Optimal Bargaining on eBay Using Deep Reinforcement Learning

Etan Green[1] E. Barry Plunkett[1,2]

[1]University of Pennsylvania

[2]D. E. Shaw

November 2020

What offer should I make?

“Best Offer” listings on eBay

This project

For any listing and at any point in the offer history, what offer maximizes the eventual payoff? For the seller? For the buyer?

- Characterize optimal behavior in a way that humans can use.

Larger goal

Tutorial for solving real-world dynamic decision problems—and characterizing the solutions.

Reinforcement learning

In each state, finds the action that maximizes the eventual payoff.

$$\pi(s) : s \rightarrow a$$

Chess

- State: board position
- Action: allowable move
- Payoff: 1 for a win, 0 for a loss

Play randomly at first, reinforce actions that lead to higher payoffs.

Deep reinforcement learning

Approximates states from features.

$$\pi(\boldsymbol{x}) : \boldsymbol{x} \rightarrow a$$

AlphaZero

“Best Offer” listings on eBay

- State: listing features + offer history
- Action: an offer
- Payoff: (tbd)

Can we train an algorithm to play optimally against human buyers and sellers on eBay?

"Best Offer" listings on eBay

- State: listing features + offer history
- Action: an offer
- Payoff: (tbd)

Can we train an algorithm to play optimally against human buyers and sellers on eBay?

- Optimal $\neq$ equilibrium

Ideally, we would train the algorithm by bargaining on eBay.

- AlphaZero achieved superhuman performance only after millions of games.

Ideally, we would train the algorithm by bargaining on eBay.

- AlphaZero achieved superhuman performance only after millions of games.

Our approach:

1. Train neural nets to mimic human buyers and sellers.
2. Train reinforcement learning agents to play optimally against these simulated buyers and sellers.

Outline

1. Gameplay
2. Data
3. Simulator
4. Payoffs
5. RL seller
6. RL buyer

Outline

1. Gameplay
2. Data
3. Simulator
4. Payoffs
5. RL seller
6. RL buyer

Seller sets three prices

1. A **list price**, at which the item may be purchased immediately.
2. An optional **auto-accept price**, above which buyer offers are immediately accepted.
3. An optional **auto-reject price**, below which buyer offers are immediately rejected.

Order of operations (circa 2013)

1. Buyer makes an offer.

Order of operations (circa 2013)

1. Buyer makes an offer.
2. If offer is between the automatic thresholds, seller may accept, counter or reject.
   - If seller does not respond in 48 hours, offer is rejected.

Order of operations (circa 2013)

1. Buyer makes an offer.
2. If offer is between the automatic thresholds, seller may accept, counter or reject.
   - If seller does not respond in 48 hours, offer is rejected.
3. If seller does not accept, buyer may accept, counter or walk.
   - If buyer does not respond in 48 hours, buyer walks.

Order of operations (circa 2013)

1. Buyer makes an offer.
2. If offer is between the automatic thresholds, seller may accept, counter or reject.
   - If seller does not respond in 48 hours, offer is rejected.
3. If seller does not accept, buyer may accept, counter or walk.
   - If buyer does not respond in 48 hours, buyer walks.
4. Repeat (2) & (3) until buyer and seller have each had 3 turns.

Order of operations (circa 2013)

1. Buyer makes an offer.
2. If offer is between the automatic thresholds, seller may accept, counter or reject.
   - If seller does not respond in 48 hours, offer is rejected.
3. If seller does not accept, buyer may accept, counter or walk.
   - If buyer does not respond in 48 hours, buyer walks.
4. Repeat (2) & (3) until buyer and seller have each had 3 turns.
5. After 3rd seller response, buyer faces take-it-or-leave-it offer.

Order of operations (circa 2013)

1. Buyer makes an offer.
2. If offer is between the automatic thresholds, seller may accept, counter or reject.
   - If seller does not respond in 48 hours, offer is rejected.
3. If seller does not accept, buyer may accept, counter or walk.
   - If buyer does not respond in 48 hours, buyer walks.
4. Repeat (2) & (3) until buyer and seller have each had 3 turns.
5. After 3rd seller response, buyer faces take-it-or-leave-it offer.

A thread ends when...

1. An offer is accepted (on any thread).
2. The buyer walks (actively or passively).
3. The listing expires.

Outline

Backus et al. (2020) data

98M "Best Offer" listings on eBay from 2012-13.

Backus et al. (2020) data

98M "Best Offer" listings on eBay from 2012-13.

1. Plausibly unique listings (unique title): XXXM

Backus et al. (2020) data

98M "Best Offer" listings on eBay from 2012-13.

1. Plausibly unique listings (unique title): XXXM
2. and list price between \$9.95 and \$1000.00: XXXM

Backus et al. (2020) data

98M "Best Offer" listings on eBay from 2012-13.

1. Plausibly unique listings (unique title): XXXM
2. and list price between $9.95 and $1000.00: XXXM
3. and fixed list price: XXXM

Backus et al. (2020) data

98M "Best Offer" listings on eBay from 2012-13.

1. Plausibly unique listings (unique title): XXXM
2. and list price between \$9.95 and \$1000.00: XXXM
3. and fixed list price: XXXM
4. and no other funny stuff: XXXM

Partitions

XXXM listings, from XXX sellers, split into 4 partitions:

1. Simulator training: 75% of sellers
2. RL training: 10%
3. Validation: 5%
4. Test: 10%

All results that follow are from validation partition.

What's in the data?

Complete offer histories for all negotiations.

- Unique among datasets of this size.

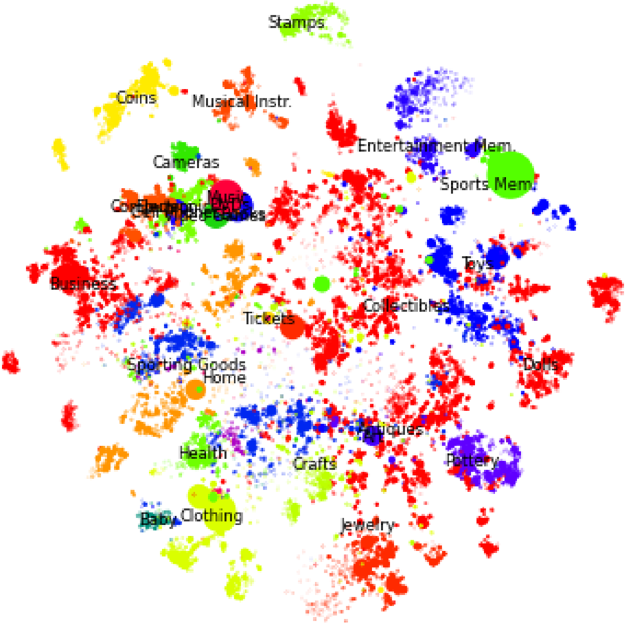
What else is in the data?

An incomplete list:

- List price and automatic thresholds.
- Category and subcategory.
- Listing start and end dates.
- Number of photos.
- Seller's rating.
- Offer timestamps.
- Whether a message is attached to the offer.

What's not in the data

- The photos themselves.
- The messages themselves.
- Item descriptions.


Stamps
Coins
Musical Instr.
Entertainment Mem.
Cameras
Sports Mem.
Music
Business
Toys
Collectibles
Tickets
Dolls
Sporting Goods
Home
Antiques
Art
Pottery
Health
Crafts
Baby
Clothing
Jewelry

Outline

1. Gameplay
2. Data
3. Simulator
4. Payoffs
5. RL seller
6. RL buyer

Goal

Generate credible counterfactual offer paths.

Example listing

- \$100 list price
- \$50 auto-reject price
- No auto-accept price

In the data, the item sells for the list price to the first buyer.

Example listing

- $100 list price
- $50 auto-reject price
- No auto-accept price

In the data, the item sells for the list price to the first buyer.

Counterfactual:

1. Buyer 1 offers 50% of list price.
2. Seller auto-rejects.
3. Buyer 1 offers 75% of list price.
4. Buyer 2 purchases the item for the list price.

Overview

Simulate:

1. The arrival of buyers.
2. The offer path of each thread.

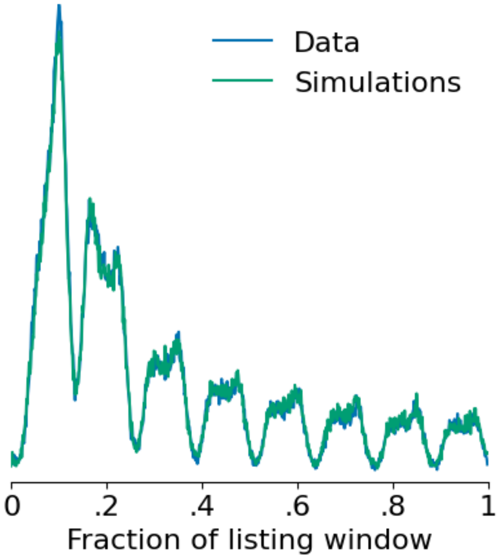
All features of the listing are held constant.

- e.g., list price and automatic thresholds

Listings expire after 1 week.

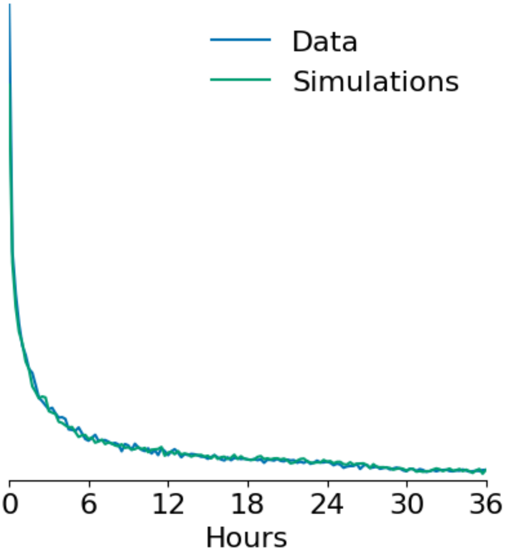
- Common listing duration in the data.

Arrival time of first buyer

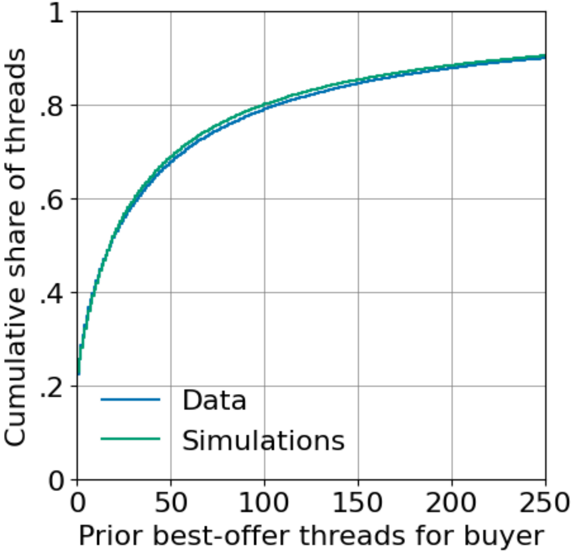


70% of listings expire without an arrival.

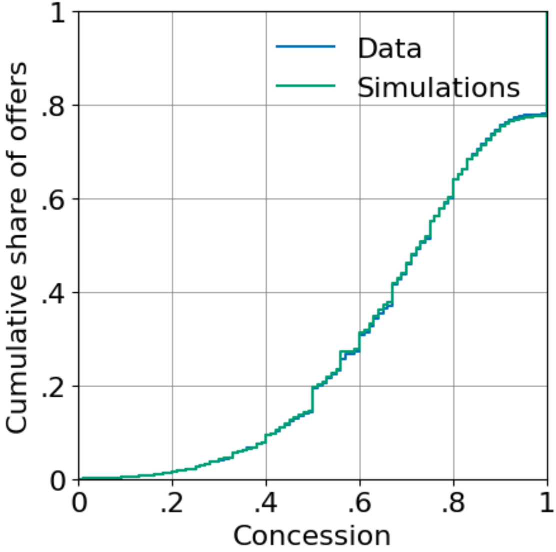
Interarrival time



Buyer experience



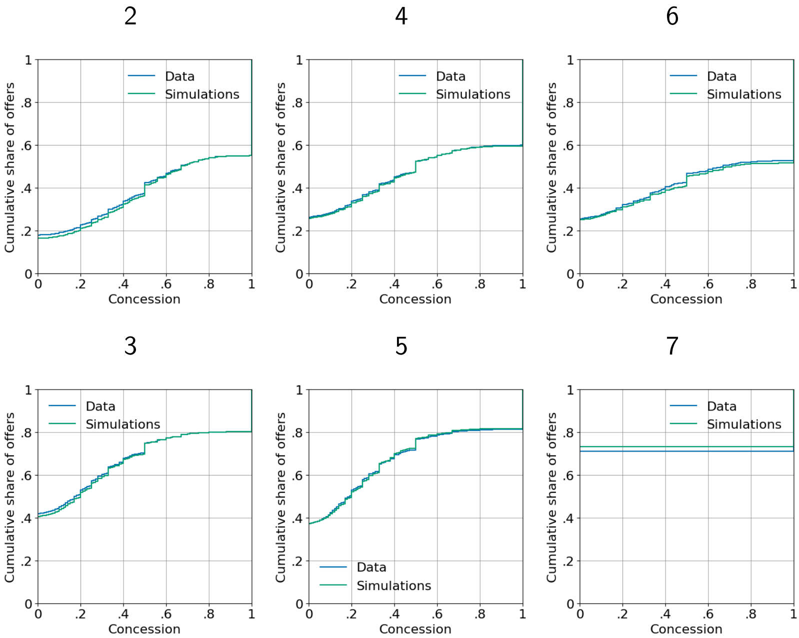
First buyer offer



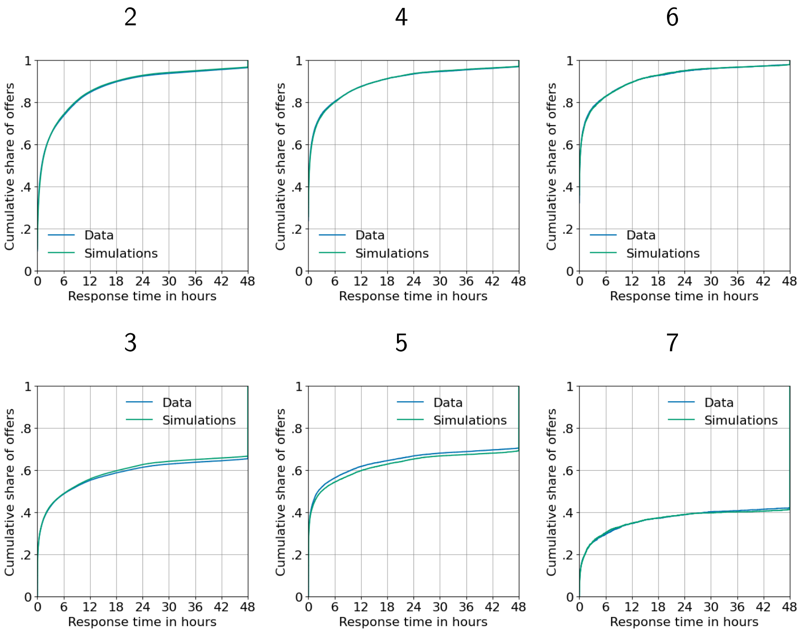
Offers as concessions

How much of the bargaining zone is conceded.

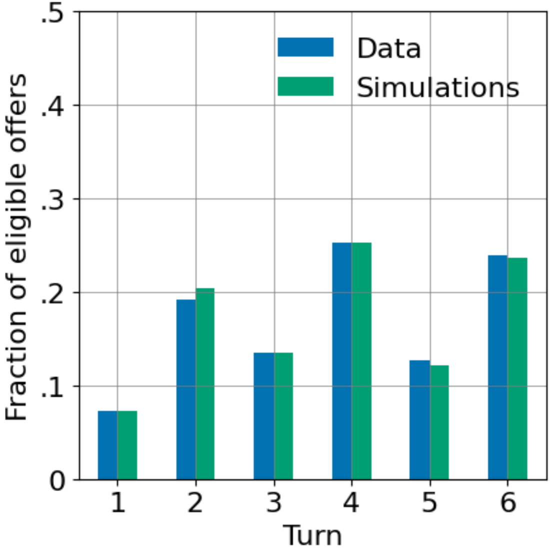
- e.g., buyer last offered \$50 and seller last offered \$100.
- \$60 buyer offer = 20% concession
- \$90 seller offer = 20% concession



2
4
6
3
5
7
Data
Simulations
Cumulative share of offers
Concession
0
.2
.4
.6
.8
1

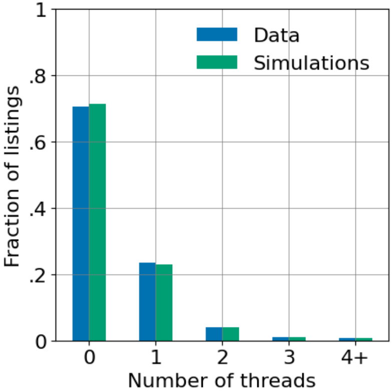

2
4
6
3
5
7
Cumulative share of offers
Response time in hours
0
.2
.4
.6
.8
1
0 6 12 18 24 30 36 42 48
Data
Simulations

Message rates

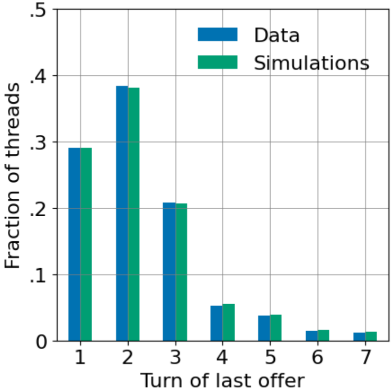


Summary statistics

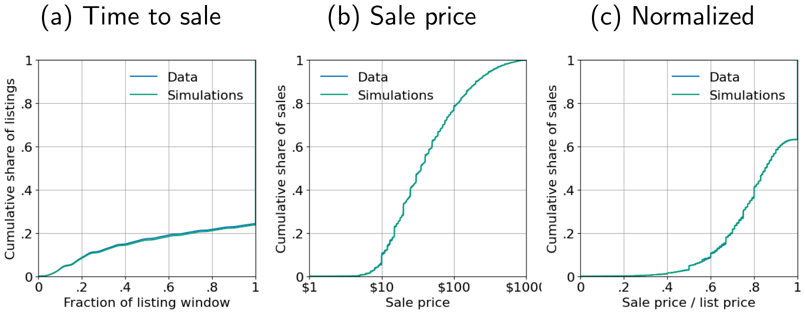
(a) Threads per listing



(b) Offers per thread

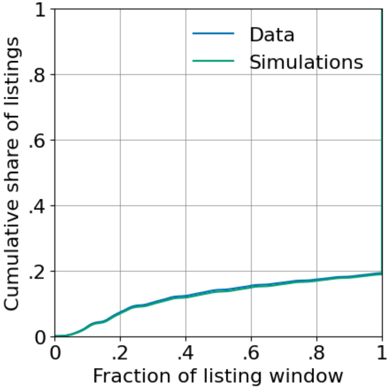


Sale time and price

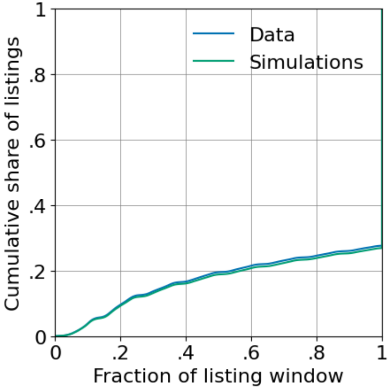


By category: time to sale

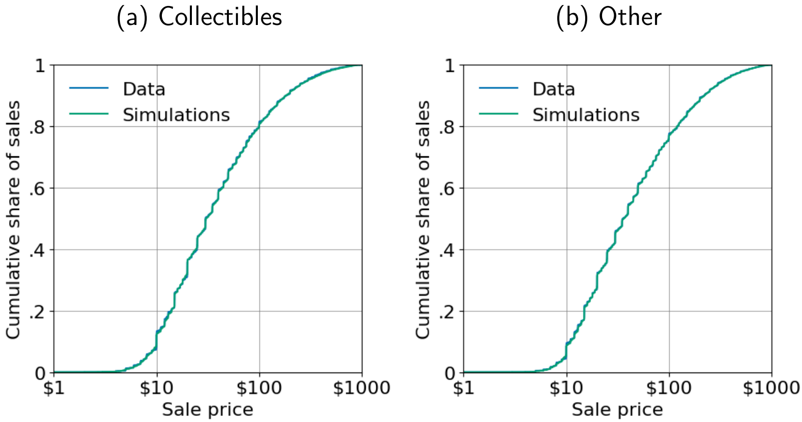
(a) Collectibles



(b) Other

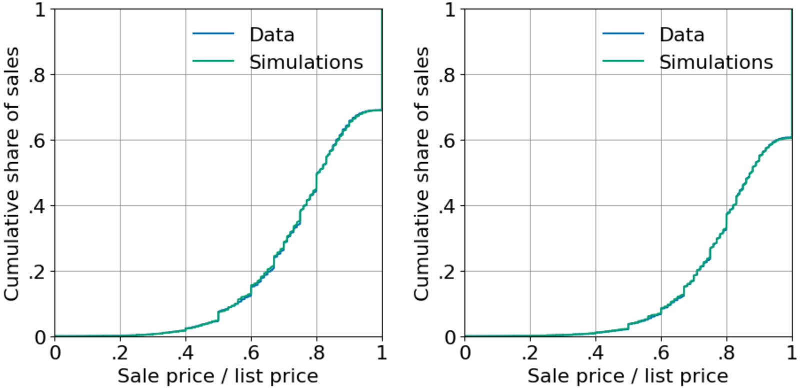


By category: sale price



By category: normalized sale price

(a) Collectibles (b) Other



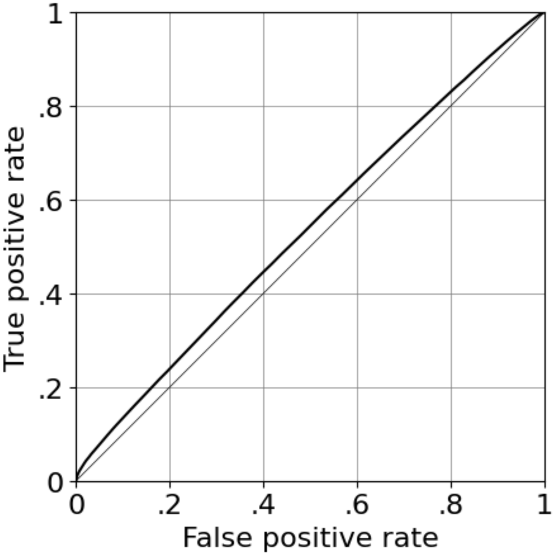
Discriminator

Observes a complete thread, either from the data or the simulation.

- Fixed listing features.
- Offer path.

Predicts the whether the thread is real or simulated.

Discriminator performance



AUC: 53.5%

Outline

Payoffs

Buyer:

- `value` − `price`, if item is purchased
- 0, otherwise

Seller:

- `price`, if item is sold
- $\delta \cdot$ `value`, otherwise

What is an item's value?

- Calculate a "market value" for each item.
- Characterize optimal behavior under these values.

Values

$$\texttt{value} = P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}] + \big(1 - P(\texttt{sale})\big) \cdot \delta \cdot \texttt{value}$$

Values

$$\texttt{value} = \frac{P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}]}{1 - \big(1 - P(\texttt{sale})\big) \cdot \delta}$$

Values

$$\texttt{value} = \frac{P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}]}{1 - \big(1 - P(\texttt{sale})\big) \cdot \delta}$$

- $\delta = 0 \to \texttt{value} = P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}]$
- $\delta = 1 \to \texttt{value} = \mathbb{E}[\texttt{price}|\texttt{sale}]$

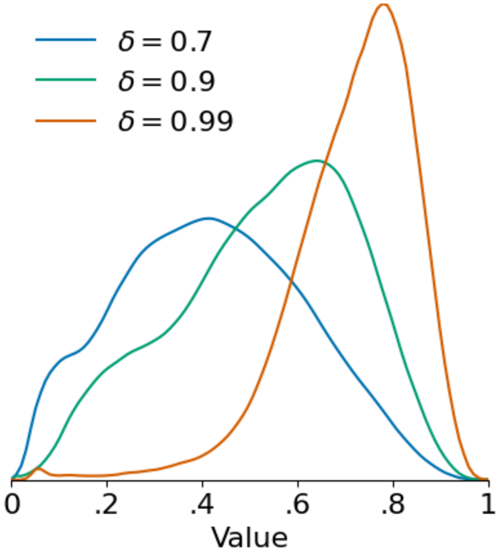
Values

$$\texttt{value} = \frac{P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}]}{1 - \big(1 - P(\texttt{sale})\big) \cdot \delta}$$

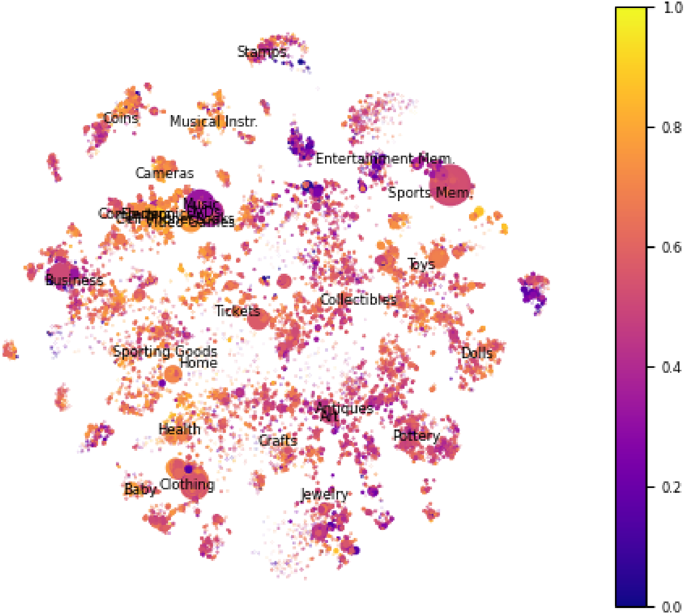
- $\delta = 0 \rightarrow \texttt{value} = P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}]$
- $\delta = 1 \rightarrow \texttt{value} = \mathbb{E}[\texttt{price}|\texttt{sale}]$

Simulate each listing to estimate $P(\texttt{sale})$ and $\mathbb{E}[\texttt{price}|\texttt{sale}]$.

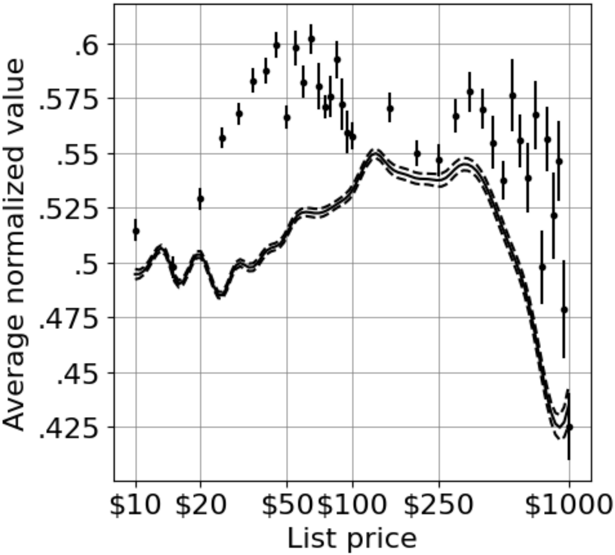
Normalized values



Category predicts value

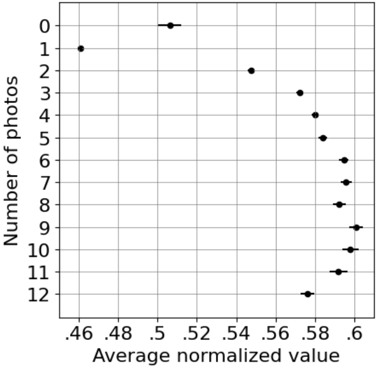


Round list prices have higher values

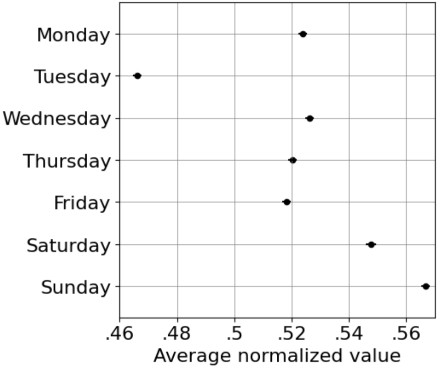


Some other predictors of value

(a) Number of photos



(b) Day of listing start



Outline

1. Gameplay
2. Data
3. Simulator
4. Payoffs
5. RL seller
6. RL buyer

$$\pi(\boldsymbol{x}) : \boldsymbol{x} \rightarrow f(a)$$

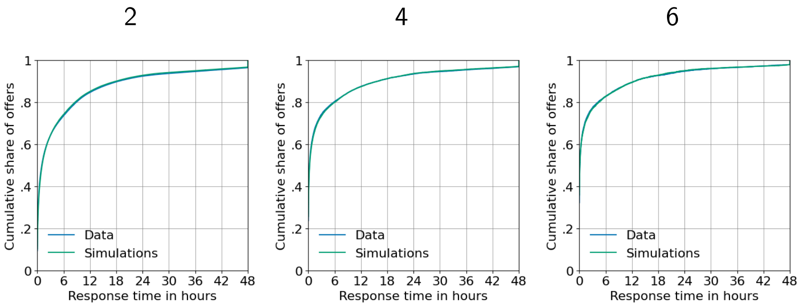
$\boldsymbol{x}$: features that are observable to seller.

- e.g., features that summarize offers on other threads.
- Seller does not observe item value.

$a \in \{\texttt{reject}, .2, .25, .33, .4, .5, .6, .67, \texttt{accept}\}$

- Cannot send a message.
- Delay drawn from simulated seller model.

Delays



Drawn from turn-specific distribution for simulated seller.

- Conditional on delay < 48 hours.

Training procedure

Initialize seller policy π.

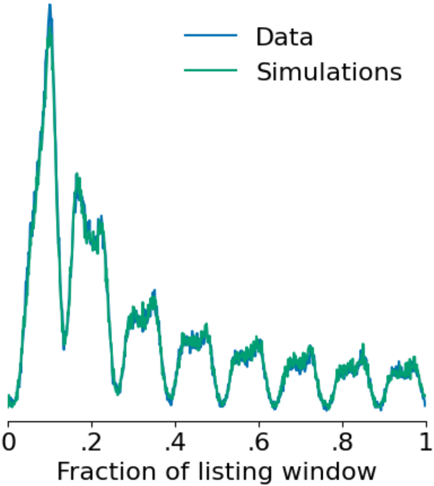
1. Draw a listing from RL Training partition.
2. Simulate until the listing ends using π to draw seller offers.
3. If listing sells, payoff is `sale price`.
   - Otherwise payoff is $\delta \cdot \texttt{value}$.
4. Update π.

Repeat until π converges to deterministic policy.

Outline

1. Gameplay
2. Data
3. Simulator
4. Payoffs
5. RL seller
6. RL buyer

Arrival process



70% of listings expire without an arrival.

- No agent arrival for 70% of simulations.

Offers

$$\pi(\boldsymbol{x}) : \boldsymbol{x} \rightarrow f(a)$$

Restrictions:

$\boldsymbol{x}$: only features that buyers can observe.

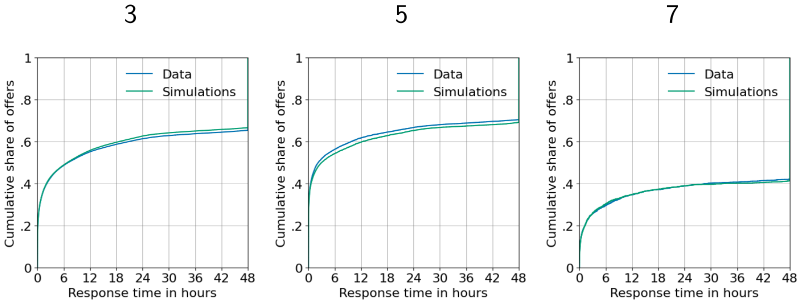
- e.g., excludes automatic thresholds, activity on other threads.

a : 7 most common concessions, plus `walk` and `accept`.

- Turn 1: $\{\texttt{walk}, .5, .6, .67, .7, .75, .8, .83, \texttt{accept}\}$
- Turns 3 & 5: $\{\texttt{walk}, .17, .2, .25, .29, .33, .4, .5, \texttt{accept}\}$
- Turn 7: $\{\texttt{walk}, \texttt{accept}\}$

- Cannot send a message.

Delays



Drawn from turn-specific distribution for simulated buyer.

- Conditional on delay < 48 hours.