Rust (1987) analyzes the decision to replace a bus engine ($d = 1$) or not ($d = 0$)

Rust (1987) analyzes the decision to replace a bus engine ($d = 1$) or not ($d = 0$)

Why is this an optimal stopping problem?

Rust (1987) analyzes the decision to replace a bus engine ($d = 1$) or not ($d = 0$)

Why is this an optimal stopping problem?

- Harold Zurcher wants to minimize operating costs

Rust (1987) analyzes the decision to replace a bus engine ($d = 1$) or not ($d = 0$)

Why is this an optimal stopping problem?

- Harold Zurcher wants to minimize operating costs
- But unexpected bus breakdowns cause "customer goodwill costs"

Rust (1987) analyzes the decision to replace a bus engine ($d = 1$) or not ($d = 0$)

Why is this an optimal stopping problem?

- Harold Zurcher wants to minimize operating costs
- But unexpected bus breakdowns cause “customer goodwill costs”
- Premature replacement is costly, but so is breakdown

Rust (1987) analyzes the decision to replace a bus engine ($d = 1$) or not ($d = 0$)

Why is this an optimal stopping problem?

- Harold Zurcher wants to minimize operating costs
- But unexpected bus breakdowns cause "customer goodwill costs"
- Premature replacement is costly, but so is breakdown
- Goal: figure out when to optimally replace engines

Rust (1987) analyzes the decision to replace a bus engine ($d = 1$) or not ($d = 0$)

Why is this an optimal stopping problem?

- Harold Zurcher wants to minimize operating costs
- But unexpected bus breakdowns cause "customer goodwill costs"
- Premature replacement is costly, but so is breakdown
- Goal: figure out when to optimally replace engines
- Some buses get driven more than others

Replacement decision depends on:

- Mileage on engine: x_t
- Cost of replacing: $\overline{P}$
- Scrap value of current engine: $\underline{P}$

Replacement decision depends on:

- Mileage on engine: x_t
- Cost of replacing: $\overline{P}$
- Scrap value of current engine: $\underline{P}$

Payoffs net of error term:

$$u_0(x_{it}, \theta) = -c(x_{it}, \theta)$$
$$u_1(x_{it}, \theta) = -[\overline{P} - \underline{P} + c(0, \theta)]$$

Replacement decision depends on:

- Mileage on engine: x_t
- Cost of replacing: $\overline{P}$
- Scrap value of current engine: $\underline{P}$

Payoffs net of error term:

$$
\begin{aligned}
u_0(x_{it}, \theta) &= -c(x_{it}, \theta) \\
u_1(x_{it}, \theta) &= -[\overline{P} - \underline{P} + c(0, \theta)]
\end{aligned}
$$

Mileage is discrete and transitions according to $f(x_{t+1}|x_t)$

Why This Matters

- Individual replacement decisions $\rightarrow$ aggregate investment volatility

Why This Matters

- Individual replacement decisions $\rightarrow$ aggregate investment volatility
- Firms replacing equipment creates economy-wide boom/bust cycles

Why This Matters

- Individual replacement decisions $\rightarrow$ aggregate investment volatility
- Firms replacing equipment creates economy-wide boom/bust cycles
- Wrong timing wastes billions in premature/delayed replacement

Why This Matters

- Individual replacement decisions $\rightarrow$ aggregate investment volatility
- Firms replacing equipment creates economy-wide boom/bust cycles
- Wrong timing wastes billions in premature/delayed replacement
- Transportation infrastructure especially critical for service quality

Estimation Procedure

Step 1: Calculate mileage transitions $f(x_{t+1}|x_t)$

Estimation Procedure

Step 1: Calculate mileage transitions $f(x_{t+1}|x_t)$

Step 2: Maximize log likelihood of choices:

$$\ell(\theta; \mathbf{d}, \mathbf{x}) = \sum_i \sum_t \sum_j (d_{it} = j) \log(p_{jt}(x_{it}, \theta))$$

Estimation Procedure

Step 1: Calculate mileage transitions $f(x_{t+1}|x_t)$

Step 2: Maximize log likelihood of choices:

$$\ell(\theta; \mathbf{d}, \mathbf{x}) = \sum_i \sum_t \sum_j (d_{it} = j) \log(p_{jt}(x_{it}, \theta))$$

Step 3: Within maximization, solve fixed point problem in the v's each time log likelihood is evaluated

Nested Fixed Point (NFXP) Algorithm

- Outer loop: maximize likelihood over parameters θ
- Inner loop: solve for value functions v given θ
- Repeat until convergence

Nested Fixed Point (NFXP) Algorithm

- Outer loop: maximize likelihood over parameters θ
- Inner loop: solve for value functions v given θ
- Repeat until convergence

Each likelihood evaluation requires solving the dynamic programming problem

Nested Fixed Point (NFXP) Algorithm

- Outer loop: maximize likelihood over parameters θ
- Inner loop: solve for value functions v given θ
- Repeat until convergence

Each likelihood evaluation requires solving the dynamic programming problem

Each outer loop iteration took 4 minutes ($N = 104$ buses, $T = 120$ months)

What does Rust do with his results?

- Examines different functional forms for $c(\cdot)$, including with unobs. heterogeneity

What does Rust do with his results?

- Examines different functional forms for $c(\cdot)$, including with unobs. heterogeneity
- Tests whether Zurcher behaves myopically (finds he doesn't)

What does Rust do with his results?

- Examines different functional forms for $c(\cdot)$, including with unobs. heterogeneity
- Tests whether Zurcher behaves myopically (finds he doesn't)
- Shows that dynamic considerations result in less replacement at high mileage

What does Rust do with his results?

- Examines different functional forms for $c(\cdot)$, including with unobs. heterogeneity
- Tests whether Zurcher behaves myopically (finds he doesn't)
- Shows that dynamic considerations result in less replacement at high mileage
- Infers implied demand for replacement as function of replacement cost

What does Rust do with his results?

- Examines different functional forms for $c(\cdot)$, including with unobs. heterogeneity
- Tests whether Zurcher behaves myopically (finds he doesn't)
- Shows that dynamic considerations result in less replacement at high mileage
- Infers implied demand for replacement as function of replacement cost
- Myopic demand curve is much more sensitive to replacement cost

Value of a structural model

Since engine replacement costs have not varied much in the past, estimating replacement demand by a "reduced-form" approach which, for example, regresses engine replacements on replacement costs, is incapable of producing reliable estimates of the replacement demand function.

In terms of Figure 7, all the data would be clustered in a small ball about the intersection of the two demand curves: obviously many different demand functions would appear to fit the data equally well.

The structural approach, on the other hand, efficiently concentrates additional information contained in the sequences $\{d_t, x_t\}$ *into estimates of a small number of primitive parameters. Despite the relatively small number of such parameters, we obtain a rich behavioral model that can be used to answer a wide range of "what if?" policy questions.*