Harold Zurcher as a Q-learner

Wending Liu

Chienhsiang Yeh

Shu Hu

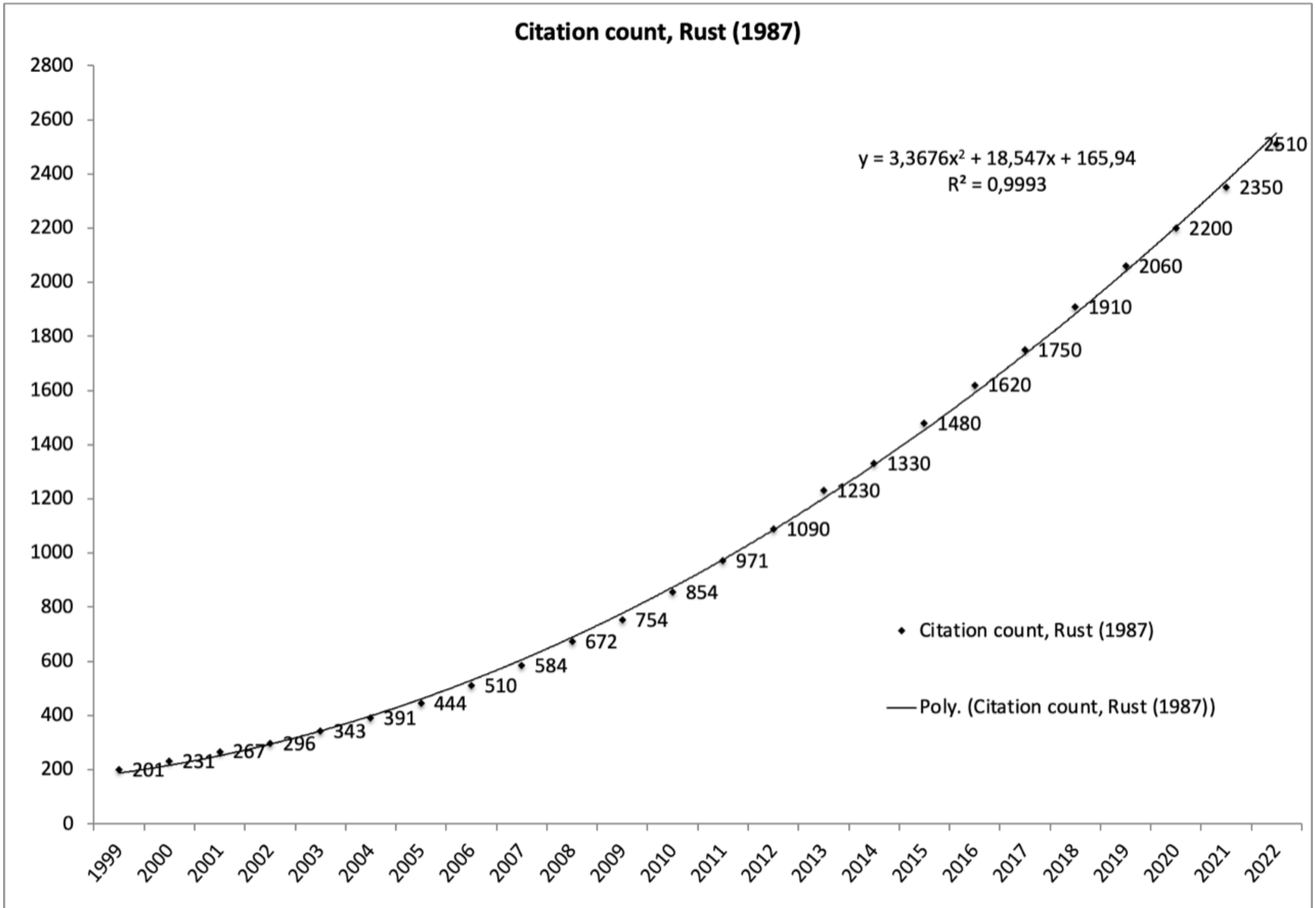
Research School of Economics, ANU

The Australasian Leadership Computing Symposium 2023

Introduction

- How to analyze the dynamic choices of agents with data?
- Dynamic Programming Approach (Rust, 1987)
  - strong assumptions for rationality and knowledge.
  - estimation based on nested fixed point algorithm.
- Q-learning Approach (This paper)
  - weak assumption for rationality.
  - agent has little knowledge of the environment.
  - simulation-based estimation.

The Importance of Rust(1987)



Zurcher's Problem

- Zurcher (a bus manager in Madison city) tries to minimize the infinite-horizon bus maintenance cost.
- He observes mileage x_t and chooses between ordinary maintenance ($d_t = 0$) and engine replacement ($d_t = 1$).
- Zurcher believes the cost function is $c(x, d) + e$, where

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

- e is an unobserved random shock, $\mathbb{E}(e|(x, d)) = \mu_e(x, d)$.

Zurcher as a DP solver

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

where $c_t = c(x_t, d_t) + e_t$.

- Bellman equation

$$C(x) = \min_d \left\{\mathbb{E}[c(x,d) + e + \beta C(x')|(x,d)]\right\}.$$

DP Estimation

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

- Zurcher can solve the Bellman equation.
- Zurcher's behavior follows the solution to DjP.
- Zurcher has complete knowledge of cost structure, distribution of cost shock, and transition kernel of mileage.
- Data is detached from solving the model, data is only useful for econometricians.

Zurcher as a Q-learner

$$C(x) = \min_d \underbrace{\{\mathbb{E}[c(x,d) + e + \beta C(x') | (x,d)]\}}_{=:Q^*(x,d)}.$$

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 $c \in \mathbb{R}$

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

  $x \leftarrow x'$

until end

Zurcher as a Q-learner

- Zurcher has initial knowledge Q_0.
  - He only observes c_t, x_t and x_{t+1}.
- Zurcher learns Q^* by Q-learning algorithm.
  - $C(x) = \min_d Q^*(x, d)$.
- Since Q_t converges to Q^*, Zurcher believes that he will learn Q^* eventually.

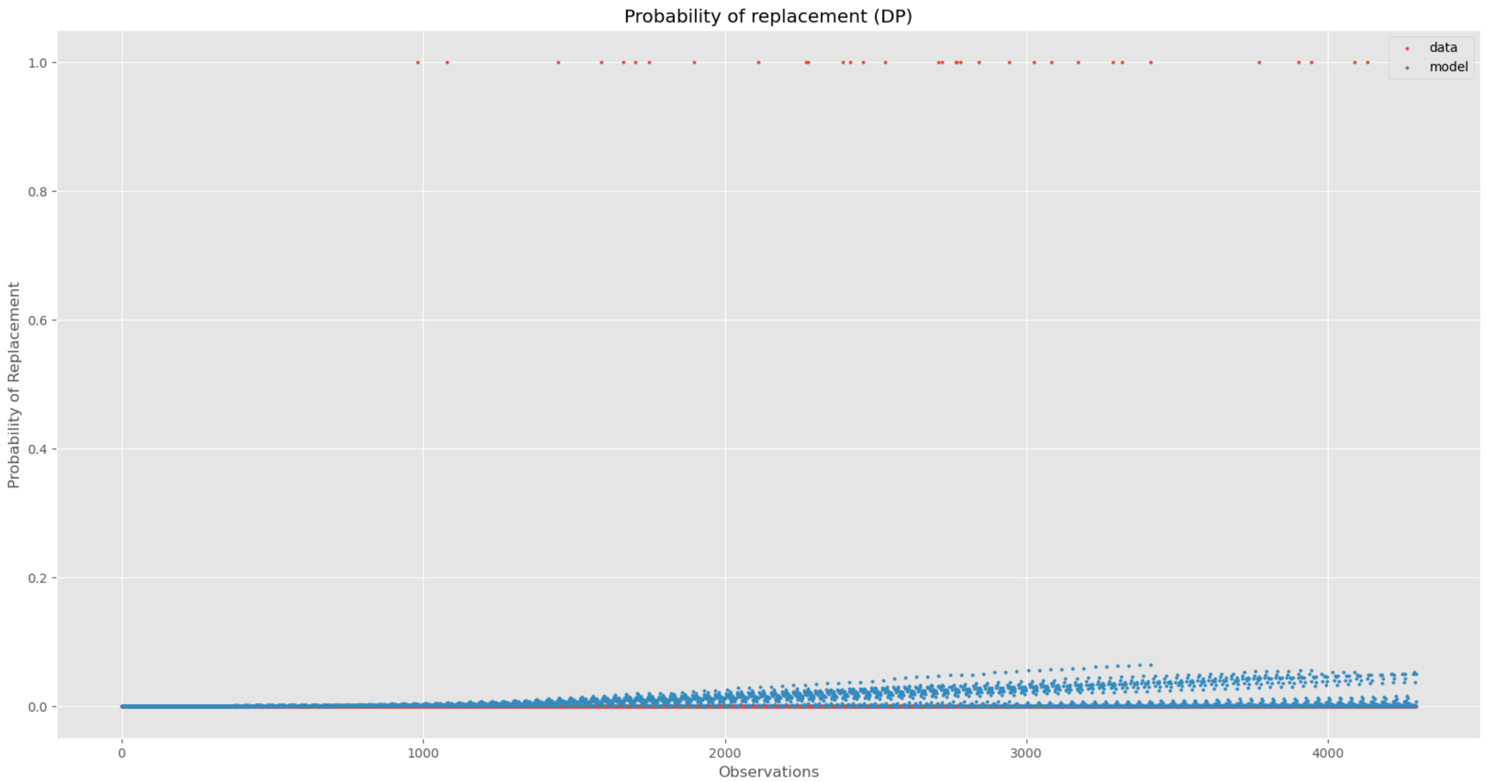
Estimation on GPU

1. Set $\beta = 0.9999, \alpha = 0.1, \varepsilon = 0.02$
2. Parameterize Q_0 as a quadratic function of (x, d).
3. Simulate many cost shock sequences, then simulate the time series of Q table and choice probabilities.
4. 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

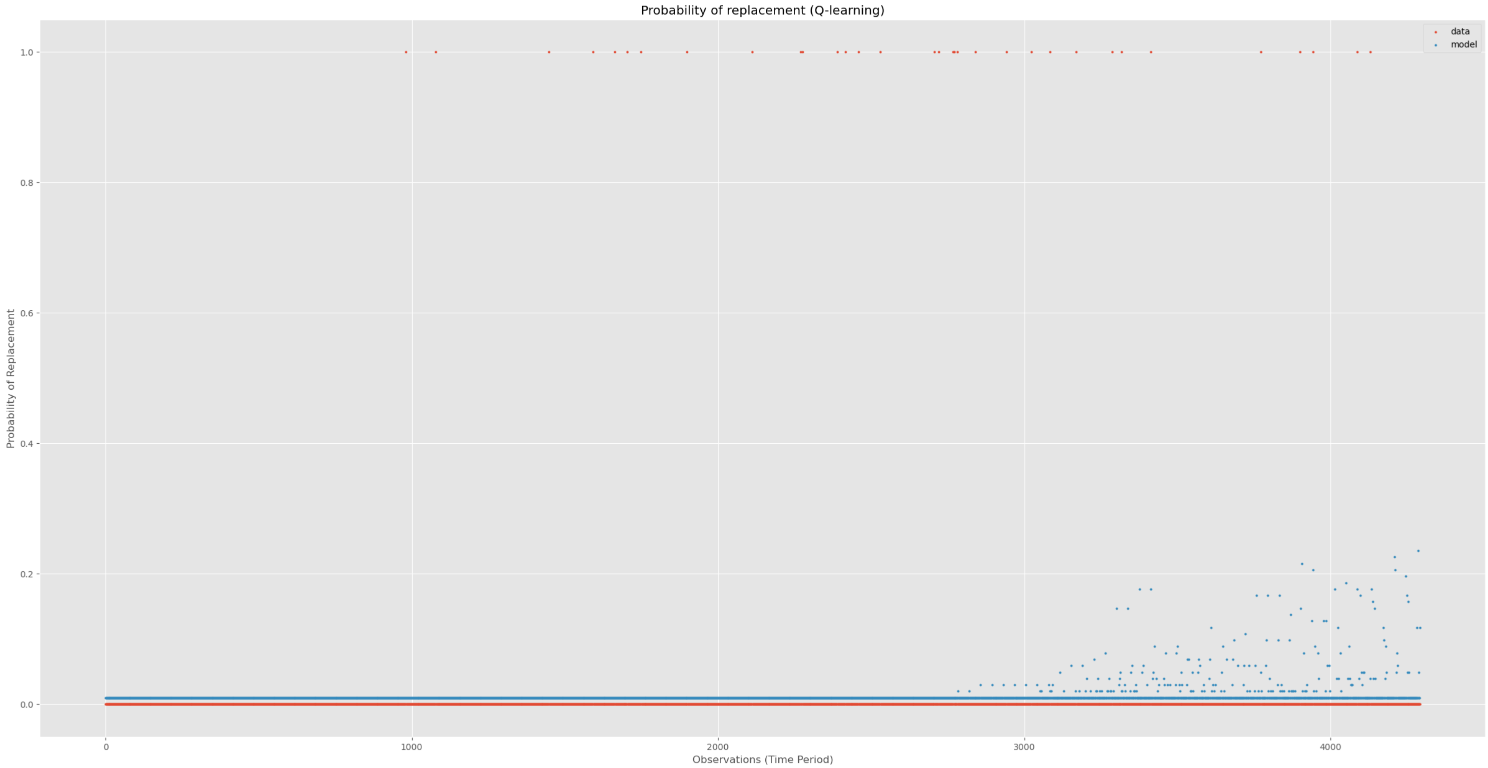
Fitness of Data (DP)

- DP: stable decision pattern.



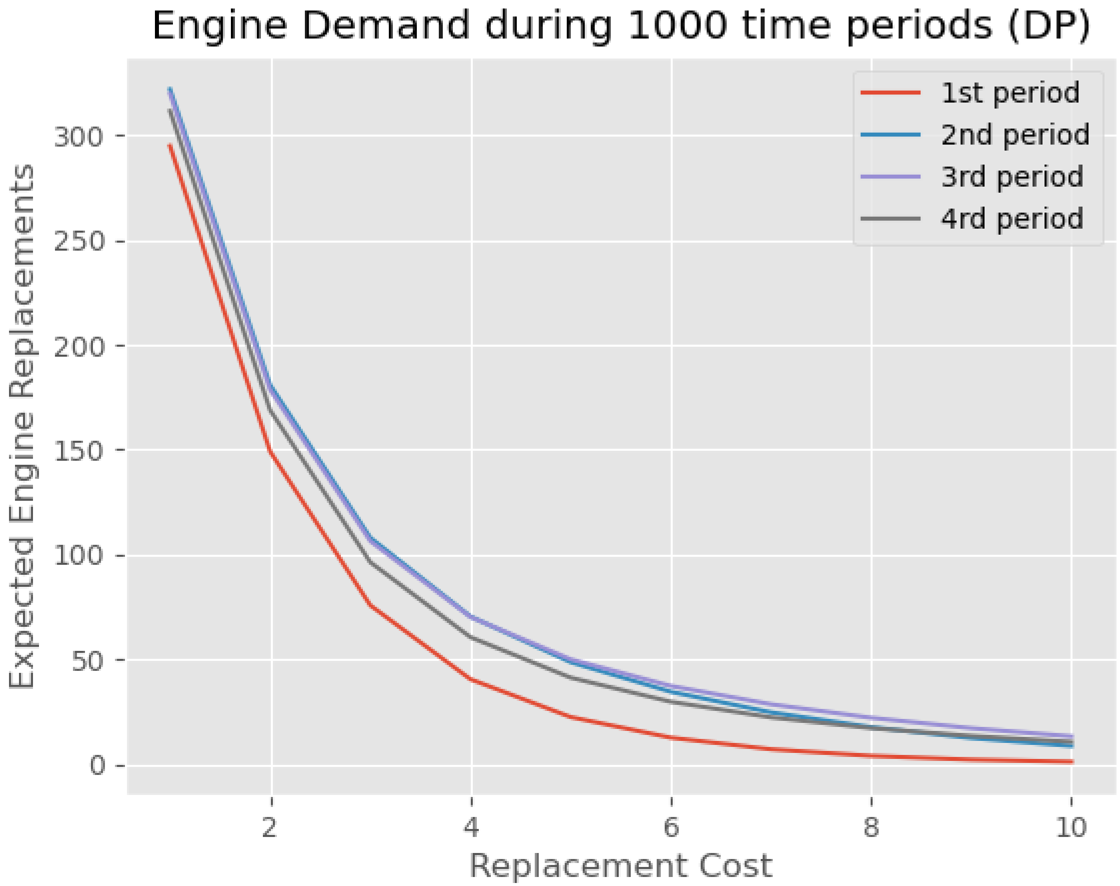
Fitness of Data (Q-learning)

- Q-learning: Zurcher learns from data!



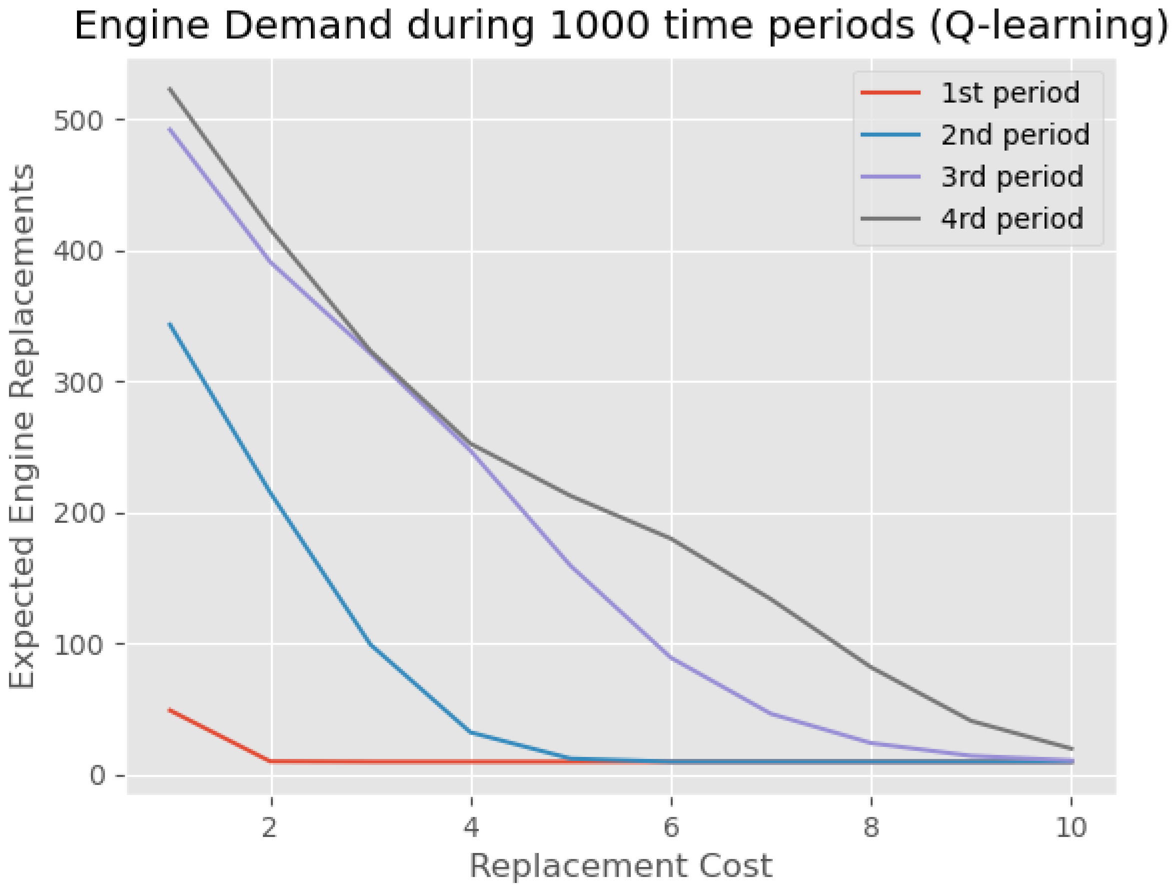
Demand for Engine Replacement

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



Demand for Engine Replacement

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



Conclusion

- "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)
- 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.