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.

- Where experimentation is infeasible but data are plentiful.

And characterizing the solutions.

Reinforcement learning

In each state, find the policy that maximizes the eventual payoff.

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

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.

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

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.

- Would require millions of negotiations.

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

- Would require millions of negotiations.

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

Outline

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

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.

A thread ends when...

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

Outline

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

Backus et al. (2020)

Universe of “Best Offer” listings on eBay from 2012-13.

- Complete offer histories for all negotiations.

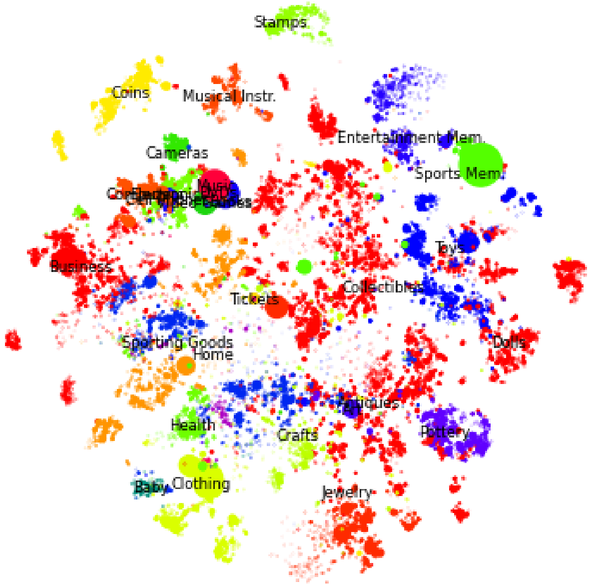
What else is in the data?

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.
Business
Toys
Collectibles
Tickets
Sporting Goods
Home
Dolls
Antiques
Health
Crafts
Pottery
Baby
Clothing
Jewelry

Data restrictions

Original data: 98.3M listings.

Data restrictions

Original data: 98.3M listings.

1. Plausibly unique listings (unique title): 28.6M

Data restrictions

Original data: 98.3M listings.

1. Plausibly unique listings (unique title): 28.6M
2. and fixed list price: 18.6M

Data restrictions

Original data: 98.3M listings.

1. Plausibly unique listings (unique title): 28.6M
2. and fixed list price: 18.6M
3. and list price between $9.95 and $1000.00: 14.6M

Data restrictions

Original data: 98.3M listings.

1. Plausibly unique listings (unique title): 28.6M
2. and fixed list price: 18.6M
3. and list price between \$9.95 and \$1000.00: 14.6M
4. and no other funny stuff: 13.4M

Partitions

13.4M listings, from 771.6k 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.

Outline

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

Example listing

- \$100 list price
- \$60 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
- $60 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.
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.

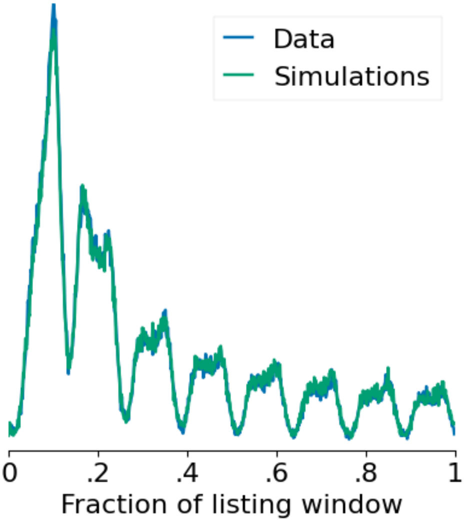
All features of the listing are held constant.

- e.g., list price and automatic thresholds

Listings expire after 1 week.

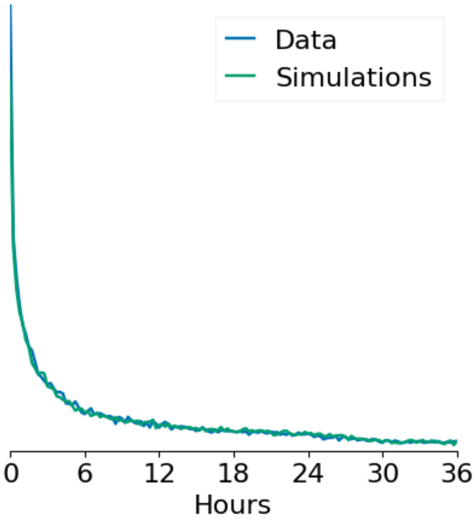
- Common listing duration in the data.

Arrival time of first buyer

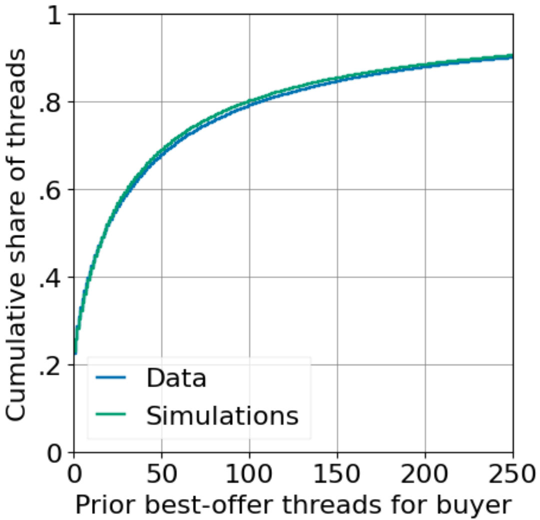


70% of listings expire without an arrival.

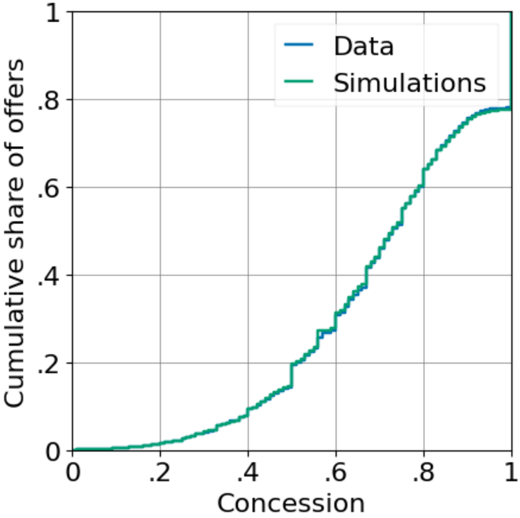
Interarrival time



Buyer experience



First buyer offer

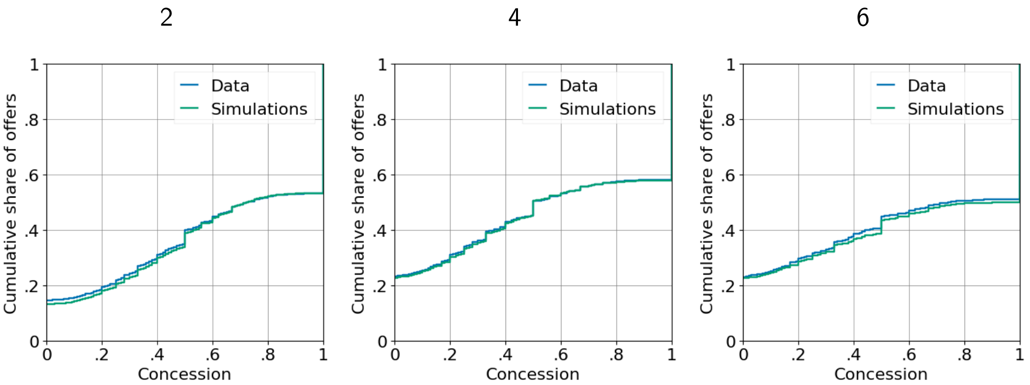


Offers as concessions

How much of the bargaining zone is conceded.

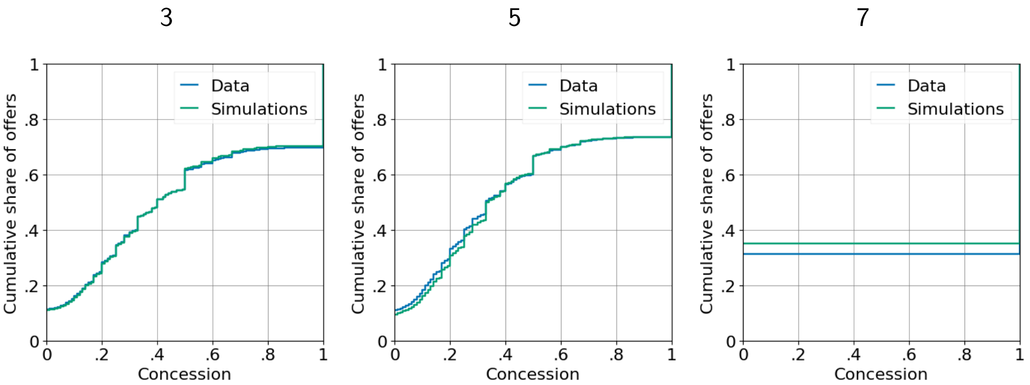
- e.g., buyer last offered \$50 and seller last offered \$100.
- \$60 buyer offer = 20% concession
- \$90 seller offer = 20% concession

Concessions: seller turns



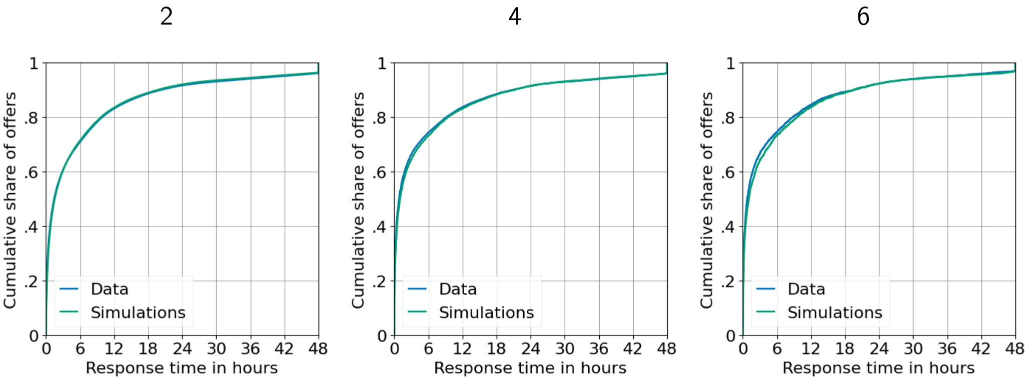
Note: excludes automatic offers and expirations.

Concessions: buyer turns



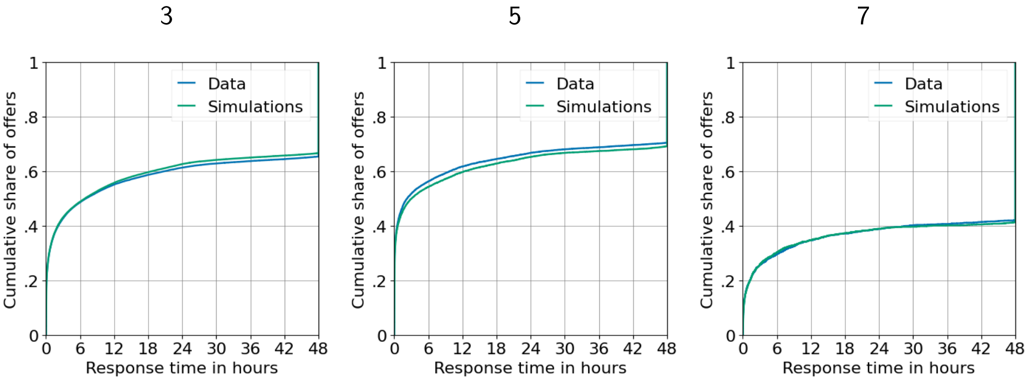
Note: excludes expirations.

Response time: seller turns

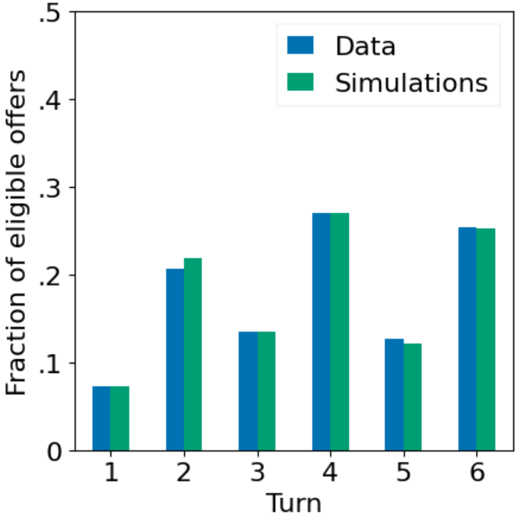


Note: excludes automatic offers.

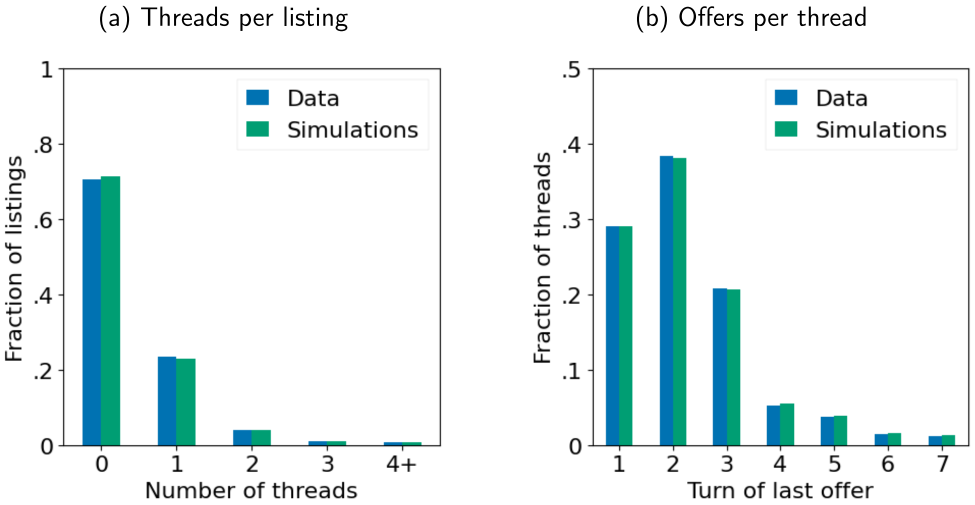
Response time: buyer turns



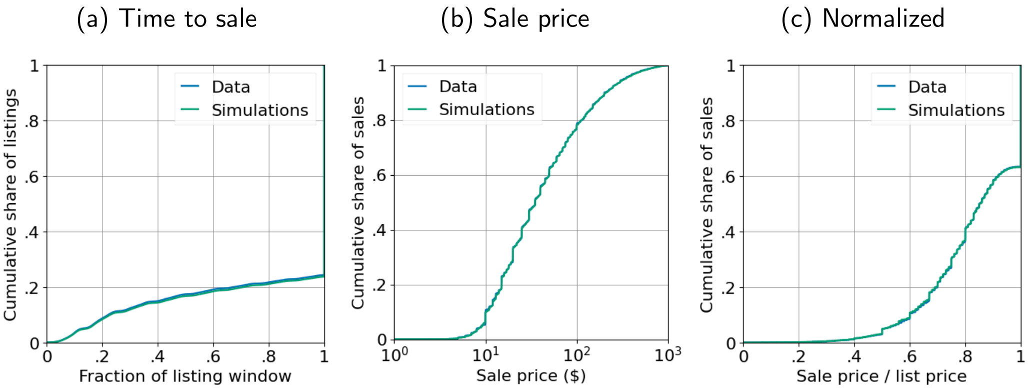
Message rates



Summary statistics

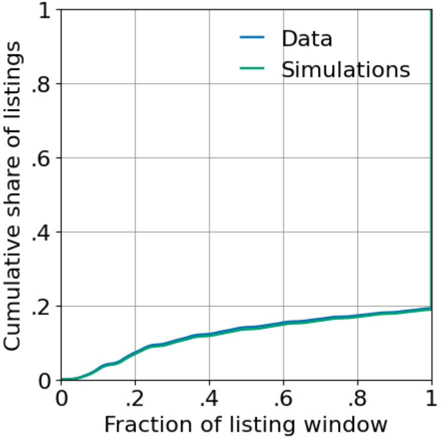


Sale time and price

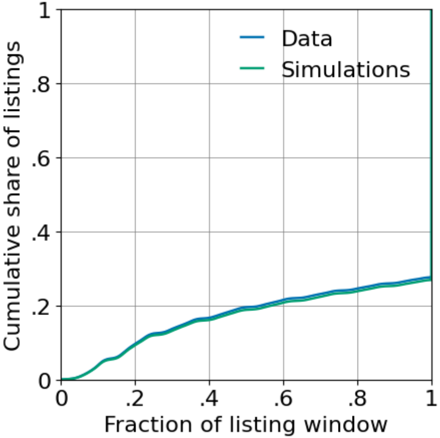


By category: time to sale

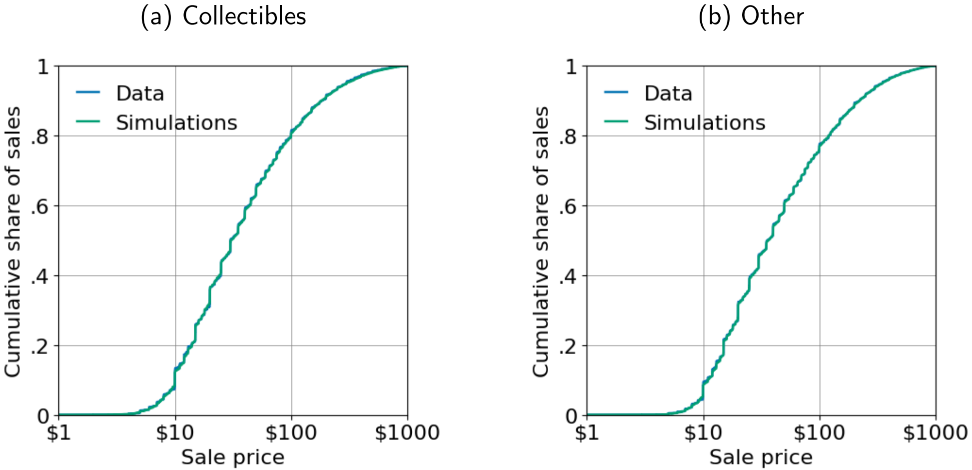
(a) Collectibles



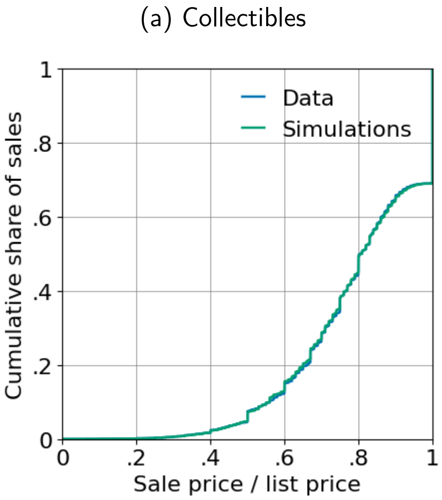
(b) Other

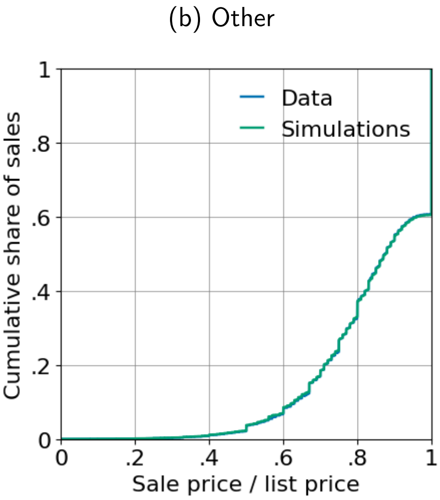


By category: sale price



By category: normalized sale price





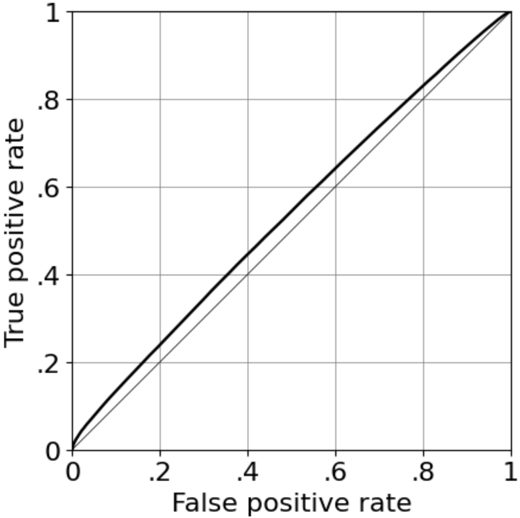
Discriminator

Observes a complete thread, either from data or simulations.

- Fixed listing features.
- Offer path.

Predicts the whether the thread is real or simulated.

Discriminator performance



AUC: 56%

Outline

Payoffs

Buyer:

- item value less sale price, if buyer purchases item
- 0, otherwise

Payoffs

Buyer:

- item value less sale price, if buyer purchases item
- 0, otherwise

Seller:

- sale price, if item sells
- item value discounted by $\delta \in [0, 1]$, otherwise

Payoffs

Buyer:

- item value less sale price, if buyer purchases item
- 0, otherwise

Seller:

- sale price, if item sells
- item value discounted by $\delta \in [0, 1]$, otherwise

What is an item's value?

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

Values

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

Values

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

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

Values

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

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

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

Values

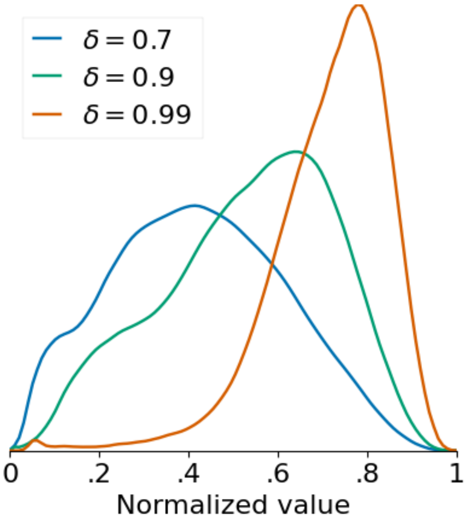
$$v = P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}] + \big(1 - P(\texttt{sale})\big) \cdot \ v$$

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

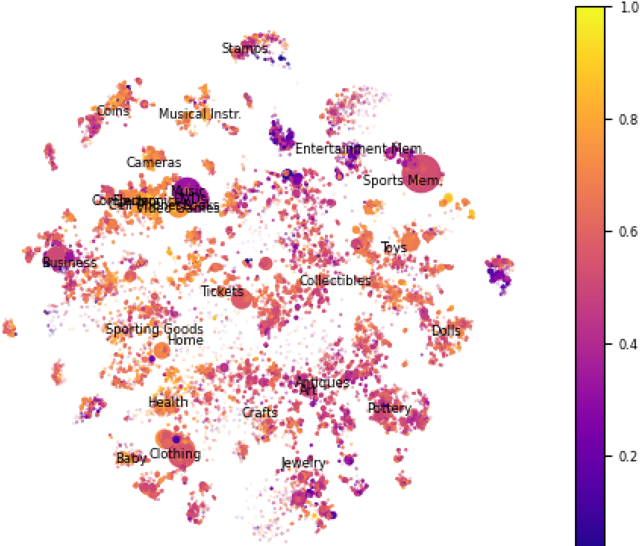
- $\delta = 0 \rightarrow \texttt{value} = P(\texttt{sale}) \cdot \mathbb{E}[\texttt{price}|\texttt{sale}]$
- $\delta = 1 \rightarrow \texttt{value} = \mathbb{E}[\texttt{price}|\texttt{sale}]$

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

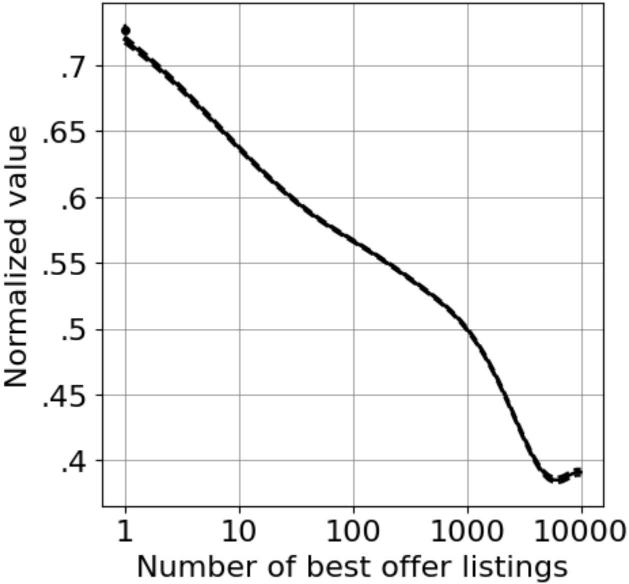
Normalized values



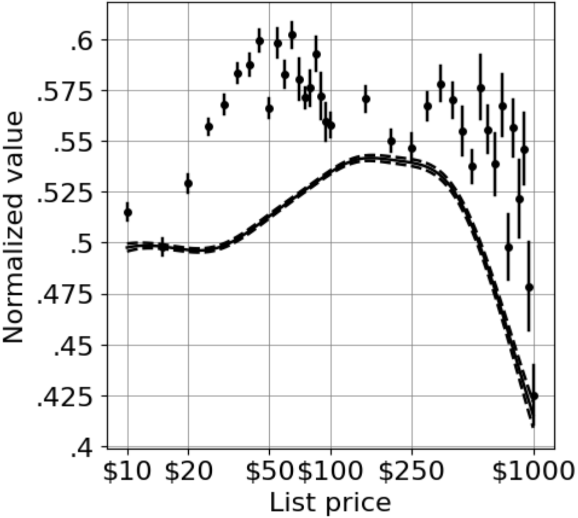
Category predicts value



Seller experience predicts value

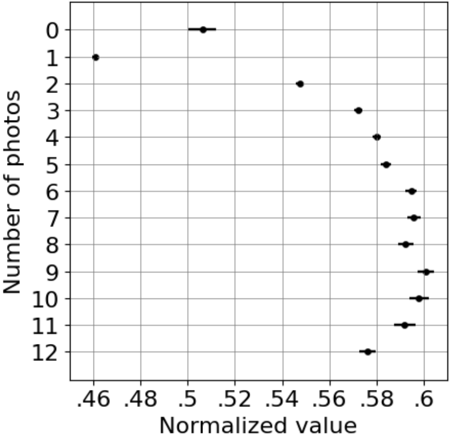


Round list prices have higher values

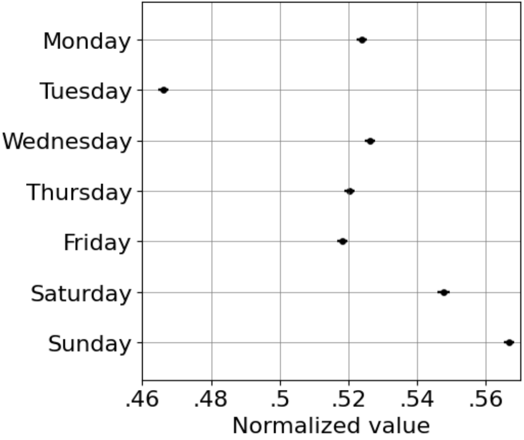


Some other predictors of value

(a) Number of photos



(b) Day of listing start



Outline

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

Training procedure

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

Initialize seller policy π.

1. Draw a listing from RL Training partition.
2. Simulate using π to draw seller offers.
3. If listing sells, payoff is `sale price`.
    - Otherwise payoff is δv.
4. Update π.

Repeat until π converges to deterministic policy.

- Simulate each listing in holdout partition 10 times.

Seller offers

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

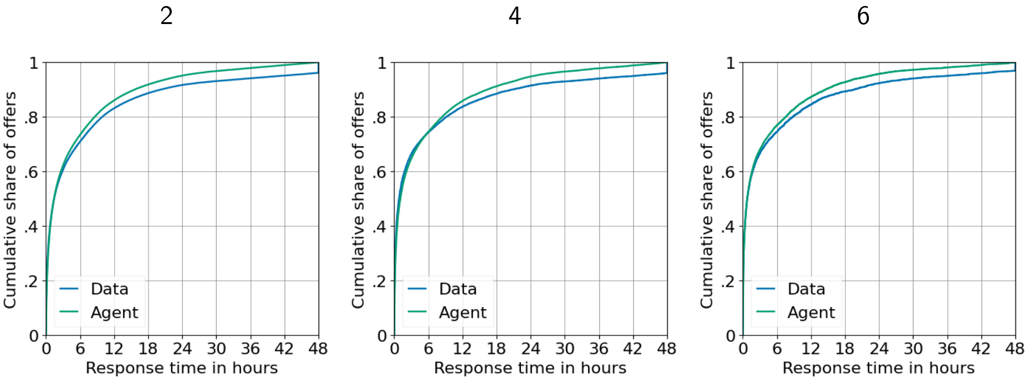
$\boldsymbol{x}$: features that are observable to seller.

- e.g., features that summarize offers on other threads.
- Excludes item value.

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

- Cannot send a message.

When does agent seller make offers?



Drawn from turn-specific distribution for simulated seller.

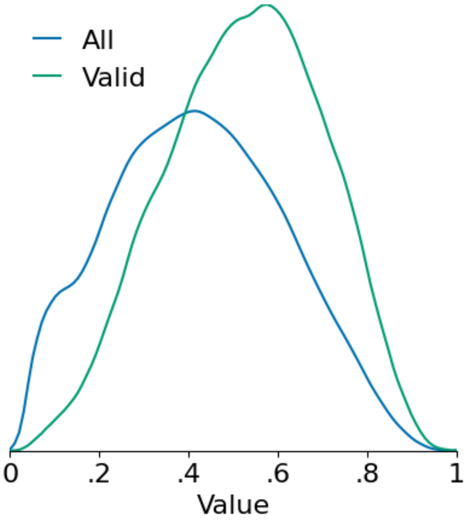
- Conditional on delay $<$ 48 hours.

Valid listings

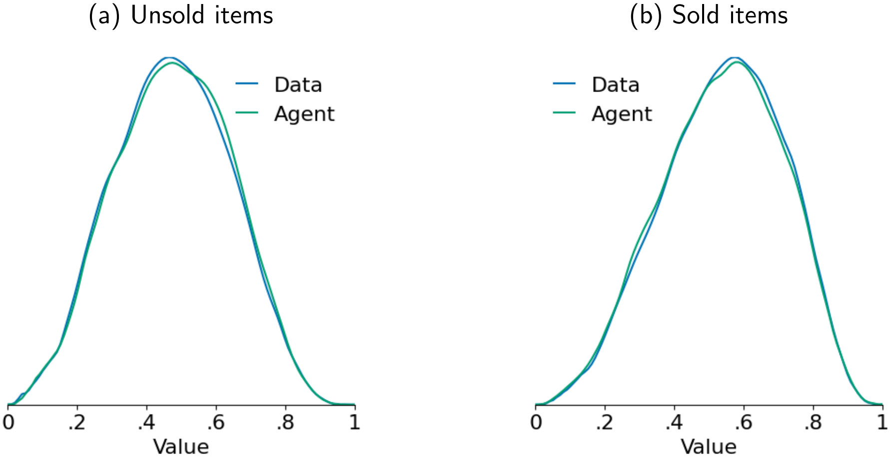
A listing is valid if agent seller makes ≥ 1 non-automatic offer.

1. i.e., a buyer arrives and
2. make an offer between the automatic thresholds and
3. seller has an opportunity to respond before listing ends.

Values ($\delta = 0.7$)



Values ($\delta = 0.7$)



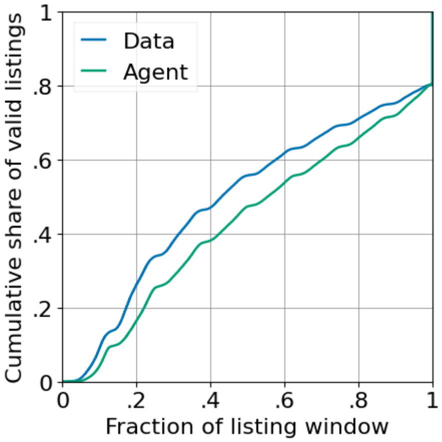
80.7% of valid listings sell in data vs. 80.3% for agent.

Sale prices

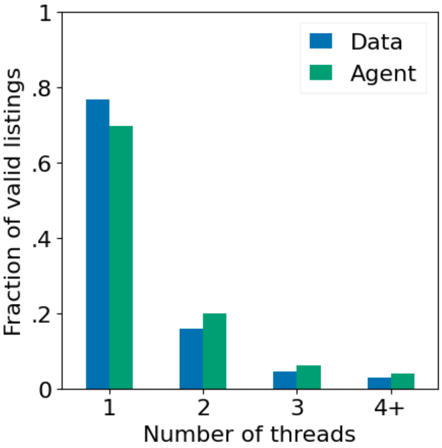
Avg normalized sale price (among sales): .76 in data / .83 for agent

Is the agent more patient than human sellers?

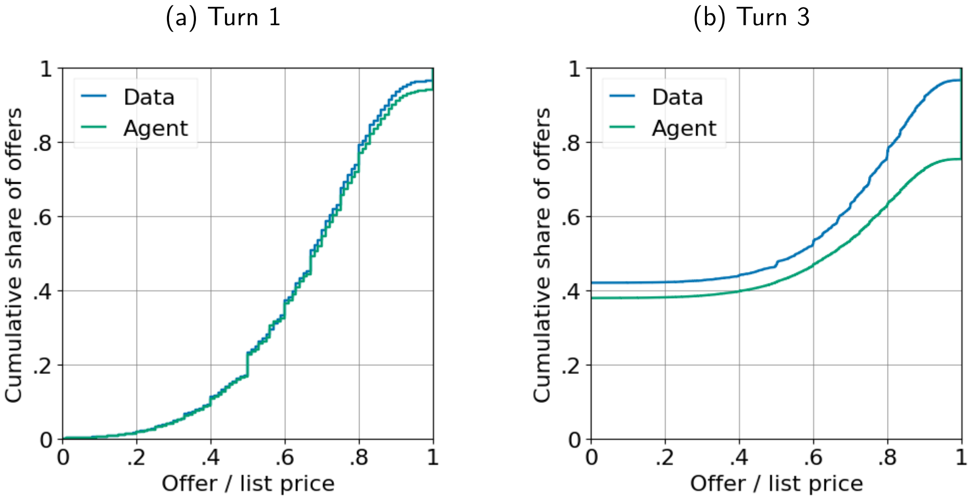
(a) Time to sale



(b) Threads per listing

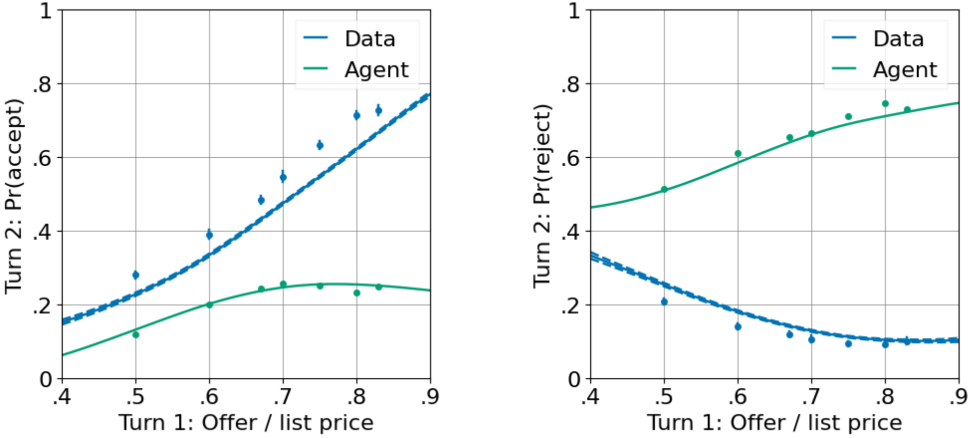


Agent seller induces full-price offers on turn 3



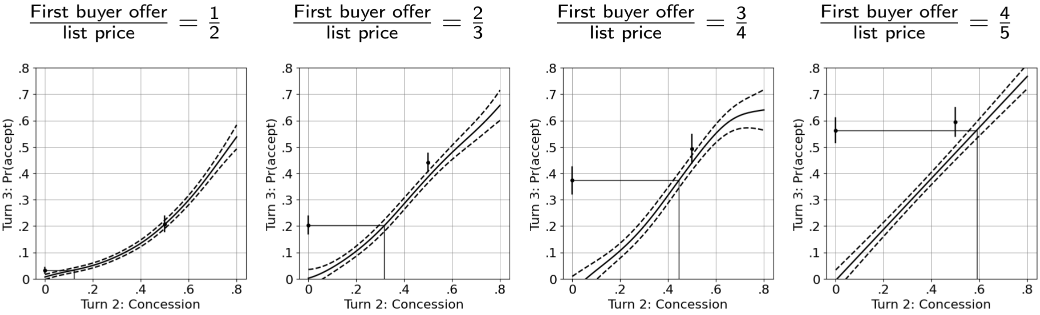
Humans and agent diverge in turn 2

(a) Accept rate (b) Reject rate



Note: excludes auto-accepts and auto-rejects.

Turn 2 rejects induce accepts



Summary so far

Human sellers accept higher first offers at higher rates.

- Consistent with comparing offer to reservation value or goal.

Summary so far

Human sellers accept higher first offers at higher rates.

- Consistent with comparing offer to reservation value or goal.

The agent seller rejects higher first offers at higher rates.

- Consistent with inferring willingness to pay from offer.

Summary so far

Human sellers accept higher first offers at higher rates.

- Consistent with comparing offer to reservation value or goal.

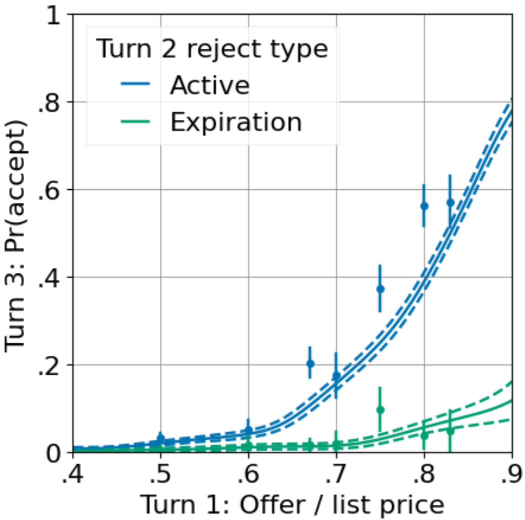
The agent seller rejects higher first offers at higher rates.

- Consistent with inferring willingness to pay from offer.

Rejections induce higher accept rates than small concessions.

- Consistent with a price integrity hypothesis.

Don't play hard to get

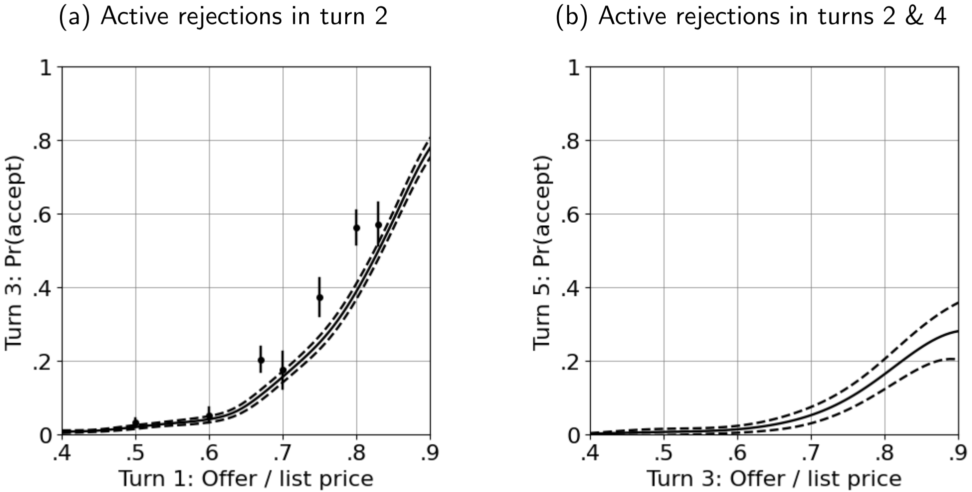


Reject rates

	Turn 2		Turn 4		Turn 6	
	Data	Agent	Data	Agent	Data	Agent
Expire	.04	-	.04	-	.03	-
Reject	.14	.63	.22	-	.22	-

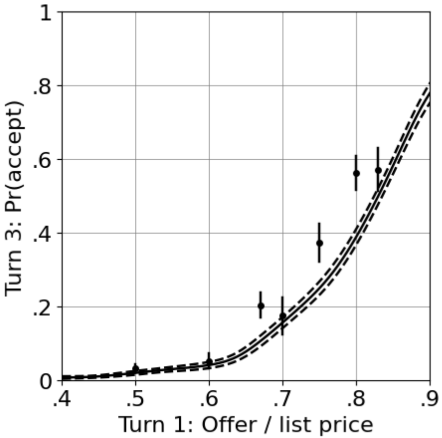
Note: excludes auto-rejects.

Active rejections don't have same effect in later turns

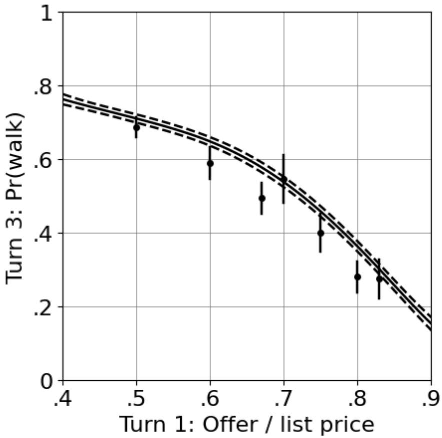


A more complete picture

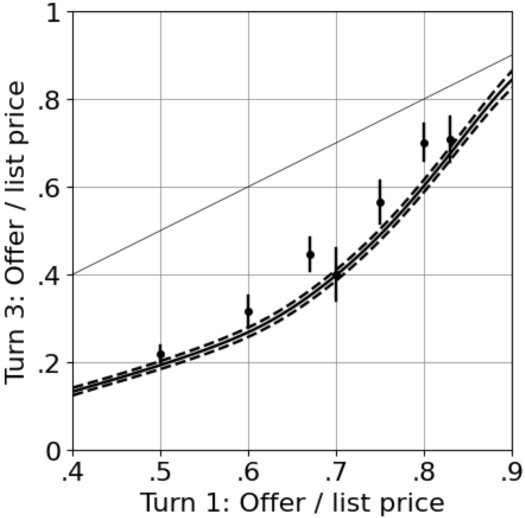
Some buyers accept...



...but others walk.



A more complete picture



Bargaining is a dynamic problem

Goal is to maximize *eventual* payoff.

- Not to maximize counterparty's response in next turn.

Bargaining is a dynamic problem

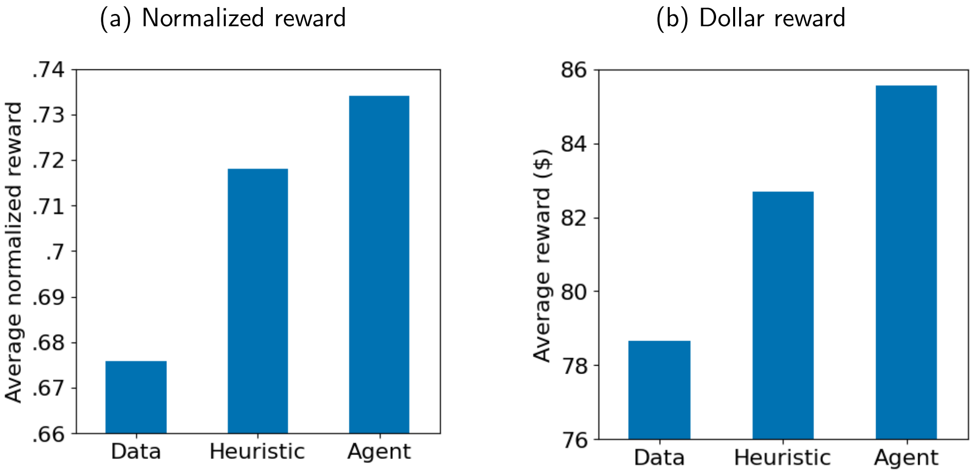
Goal is to maximize *eventual* payoff.

- Not to maximize counterparty's response in next turn.

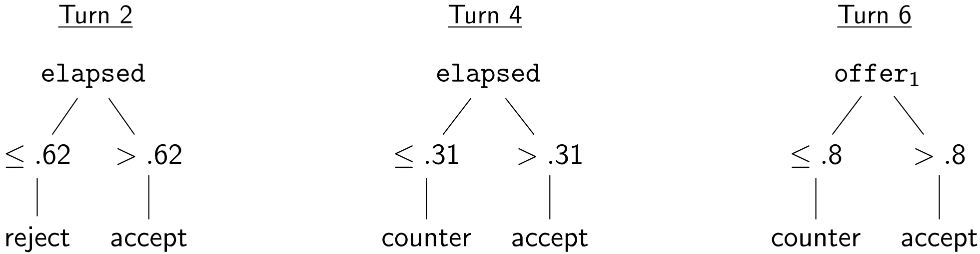
The payoff to the rejecting on turn 2 comes from:

1. Buyer acceptances on turn 3.
2. Agreements with the same buyer in a later turn.
3. An agreement with another buyer.

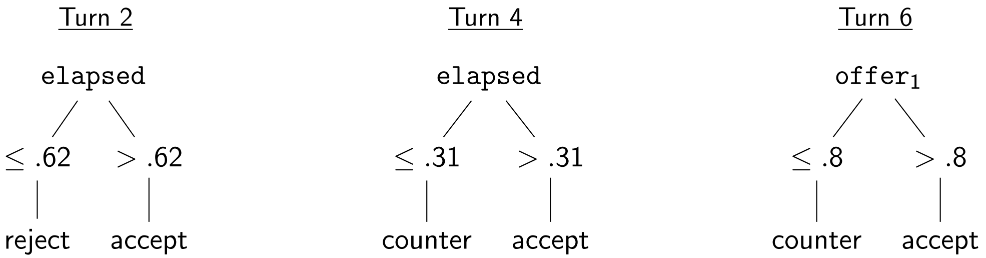
Is there a simple agent that does almost as well?



Heuristic agent



Heuristic agent



Reject early, concede later.

Heuristic human agent

Turn 2

- `offer`$_1$
  - $\leq .67$
    - counter
  - $> .67$
    - accept

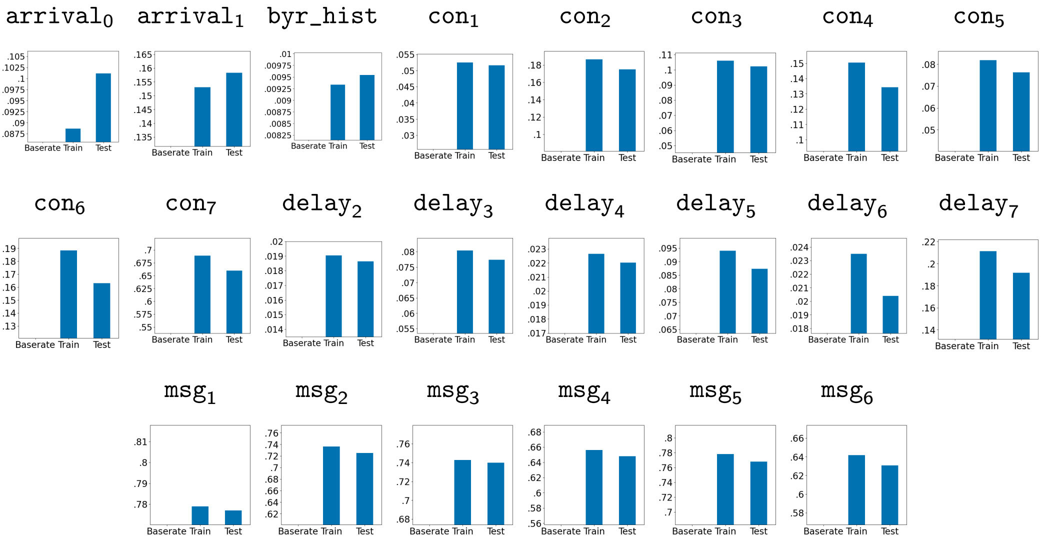
Turn 4

- `offer`$_1$
  - $\leq .66$
    - counter
  - $> .66$
    - accept

Turn 6

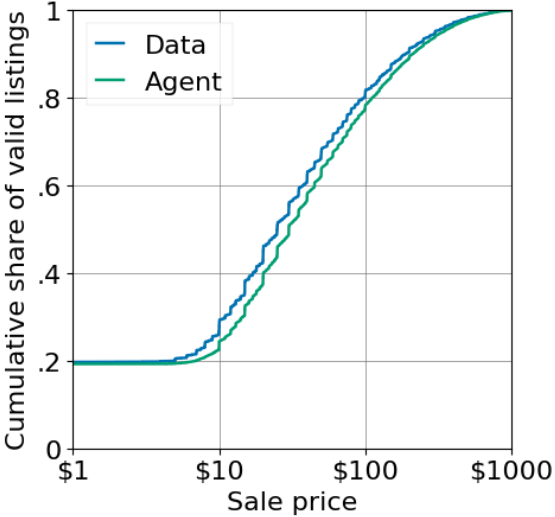
accept

Appendix



arrival0
.105 .1025 .1 .0975 .095 .0925 .09 .0875
Baserate Train Test
arrival1
.165 .16 .155 .15 .145 .14 .135
Baserate Train Test
byr_hist
.01 .00975 .0095 .00925 .009 .00875 .0085 .00825
Baserate Train Test
con1
.055 .05 .045 .04 .035 .03
Baserate Train Test
con2
.18 .16 .14 .12 .1
Baserate Train Test
con3
.11 .1 .09 .08 .07 .06 .05
Baserate Train Test
con4
.15 .14 .13 .12 .11 .1
Baserate Train Test
con5
.08 .07 .06 .05
Baserate Train Test
con6
.19 .18 .17 .16 .15 .14 .13
Baserate Train Test
con7
.7 .675 .65 .625 .6 .575 .55
Baserate Train Test
delay2
.02 .019 .018 .017 .016 .015 .014
Baserate Train Test
delay3
.08 .075 .07 .065 .06 .055
Baserate Train Test
delay4
.023 .022 .021 .02 .019 .018 .017
Baserate Train Test
delay5
.095 .09 .085 .08 .075 .07 .065
Baserate Train Test
delay6
.024 .023 .022 .021 .02 .019 .018
Baserate Train Test
delay7
.22 .2 .18 .16 .14
Baserate Train Test
msg1
.81 .8 .79 .78
Baserate Train Test
msg2
.76 .74 .72 .7 .68 .66 .64 .62
Baserate Train Test
msg3
.76 .74 .72 .7 .68
Baserate Train Test
msg4
.68 .66 .64 .62 .6 .58 .56
Baserate Train Test
msg5
.8 .78 .76 .74 .72 .7
Baserate Train Test
msg6
.66 .64 .62 .6 .58
Baserate Train Test

Sale prices

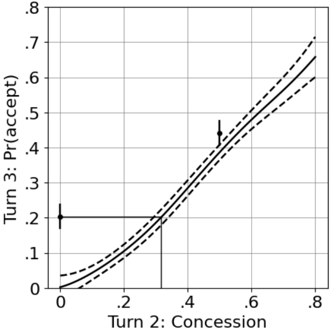


Data
Agent
Cumulative share of valid listings
0
.2
.4
.6
.8
1
$1
$10
$100
$1000
Sale price

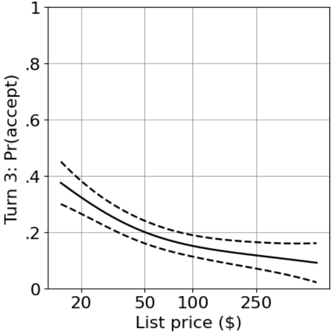
Turn 3 accept rates

$$\frac{\text{First buyer offer}}{\text{list price}} = \frac{2}{3}$$

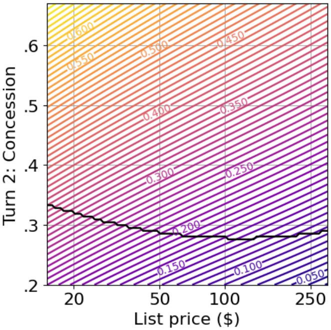
Overall



After a reject, by price

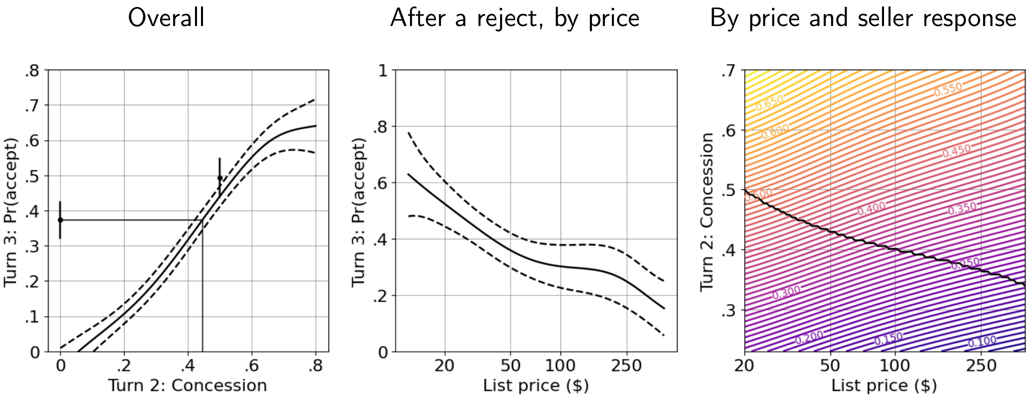


By price and seller response



Turn 3 accept rates

$$\frac{\text{First buyer offer}}{\text{list price}} = \frac{3}{4}$$



Turn 2 rejection rates