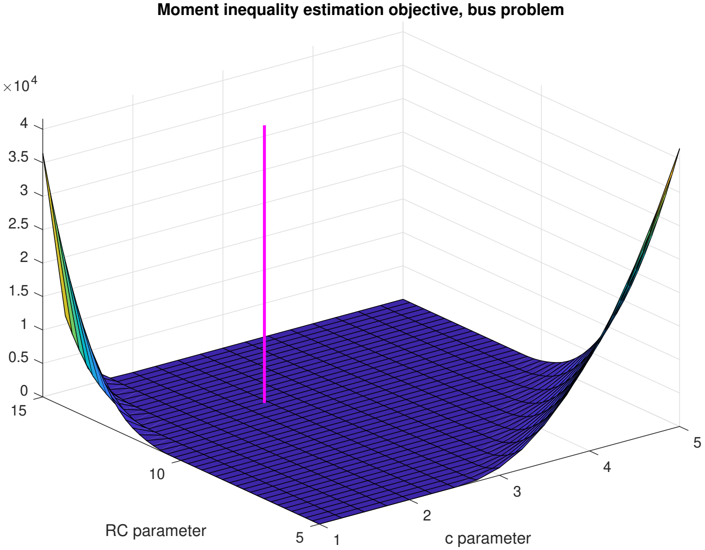
Application of BBL approach to the bus engine problem

- Now suppose we can also solve for $(v(x,1,\theta), v(x,0,\theta))$ for any θ. Then it is easy to see we can evaluate the moment inequality function $g(x,\epsilon,\theta)$ for any θ and any (x,ϵ) and hence we can evaluate the MIE objective function.
- In actual estimation, we cannot exactly recover $P(1|x,\theta^*)$ from data, but only a relatively noisy non-parametric estimate of it, $\hat{P}(1|x,\theta^*)$. This implies we have a consistent but potentially noisy estimate of the optimal decision rule, $\hat{\delta}(x,\epsilon)$ that (hopefully) converges uniformly to the true decision rule $\delta(x,\epsilon,\theta^*)$ as the number of observations N used to estimate $P(1|x,\theta^*)$ tends to infinity.
- We may also have noise due to numerical solution error or simulation error in the value functions $(\hat{v}(x,1,\theta), \hat{v}(x,0,\theta))$. As a result the moment inequality function we actually estimate $\hat{g}(x,\epsilon,\theta)$ has a mix of simulation and estimation errors in it, so it is not necessarily the case that $\hat{g}(x,\epsilon,\theta) \geq 0$ at $\theta = \theta^*$.
- However as $N \to \infty$ we can expect that $\hat{g}(x,\epsilon,\theta^*) \to g(x,\epsilon,\theta^*) \geq 0$ with probability 1, so the moment inequalities will be satisfied at the true parameter values.

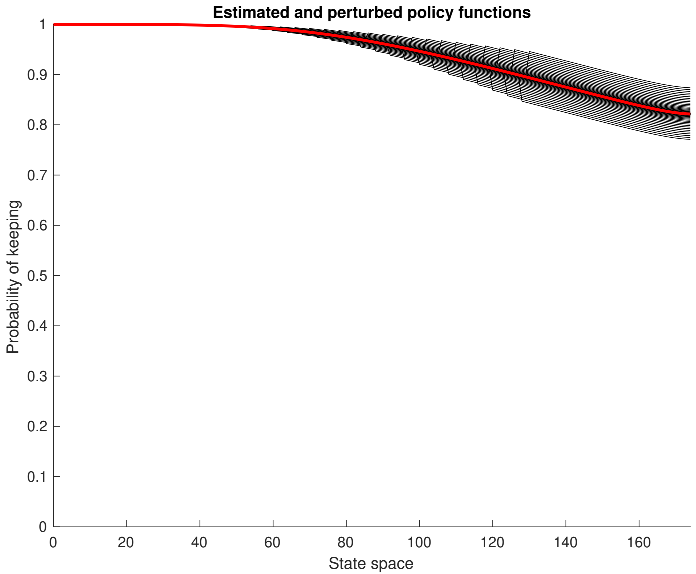
Policy perturbations, bus engine problem



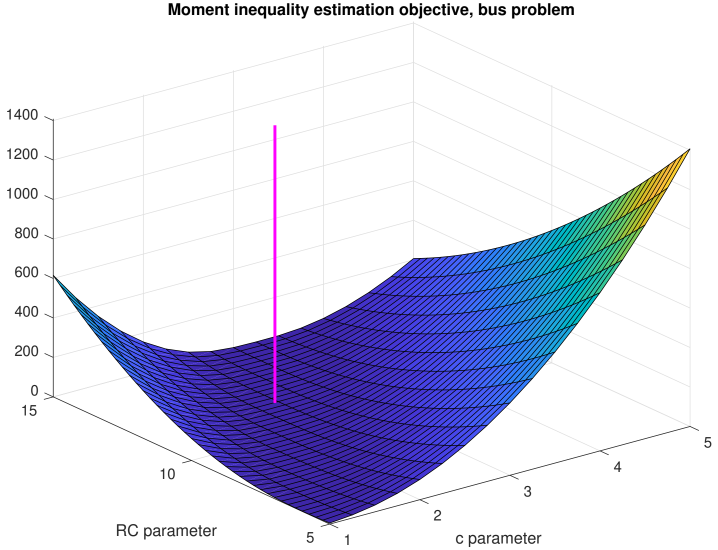
Moment inequality objective, bus engine problem



Policy perturbations, bus engine problem



Moment inequality objective, bus engine problem



Matlab Implementation

BBL illustration:
run_bbl.m
(see switches in lines 15-16)

McFadden's MSM estimator for discrete models

- McFadden (1989) introduced the method of simulated moments (MSM) which can enable consistent estimation of discrete choice models with only 1 simulated utility draw per observation. The basic idea of MSM is that we can use the law of large numbers to average out simulation error in the same way we use the LLN to average out sampling error.
- MSM first introduced for estimation of static discrete choice models with non-extreme valued error terms with high dimensional choice sets
- However people soon realized that the idea of MSM is very general and extends to a much broader class of problem
- Jean-Francois Richard: "If you can simulate it, you can estimate it"
- MSM – "Swiss Army Knife" of structural estimation

McFadden's MSM estimator for discrete models

- Basic idea is very similar to *calibration* used in macroeconomics:
  1. you, as econometrician, determine a vector of *actual moments* characterizing/summarizing the data you observe and want to explain with your model
  2. For any parameter value θ for your structural model, solve the model and "simulate data" from the model to construct a correponding vector of *simulated moments*
  3. Search over θ for a value $\hat{\theta}_{\text{msm}}$ that enables the simulated moments to "best fit" the actual moments
- Actual data: $\{x_t\}$, $t = 1, \ldots, T$ (cross section, panel, or time series)
- Define moments as a $J \times 1$ mapping $h(x_t) \in R^J$. The moments are

$$\overline{h}_T = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$$

Solving and simulating your structural model

- Now, suppose we have a structural model that depends on a vector θ of "structural parameters" we want to estimate
- Can by dynamic/static, singe/multiple agent, discrete/continuous choices and states
- Let $\{\tilde{x}_t^s(\theta)\}$, $t = 1, \ldots, T$, $s = 1, \ldots, S$ be S *IID* sets of *simulated data* from your structural model for the guess of the parameters θ.
- Similar to McFadden's case, we can get by with as few as $S = 1$ simulation of the model for the T observations/time periods.
- Form *simulated moments* $\overline{h}_T^S(\theta)$ as follows

$$\overline{h}_T^S(\theta) = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{T}\sum_{t=1}^{T} h(\tilde{x}_t^s(\theta)).$$

Equations and unknowns

- Let there be K unknown parameters to be estimated, $\theta \in R^K$.
- Must have $J \geq K$ "equations"

$$\overline{h}_T = \overline{h}_T^S(\hat{\theta}_{\text{msm}}).$$

This is the simulation analog of the classical "method of moments" introduced by Pafnuty Chebyshev in 1887.

- Usually it's the *overidentified case* where $J > K$
- So in the general case, MSM mimics that GMM strategy of Sargan/Hansen and we define the MSM estimator as

$$\hat{\theta}_{\text{msm}} = \underset{\theta}{argmin} \left([\overline{h}_T - \overline{h}_T^S(\theta)]' W_T [\overline{h}_T - \overline{h}_T^S(\theta)] \right)$$

where W_T is a $J \times J$ positive semi-definitive weighting matrix.

Key Assumptions for MSM Estimatation

- **Correct specification** There is a value θ^* such that $\{\tilde{x}_t\} \sim \{\tilde{x}_t(\theta^*)\}$, i.e. that simulated data from the structural model at θ^* has the same probability distribution (stochastic process) as the actual data.
- **Identification** If $\theta \neq \theta^*$ then $E\{h(\tilde{x}(\theta)\} \neq E\{h(\tilde{x}(\theta^*))\}$.
- **Differentiability** The gradient $\nabla E\{h(\tilde{x}(\theta)\}$ $(J \times K)$ exists and is a continuous function of θ.
- **Full Rank** The generalized inverse of the asymptotic covariance matrix of the moments Ω, Ω^+, exists and the $K \times K$ matrix Λ given by

$$\Lambda = [\nabla E\{h(\tilde{x}(\theta^*))\}'[\Omega^+]\nabla E\{h(\tilde{x}(\theta^*))\}]$$

  exists and is invertible.
- **Theorem** *Under the assumptions above (and other technical assumptions to guarantee LLN/CLT hold) we have*

$$\sqrt{T}[\hat{\theta}_{\text{msm}} - \theta^*] \Longrightarrow N(0, (1 + 1/S)\Lambda^{-1}).$$

Low cost/penalty for simulating moments

- Notice that even with a single simulation "per observation" ($S = 1$), the MSM is consistent and asymptotically normally distributed and the penalty for using only a single simulation is just a doubling of the asymptotic covariance matrix of $\hat{\theta}_{\text{msm}}$ compared to the case of "exact integration" ($S = \infty$), since $(1 + 1/S) = 2$ when $S = 1$.
- Power of MSM approach: we can estimate models where it is very difficult/impossible to form a likelihood, so it is a flexible/convenient way to handle a host of econometric problems such as
    1. Measurement error
    2. Censoring/attrition/missing data
    3. Endogeneity
- Downside of MSM: its power can enable you to estimate *statistically degenerate models* i.e. models where certain observations have zero probability of occurring under the model due to lack of sufficient flexibility, no unobserved shocks, etc

Example: MSM estimation of the bus replacement problem

- The data: T buses, N_i observed pairs of mileage and replacement decisions

$$\{x_{it}, d_{it}\}_{i=1,..,T;t=1,..,N_i}$$

- **Moments**
  - Consider $J+1$ *mileage bins* defined by mileage cutoffs $(c_0, c_1, c_2, \dots, c_{J+1})$ where $c_0 = 0$ and $c_{J+1} =$ maximum mileage
  - J moments defined as fractions of buses in each but one of the mileage bins in the long run (stationary disctribution)

$$\bar{h}_j = \frac{1}{T}\sum_{i=1}^{T}\Big[\frac{1}{N_i}\sum_{t=1}^{N_i} 1\{c_{j-1} \le x_{it} \le c_j\}\Big]$$

  - In the example we can also use the stationary distribution over the mileage that can be computed from the solution of the model (mileage transition process + replacement choice probabilities)

Example: MSM estimation of the bus replacement problem

- **Simulated moments**
  1. Solve the model to compute choice and transition probabilities for give parameter θ
  2. Simulate mileage $\tilde{x}_{it}(\theta)$ and replacements $\tilde{d}_{it}(\theta)$ for N buses and T periods using a collection of IID random numbers *with fixed seed* to avoid simulation noise during the estimation
  3. Compute simulated moments

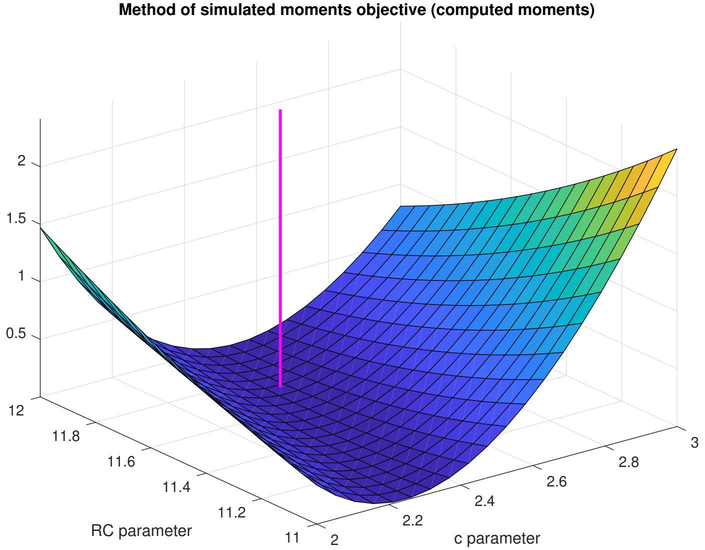
$$\bar{h}_j^S(\theta) = \frac{1}{T}\sum_{i=1}^{T}\left[\frac{1}{N_i}\sum_{t=1}^{N_i} 1\{c_{j-1} \leq \tilde{x}_{it}(\theta) \leq c_j\}\right]$$

- The MSM criterion

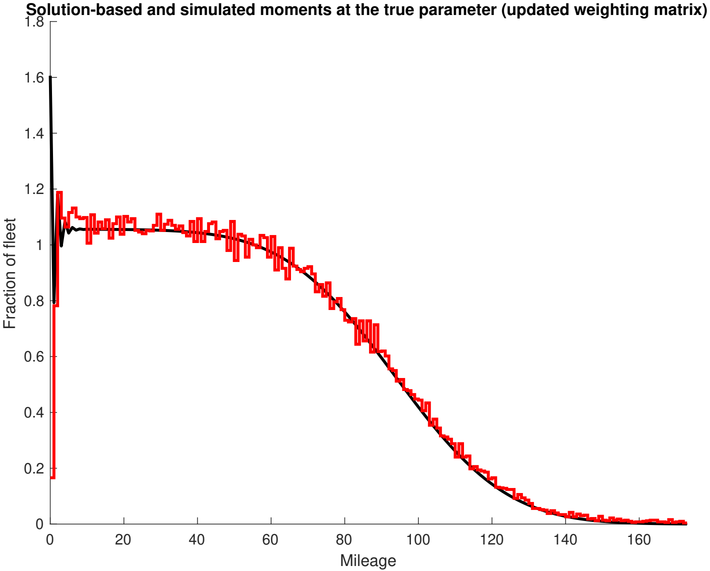
$$Q(\theta) = g(\theta)Wg(\theta)', \quad g(\theta) = \bar{h} - \bar{h}^S(\theta)$$

$g(\theta)$ is a 1 by J vector
W is J by J weighting matrix

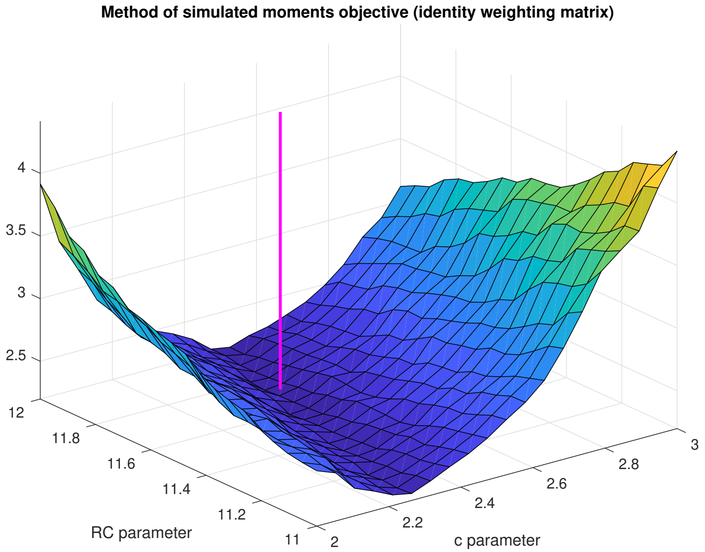
MSM criterion, predicted moments, bus engine problem



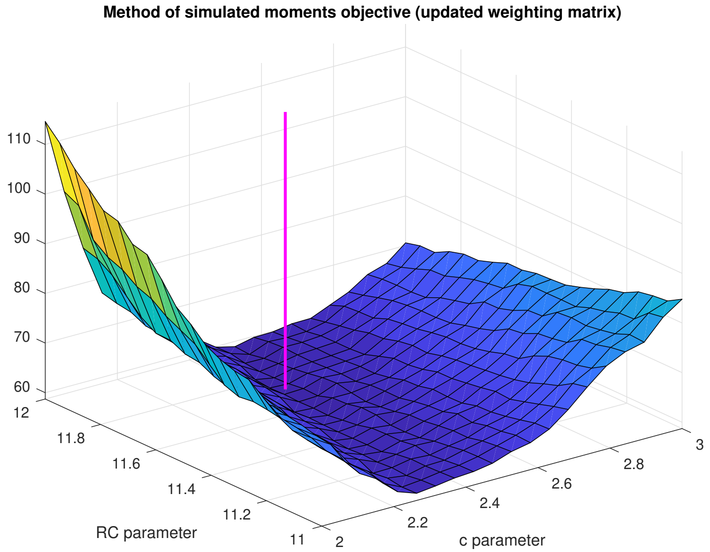
Data and simulated moments, bus engine problem



MSM criterion, identity weighting matrix



MSM criterion, updated weighting matrix



Matlab Implementation

MSM illustration:
run_msm.m
(see switches in lines 13-15)