Problem Set 4 ECON8853*

Joe Jourden, Federico Rodari, and Tuo Yang

December 20, 2022

1. Write down the sequence problem for the firm

Let $\varepsilon_t = (\varepsilon_t(0), \varepsilon_t(1))$. The firm's problem can be written as

$$V_t(a_t, \varepsilon_t) = \sup_{\{f_t, f_{t+1}, \dots\}} \mathbb{E}\left\{\sum_{\tau=t}^{\infty} \beta^{\tau-t} u(a_\tau, \varepsilon_\tau, f_\tau, \mu) \mid a_t, \varepsilon_t\right\}$$

where f_t is an action rule that is a function of the entire history of the process,

$$i_t = f_t(a_t, i_{t-1}, a_{t-1}, i_{t-2}, a_{t-2}, \dots),$$

expectation is taken with respect to $\{a_t\}$, and utility is defined by

$$u(a_t, \varepsilon_t, i_t, \mu) = \begin{cases} -\mu a_t + \varepsilon_t(0), & \text{if } i_t = 0 \\ -R + \varepsilon_t(1), & \text{if } i_t = 1. \end{cases}$$

2. Write down Bellman's equation for the value function of the firm

In this case, we do not need to specify a probability transition matrix since it is degenerate. Conditional on the decision, we will have a deterministic Bellman equation (here we must rely on conditional value functions and cannot use Rust's result of action dimensionality reduction for which $\forall x,\ \text{EV}(x, 1) = \text{EV}(0, 0)$, since no action value is nested in any other action)

$$V(a_t) = \begin{cases} -\mu a_t + \varepsilon_t(0) + \beta V(\min\{5, a_t + 1\}), & \text{if } i_t = 0 \\ -R + \varepsilon_t(1) + \beta V(1), & \text{if } i_t = 1. \end{cases}$$

Given the conditional deterministic nature of the state dynamics, the expected value function in the Bellman's equation is itself deterministic, though we can still compute it using degenerate transition matrices.

3. Contraction mapping

We are looking for a stationary (time independent) policy/value function, and to do so we can show that the optimal conditional value function is the fixed point of a specific contraction mapping. Solving

*We have benefited from the following slides, and from different Github repositories that replicated Rust 1987 (though as mentioned later this problem cannot be exactly formulated as in the original paper).

for the fixed points allows to generate forward-looking probabilities. In doing so, we compute the non-myopic expected value of the agent for each possible decision and each possible state of the machine. The good thing is that we can find a closed form expression for the contraction mapping. The general idea starts from writing the Bellman's equation as

$$V(a,\varepsilon) = \max_{i \in I(a)} \left\{ u(a,i) + \beta \int \underbrace{\left[\int V(a',\varepsilon') q(\varepsilon' \mid a') d\varepsilon' \right]}_{V(a')} f(a' \mid a,i) da' + \varepsilon(i) \right\}$$

where

$$v(a,i) = u(a,i) + \beta \int V(a') f(a' \mid a,i) da'$$

is the conditional value function (each column is the value function associated with choice i). So far we have exploited additive separability in the utility function and the conditional independence assumption from Rust (1987) to integrate out the unobserved state ε. We can now use the property of the expectation of an extreme value random variable[1] to get rid of the maximization and to have the Bellman in closed form. In other words, since

$$V(a) = \int \max_i \{v(a,i) + \varepsilon(i)\} \, q(\varepsilon \mid a) \mathrm{d}\varepsilon$$

we can show that

$$V(a) = \gamma + \log \left(\sum_{i'} \exp v(a,i') \right)$$

which implies

$$v(a,i) = u(a,i) + \beta \int \gamma + \log \left(\sum_{i'} \exp v(a',i') \right) f(a' \mid a,i) \mathrm{d}a'$$

where $f(a' \mid a,i)$ can be represented by sparse and degenerate transition probabilities. In other words, the density is a Dirac delta function. The conditional transition probabilities are

$$p(a' \mid i=0,a) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \!\! \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \quad p(a' \mid i=1,a) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \!\! \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

The results we get is a 5×2 value function that solves the functional equation of the Bellman.

- *Suppose $a_t = 2$. For what value of $\varepsilon_{0t} - \varepsilon_{1t}$ is the firm indiffernet between replacing its machine and not?* We are looking for the following condition

$$V(a_t = 2, i = 0) + \varepsilon_{0t} = V(a_t = 2, i = 1) + \varepsilon_{1t}$$

[1] The expectation of the maximum of T1EV random variables is itself a T1EV random variable with appropriate scale parameter.

which implies

$$\varepsilon_{0t} - \varepsilon_{1t} = V(a_t = 2, i = 1) - V(a_t = 2, i = 0)$$

so, given our optimal value function, $\varepsilon_{0t} - \varepsilon_{1t} = 0.1145$

- *Suppose $a_t = 2$. What is the probability that this firm will replace its machine?* Using extreme value distributed errors, we can again derive a closed form expression for the conditional choice probabilities

$$P(i = 1 \mid a = 2) = \frac{\exp(v(a = 2, i = 1))}{\exp(v(a = 2, i = 0)) + \exp(v(a = 2, i = 1))} = 0.528591$$

- *What is the value of a firm at state $a_t = 4, \varepsilon_{0,t} = 1, \varepsilon_{1,t} = 1.5$?* The probability is

$$p(i_t = 1 \mid a_t = 4)\left[v(a_t = 4, i_t = 1) + \varepsilon(i_t = 1)\right] +$$
$$p(i_t = 0 \mid a_t = 4)\left[v(a_t = 4, i_t = 0) + \varepsilon(i_t = 0)\right] = -10.15665$$

4. Simulate data

Note that here the decision rule is not deterministic due to the unobserved states, i.e. for the same age of two different machines we can observe different actions. What reason why this happens is due to a wedge between the agent's information set and the econometrician's one: conditioning only on the observed state a, the econometrician is not fully able to fully recover the deterministic policy function of the agent. We simulate the data according to the following algorithm

Algorithm 1 Data Generating Process

Require: N observations, $1 \leq j \leq N$, θ parameter vector
Require: V^* optimal value function of the DP problem $\mid \theta$ (max$(a) \times \mid i \mid$)
Require: a vector of all possible states (max$(a) \times 1$)
Require: $\varepsilon \sim$ Gumbel$(0, 1)$, $(N \times \mid i \mid)$
Require: $S \sim$ Unif$\{1, \max(a)\}$ random starting states $(N \times 1)$
  if $V(s_j, i = 1) + \varepsilon_{1j} > V(s_j, i = 0) + \varepsilon_{0j}$ **then**
    $1 \leftarrow A_j$ (replace)
  else
    $0 \leftarrow A_j$ (do not replace)
  end if
  $S'_j = \min((S_j + 1) \cdot \mathbb{1}(A_j = 0) + 1 \cdot \mathbb{1}(A_j = 1), \max(a))$ (next state)

In simple words, following Rust (1987) assumptions, each row is an independent observation representing machine facing (1) a given state, (2) a decision (conditional on both observed and unobserved state) of either replacing the machine or not, (3) the next state conditional on the decision.
The data look like this

S	A	S'
1	0	2
3	1	1
4	1	1
3	0	4

5. Implement Rust's NFP approach

The nested fixed point algorithm consists of an inner and an outer loop. The inner loop solves the contraction mapping $v(a, i, \theta^c) = T(v(a, i, \theta^c))$ for a candidate parameter vector and computes the conditional choice probabilities, while the outer loop searches over the parameter space for the vector that maximizes the partial loglikelihood computed using the conditional choice probabilities.
Given the random seed (2022) and initial candidate vector $\theta^c = [0, 0]$ the results we obtain are $\hat{\theta} = [0.9963, 2.993]$, compared to $\theta = [1, 3]$ in the data generating process.

A. Coding visuals

How to see the unconditional value function in the code (numbers are states (rows) and actions (columns))

$$V\begin{pmatrix}1\\2\\3\\4\\5\end{pmatrix} = \begin{pmatrix}\gamma\\\gamma\\\gamma\\\gamma\\\gamma\end{pmatrix} + \log\left[\exp v\begin{pmatrix}1,0\\2,0\\3,0\\4,0\\5,0\end{pmatrix} + \exp v\begin{pmatrix}1,1\\2,1\\3,1\\4,1\\5,1\end{pmatrix}\right]$$

where $v(\cdot)$ is the candidate solution fed to the algorithm.
How to see the updating of the conditional value function in the code

$$v(a,i) = u\begin{pmatrix}1,0 & 1,1\\2,0 & 2,1\\3,0 & 3,1\\4,0 & 4,1\\5,0 & 5,1\end{pmatrix} + \beta V\begin{pmatrix}1\\2\\3\\4\\5\end{pmatrix}\left[\begin{pmatrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\0&0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\end{pmatrix}\right]$$

where the second component shows how the value function evaluated at the next state is computed.