A Game-Theoretic Perspective on Trustworthy Data-Driven Algorithms

Sarah H. Cen, Andrew Ilyas, and Aleksander Mądry

MIT EECS

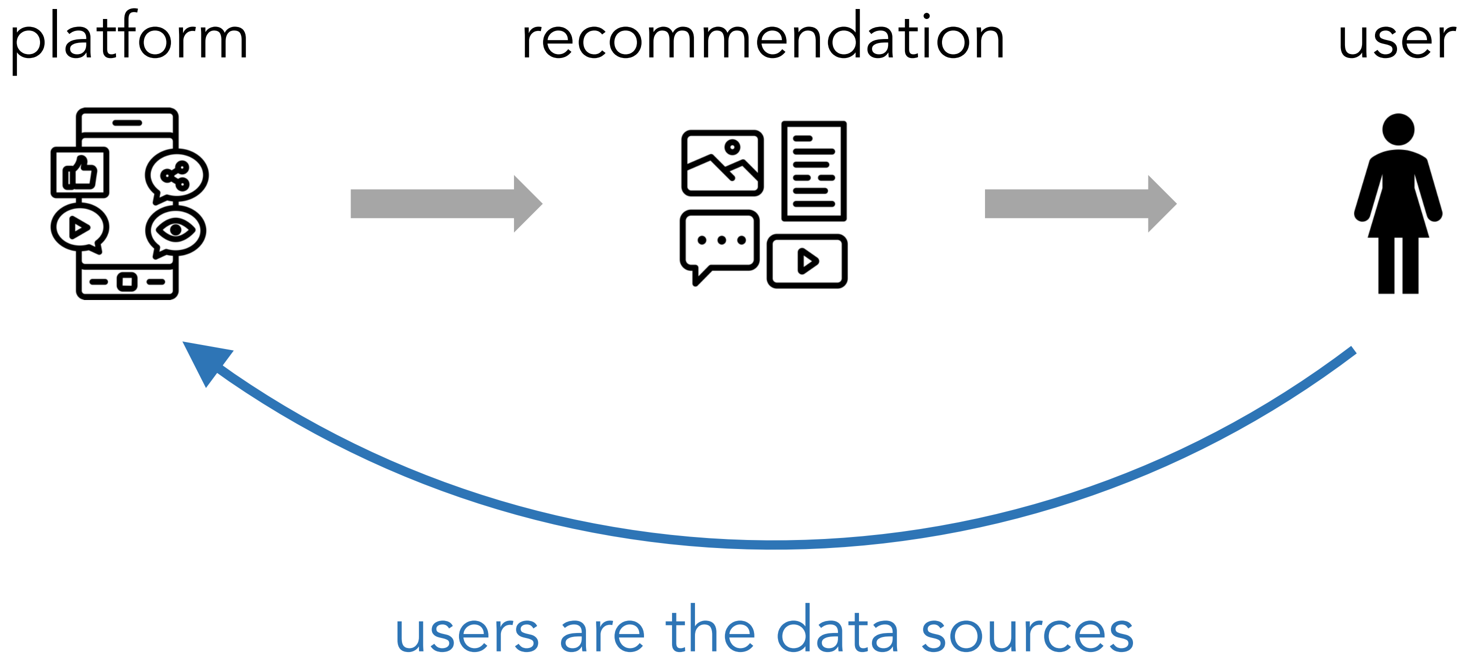
INFORMS Annual Meeting

October 16, 2023

Data-driven algorithms are built on, well, data.

Where does the data come from?

In many settings, the data comes from humans



In many settings, the data comes from humans

To make this work, typically assume that user behavior is **exogenous**

(i.e., if a *different* platform issues the same recommendation,
the user would respond in the same way)

In many settings, the data comes from humans

In practice, users can **learn**, **adapt**, and **strategize**.

(i.e., they can respond to the same recommendation differently based on the algorithm that generated it!)

Strategization is common

Example 1: Social media users

User believes platform pays too much attention to their clicks.

Avoid clicking

Search links in private mode

“Sometimes I may like a song but not thumbs-up the song because I don't want my feed filled with similar artists/videos”

[Cen, Ilyas, Allen, Li & Madry, ‘23]

Strategization is common

Example 1: Social media users

User believes platform pays too much attention to their clicks.

Avoid clicking

Search links in private mode

"I avoid reading certain news stories on Google news because I know I will be bombarded with similar articles. Instead I switch to an untracked browser to read the story."

[Cen, Ilyas, Allen, Li & Madry, '23]

Strategization is common

Example 1: Social media users

User believes platform pays too much attention to their clicks.

Avoid clicking

Search links in private mode

"I have many YouTube accounts so my algorithm does not pick up a YouTube link a friend sends me to watch"

[Cen, Ilyas, Allen, Li & Madry, '23]

Strategization is common

Example 1: Social media users

User believes platform pays too much attention to their clicks.

Avoid clicking

Search links in private mode

Example 2: Uber drivers

Driver learns that Uber represents their preferences as unimodal.

Uber ← for longer rides

for shorter rides → lyft

Strategization is common

Example 1: Social media users

User believes platform pays too much attention to their clicks.

Is user strategization a problem?

Ex

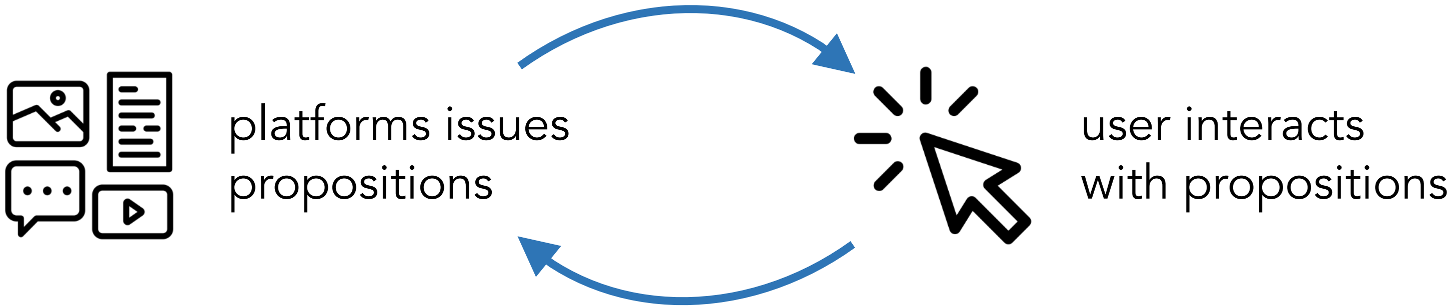
Driver learns that Uber represents their preferences as unimodal.

Uber ← for longer rides

for shorter rides → lyft

Contributions

Model: Repeated, two-player game



- Propose a model that captures user strategization
- Show strategization can help platform in short-term
- Show strategization hurts platform by misleading them
- Connect to designing **trustworthy algorithms**

Related work

Mechanism design & strategic behavior. [Myerson ’89; Nisan & Ronen ‘99; Borgers & Krahmer ‘15]

Repeated, alternating games. [Roth, et al. ‘10; Fudenberg & Tirole ‘05; Tuyls, et al. ‘18]

Strategic classification. [Hardt, et al. ’15; Levanon & Rosenfeld ‘22]

Model

Repeated, two-player game

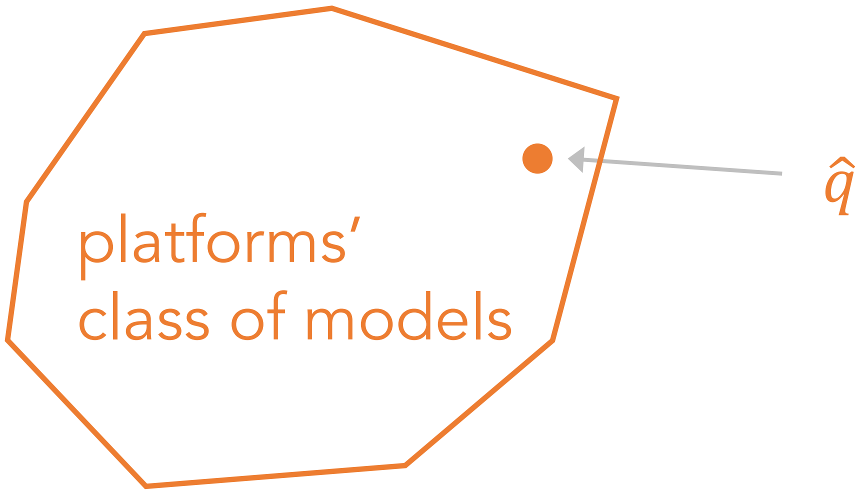
At each time step $t = 1, 2, \ldots$

Platform generates propositions Z_t

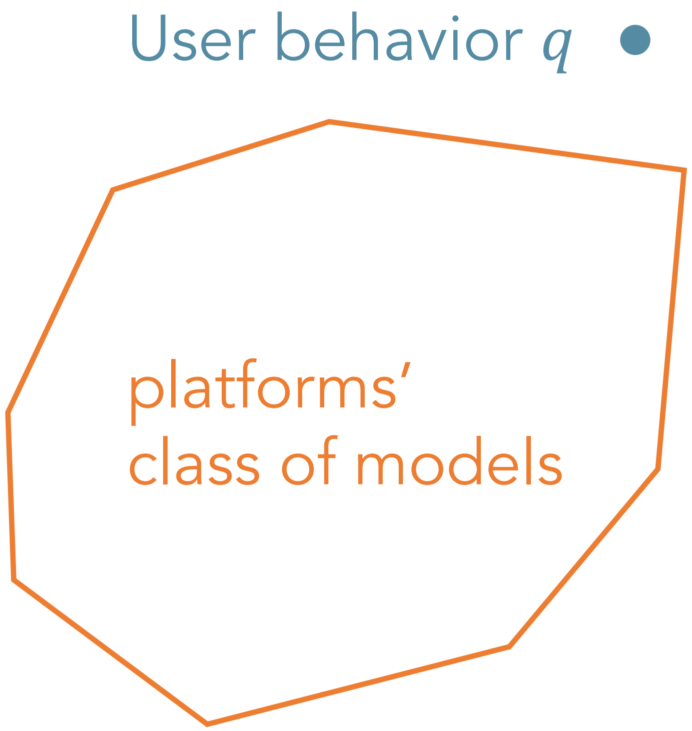
User responds with behavior $B_t \sim q(\cdot \mid Z_t)$

Platform and user collect payoffs $V(Z_t, B_t)$ and $U(Z_t, B_t)$.

To generate props, platform tries to learn model of user behavior q



Platform behavior



$\hat{Q} = \{\hat{q}_i : i \in \Omega\}$

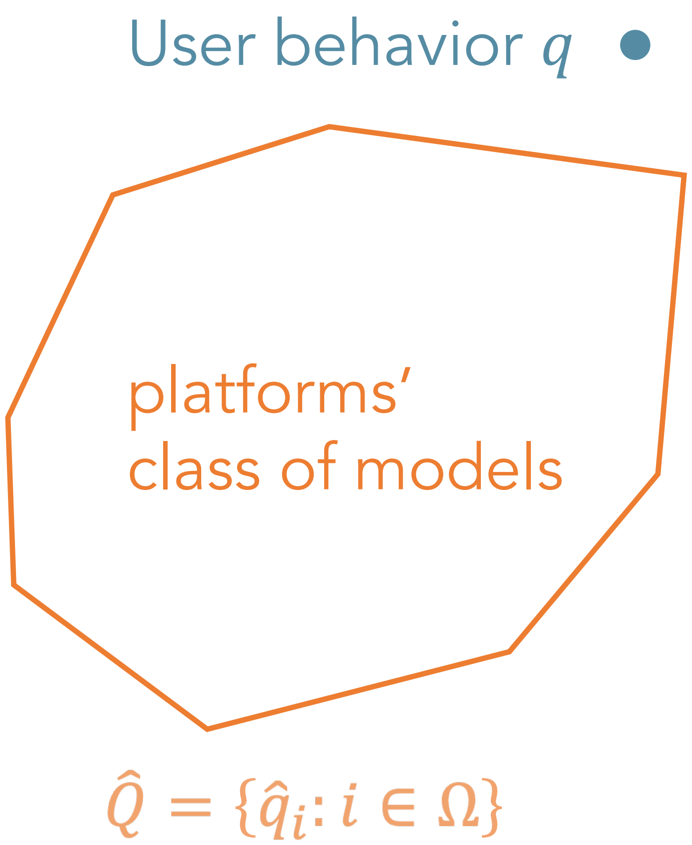
Platform maintains estimate of q

Platform updates estimate at every t based on Z_t and B_t

Bayesian update:

$$\mu_{t+1}(\omega) = \frac{\mu_t(\omega)\hat{q}_\omega(B_t|Z_t)}{\sum_{\omega' \in \Omega} \mu_t(\omega')\hat{q}_{\omega'}(B_t|Z_t)}, \quad \forall \omega \in \Omega.$$

Platform behavior



Platform maintains estimate of q

Platform updates estimate μ_t at every t based on Z_t and B_t

Platform generates propositions Z_t using an algorithm p

e.g., if it believes you like cat videos, does it show you cat videos or animal videos

That is, $Z_t \sim p(\cdot\,; \mu_t)$

Important detail

Before the game,

Platform declares $(p, \hat{Q})$

User decides q ⟵ **may depend on $(\mathbf{p}, \hat{\mathbf{Q}})$**

At each time step $t = 1, 2, \ldots$

Platform generates propositions $Z_t \sim p(\cdot\,; \mu_t)$

User responds with behavior $B_t \sim q(\cdot \mid Z_t)$

Platform and user collect payoffs $V(Z_t, B_t)$ and $U(Z_t, B_t)$.

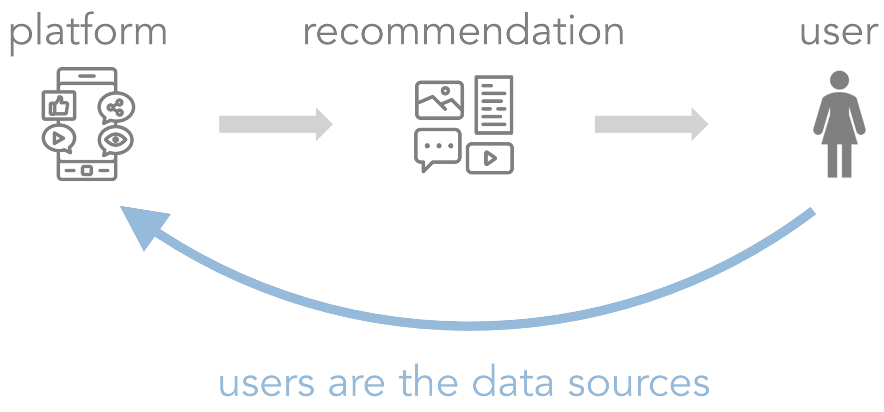
How should we model user strategization?

How should we model user strategization?

Recall: We want to capture behavior, like…

"Sometimes I may like a song but not thumbs-up the song because I don't want my feed filled with similar artists/videos"

"I have many YouTube accounts so my algorithm does not pick up a YouTube link a friend sends me to watch"



Users know that their **current** actions affect their **downstream** outcomes

Naive user vs. Strategic user

Naive user: Behaves as if they are only interacting once

$$q^{\mathrm{BR}} \propto \underbrace{\arg\max_{B \in \mathcal{B}}}_{\text{user action}} \underbrace{U(Z, B)}_{\text{user payoff}}$$

Naive user vs. Strategic user

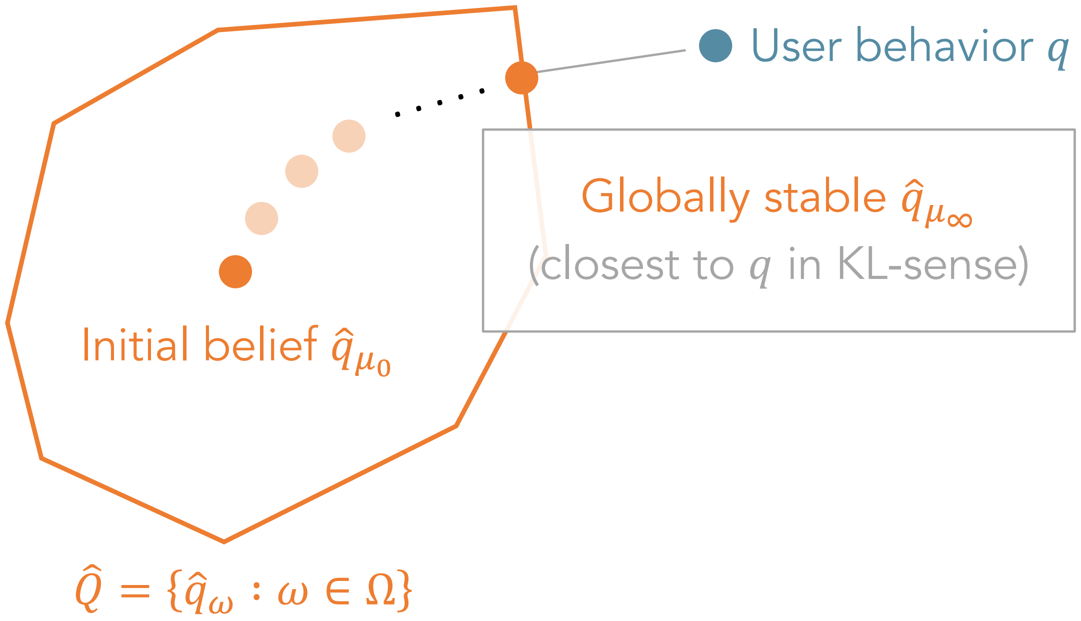
Naive user: Behaves as if they are only interacting once

Strategic user: Chooses behavior that optimizes their downstream (limiting) outcome

$$q^*(p,\Omega) \in \arg\max_{q\in\mathcal{Q}} \underbrace{\min_{\mu\in\Delta(S^{\infty}_{p,q,\Omega})} \bar{U}(p^{\mu},q)}_{\text{worst-case, limiting payoff under } q}$$

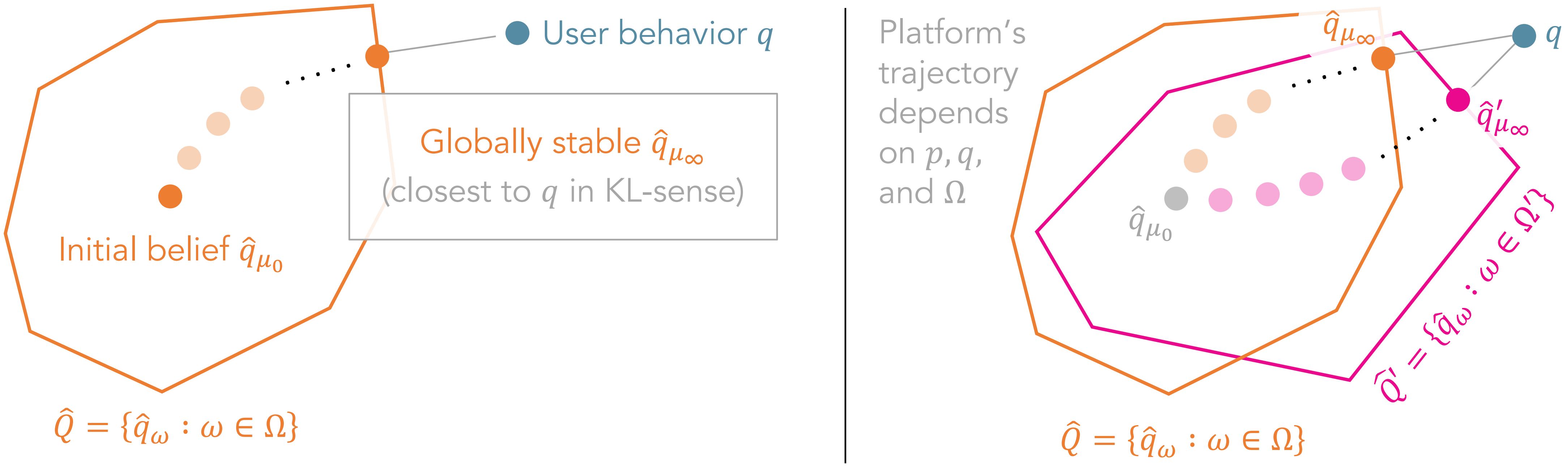
user behavior

Limiting behavior



Theorem (informal). $\hat{q}_t$ converges to models closest to q in KL-sense [Frick, Iijima & Ishii '20]

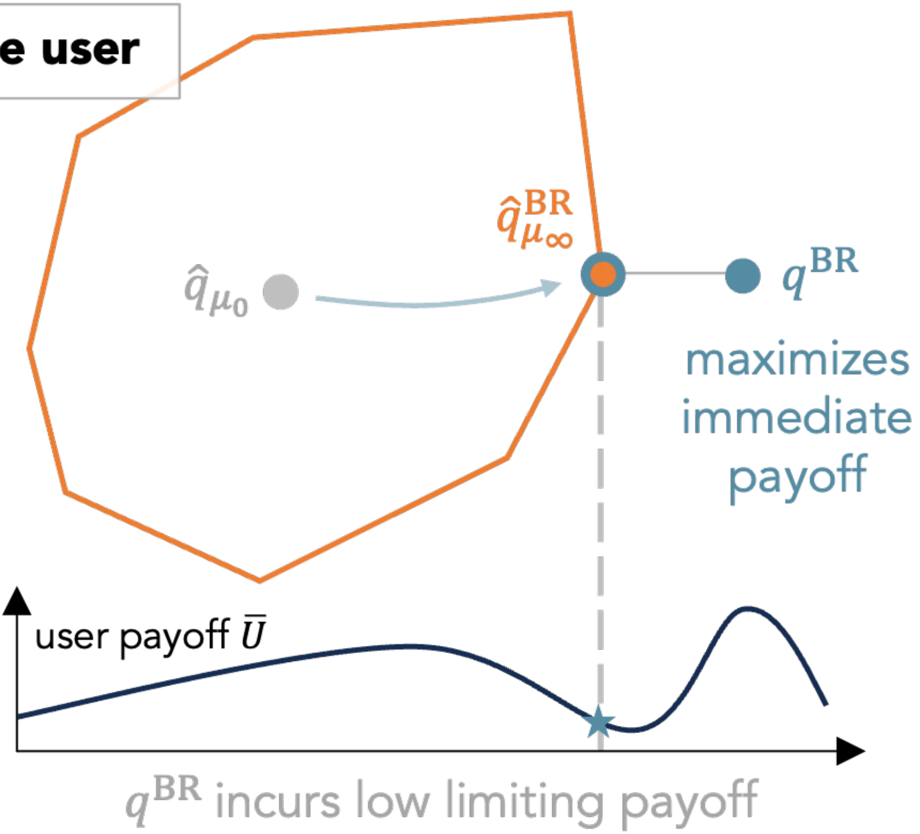
Limiting behavior



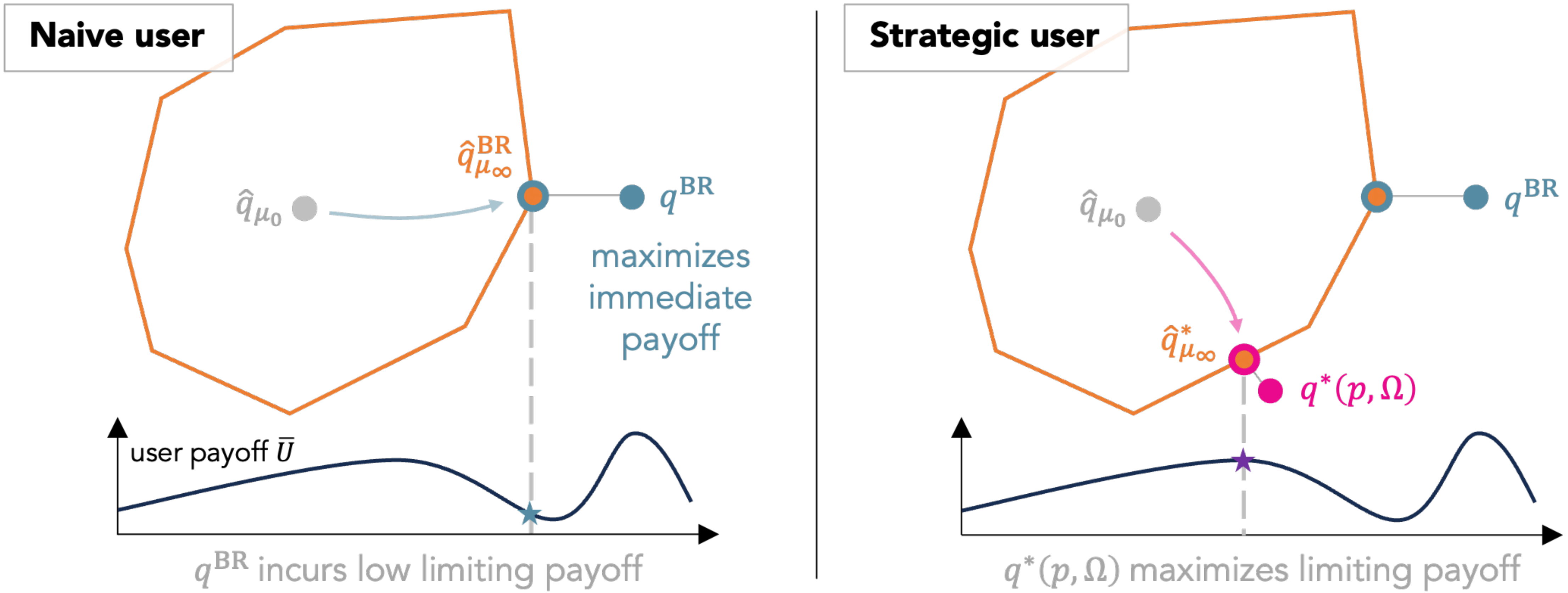
Theorem (informal). $\hat{q}_t$ converges to models closest to q in KL-sense [Frick, Iijima & Ishii '20]

Geometric intuition

Naive user



Geometric intuition



Effects of strategization

Theorem (informal). When platform and user payoffs are sufficiently aligned but platform is mis-specified, then user strategization increases the platform's payoff.

Theorem (informal). When a platform collects data under one algorithm, its estimate of its payoff under a different algorithm can be arbitrarily bad <u>when the user is strategic</u>.

Theorem (informal). A platform's payoff can <u>decrease</u> when it expands it model family if <u>the user is strategic</u>.

Trustworthy algorithms

Encapsulated interest! (Hardin)

Definition. A user trusts their platform if she is incentivized to be BR.

That is, user trusts that the platform will not misinterpret user's best-response behavior and behave optimally for the user in the long-run.

Interventions:

- Eliciting more active feedback
- Multiplicity of algorithms

Summary

Users are main data sources in many settings.

Users can adapt to platform.

This makes the data platforms collect unreliable.

How should we design algorithms under strategic users?

We provide a framework for algorithm design under strategic users. We find that participatory design improves outcomes.

Thank you!

shcen@mit.edu