Assignment 2

Luke Motley, Jeffrey Ohl

2/20/25

Exercise 1

Question 1

We first generate plots of the implied true conditional choice probabilities of maintaining the engine for each mileage state x_t under different parameter values of $(\beta, \theta_1, \theta_2)$.

Figure 1 shows the probability $\Pr(\text{maintain} \mid x)$ for several parameter combinations. As seen in the figure:

- **Varying β:** As β increases (i.e., the agent discounts the future less), the probability of maintenance decreases at all mileage levels.
- **Varying θ_1:** If θ_1 is more negative, so maintenance costs are higher, the probability of maintenance decreases at all mileage levels.
- **Varying θ_2:** Higher θ_2 (higher replacement cost) raises the probability of maintenance at all mileage levels. This is obvious: more costly replacement makes maintenance more attractive.

Overall, the patterns match economic intuition.

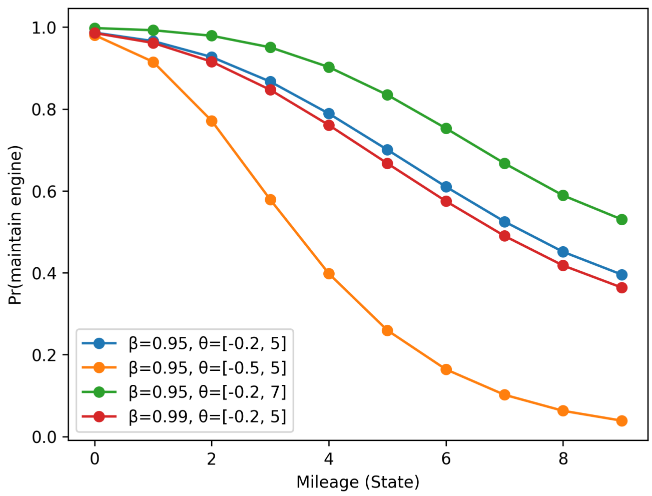

Figure 1: Probability of maintaining vs. mileage, under different $(\beta, \theta_1, \theta_2)$.

Question 2

The key idea is that under T1EV errors, the ratio of the CCPs identifies the difference in choice-specific value functions:

$$\ln\left(\frac{\Pr(a = j \mid x)}{\Pr(a = J \mid x)}\right) = \overline{v}_j(x) - \overline{v}_J(x).$$

With a reference choice J normalized to zero, we recover $\overline{v}_j(x)$ from the sample analog $\widehat{\Pr}(a = j \mid x)$. The code calculates these inverted value functions and then sets up the (partial) likelihood for the structural parameters θ_1 and θ_2.

Question 3

We simulate data under "true" parameters (e.g. $\theta_1 = -0.25$, $\theta_2 = 5$, $\beta = 0.95$). We estimate $\theta = (\theta_1, \theta_2)$ by maximizing the two-step Hotz–Miller likelihood. Figure 3 compares the true CCP and the estimated CCP from the two-step procedure. The estimated parameters are close to the true values, and the estimated CCP curve aligns well with the true CCP curve across the mileage states. However, there does appear to be downward bias in the probability of maintenance at higher mileage states. This is consistent with the data though.

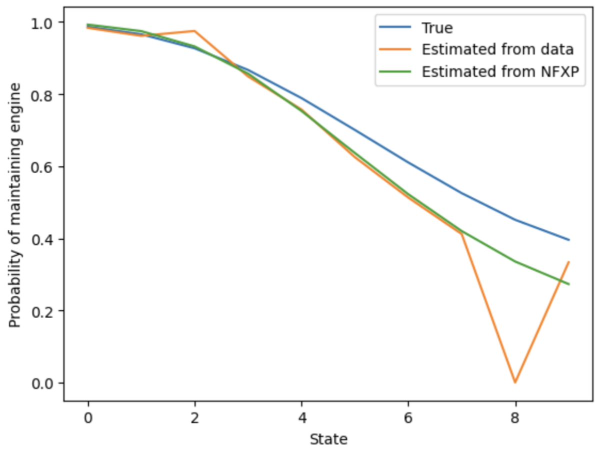

Figure 2: True vs NFXP vs data

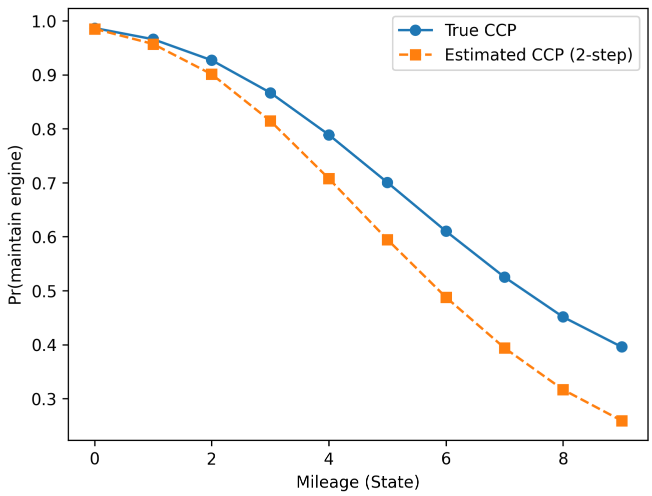

Figure 3: True CCP vs. estimated CCP (two-step). Parameter estimate: $(\theta_1, \theta_2) \approx (-0.258, 4.998)$.

Question 4

We repeat the estimation on the same simulated data using:

1. **Nested Fixed-Point (NFXP):** The classical approach that repeatedly solves the Bellman equation in an inner loop while searching over (θ_1, θ_2) in the outer loop.
2. **Nested Pseudo-Likelihood (NPL):** An iterative procedure that updates CCPs and parameters in turn, often faster than full NFXP if well-implemented.
3. **MPEC (Mathematical Programming with Equilibrium Constraints):** A constrained optimization approach that imposes the Bellman fixed-point as constraints and maximizes the likelihood subject to those constraints.

We time each method. Below is the output (with parameter estimates and runtimes). HOWEVER, NOTE THAT THE NFXP RUNTIME IS AFTER WE HAVE STORED THE FIXED POINT. The first time we ran NFXP, it took over param$_c cp!(nochunking)$

```
True theta: [-0.25  5.  ]
Beta: 0.95
NFXP estimated theta: [-0.27103871  5.8225678 ]
NFXP runtime: 8.1379 seconds (BUT REALLY TOOK OVER TWO HOURS)

NPL estimated theta: [-0.29438793  5.38765412]
NPL runtime: 94.7935 seconds

MPEC estimated theta: [-0.34869353  5.43107649]
MPEC runtime: 119.3358 seconds
```

Results

- All three methods yield similar estimates, which are close to the true values.
- The differences in run times is stark. NPL is the best in this specific case, though MPEC is not far behind and would likely be faster for problems that have nice constraints or in cases when the state space is large so it takes long for NPL to converge.
- In practice, I guess we balance the computational cost with implementation complexity; NPL and the two-step CCP approach can be more scalable in certain applications, while NFXP remains a benchmark method, and MPEC can be elegant with powerful solvers.

Exercise 2

2a

We just look at each time that it hits a local max and assume it's replaced then. To be precise, if $x_{t-1} < x_t$ we will say the engine was replaced at t.

2b

Conditional independence: Given $(x_t, d_t), \epsilon_t$ does not influence the realization of x_{t+1}

i.e. we can express transition probabilities as a product $p(x_{t+1}, \epsilon_{t+1}|x_t, \epsilon_t, d_t) = \rho(x_{t+1}|x_t, d_t)g(\epsilon_{t+1})$

2c

See code notebook. We choose $K = 10$ for chunking.

```
(array([[0.43846657, 0.56000871, 0.00152472, 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ],
        [0.        , 0.43846657, 0.56000871, 0.00152472, 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.43846657, 0.56000871, 0.00152472,
         0.        , 0.        , 0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.43846657, 0.56000871,
         0.00152472, 0.        , 0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.43846657,
         0.56000871, 0.00152472, 0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.43846657, 0.56000871, 0.00152472, 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.43846657, 0.56000871, 0.00152472, 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.43846657, 0.56000871, 0.00152472],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.43846657, 0.56153343],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 1.        ]]),
 array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]]))
```

Figure 4: First matrix is transition matrix when $a = 0$ (maintain). Second matrix is transition matrix when $a = 1$ (repair). Clearly, we always go back to the lowest-milage state for the $a = 1$ matrix.

2d

We want to show that the function

$$\mathrm{EV}(x,d) \;=\; \int \ln\Bigl(\sum_j \exp\bigl[u(y,j)+\beta\,\mathrm{EV}(y,j)\bigr]\Bigr)\,p(\mathrm{d}y \mid x,d)$$

arises naturally as the fixed-point equation in a dynamic discrete choice problem with T1EV errors.

Once Harold chooses an action d in state x, the next state y is drawn according to $p(\mathrm{d}y \mid x,d)$. Then, given y, Harold chooses the action j that maximizes

$$u(y,j)+\beta\,\mathrm{EV}(y,j)+\epsilon_j,$$

where $\{\epsilon_j\}$ are i.i.d. Type-1 Extreme Value (T1EV) shocks. Hence, we can write

$$\mathrm{EV}(x,d) \;=\; \mathbb{E}\Bigl[\max_j \bigl\{u(y,j)+\beta\,\mathrm{EV}(y,j)+\epsilon_j\bigr\} \Bigm| x,d\Bigr].$$

Here, $\mathbb{E}[\cdot \mid x,d]$ is the expectation over both the random state y and the shocks $\{\epsilon_j\}$.

Recall the classic T1EV result:

$$\mathbb{E}\Bigl[\max_j\,(v_j+\epsilon_j)\Bigr] \;=\; \gamma \;+\; \ln\Bigl(\sum_j e^{v_j}\Bigr),$$

where γ is the Euler–Mascheroni constant. In many treatments, γ is absorbed into an overall constant term and does not affect choice probabilities, so it is often omitted.

Applying the above formula with $v_j = u(y,j)+\beta\,\mathrm{EV}(y,j)$, we have

$$\mathbb{E}\Bigl[\max_j \bigl\{u(y,j)+\beta\,\mathrm{EV}(y,j)+\epsilon_j\bigr\}\Bigr] \;=\; \ln\Bigl(\sum_j e^{\,u(y,j)+\beta\,\mathrm{EV}(y,j)}\Bigr) \;+\; (\text{constant}).$$

Hence,

$$\mathrm{EV}(x,d) \;=\; \int \ln\Bigl(\sum_j \exp\bigl[u(y,j)+\beta\,\mathrm{EV}(y,j)\bigr]\Bigr)\,p(\mathrm{d}y \mid x,d) \;+\; (\text{constant}).$$

Dropping or absorbing the constant term (since it does not affect optimal choices), we obtain the key fixed-point equation:

$$\mathrm{EV}(x,d) \;=\; \int \ln\Bigl(\sum_j \exp\bigl[u(y,j)+\beta\,\mathrm{EV}(y,j)\bigr]\Bigr)\,p(\mathrm{d}y \mid x,d).$$

In class we showed that that the mapping

$$\mathrm{EV} \;\mapsto\; \mathrm{EV}$$

is a contraction, hence a fixed point exists and is unique.

2e

We know following from the derivations from class that

$Pr(a_t|x_t,\theta) = \frac{\exp[u(x_{it},a_{it},\theta)+\beta E(v(x_{it},a_t))]}{\sum_{a' \in A}\exp[u(x_{it},a',\theta)+\beta E(v(x_{it},a'))]}$

2f

The process has the regenerative property, because we only need to track the state (miles) since the last replacement.
This means that it is not truly an infinite time horizon problem.
We can combine (i) our choice from part 3 to discretize the region into K chunks, with the highest chunk being informed by $\hat{X}$, the highest mileage we observe in the data, and (ii) this observation, to note there are effectively only K mileage states, and 2 action states for d. Hence we only had 2K states for EV.

2g

Let:

$$x \in \{1,\dots,K\}, \quad d \in \{1,\dots,D\},$$

and define:

$$\mathbf{EV} \in \mathbb{R}^{K\cdot D},$$

to be the stacked vector of values $EV(x,d)$. Let

$$\mathbf{P} \in \mathbb{R}^{(K\cdot D)\times K} \quad \text{with} \quad \mathbf{P}_{(x,d),\,y} = p\big(y \mid x,d\big).$$

Next, define the function $\Gamma : \mathbb{R}^{K\cdot D} \to \mathbb{R}^K$ by

$$\Gamma(\mathbf{EV})_y \;=\; \ln\Bigg(\sum_{j=1}^{D} \exp\big(u(y,j) + \beta\, EV(y,j)\big)\Bigg), \quad y = 1,\dots,K.$$

Then, the Bellman/log–sum–exp fixed–point equation

$$EV(x,d) \;=\; \sum_{y=1}^{K} p(y \;\mid\; x,d)\ln\Big(\sum_{j=1}^{D} \exp\big(u(y,j) + \beta\, EV(y,j)\big)\Big),$$

can be written in matrix form as:

$$\mathbf{EV} \;=\; \mathbf{P}\,\Gamma(\mathbf{EV}).$$

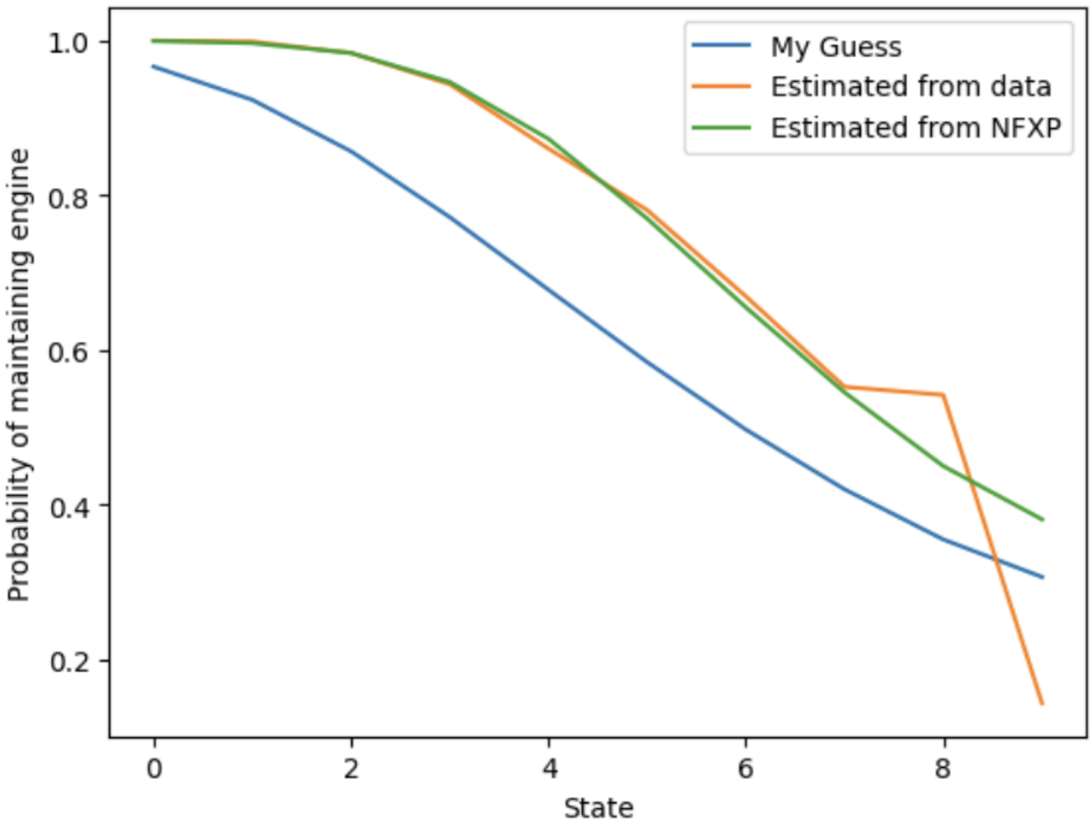

Figure 5: Estimated Parameters from NFXP vs. data. "My guess' was from a few rounds of guessing and checking. This shows we get a much better fit from NFXP than merely guessing and checking.

2h

See function `vfi` in code.

2i

See function `likelihood_nfxp` in code.

2j

We maximize the function in 2i and get the following:

$$\hat{\theta}_{nfxp} = \begin{pmatrix} 0.234 \\ 1.44 \\ 7.44 \end{pmatrix}$$

Hence, the linear portion of the cost is parameterized by 0.23, the curvature, in miles^2, by $\frac{1.44}{100^2}$, and the one-time replacement cost is 7.44. We plot how well this fits the data in Figure 5.

Exercise 3

Question 1

For $J = 2$, denote the entry decisions by $y_{1t}, y_{2t} \in \{0, 1\}$. If a firm does not enter, its profit is normalized to zero. When firm j enters, its profit depends on whether the other firm enters:

- For firm 1:

$$\pi_{1t}(0) = x_{1t}\beta - \phi_1 + \epsilon_{1t} \quad \text{(if firm 2 does not enter)},$$
$$\pi_{1t}(1) = x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t} \quad \text{(if firm 2 enters)}.$$

- For firm 2:

$$\pi_{2t}(0) = x_{2t}\beta - \phi_2 + \epsilon_{2t},$$
$$\pi_{2t}(1) = x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t}.$$

A pure strategy Nash equilibrium (NE) must satisfy, for each firm, that given the other firm's decision, its own action is optimal. For instance:

- **Equilibrium** $(0,0)$**:** Both firms choose not to enter if

$$0 \geq x_{1t}\beta - \phi_1 + \epsilon_{1t},$$
$$0 \geq x_{2t}\beta - \phi_2 + \epsilon_{2t}.$$

- **Equilibrium** $(1,0)$**:** Firm 1 enters and firm 2 does not if

$$x_{1t}\beta - \phi_1 + \epsilon_{1t} \geq 0,$$
$$0 \geq x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t}.$$

- **Equilibrium** $(0,1)$**:** Firm 2 enters and firm 1 does not if

$$0 \geq x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t},$$
$$x_{2t}\beta - \phi_2 + \epsilon_{2t} \geq 0.$$

- **Equilibrium** $(1,1)$**:** Both enter if

$$x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t} \geq 0,$$
$$x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t} \geq 0.$$

Depending on the realizations of $(\epsilon_{1t}, \epsilon_{2t})$, there may be a unique equilibrium or multiple equilibria (for instance, when the shocks are such that a firm is indifferent). Specifically, that if an ϵ_{jt} falls in the intermediate interval between the two thresholds, the firm's best response may depend on its beliefs about the other's action. This can lead to regions of multiple Nash equilibria.

Question 2

Suppose we observe market-level data $\{y_{1t}, y_{2t}, x_{1t}, x_{2t}\}_{t=1}^T$, where $y_{jt} = 1$ if firm j enters. The likelihood is written as

$$L(\beta, \delta_1, \delta_2) = \prod_{t=1}^{T} l(y_{1t}, y_{2t} \mid x_{1t}, x_{2t}, \beta, \delta_1, \delta_2).$$

However, because the game is simultaneous and multiple Nash equilibria may occur for some realizations of the shocks, we cannot map shocks to unique outcomes. Thus, the likelihood is not well-defined.

Question 3

Now assume entry is sequential with firm 1 moving before firm 2. In this case,we think of Subgame Perfect Nash Equilibrium (SPNE). Firm 1 chooses its action anticipating firm 2's best response. Under standard conditions, the outcome is unique. Consequently, the likelihood

$$L(\beta, \delta_1, \delta_2) = \prod_{t=1}^{T} l(y_{1t}, y_{2t} \mid x_{1t}, x_{2t}, \beta, \delta_1, \delta_2)$$

is well-defined. There is a unique equilibrium outcome in each market given the observed data.

Question 4

Now consider market-level data where, for each market t, we observe only the number of entrants n_t and a market characteristic x_t. Assume firms are homogeneous and the profit function is

$$\pi_t(n_t) = x_t\beta - \phi - \delta \log(n_t) + \epsilon_t,$$

with $\epsilon_t \overset{i.i.d.}{\sim} N(0, 1)$.

(a)

In a sequential entry game, firms enter until the marginal entrant earns nonnegative profit. Let n_t denote the equilibrium number of entrants. Then the following conditions must hold:

- The n_t^{th} entrant earns nonnegative profit:

$$x_t\beta - \phi - \delta \log(n_t) + \epsilon_t \geq 0.$$

- If one more firm were to enter, its profit would be negative:

$$x_t\beta - \phi - \delta \log(n_t + 1) + \epsilon_t < 0.$$

These conditions determine the unique equilibrium.

(b)

Given the normality assumption, the probability that market t observes exactly n_t entrants is the probability that the common shock ϵ_t falls between the thresholds defined by the equilibrium conditions:

$$P(n_t \mid x_t, \beta, \phi, \delta) = \Phi\Big(\phi + \delta \log(n_t + 1) - x_t\beta\Big) - \Phi\Big(\phi + \delta \log(n_t) - x_t\beta\Big),$$

The likelihood over T markets is then:

$$L(\beta, \phi, \delta) = \prod_{t=1}^{T} \Big[\Phi\Big(\phi + \delta \log(n_t + 1) - x_t\beta\Big) - \Phi\Big(\phi + \delta \log(n_t) - x_t\beta\Big)\Big].$$

Implementing in R, the results are

```
beta = 2.152689
phi  = 1.571236
delta = 10.60228
```

This means that firms initially need to overcome a cost of 1.57 for entry to be possible. A δ of 10.6 means that additional entrants have a large negative effect on profits. This indicates competitive pressure. We can interpret β as a coefficient on market "attractiveness" (not including the number of competitors), where larger β implies more sensitivity to these conditions.