Reliable and Informative Interaction with Misaligned Agents

Nischal Aryal
Oregon State University
Corvallis, OR 97331
aryaln@oregonstate.edu

Arash Termehchy
Oregon State University
Corvallis, OR 97331
termehca@oregonstate.edu

Marianne Winslett
University of Illinois Urbana-Champaign
Urbana, IL 61801
winslett@illinois.edu

Abstract

Conflicts of interest often arise between data sources and their users regarding how the users' information needs should be interpreted by the data source. For example, an online product search might be biased towards presenting certain products higher in its list of results to improve its revenue, which may not follow the user's desired ranking expressed in their query. The research community has proposed schemes for data sources to implement to ensure unbiased results. However, data sources usually have little or no incentive to implement these measures, e.g., their biases often increase their profits. Thus, we propose a novel framework for querying in settings where the data source has incentives to return biased answers intentionally due to the conflict of interest between the user and the data source. We propose efficient algorithms that reformulate input queries to increase the amount of relevant information in the returned results over biased data sources. We also propose methods to detect biased information in the results of a query efficiently.

1 Introduction

Large-scale data sources, e.g., online databases and LLM-powered AI agents, are widely adopted across almost all sectors of our society, and their influence on users' decision-making continues to grow. Due to financial, social, and political factors, the incentives of data sources and their users are often not perfectly aligned. For example, companies are often said to rework their recommendation and ranking models to promote their own products and undermine those of their competitors [17, 20].

Such incentives to manipulate information are inherent at some data sources, such as shopping websites, which charge sellers a 'referral fee' that is a fraction of the price of items sold on its site [2], so that it profits more when users buy higher-priced versions of a given product. Cost-conscious users who re-sort the results by price may have to wade through many undesirable results before reaching the first good match.

Example 1.1. *A well-known shopping website (**SW** for short) provides a product-search form with sorting options and checkbox filters on attribute values such as brand, rating, and price range. On the day of this writing, a search for 'headset' returns headphones priced between $24.99 and $39.95 near the top of its Results section, and the remaining items on the page were also headphones. When re-sorted by price, the top hit on the first page is a charging cord, followed by microphones,*

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Workshop: Dynamic Alignment on Human-AI Coupled System (BiAlign Workshop).

cables, and one $1.98 headset. Only on the seventh page do headphones start to outnumber other accessories.

Data sources' incentives to manipulate information may also arise from non-financial considerations. To maximize user engagement, some social media sites may promote extreme content, which is not necessarily in anyone else's best interest [29]. More generally, data sources may return information that is biased toward or against controversial topics, political groups, or demographic groups, e.g., giving a low rank to information related to these [12, 29].

In the aforementioned settings, data sources interpret the information in user queries to take actions that do not exactly meet users' preferences and objectives and, in fact, might largely deviate from what users deem as desirable. Current proposals to combat undesirable data source actions usually require the data source to follow and implement certain protocols [29]. For example, researchers have proposed methods to remove bias and ensure fairness for data sources to implement [16, 23, 26–29]. However, data sources often do not have any incentive to implement these proposals because they conflict with their objectives and business models. Some might also argue that such restrictions limit the freedom to disseminate information or conduct business and may not be ethical or have long-term adverse impacts on our society.

Data sources and users do share at least one interest: a data source would like to satisfy the information needs behind users' queries well enough to keep them engaged. Users can leverage this partial common interest and may combat manipulation by modifying their queries so that the data source interpretations of the submitted queries are those desired by users.

Example 1.2. *Consider again the user in Example 1.1 who looks for affordable headphones on SW. If the results are biased towards relatively more expensive products, she can add a price cap to her query. For example, when we reissued the SW headset query with a $19 maximum price, the second result returned cost $21, but the others on the first page cost less than $19. Thus, SW does not take the user's query literally, but they do influence the results.*

But two can play at this game. The data source may be aware of the fact that users know about its manipulations, e.g., users have observed undesirable biases in their previous experiences. Thus, it may reason that users aim at guiding its results towards the users' desired results. Since it is not in the data source's best interest to accept the modified query as precisely the user's true information need, the data source may try to recover the original intent behind the modified query. For instance, it might also use its prior belief about users' intents, e.g., gained from external sources or previous interactions with users, to interpret the true intent behind the modified query. Consequently, users may reason about this process and modify their queries further. The agents may reason and make decisions based on possible strategies of each other iteratively to find the best method to formulate and interpret queries.

Example 1.3. *In Example 1.2, given that the user has added a degree of bias in the modified query, the data source may reason that the user has overstated their preference for headphones of a particular price, just to offset the data source's bias. Hence, it might avoid placing such headphones in the top positions, as we saw when SW still ranked a $21 headset second even after we limited the price to $19 in Example 1.2. The user might counteract this by further reducing her submitted maximum price for a headset.*

We propose a framework that models the communication of the user and the data source with conflicts of interest using queries that return ranked lists of results. We present algorithms that lead the aforementioned recursive reasoning of the users, e.g., client or middleware agent, and data source to a stable state and return relevant information for the original user query. We also propose methods to detect biased information in the results of a query efficiently. Our experiments on real-world datasets indicate that our algorithms scale to large data.

2 Framework

Our framework models the user and the data source as agents with different objectives and uses the current body of research in game theory on how agents with different goals establish informative communication [7, 8].

Query Language. A schema $\mathcal{R}$ is a set of relation symbols, each with a fixed arity and attributes whose domains are subsets of a common domain $\mathbf{dom}$; an instance I of $\mathcal{R}$ assigns a finite relation to

Table 1: Returned results for $\mathbf{q_{rating}}$ on SW.

id	model	brand	rating	price	best seller
Page 1					
e_1	Tune 770	JBL	4.5	149.95	Y
e_2	Q20i	Soundcore	4.6	39.99	N
e_3	A10 Pro	Sisism	4.7	39.50	N
e_4	Sports	Haoyuyan	3.9	299.99	N
Page 3					
e_{39}	Riff	Skullcandy	4.8	21.99	N

each symbol. Let $\mathcal{Q}$ be a query language. An a-ary query $q \in \mathcal{Q}$ maps every instance I to a totally ordered list $q(I)$ of tuples in $\mathbf{dom}^a$. We write $e \in q(I)$ for membership, $e \prec e' \in q(I)$ when e precedes e', and $e \sim e' \in q(I)$ when they share a position, and we let $\mathrm{r}(e, q(I)) \in \{1, 2, \dots\}$ denote the rank of e in $q(I)$, smaller being better. Two queries are **equivalent** if they return the same list over every instance of $\mathcal{R}$, and **set-equivalent** if they return the same set of tuples over every instance. Users express such queries in languages such as SQL, whose ranking is given by an `ORDER BY` or `ORDER BY CASE` clause [19, 21]. Our framework extends to other query languages, e.g., languages with explicit preference constructs [10, 18].

Intent and Interpretation. We call the user's true information need, which is expressed in $\mathcal{Q}$, their intent τ. Given intent τ, the user submits the query $q \in \mathcal{Q}$ to the data source. The query q may *not* be equivalent to τ, e.g., to hide the user's intent τ to offset the data source bias in its returned results for q. The data source interprets the query q submitted by the user to obtain its own desired query, i.e., the interpretation β, and returns $\beta(I)$. If there is a conflict of interest between the user and the data source, the interpreted query β may *not* be equivalent to q, e.g., if the data source is biased towards ranking certain tuples high.

Example 2.1. *Consider a user who queries SW to search for headphones. The user's intent τ might be to view headphones ordered by customer rating, which can be expressed as an SQL query with an `ORDER BY` clause on rating. To express this in SW's real-world interface, the user would first type 'headphones' into the search bar, then sorting by rating ($\mathbf{q_{rating}}$). As of this writing, when we submit $\mathbf{q_{rating}}$ to SW, the first page of interpretation β results includes top-16 results representing a variety of brands and price points, as partially shown in Table 1. The results page says that it is showing only products with ratings of 4 and up, though the Haoyuyan brand is rated 3.9. Additionally, a relatively expensive JBL product is promoted to the top. Similarly, a Skullcandy headphone is only returned on page 3 of results, though it is rated* 4.8.

Strategies. User's submitted query q is often a reformulation of the user's intent τ, e.g., to hide their intent or to offset data-source bias. The user strategy is a mapping from intents to queries. The data source interpretation is not equivalent to the submitted query. For example, if the data source is biased, given q, it computes a biased query β and returns the result of β on its database instance. The data source strategy is mapping from queries to interpretation.

Utility Functions. The utility functions formalize the agents' objectives. We denote the user's utility and the data source's utility by $U^r(\tau, \beta)$ and $U^s(\tau, \beta)$, respectively. Each maps an intent τ and an interpretation β to a real value. When the agents' objectives differ, U^r and U^s may be maximized by different interpretations for the same intent. To model this conflict, we assume the data source's utility includes a *bias function* $b(e)$ that maps each tuple e of the relation schema $\mathcal{R}$ to real values. We assume that both agents know each other's utility functions. Learning utility functions from interaction traces is a separate problem. Researchers have proposed inverse game theory and maximum-likelihood methods for learning the utility functions from traces of past agent interaction [11, 24, 25].

Example 2.2. *Continuing the running example, the user prefers the data source's results to match the user's intent. To explain this, let $U^r(\tau, \beta) = -\sum_e \big(\mathrm{r}(e, \tau) - \mathrm{r}(e, \beta)\big)^2$, where $\mathrm{r}(e, .)$ denotes the rank of the tuple e under a query. To explain the data source's objective for demoting brand B and promoting expensive products, let $U^s(\tau, \beta) = -\sum_e \big(\mathrm{r}(e, \tau) - b(e) - \mathrm{r}(e, \beta)\big)^2$, where*

$b(e) = -\mathbf{1}[\text{brand}(e) = B] + \mathbf{1}[\text{price}(e) \geq 100]$. *Then U^s is maximized by ranking tuples in increasing order of* $\mathrm{r}(e, \tau) - b(e)$.

Utility of Strategies. The data source does not know the user's intent and only observes the query q. However, we assume the data source knows the prior over the user's intent based on past interaction. Given a submitted query q, the data source updates its belief using Bayes' rule [8], obtaining the posterior $\Phi(\tau \mid q)$ and then computes its expected utility $\mathbb{E}[U^s(\tau, \beta') \mid q]$. The user's expected utility $\mathbb{E}[U^r(\tau, \beta) \mid q]$ is over the data source's strategy.

Equilibrium. An equilibrium is a stable state in which neither agent can improve their utility by changing strategy. In equilibrium, the user finds the query q that (partially) satisfies the user's objective, $q = \arg\max_{q'} \mathbb{E}[U^r(\tau, \beta) \mid q']$. The data source returns results based on an interpretation β that (partially) satisfies its objectives, $\beta = \arg\max_{\beta'} \mathbb{E}[U^s(\tau, \beta') \mid q]$. An equilibrium is influential if the data source's results change based on the queries, and users can combat manipulation by sending appropriate queries.

3 Finding Influential Queries

The data source may steer results toward sponsored, high-margin, or otherwise biased products. As a result, directly expressing an intent, e.g., searching for a particular low-priced product, may yield a ranked list dominated by promoted alternatives. A user may, therefore, submit a query that strategically adjusts the ranking of products revealed to the data source to recover relevant results.

Problem 1. *Given the interaction setting $(\mathcal{R}, \tau, U^r, U^s)$, find a query that leads to an influential equilibrium.*

We first characterize what information about the user's intent can be credibly communicated to the data source. Building on this, we present an algorithm for finding such a query. If a shopping website has an incentive to promote a product e' over e, the user's query should rank e above e' only when e is ranked sufficiently higher in the intent, so that the claim survives the data source's bias. We formalize this using relative rank constraints.

Relative rank constraints. A user aware of biased rankings may need to communicate not only that a tuple e is preferred to e', but that this preference is strong. For a result $\tau(I)$, let $\mathrm{r}(e, \tau(I))$ denote the rank of e, smaller being better. Given $e, e' \in \tau(I)$ and $\delta \geq 1$, a *relative rank constraint* is $\mathrm{r}(e', \tau(I)) - \mathrm{r}(e, \tau(I)) \geq \delta$, which says that e is ranked at least δ positions above e' in the user's intent.

Possible intent ranks. As discussed in Section 2, the data source does not know the user's intent τ, so it maintains a prior over the user's intent. Specifically, it maintains a prior over the *ranks* of tuples in $\tau(I)$. We assume each rank $\mathrm{r}(e, \tau(I))$ is uniform on $[z] = \{1, \ldots, z\}$. Given $\delta \in \{1, \ldots, z-1\}$, define $g(z, \delta) = \frac{-\delta^3 + 3\delta^2 z + \delta z^2 + \delta + z^2 - z}{3D(z,\delta)}$, $\quad s(z, \delta) = \frac{(z-\delta)(z-\delta+1)(z+2\delta-1)}{3D(z,\delta)}$, where $D(z, \delta) = z^2 - z + \delta(2z+1) - \delta^2$.

Theorem 3.1. *Given the interaction setting ($\mathcal{R}$,τ, U^r,U^s) suppose the user and data source have quadratic utility functions. For an ordered tuple pair (e, e'), let $b = b(e) - b(e')$. Let $\delta^\star(e, e')$ be the smallest integer $\delta \in \{1, \ldots, z-1\}$ satisfying*

$$g(z, \delta) - 1 \;<\; b \;\leq\; \min\big\{g(z, \delta),\, s(z, \delta)\big\}, \tag{1}$$

and set $\delta^\star(e, e') = \perp$ if no such integer exists. If $\delta^\star(e, e') \neq \perp$, then the relative rank constraint for (e, e') is $\mathrm{r}(e', \tau(I)) - \mathrm{r}(e, \tau(I)) \geq \delta^\star(e, e')$.

Informally, the bound $g(z, \delta)$ says that the data source's larger bias in favor of e requires the user to provide stronger evidence that e truly outranks e' in the intent, i.e., a larger $\delta^\star$. The bound $s(z, \delta)$ is the posterior rank gap between e and e' among the intents that violate the constraint. If $b > s(z, \delta)$, the source promotes e over e' whether or not the constraint is asserted, so both queries are sent but the returned ranking never changes and the equilibrium is not influential. Since s decreases in δ while g increases, the window (1) shrinks as the required separation grows and is empty past a threshold, and the largest usable δ grows linearly in z at a rate of about $0.36z$. Bias gaps beyond this cannot be corrected by any relative rank constraint on the pair.

Proposition 3.2. *If some pair (e, e') has $\delta^\star(e, e') \neq \perp$, then the interaction has an influential equilibrium, supported by the query asserting $\mathrm{r}(e', \tau(I)) - \mathrm{r}(e, \tau(I)) \geq \delta^\star(e, e')$ and its negation.*

Algorithm 1 Efficiently Finding an Influential Query

Input: interaction setting $(\mathcal{R}, \tau, U^r, U^s)$
Output: influential δ-query q^δ, or $\perp$
for each ordered pair (e, e') in the rank domain, in a fixed order **do**
    $\Delta b \leftarrow b(e) - b(e')$
    Find the smallest $\delta^\star \in \{1, \dots, z-1\}$ with $g(z, \delta^\star) - 1 < \Delta b \leq \min\{g(z, \delta^\star),\ s(z, \delta^\star)\}$ ▷ Theorem 3.1
    if $\delta^\star = \perp$ **then**
        continue ▷ no constraint on (e, e') can influence the data source
    end if
    if $\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) \geq \delta^\star$ **then** ▷ first pair with $\delta^\star \neq \perp$using τ
        return $q^\delta \leftarrow \big(\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) \geq \delta^\star\big)$
    else
        return $q^\delta \leftarrow \big(\mathrm{r}(e, \tau) - \mathrm{r}(e', \tau) \geq 1 - \delta^\star\big)$ ▷ negation, over integer ranks
    end if
end for
return $\perp$ ▷ no pair has $\delta^\star \neq \perp$

Definition 3.1. Given an intent τ over an instance I of schema $\mathcal{R}$, a **δ-query** q^δ is a relative rank constraint, i.e., for some $e, e' \in \tau(I)$ and $\delta \in \mathbb{Z}$, $q^\delta = \big(\mathrm{r}(e', \tau(I)) - \mathrm{r}(e, \tau(I)) \geq \delta\big)$.

Algorithm. By Proposition 3.2, one pair with $\delta^\star \neq \perp$ suffices for an influential equilibrium. Algorithm 1 therefore finds one such pair and returns a δ-query on it that is satisfied by the input intent τ. Because the user may not know the exact instance I, it operates over the rank domain induced by the schema and the intent's ranking attributes. It scans the ordered pairs in a fixed order, computes $\delta^\star$ by Theorem 3.1, and skips pairs with $\delta^\star = \perp$. At the first pair with $\delta^\star \neq \perp$, it asserts either the constraint or its negation according to whether τ satisfies it, and stops. This pair and its $\delta^\star$ depend only on the biases and z, not on τ, so every intent reports on the same pair, as in Proposition 3.2. If no pair qualifies, the algorithm returns $\perp$. Let $m = \prod_{i=1}^{l} |\operatorname{dom}(A_i)|$ be the size of the domain induced by the ranking attributes $A_1, \dots, A_l$. In the worst case, when no pair qualifies, the algorithm examines all $m(m-1) = O(m^2)$ ordered pairs, and for each it finds $\delta^\star$ by scanning $\delta \in \{1, \dots, z-1\}$ in $O(z)$ time, since g and s are closed-form. The overall time complexity is $O(m^2 z)$. Finding all such pairs is the same scan without the early stop, in the same time.

Example 3.1. *Suppose Algorithm 1 outputs, over the* `Headphones` *instance I, the query $q^\delta = \big(\mathrm{r}(e_2, \tau(I)) - \mathrm{r}(e_1, \tau(I)) \geq 1\big)$. It requires e_1 above e_2, but says nothing about e_3 and e_4. Hence q^δ is consistent with several ranked lists, e.g., $e_1 \prec e_2 \prec e_3 \prec e_4$ and $e_4 \prec e_3 \prec e_1 \prec e_2$.*

4 Empirical Study

4.1 Experiment Setup

Datasets and Queries. We evaluate our method on three real-world datasets of varying sizes, domains, and conflict-of-interest scenarios. *Amazon:* This dataset contains over 1.4M product listings with 11 attributes from Amazon [9]. We test for conflict of interest based on attributes: rating, sales in the last month, and price. *PriceRunner:* The dataset is collected from PriceRunner[1], a popular product comparison platform [1]. It includes 35k product offers with 7 attributes provided by different sellers. We test for conflict of interest based on: seller, product model, and product category. *Flights:* The dataset was scraped from EaseMyTrip[2], a flight booking service provider [5]. It contains 300k flight bookings and 12 attributes. We test for conflict of interest based on attributes: airlines, days from trip, and price. For each dataset, we generate queries using the conflict of interest attributes described above as candidate ranking attributes. Each query has the form `ORDER BY A_1, ..., A_l`, where $l \in \{1, 2, 3\}$. Varying l lets us study scalability against the number of attributes.

Bucketization. The complexity of Algorithm 1 depends on the size of the rank domain induced by the intent's ranking attributes, which can be large when attributes have high-cardinality domains. Following standard practice [3, 12], we therefore bucketize large attribute domains into a small number of groups. For example, continuous attributes such as rating can be discretized into 3–4 buckets. This choice is also motivated by practice. In many data sources, bias is naturally expressed

[1] `https://www.pricerunner.com/`
[2] `https://www.easymytrip.us/`

Table 2: Runtime by number of attributes. Runtime is in 10^{-3} seconds.

Dataset	# Attributes		
	1	2	3
Amazon	0.04	1.18	1474
Flights	0.02	0.37	60.6
PriceRunner	0.05	3.04	396.7

Table 3: Runtime and utility gain across bucket sizes. Runtime is in 10^{-3} seconds.

Dataset	Metric	# Buckets			
		2	5	12	1065
Amazon	Time	5.83	8.91	10.48	13.59
	Gain	0	0	5007	6384
Flights	Time	1.55	2.51	2.70	4.49
	Gain	0	0	34778	41576
PriceRunner	Time	1.63	2.43	2.83	4.81
	Gain	0	5536	5693	5787

at the level of *groups* of tuples, e.g., owners' brands, sponsored sellers, or coarse price bands, rather than at the level of individual tuples [12]. Since the selection of bucketization may affect the user's utility and the running time of our algorithm, we also evaluate this effect.

Baseline and metrics. We compare Algorithm 1 (IQ) against the baseline that directly submits the user's intent without reformulation. With quadratic user utility, $U^r(\tau, \beta) = -\sum_{e \in \tau(I)} (\mathrm{r}(e, \beta(I)) - \mathrm{r}(e, \tau(I)))^2$. We report utility gain as $U^r(\tau, \beta_{\mathrm{IQ}}) - U^r(\tau, \beta_{\mathrm{base}})$, so positive gain means that IQ reduces squared rank distortion relative to the original query. To model data-source bias consistently across datasets, we randomly sample normalized bias values from $[0, 1]$ and scale them to the domain of attributes in the query. All our reported results are averaged over five runs.

4.2 Experiment Results

Impact of number of attributes on execution time. We study how the rank-domain size affects runtime by varying the number of ranking attributes from 1 to 3. Table 2 shows that the algorithm is extremely fast for one and two attributes, with runtimes below 3.1×10^{-3} seconds. Runtime increases for three attributes, as expected from the larger search space, but remains practical: even the largest case, Amazon, finishes in under 1.5 seconds.

Impact of bucketization on execution time and utility. We evaluate bucketization on the highest-cardinality attribute in each dataset. We use dyadic refinement, starting from two contiguous buckets and repeatedly splitting buckets to obtain larger bucket counts. For each level, we run Algorithm 1 (IQ) and evaluate the resulting query on the full unbucketized domain, with a 10-minute budget per level. Table 3 shows that IQ scales smoothly with bucket size. Runtime remains below 14×10^{-3} seconds across all datasets, while utility gain rises quickly at small to moderate bucket sizes and then plateaus. Thus, moderate bucketization captures most of the utility gain with little runtime overhead.

5 Related Work

Responsible Data Management. As explained in Section 1, our work shares the same goal as efforts in responsible data management and science [22]. In particular, researchers have proposed several fair ranking schemes [13, 16, 23, 26, 27]. Instead of designing a fair ranking scheme, our goal is to investigate whether informative querying is possible in the presence of a conflict of interest and to enable users to extract reliable information from biased data sources. Also, unlike our setting, these algorithms often require complete information about the underlying dataset. Researchers have investigated the trade-off between the efforts of agents, e.g., school applicants, and their ranks in a ranked-based reward system, e.g., school admission, and proposed methods to design robust ranking systems [15]. However, we focus on the challenges of querying a data source where there are conflicts of interest between the user and the data source. Unlike our settings, the efforts of agents in [15] may be highly costly, e.g., time on extracurricular activities.

Strategic Communication. There is substantial research on strategic communication between agents with non-identical interests in economics and game theory [4, 6–8, 14]. Our problem and approach have important differences with the settings of strategic communication in these fields. First,

we propose computational methods to find the influential equilibria and strategies for ranking queries. Second, we address the challenges of querying with conflict of interest over large (structured) data.

References

[1] Leonidas Akritidis. Product Classification and Clustering. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C5M91Z.

[2] Amazon. Amazon selling fees, 2025. URL https://sell.amazon.com/pricing. Accessed: 2025-01-21.

[3] Abolfazl Asudeh, H. V. Jagadish, Gerome Miklau, and Julia Stoyanovich. On obtaining stable rankings. *CoRR*, abs/1804.10990, 2018. URL http://arxiv.org/abs/1804.10990.

[4] Matthew Backus, Thomas Blake, and Steven Tadelis. On the empirical content of cheap-talk signaling: An application to bargaining. *Journal of Political Economy*, 127(4):1599–1628, None 2019. doi: 10.1086/701699. URL https://ideas.repec.org/a/ucp/jpolec/doi10.1086-701699.html.

[5] Shubham Bhatwal. Flight Price Prediction. Kaggle, 2021. https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction.

[6] Andreas Blume, Ernest K. Lai, and Wooyoung Lim. Chapter 13 strategic information transmission: a survey of experiments and theoretical foundations. In *Handbook of Experimental Game Theory*. Edward Elgar Publishing, Cheltenham, UK, 2020. ISBN 9781785363320. doi: 10.4337/9781785363337.00022. URL https://china.elgaronline.com/view/edcoll/9781785363320/9781785363320.00022.xml.

[7] Archishman Chakraborty and Rick Harbaugh. Comparative cheap talk. *Journal of Economic Theory*, 132(1):70–94, 2007. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2005.07.003. URL https://www.sciencedirect.com/science/article/pii/S0022053105001729.

[8] Vincent P. Crawford and Joel Sobel. Strategic information transmission. *Econometrica*, 50 (6):1431–1451, 1982. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913390.

[9] Aaron Frias. Amazon Products Dataset. Kaggle, 2025. https://www.kaggle.com/datasets/aaronfriasr/amazon-products-dataset/data.

[10] Werner Kießling and Gerhard Köstler. Chapter 91 - preference sql — design, implementation, experiences. In Philip A. Bernstein, Yannis E. Ioannidis, Raghu Ramakrishnan, and Dimitris Papadias, editors, *VLDB '02: Proceedings of the 28th International Conference on Very Large Databases*, pages 990–1001. Morgan Kaufmann, San Francisco, 2002. ISBN 978-1-55860-869-6. doi: https://doi.org/10.1016/B978-155860869-6/50098-6. URL https://www.sciencedirect.com/science/article/pii/B9781558608696500986.

[11] Volodymyr Kuleshov and Okke Schrijvers. Inverse game theory: Learning utilities innbsp;succinct games. In *Web and Internet Economics: 11th International Conference, WINE 2015, Amsterdam, The Netherlands, December 9-12, 2015, Proceedings*, page 413–427, Berlin, Heidelberg, 2015. Springer-Verlag. ISBN 978-3-662-48994-9. doi: 10.1007/978-3-662-48995-6_30. URL https://doi.org/10.1007/978-3-662-48995-6_30.

[12] Jinyang Li, Yuval Moskovitch, and H. V. Jagadish. Detection of Groups with Biased Representation in Ranking . In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2167–2179, Los Alamitos, CA, USA, April 2023. IEEE Computer Society. doi: 10.1109/ICDE55515.2023.00168. URL https://doi.ieeecomputersociety.org/10.1109/ICDE55515.2023.00168.

[13] Jinyang Li, Yuval Moskovitch, and H. V. Jagadish. Detection of groups with biased representation in ranking, 2023. URL https://arxiv.org/abs/2301.00719.

[14] Lingxiao Li and Jia Xie. Comparative cheap talk: Evidence from the housing market. *International Real Estate Review*, 28(2):117–148, None 2025. doi: None. URL https://ideas.repec.org/a/ire/issued/v28n022025p117-148.html.

[15] Lydia T. Liu, Nikhil Garg, and Christian Borgs. Strategic ranking. *Proceedings of Machine Learning Research*, 151:2489–2518, 2022. ISSN 2640-3498. Publisher Copyright: Copyright © 2022 by the author(s); 25th International Conference on Artificial Intelligence and Statistics, AISTATS 2022 ; Conference date: 28-03-2022 Through 30-03-2022.

[16] Michael Mathioudakis, Carlos Castillo, Giorgio Barnabò, and Sergio Celis. Affirmative action policies for top-k candidates selection: With an application to the design of policies for university admissions. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, page 1640–1649, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368667. doi: 10.1145/3341105.3373878. URL https://doi.org/10.1145/3341105.3373878.

[17] Dana Mattioli. Amazon changed search algorithm in ways that boost its own products. *Wall Street Journal*, September 2019. URL https://www.wsj.com/articles/amazon-changed-search-algorithm-in-ways-that-boost-its-own-products-11568645345.

[18] Niccolò Meneghetti, Denis Mindolin, Paolo Ciaccia, and Jan Chomicki. Output-sensitive evaluation of prioritized skyline queries. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, page 1955–1967, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450327589. doi: 10.1145/2723372.2723736. URL https://doi.org/10.1145/2723372.2723736.

[19] MySQL. MySQL. https://dev.mysql.com/doc/refman/8.4/en/case.html. Accessed: 2025-02-10.

[20] Jack Nicas and Keith Collins. How apple's apps topped rivals in the app store it controls. *The New York Times*, September 2019. URL https://www.nytimes.com/interactive/2019/09/09/technology/apple-app-store-competition.html.

[21] Oracle. Oracle. https://docs.oracle.com/cd/B13789_01/appdev.101/b10807/13_elems004.htm. Accessed: 2025-02-10.

[22] Julia Stoyanovich, Serge Abiteboul, Bill Howe, H. V. Jagadish, and Sebastian Schelter. Responsible data management. *Commun. ACM*, 65(6):64–74, May 2022. ISSN 0001-0782. doi: 10.1145/3488717. URL https://doi.org/10.1145/3488717.

[23] Behzad Tabibian, Vicenç Gomez, Abir De, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. On the design of consequential ranking algorithms. volume 124, pages 171–180. PMLR, 2020. URL http://proceedings.mlr.press.

[24] Jibang Wu, Weiran Shen, Fei Fang, and Haifeng Xu. Inverse game theory for stackelberg games: the blessing of bounded rationality. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[25] Yue Yu, Jonathan Salfity, David Fridovich-Keil, and Ufuk Topcu. Inverse matrix games with unique quantal response equilibrium. *IEEE Control Systems Letters*, 7:643–648, 2023. doi: 10.1109/LCSYS.2022.3214857.

[26] Meike Zehlike and Carlos Castillo. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*, WWW '20, page 2849–2855, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366424.3380048. URL https://doi.org/10.1145/3366424.3380048.

[27] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1569–1578, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3132938. URL https://doi.org/10.1145/3132847.3132938.

[28] Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking, part i: Score-based ranking. *ACM Comput. Surv.*, 55(6), dec 2022. ISSN 0360-0300. doi: 10.1145/3533379. URL https://doi.org/10.1145/3533379.

[29] Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking, part ii: Learning-to-rank and recommender systems. *ACM Comput. Surv.*, 55(6), dec 2022. ISSN 0360-0300. doi: 10.1145/3533380. URL https://doi.org/10.1145/3533380.

A Proofs of Section 3

Proof of Theorem 3.1. Write $r = \mathrm{r}(e, \tau(I))$, $r' = \mathrm{r}(e', \tau(I))$, so (r, r') is uniform on $[z]^2$, and let $\mathcal{H} = \{(r, r') : r' - r \geq \delta\}$ and $\mathcal{H}' = [z]^2 \setminus \mathcal{H}$. Let $\Phi_x(X)$ be the posterior mean rank of $x \in \{e, e'\}$ given X, and $\Delta(X) = \Phi_e(X) - \Phi_{e'}(X)$.

(i) Best responses. Under $u_x^s = -(r_x^\tau - (r_x^\beta + b_x))^2$ with real-valued interpretation scores, the posterior expected loss is strictly convex in r_x^β, so the unique optimal response to a query with satisfying set X is $\mathrm{r}(x, \beta_X) = \Phi_x(X) - b_x$. Hence β_X ranks e ahead of e' iff $b > \Delta(X)$.

(ii) Posterior means. Summing over $\{r' - r \geq \delta\}$ gives $\Phi_e(\mathcal{H}) = \frac{z-\delta+2}{3}$ and $\Phi_{e'}(\mathcal{H}) = \frac{2z+\delta+1}{3}$, so $\Delta(\mathcal{H}) = -\frac{z+2\delta-1}{3}$. Since $\mathbb{E}[r - r'] = 0$, the law of total expectation gives $|\mathcal{H}|\Delta(\mathcal{H}) + |\mathcal{H}'|\Delta(\mathcal{H}') = 0$, with $|\mathcal{H}| = \frac{(z-\delta)(z-\delta+1)}{2}$ and $|\mathcal{H}'| = z^2 - |\mathcal{H}|$, whence $\Delta(\mathcal{H}') = s(z, \delta) > 0$.

(iii) The user's split is a threshold. The involution $\sigma(r, r') = (z + 1 - r', z + 1 - r)$ preserves $r' - r$, hence fixes $\mathcal{H}$ and $\mathcal{H}'$, and exchanges the coordinates. As the prior is uniform, $\Phi_e(X) + \Phi_{e'}(X) = z + 1$ on both cells. Under $u^r = -(r_x^\tau - r_x^\beta)^2$, the user's loss difference between $\beta_{\mathcal{H}}$ and $\beta_{\mathcal{H}'}$ is affine in (r, r') with coefficients $-2(\Phi_e(\mathcal{H}) - \Phi_e(\mathcal{H}'))$ and $-2(\Phi_{e'}(\mathcal{H}) - \Phi_{e'}(\mathcal{H}'))$; the biases cancel, and the two are negatives of each other by the previous identity. So the difference depends on (r, r') only through $r' - r$, and the user prefers $\beta_{\mathcal{H}}$ exactly when $r' - r > t$, where

$$t = \tfrac{1}{2}\big[(\mathrm{r}(e', \beta_{\mathcal{H}}) + \mathrm{r}(e', \beta_{\mathcal{H}'})) - (\mathrm{r}(e, \beta_{\mathcal{H}}) + \mathrm{r}(e, \beta_{\mathcal{H}'}))\big] = \delta - g(z, \delta) + b.$$

Because ranks are integers, t induces the partition at δ iff $t \in (\delta - 1, \delta]$, i.e., iff $b \in (g(z, \delta) - 1,\ g(z, \delta)]$.

(iv) The data source answers differently. By (i), $\beta_{\mathcal{H}} \neq \beta_{\mathcal{H}'}$ iff exactly one of $b > \Delta(\mathcal{H})$ and $b > \Delta(\mathcal{H}')$ holds, i.e., iff $\Delta(\mathcal{H}) < b \leq s(z, \delta)$. The lower bound is implied by (iii), since $g(z, \delta) - 1 \geq \Delta(\mathcal{H})$ for all $1 \leq \delta \leq z - 1$. Combining (iii) and (iv) gives (1). □

Proof of Proposition 3.2. Let $\delta = \delta^\star(e, e')$ and let $\mathcal{H} = \{(r, r') : r' - r \geq \delta\}$, with complement $\mathcal{H}'$. By Theorem 3.1, the window (1) guarantees both that users in $\mathcal{H}$ prefer the query asserting the constraint while users in $\mathcal{H}'$ prefer its negation, and that the data source's optimal interpretations on the two posterior cells are distinct. Hence the two messages and their induced posteriors form an equilibrium with different on-path responses, and therefore the equilibrium is influential. □