Carnegie Mellon University
School of Computer Science

Algorithmic Persuasion Through Simulation

Keegan Harris Nicole Immorlica Brendan Lucier Alex Slivkins

Microsoft

Bayesian Persuasion with Unknown Types

Binary setting: State $\omega \in \{0,1\}$, Action $a \in \{0,1\}$, Sender utility $u_S(\omega, a) = a$, Receiver utility $u_R(\omega, a) = 1\{\omega = a\}$

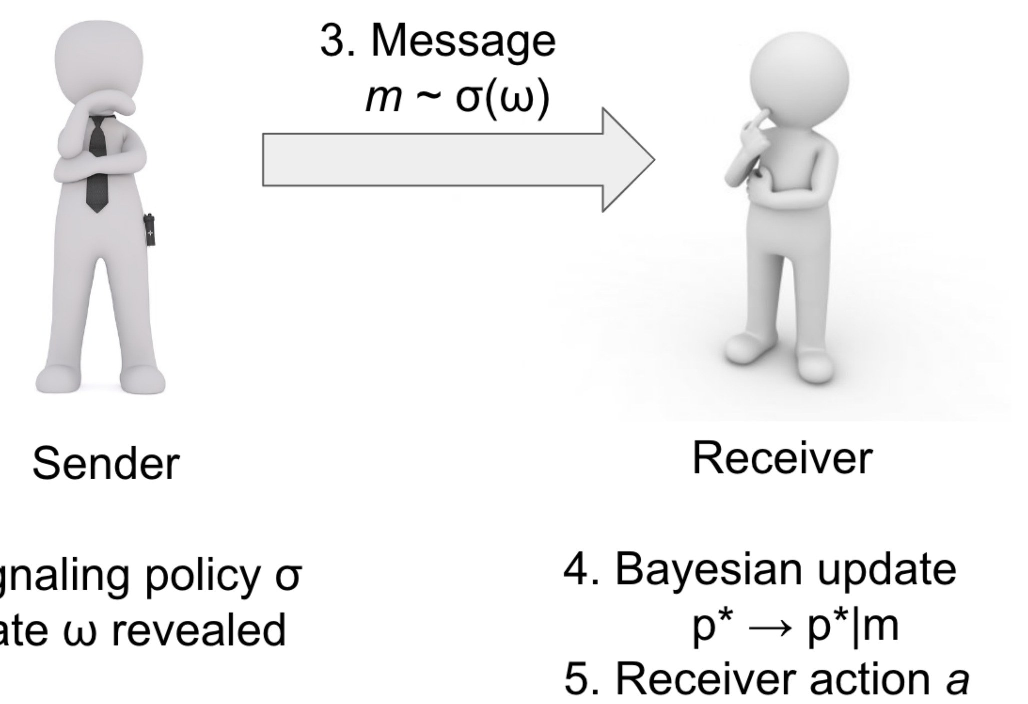


Figure 1. The Bayesian persuasion setup. The form of the sender's optimal signaling policy π^* generally depends on the receiver's prior p.

Proposition: Given a simulation query $q = (\sigma, m)$, the response a_q is equal to 1 if and only if the belief $p = Pr[\omega = 1|s]$ of the receiver conditional on the private signal s satisfies $p \geq \theta_q$, where

$$\theta_q = \frac{\sigma(m|\omega=1)}{\sigma(m|\omega=1) + \sigma(m|\omega=0)}.$$

Optimal messaging policy: $\sigma(m = m_i^*|\omega = 1) = 1$, $\sigma(m = m_i^*|\omega = 0) = \frac{p_{i^*}}{1-p_{i^*}}$, where

$$i^* := \arg\max_{i' \in [L,H]} \sum_{i=L}^{i'} \mathbb{P}_{\mathcal{P}(\mathcal{T}')}(\tau_i) \cdot \left(p_i + (1-p_i) \cdot \frac{p_{i'}}{1-p_{i'}}\right)$$

Interpretation: "give up" on persuading hard-to-persuade types in order to better-persuade other types

Our Setting

1. The sender uses querying policy π to query the simulation oracle up to K times, resulting in query history H;

2. The sender commits to a messaging policy σ_H, which is visible to the receiver;

3. The state ω is revealed privately to the sender;

4. The message $m \sim \sigma_H(\omega)$ is sent to the receiver;

5. The receiver chooses an action $a = a(m) \in \mathcal{A}$.

TL;DR

1. We study a Bayesian persuasion problem in which the sender has **incomplete information** about the receiver.
2. Motivated by customer surveys, user studies, and recent advances in generative AI, we allow the sender to learn more about the receiver by **querying** an **oracle** that simulates the receiver's behavior.
3. We design a polynomial-time **querying algorithm** that optimizes the sender's utility.
4. We also consider **approximate oracles**, more **general query structures**, and **costly queries**.

Equilibrium Computation

An **adaptive querying policy** $\pi : \mathcal{H} \to \mathcal{Q}$ is a mapping from histories to queries.

A **non-adaptive querying policy** $\pi : \mathcal{H} \times \mathbb{N} \to \mathcal{Q}$ is a mapping from the set of possible *histories* of queries $\mathcal{H}$ to queries.

Idea: Leverage **total ordering** over receiver types to compute (offline) a collection of $\min\{2^K - 1, T\}$ *possible* queries, then use **binary search** to construct an adaptive procedure given any history of responses.

Theorem: Fix $K \geq 1$. Suppose that π is the optimal non-adaptive querying policy with $\min\{T, 2^K - 1\}$ queries. There exists an optimal (adaptive) querying policy π' with K queries that only makes queries in the support of π. π' can be simulated in time $O(\min\{T, 2^K\})$ given access to π.

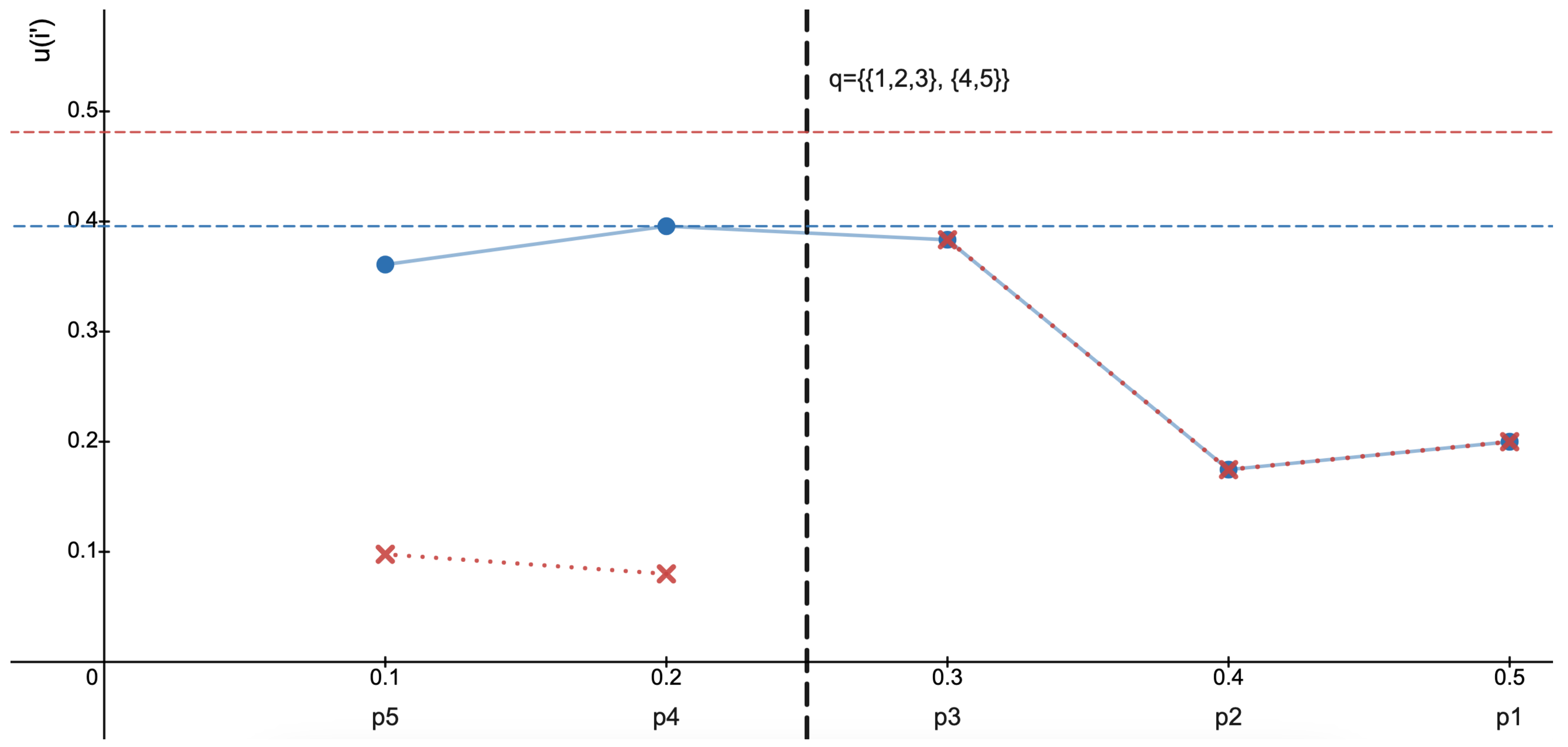
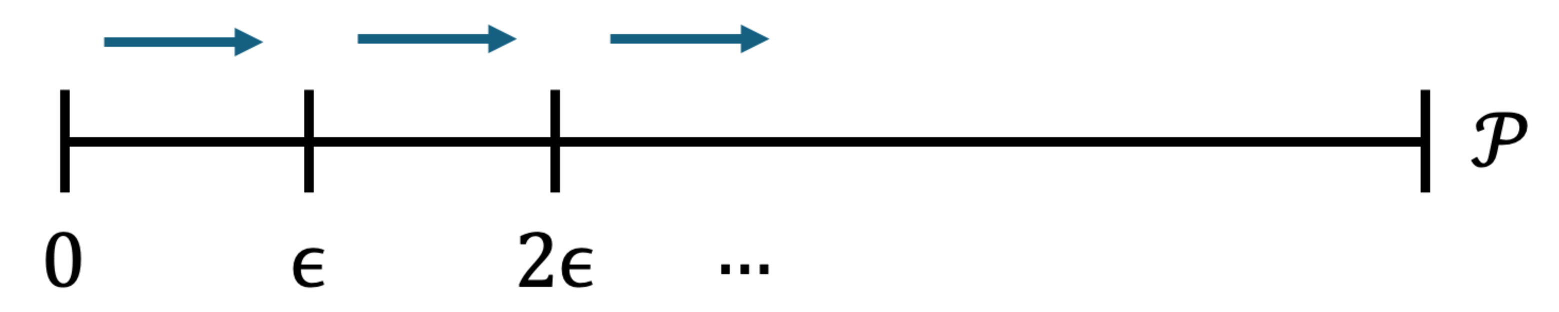


Figure 2. Sender's utility as a function of the cutoff belief. Blue solid line: sender's utility when they make no queries. Blue dashed line: sender's optimal utility. Red dotted line: sender's utility when they make the query q which separates the three highest beliefs from the two lowest. Red dashed line: sender's ex-ante utility from messaging optimally after making query q.

Approximate Oracles

Idea: Round second-order prior $\mathcal{P}$ by **increasing** each receiver belief under $\mathcal{P}$ up to ϵ, additively.



Theorem: Choose any $\epsilon > 0$. One can compute a querying policy in $O(K\epsilon^{-2})$ time and a messaging policy in $O(\epsilon^{-2})$ time, for which the sender's expected utility is at least $OPT - \epsilon$.

Corollary: Suppose that the receiver's belief is p and the oracle simulates a receiver with belief p', where $\Pr[|p' - p| > \delta] < \gamma$ for some $\delta, \gamma > 0$. Then we can compute querying and messaging policies for the sender that obtain expected payoff $(1-\gamma)OPT - O(\delta)$.

Partition Queries

Definition: A **partition oracle** with query space inputs a query $Q \in$ and returns the subset $q \in Q$ such that the receiver's belief $p \in q$.

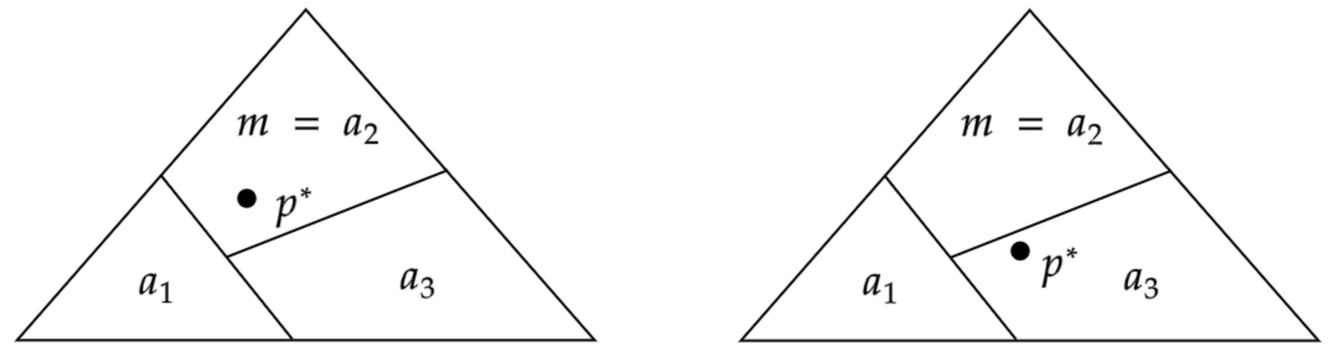


Figure 3. Visualization of receiver best-response regions of Δ^d for simulation query (σ, m). Left: A setting in which the receiver follows the sender's recommendation. Right: A setting in which the receiver does not follow the sender's recommendation.

Theorem: Finding the optimal querying policy is **NP-Complete** in Binary BP with partition queries.

Idea: Reduce from Set Cover.

- Set elements are receiver types
- Subsets are queries
- Construct instance in a way such that optimal utility is achieved iff the set is covered

keeganh@cs.cmu.edu, nicimm@microsoft.com, brlucier@microsoft.com, slivkins@microsoft.com