Zurcher as a Q-learner

Wending Liu

Chien-Hsiang Yeh

Shu Hu

Research School of Economics
Australian National University

Motivation

- How to analyze the dynamic choices of agents with data?
  - Zurcher is a bus manager,
  - observes mileage x_t,
  - chooses between ordinary maintenance ($d_t = 0$) and overhaul/engine replacement ($d_t = 1$),
  - minimizes infinite-horizon discounted costs

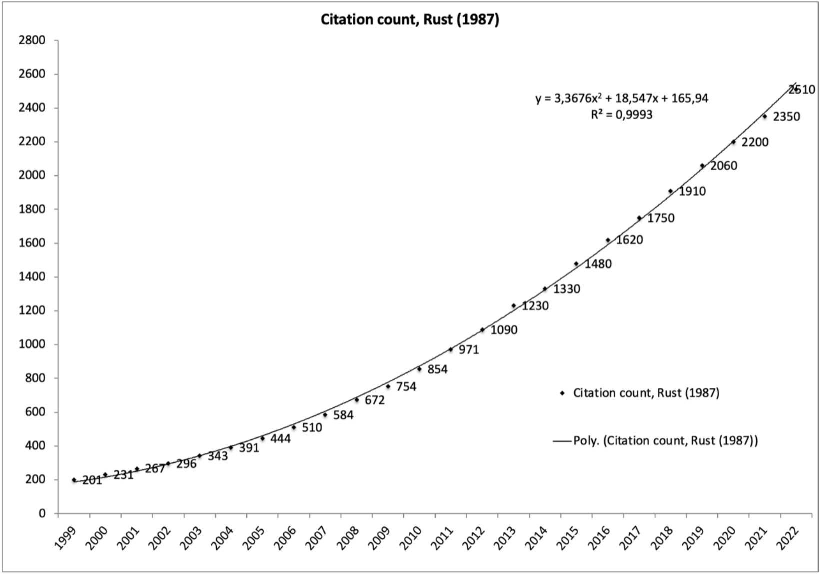
$$C(x) := \min_{\{d_t\}_{t \geq 0}} \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t c_t \middle| x_0 = x \right] \qquad (x \in \mathsf{X})$$

$c_t := c(x_t, d_t) + e_t$: cost of engine maintenance or replacement, with unobserved shocks e_t

Motivation

- Observe $\{(x_t, d_t)\}_{t\geq 0}$, how to predict Zurcher's behavior?
- What are costs? How to take expectations?
  - parameterize: $c(x, d, \theta), P(x'|x, d, \theta)$
  - How to estimate the primitives from data?
- Rust (1987) proposes nested fixed point algorithms
  - given parameters, solve DP
  - obtain closed forms of choice probability functions
  - maximize likelihood function, with observed choice

The Importance of Rust (1987)



Assumptions in Rust (1987)

- Zurcher's decisions coincide with a solution of DP
- Zurcher has perfect information on cost and state transition
- e.g., transition matrix for mileage

$$\Pi_{(d=0)} = \begin{pmatrix} p_1 & p_2 & p_3 & 0 & \cdots & 0 \\ 0 & p_1 & p_2 & p_3 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & 0 & p_1 & 1 - p_1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

- Extreme value Type 1 distribution of shocks
  - logit structure of optimal choice probability

Zurcher as a DP solver

Bellman equation in Rust (1987):

$$C(x, e; \theta) = \min_{d \in \{0,1\}} c(x, d) + e + \beta \mathbb{E}[C(x', e'; \theta) | (x, e, d)]$$

where $e \sim Gumbel(0, 1)$ and

$$c(x, d) := \begin{cases} RC + c_m(0; \theta), & d = 1 \\ c_m(x; \theta), & d = 0 \end{cases}$$

e.g., $c_m(x; \theta) = \theta_1 x$

DP Estimation on GPU with JAX

1. Fix $\beta = 0.9999$.
2. Estimates transition kernel of mileage by MLE.
3. Estimates cost functions by NFXP algorithm.

Parameter	Interpretation	Estimate	Std
p_1	$Pr(x_{t+1} = x_t)$	0.3919	0.0096
p_2	$Pr(x_{t+1} = x_t + 1)$	0.5953	0.0118
p_3	$Pr(x_{t+1} = x_t + 2)$	0.0129	0.0017
θ_1	$c_m(x) = \theta_1 x$	0.0023	0.0006
RC	Replacement Cost	10.0562	1.3576

Limitations of DP approach

Many assumptions.

- Zurcher can solve the Bellman equation.
- Zurcher's behavior follows the solution to DP.
- Zurcher has complete knowledge of the environment.
- Data is detached from solving the model, data is only useful for econometricians.

Why not let reality/data speak for itself?

Introduction of This Project

Implement structural estimation using Q-learning.

Q-learning (Watkins and Dayan, 1992).

- Model-free:
  do not require prior knowledge about an environment

- Applications in economics and finance
  Cournot model (Waltman and Kaymak, 2008),
  financial trading (Lee et al., 2007; Jeong and Kim, 2019; Chakole et al., 2021),
  pricing (Tesauro, 2001)

Zurcher as a Q-learner

Shocks e_t are unobservable to agents/econometricians: observe c_t.

Instead of learning

$$C(x) = \min_{d \in \{0,1\}} \underbrace{\mathbb{E}[c(x,d) + e + \beta C(x') | (x,d)]}_{Q(x,d)}$$

we learn

$$Q(x,d) = \mathbb{E}\left[c(x,d) + e + \min_{a \in \{0,1\}} Q(x',a) \middle| (x,d) \right]$$

Zurcher as a Q-learner

- For each time t, Zurcher learns mileage x_t and takes maintenance/replacement decision d_t according to his experience Q_t

- Zurcher then observes cost c_t and next period mileage x_{t+1}.

- Zurcher updates his experience $Q_{t+1}(x_t, d_t)$ with learning rate α_t by $c_t + \beta \min_a Q_t(x_{t+1}, a)$

Zurcher as a Q-learner

$$Q_{t+1}(x_t, d_t) = (1-\alpha_t)Q_t(x_t, d_t) + \alpha_t \underbrace{\left[c_t + \beta \min_{a\in\{0,1\}} Q_t(x_{t+1}, a)\right]}_{Y_t}$$

where $\alpha_t(x, a) \in [0, 1]$ is the learning rate, setting $\alpha_t(x, d) = 0$ if $(x, d) \neq (x_t, d_t)$.

Q_t is a sample mean:

$$\begin{aligned} Q(x, d) &= \mathbb{E}\left[c(x, d) + e + \min_{a\in\{0,1\}} Q(x', a)\middle|(x, d)\right] \\ &\approx \frac{1}{T}\sum_{t=1}^{T} Y_t = \frac{1}{T}\sum_{t=1}^{T-1} Y_t + \frac{1}{T} Y_T \qquad (T \to \infty) \\ &= (1 - \frac{1}{T})\bar{Y}_{T-1} + \frac{1}{T} Y_T \end{aligned}$$

Assumption 0.1 (Robbins-Monro)

1. $\sum_t \alpha_t(x,d) = \infty$ and $\sum_t \alpha_t^2(x,a) < \infty$ for all $(x,a) \in$ G w.p.1
2. $\mathbb{E}(e|(x,d)) = \mu_e(x,d)$ and $\mathrm{Var}(e_t|\mathscr{F}_t) < \infty$

Lemma 0.1 (Watkins and Dayan (1992); Tsitsiklis (1994))

If Assumption 0.1 holds, then $\{Q_t\}_{t\geq 0}$ *converges to* Q *w.p.1.*

Zurcher as a Q-learner

Zurcher knows that

- If he updates his experience following Q-learning, he will learn the optimal strategy eventually.
- Allows himself to make mistakes to learn optimal strategy.
- Does not need to know perfect information about environments.
- Can also learn cost function and probability transition kernel if he observes $\{c_t\}_{t \geq 0}$.

Unfortunately, econometricians do not observe $\{c_t\}_{t \geq 0}$, and do not have large enough data.

Fortunately, econometricians observe $\{d_t\}_{t \geq 0}$ and can estimate $c(x, d; \theta)$.

Q-learning Algorithm

Algorithm 1: Q-learning

Initialize $Q \in \mathbb{R}^G, x \in \mathsf{X}$

repeat

  Take action d, based on $Q(x, \cdot)$ using ϵ-greedy policy

  Observe $x' \in \mathsf{X}$ and $c \in \mathbb{R}$

  $Q(x,d) \leftarrow (1-\alpha(x,d))Q(x,d) + \alpha(x,d)\left(c + \beta \min_{a \in \{0,1\}} Q(x',a)\right)$

  $x \leftarrow x'$

end

- (x_t, d_t, x_{t+1}) are observed from data.
- $c_t = c(x_t, d_t; \theta) + e_t$ and e_t are simulated.
- ϵ-greedy policy: $1-\epsilon$ probability that $d_t = \arg\min_a Q_t(x_t, a)$ and ϵ probability that d_t is randomly chosen.

Q-learning Algorithm

- Since $P(x'|x, d=1) = P(x'|0, d=0)$, we have $Q(x,1) = RC + Q(0,0)$.

Algorithm 2: Q-learning with q

Initialize $Q \in \mathbb{R}^G$ and set $q(x) = Q(x,0)$ for all $x \in \mathrm{X} = \{0, m_1, \ldots, m_n\}$

Initialize $x \in \mathrm{X}$

repeat

  $d \leftarrow \mathbb{1}\{RC + q(0) > q(x)\}$ or randomly choose d with ϵ-greedy policy

  Observe $x' \in \mathrm{X}$ and $c \in \mathbb{R}$

  if $d = 0$ **then**

    $q(x) \leftarrow (1-\alpha)q(x) + \alpha[c + \beta \min\{q(x'), RC + q(0)\}]$

  else

    $q(0) \leftarrow (1-\alpha)q(0) + \alpha[c + \beta \min\{q(x'), RC + q(0)\} - RC]$

  $x \leftarrow x'$

end

$Q(x,d) \leftarrow \mathbb{1}_{\{d=1\}}(RC + q(0)) + \mathbb{1}_{\{d=0\}} q(x) \qquad ((x,d) \in G)$

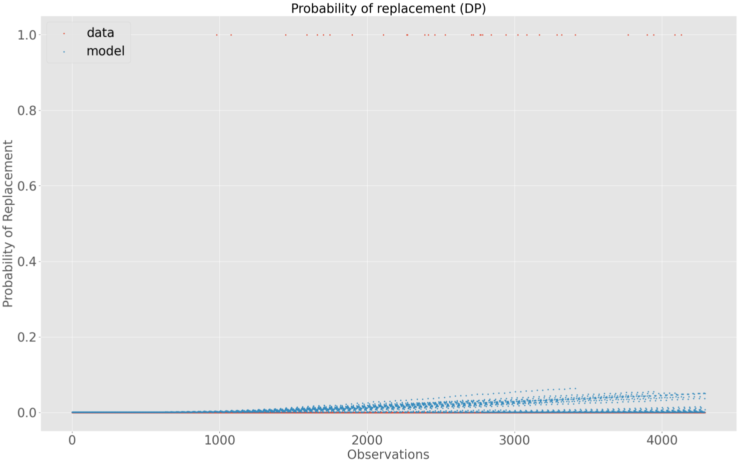
Estimation on GPU with JAX

1. Set $\beta = 0.9999, \alpha = 0.1, \epsilon = 0.02$.
2. Parameterize Q_0 as a quadratic function of (x, a).
3. Simulate cost shock sequences, $\{(x_t, c_t, x_{t+1}\}_{t \geq 0}$.
4. Simulate the time series of Q-table and choice probability.
5. Simulated maximum likelihood estimation.

Parameter	Interpretation	Estimate	Std
δ_0	$Q_0(x, 0) = \delta_0 + \delta_1 x + \delta_2 x^2$	0.0010	0.0002
δ_1	$Q_0(x, 0) = \delta_0 + \delta_1 x + \delta_2 x^2$	0.0021	0.0004
δ_2	$Q_0(x, 0) = \delta_0 + \delta_1 x + \delta_2 x^2$	0.0004	0.00007
θ_1	$c_m(x) = \theta_1 x$	0.0011	0.0002
RC	Replacement Cost	7.2174	1.3391

Fitness of Data: DP approach

- DP: stable decision pattern.

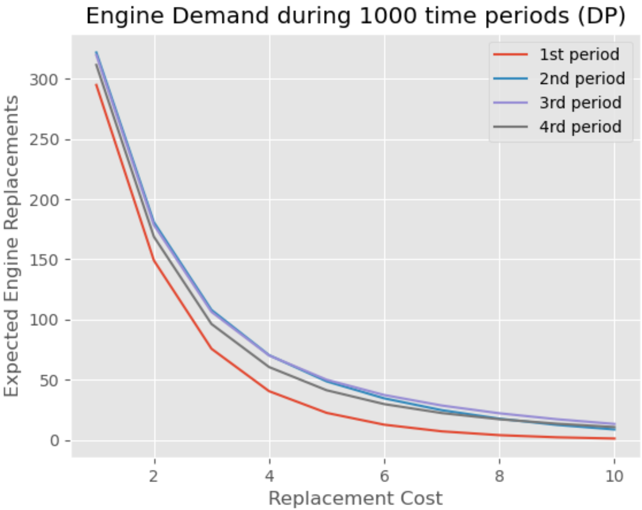


Fitness of Data: Q-learning

- Q-learning: Zurcher learns from data.

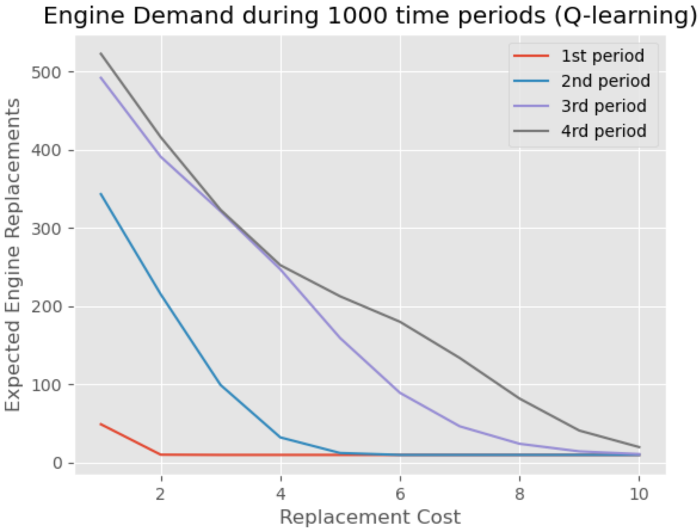
Demand For Engine Replacement: DP approach

- DP: stable engine demand acoss time, $d = f(x, RC)$.



Demand For Engine Replacement: Q-learning

- Q-learning: engine demand curve shifts through time, $d = f(x, RC, t)$.



"The majority of the modern economics literature can be regarded as a type of applied DP, ..., However, my impression is that formal DP has not been widely adopted to improve decision making by individuals and firms."

— Rust (2019)

Conclusion

Q-learning is a promising complement to DP

- more realistic assumptions for rationality.
- evolving decision rules over time.
- more flexible in modeling complex decisions.
- GPU makes simulation-based estimation fast.

Future works

- Applications in other economic problems.
- Q-learning with function approximation.
- Show that Q-learning has better performance.

Thank you! We can go home now!

References I

Chakole, J. B., M. S. Kolhe, G. D. Mahapurush, A. Yadav, and M. P. Kurhekar (2021): "A Q-learning agent for automated trading in equity stock markets," *Expert Systems with Applications*, 163, 113761.

Jeong, G. and H. Y. Kim (2019): "Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning," *Expert Systems with Applications*, 117, 125–138.

Lee, J. W., J. Park, O. Jangmin, J. Lee, and E. Hong (2007): "A multiagent approach to q-learning for daily stock trading," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37, 864–877.

Rust, J. (1987): "Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher," *Econometrica: Journal of the Econometric Society*, 999–1033.

——— (2019): "Has dynamic programming improved decision making?" *Annual Review of Economics*, 11, 833–858.

Tesauro, G. (2001): "Pricing in agent economies using neural networks and multi-agent Q-learning," *Sequence Learning: Paradigms, Algorithms, and Applications*, 288–307.

Tsitsiklis, J. N. (1994): "Asynchronous stochastic approximation and Q-learning," *Machine learning*, 16, 185–202.

Waltman, L. and U. Kaymak (2008): "Q-learning agents in a Cournot oligopoly model," *Journal of Economic Dynamics and Control*, 32, 3275–3293.

Watkins, C. J. and P. Dayan (1992): "Q-learning," *Machine learning*, 8, 279–292.