存在利益冲突的查询

Nischal Aryal
俄勒冈州立大学
aryaln@oregonstate.edu

Arash Termehchy 俄勒冈州立大学
termehca@oregonstate.edu

Marianne Winslett 伊利诺伊大学
winslett@illinois.edu

arXiv:2603.05704v1 [cs.DB] 5 Mar 2026

摘要

数据源与其用户之间经常会在数据源应如何解读用户的信息需求这一问题上产生利益冲突。例如，为了提高收入，在线产品搜索可能会倾向于在结果列表中将某些产品排得更靠前，这可能不符合用户在查询中表达的期望排名。研究界已提出一些方案，供数据系统实施，以确保结果无偏。然而，数据系统和服务通常几乎没有或完全没有动力实施这些措施，例如，这些偏差往往会增加其利润。本文提出了一个新颖的形式化框架，用于处理数据源因用户与数据源之间的利益冲突而有意返回有偏答案的情形。我们提出了高效算法，用于检测用户是否能够从有偏数据源中提取相关信息。我们提出了能够高效检测查询结果中有偏信息的方法。我们还提出了算法，用于改写输入查询，以增加从有偏数据源返回的结果中相关信息的数量。通过在真实世界数据集上的实验，我们表明这些算法效率高，并且能够从大型数据中返回相关信息。

1 引言

由于财务、社会和政治因素，数据源所有者（简称数据源）与其用户的激励往往并不完全一致。例如，提供产品信息的数据源可能会通过说服用户购买某些产品而在经济上获益。事实上，亚马逊、谷歌和苹果已被指控以这种方式返回存在偏差的结果 [12, 22, 24, 26]。例如，据称苹果重新调整了其排名算法，使得用户在 App Store 中搜索音乐应用时，苹果音乐排名第一，Spotify 排名第23 [26]。同样，搜索播客应用时，苹果播客排在首位，随后是12个不相关的苹果应用，最后才是其他发布者的播客应用。2018-9年搜索“TV”会将苹果应用排在最前面，而奈飞则排在一百多名 [26]。正如专家当时所评论的：“我很难相信，某些苹果应用会自然地排名高于评价更高、下载量更大的竞争对手”，以及“主导搜索结果在商业上有巨大的收益空间” [26]。

这种利益冲突在某些数据源中是固有的，例如购物网站会向卖家收取一笔推荐费，占售出商品价格的相当大比例 [2], , 因此用户购买某种产品的高价版本时，网站获利更多。注重成本的用户按价格重新排序结果后，可能要浏览许多不理想的结果，才能找到第一个合适的结果。

例 *1.1.* 写作当天，在一家知名购物网站（简称**SW**）上搜索“头戴式耳机”，其顶部的“结果”栏显示的耳机价格从 24.99 美元到 39.95 美元不等，页面其余部分的商品也确实都是耳机。按价格重新排序后，第一页排名第一的是一根充电线，接下来是入耳式耳机、麦克风、线缆，还有一款售价 1.98 美元的头戴式耳机。直到结果的第七页，耳机的数量才开始超过其他配件。

利益冲突也可能源于非财务方面的考量。为了最大限度地提高用户参与度，一些社交媒体网站可能会推广极端内容，而这未必符合其他任何人的最大利益。更一般地说，数据源返回的信息可能会对争议性话题和事业、政治团体或人口群体持有偏向或反对的偏差，例如，不返回与这些 [36, 37]相关的信息，或降低其排名。近期最广为人知的例子或许是，美国政府网站移除了所有与气候变化相关的话题 [38]。属于外国对手的数据源也可能推广有利于数据源所有者目标的内容。

在上述情境中，数据源会解读用户查询中的信息，并采取无法完全满足用户偏好和目标的行动；事实上，这些行动可能会大幅偏离用户认为可取的结果。当前应对不当数据源行为的提议，通常要求数据源遵循并实施某些协议 [32]。例如，研究人员提出了供数据源实施的消除偏差并确保公平性的方法 [21, 33–37]。然而，数据源通常没有动力实施这些提议，因为这些提议往往与其目标和商业模式相悖。有些人还可能认为，这些限制会限制传播信息或开展业务的自由，因而可能不符合伦理，或会对我们的社会产生长期不利影响。

在这种情况下，数据源和用户确实有一些共同利益：数据源希望充分满足用户查询背后的信息需求，以保持用户的参与度。用户可以利用这种部分共同利益，修改查询，使数据源对所提交查询的解释符合用户的期望。

例 *1.2*。 再来看例 1.1 中那位在 SW 上寻找价格实惠的耳机的用户。如果结果偏向价格相对较高的产品，她可以在查询中添加价格上限。例如，当我们重新提交带有 19 美元最高价格限制的 SW 头戴式耳机查询时，第二个 返回结果的价格为 21 美元，但第一页上的其他结果都低于 19 美元。因此，SW 并未严格按照价格上限筛选，但价格上限确实会影响结果。

但这场博弈并非只有一方会玩。数据源可能知道用户了解它的不良行为，例如，用户在以往使用该数据源的过程中，已经观察到它存在不良偏差。因此，数据源可能会推断，用户试图引导

数据源的解释和结果，使其朝着用户期望的结果发展，方法是修改原始查询。由于接受修改后的查询并将其视为用户真实的信息需求，并不符合数据源的最佳利益，因此数据源可能会试图还原修改后的查询背后的原始意图。例如，它也可能利用对用户意图的先验信念（例如从外部来源或与用户以往的交互中获得），来解释修改后的查询背后的真实意图。因此，用户可能会推断数据源的推理过程，并尝试更大幅度地修改查询，以影响数据源的解释。用户和数据源会基于对彼此可能采取的推理方式和策略的判断，反复推理并作出决策，以找到构造和解释查询的最佳方法。

示例 *1.3*。在示例 1.2 中，鉴于用户在修改后的查询中加入了一定程度的偏差，数据源可能会推断，用户夸大了自己对某一价格耳机的偏好，只是为了抵消数据源偏差。因此，它可能会避免将这类耳机放在排名靠前的位置；正如我们在示例 1.1 中所看到的，即使将最高价格限制为 19 美元，SW 仍将一款 21 美元的头戴式耳机排在第二位。用户可以通过进一步降低其提交的头戴式耳机最高价格来抵消这一影响。

在上述某些场景中，用户若想在不让数据源返回不理想结果的情况下获取所需信息，可以采用的暴力方法是提交一个能够检索大部分或全部数据的查询，然后对这些数据进行后处理，以提取所需信息。以示例 1.3 为极端情况，搜索数据源的用户可能会提交一个返回所有产品的查询。但这通常并不实际，因为这些数据源管理的数据量十分庞大。此外，数据源可能不会返回如此大的数据集，因为这有悖于其目标；例如，它们可能会要求用户重新提交一个只返回相对较小数据子集的查询。另外，数据源通常会限制检索信息的速率，使用户难以通过多个查询下载非常大的数据集。更重要的是，用户通常缺乏存储和后处理大型数据集所需的计算资源、时间和技能。

在这种垄断设定下，有四个重要问题。首先，我们想知道在什么条件下，用户与数据源前述的递归推理会收敛到一种稳定状态，使用户找到一个能为她提供有用信息的查询。用户或许能够利用数据源希望让用户持续参与这一点，即双方存在部分共同利益，在一定程度上说服数据源返回她想要的结果。其次，我们希望告知用户数据源返回的结果中是否包含不可信信息，例如，返回的 top-k 结果是否确实是针对该查询、基于全部数据得出的前 k 个结果。这颇具挑战性，因为数据源可能不会向用户展示所有相关信息，导致用户无法核实信息的有效性。第三，给定用户的查询，我们想知道用户能从数据源获得多少可靠信息。最后，给定一个查询，我们希望找到另一个查询，使数据源能够返回比原始查询更多的相关且可靠的信息。

本文提出了一个框架，用于对用户与数据源之间存在利益冲突时双方的策略性信息共享进行建模。该框架将用户和数据源建模为目标不同的代理，并借鉴博弈论领域关于目标不同的代理如何建立信息性沟通的现有研究 [8, 9]。基于这一框架，我们提出了能够高效解决上述问题的高效算法。我们的贡献如下：

- 我们提出了一个形式化框架，用于对用户与存在利益冲突的数据源之间的通信进行建模，其中通信通过查询返回结果的排序列表（第2节）。我们的框架定义了用户（例如客户端或中间件系统）与存在利益冲突的数据源在回答查询时进行递归推理所达到的稳定状态（不动点）。
- 如果用户与数据源之间的利益冲突过于严重，数据源可能会忽略用户查询，并不顾输入查询中表达的偏好，返回其自身偏好的排序。我们提出了高效算法，用于检查通信是否存在稳定状态，使得输入查询能够影响数据源返回答案的决策（第3节）。
- 我们提出了一种高效算法，用于检测查询答案中的不可信结果（第4节）。
- 我们证明，寻找能够为用户原始意图返回最大信息量的查询这一问题是NP难的。因此，我们提出了高效算法，用于求解若干一般类别的效用函数对应的问题（第5节）。
- 我们使用种类繁多的真实世界数据集进行了大量实证研究。实证研究表明，我们的算法能够扩展到大规模数据（第 6 节）。

2 框架

2.1 数据模型与查询语言

模式 R 是由关系符号 R_i、$1 \le i \le n$ 组成的集合，其中每个 R_i 的元数固定为 a_i。每个关系符号 R_i 都与一组属性符号（$A_{i,\ j}$、$1 \le j \le a_i$）相关联。为简化说明，我们将每个关系中各属性 $\mathbf{dom}(A_{i,\ j})$ 的可能取值域定义为单个集合 **dom** 的子集或该集合本身。于是，关系符号 R_i 的每个元组都是笛卡尔积 $\prod_{j=1}^{z_i} \mathbf{dom}(A_{i,\ j})$ 中的一个元素。R_i 的关系是这类元组组成的有限集合。R 的一个实例是其各关系符号对应的关系组成的集合。

令 Q 表示一种查询语言。模式 R 上的一个 a 元查询 $q \in \mathrm{Q}$，$a > 0$ 是一个函数，它将 R 的每个实例 I 映射为一个全序列表，列表中的每个成员都属于 dom^z。我们将查询 q 在 I 上的结果记作 $q(I)$。令 $elm(l)$ 表示全序列表 l 中的元组集合。对于 e、$e' \in elm(q(I))$，我们用 $e \prec e'$ ($e \sim e'$) 表示 e 位于 e' 之前（且 e 和 e' 处于相同位置）的情况，见于 $q(I)$。为简洁起见，我们稍作记号上的滥用，将 $e \in elm(q(I))$ 写作 $e \in q(I)$。若成对顺序 $e \prec e'$ 出现在 $q(I)$ 的结果中，则称其属于 $q(I)$。为简洁起见，我们将 $e \prec e'$ 属于 $q(I)$ 写作 $e \prec e' \in q(I)$。

用户可以在我们的框架中使用查询语言（如 SQL）来表达查询，这些查询会返回一组结果或一个排序列表，例如使用诸如 *ORDER BY CASE* 子句[13, 14, 14, 25, 27, 31]之类的排序构造。他们还可以使用所提出的构造，在查询语言中表达偏好关系 [14, 15, 23, 29]。

表1：耳机

id	型号	品牌	评分	价格	畅销品
e_1	Tune 510BT	JBL	4.5	25.20	Y
e_2	Q20i	Soundcore	4.6	39.99	N
e_3	A10 Pro	Sisism	4.8	21.99	N
e_4	运动	Haoyuyan	3.9	299.99	N

2.2 意图、查询与结果

用户意图与查询。 我们将用户在 Q 中表达的真实信息需求称为其意图。数据源不知道用户意图。用户从先验分布 π 中选取其意图，而该先验分布为数据源所知。给定意图 τ，用户向数据源提交查询 $q \in \mathrm{Q}$。查询 q 可能不等价于 τ，例如为了隐藏用户意图 τ，以抵消数据源在针对 q 返回的结果中的数据源偏差。

示例 *2.1.* 设有一位用户在 SW 中查询耳机。用户意图 τ 可能是查看按顾客评分排序的耳机：SELECT DISTINCT brand, model, rating, price FROM Products WHERE type = 'headphones' ORDER BY rating DESC。要在 SW 的实际界面中表达这一意图，用户会先在搜索栏中输入'headphones'，然后要求按评分排序（$\mathbf{q}^{\mathbf{rating}}$）。截至本文撰写时，我们向 SW 提交$\mathbf{q}^{\mathbf{rating}}$ 后，解释 β结果的第一页包含 16 种型号，涵盖多个品牌和价格区间，如表1所示的部分内容。结果页面上的查询面板显示，它只展示评分为 4 分及以上的产品，尽管 Haoyuyan 品牌的评分为 3.9。

数据源解释。 数据源会解释用户提交的查询 $q \in \mathrm{Q}$，以生成其自身期望的查询 β。如果用户与数据源之间存在利益冲突，解释后的查询 β 可能不等同于 q，例如，数据源可能倾向于将某些元组排在较高位置。

2.3 策略

用户策略。 用户策略 $P^r(q|\tau)$ 是从 Q 到 Q 的（随机）映射。给定意图 τ, P^r 返回用户提交给数据源的查询。

例 *2.2.* 接续例 2.1，为了抵消 SW 的偏差，我们提交了另一个查询$\mathbf{q}^{\mathbf{'rating}}$，要求列出非主流品牌（非 JBL）的产品，并按评分排序。结果，Skullcandy 评分为 4.8 和 4.9 的产品分别排在第一页的第 12 位和第 14 位，尽管该页上的其他所有产品评分都只有 4.7。同样，第二页的第一项是一款评分为 4.8 的 Skullcandy 头戴式耳机，而它并未出现在前一页。然而，使用 SW 的默认搜索顺序搜索"耳机"时，前 10 页结果中都没有出现 Skullcandy 的产品。这让我们设想一种情况：某个数据源对某品牌 B 的产品存在偏差，并在解释 ORDER BY子句时系统性地降低这些产品的排名，例如：ORDER BY CASE WHEN brand = 'B' THEN rating - .2 ELSE rating END DESC。在这种情况下，如果用户知道这种偏差，并且特别喜欢 B的产品，那么更符合她意图的查询是$\mathbf{q}^{\mathbf{"rating}}$ ORDER BY CASE WHEN brand = 'B' THEN .2 + rating ELSE rating END DESC。这样会先按评分顺序列出所有 B产品，然后再按评分降序列出其他品牌的型号。在 SW 界面中，用户可以通过以下方式近似实现这一点先使用搜索栏查询"耳机"，按评分排序，然后勾选优先显示 Skullcandy 耳机的复选框。我们这样操作后，第一页的结果全是 Skullcandy 产品，尽管它们实际上并非按评分顺序排列。

数据源策略。 数据源策略 $P^s(\beta|q)$ 是从 Q 到 Q 的（随机）映射。数据源使用 P^s来解释提交的查询 q背后的意图。例如，如果数据源存在偏差，给定 q，它会计算出有偏差的查询 β，并返回其数据库实例上 β的结果。

2.4 交互设置

效用函数。 用户和数据源的效用函数分别衡量对用户和数据源而言，对用户意图的解释是否理想。我们分别用 $U^r(\tau, \beta)$ 和 $U^s(\tau, \beta)$表示用户和数据源的效用函数，它们将 $\mathrm{Q} \times \mathrm{Q}$ 中的查询对映射为实数。如果用户和数据源之间存在利益冲突，那么对于同一意图 τ，U^r和 U^s会在不同的 β取值下达到各自的最大值：当 β和 τ等价时，U^r达到最大值；而对于某个不等价于 τ的 β'，U^s会达到最大值，并返回符合数据源偏差的结果。U^r和 $U^r s$的具体设定因应用和数据域而异。

示例*2.3*。 考虑示例2.1中的意图 τ，但无法访问耳机实例中的底层元组。假设只有两种可能的数据源解释：β_1：ORDER BY CASE WHEN brand='Skullcandy' THEN rating - .2 ELSE rating END，以及β_2：ORDER BY rating DESC。解释 β_2 比 β_1 更符合用户意图。为了输出一个实数评分，假设效用函数衡量排名位置上的绝对误差（斯皮尔曼脚规距离 [30]）。具体而言，假设可以访问已知模式 R 中评分的域，其中数据源的四个耳机在 τ 下的真实排名为 $[1, 2, 3, 4]$，在 β_1 下的真实排名为 $[4, 3, 2, 1]$。那么 $U^r(\tau,\ \beta_1) = -(|1-4| + |2-3| + |3-2| + |4-1|) = -8$，类似地，$U^r(\tau,\ \beta_2) = -2$，因此解释$\beta_2$ 提供了更高的用户效用。

交互设置。 一个交互设置（简称交互）是元组 $(\mathrm{R}\ \tau, U^r, U^s)$，其中 R 是数据库的模式，$\tau$是用户意图，$U^r$和 U^s分别是用户和数据源的效用函数。U^r和 U^s对于 R 上的所有用户意图都相同。

共同知识。 我们假设用户知道模式，但不知道数据源中维护的数据库实例。数据源和用户都知道所有效用函数。用户可以通过查询数据源并利用其与数据源的过往交互来了解数据源的效用函数。

2.5 交互均衡

稳定状态。如第1节所述，用户和数据源会利用各自掌握的信息，对另一方的策略进行推理，以最大化自己的效用函数。我们希望确定，这种推理是否会收敛到某种最终稳定状态，使用户和数据源最终确定一对固定策略，且任何一方都无法通过偏离自己的策略来提高效用。了解这些状态将使我们能够分析

为用户和数据源选定的策略，并设计算法，为用户在这些情境中找到理想的策略。

例 *2.4.* 考虑以下场景。耳机实例 I仅有两个元组 e_1（JBL，评分 4.5，$25.20）和 e_2（Soundcore，评分 4.6，$39.99）。用户具有两种意图之一：价格敏感 $\tau(I)$ $e_1 \prec e_2$或评分敏感 $\tau'(I)$：$e_2 \prec e_1$。数据源在两种解释之间进行选择，$\beta(I)$：$e_2 \prec e_1$和$\beta'(I)$：$e_1 \prec e_2$。假设若解释与其意图匹配，用户获得 $U^r = 2$，否则获得 $U^r = 0$。当较贵的耳机 e_2 排名第一时，数据源额外赚取佣金 $c > 0$，并在用户对结果不满时遭受损失 $L > 0$。数据源效用为佣金减去不满损失。由此得到支付矩阵：

	β $(e_2 \prec e_1)$	β' $(e_1 \prec e_2)$
$\tau(e_1 \prec e_2$	$(0,\ c-L)$	$(2,\ 0)$
τ' $(e_2 \prec e_1)$	$(2,\ c)$	$(0,\ -L)$

其中每个单元格显示（U^r，U^s）。如果 $c < L$，那么在 τ下，数据源更偏好β'（因为 $0 > c - L$），而在 τ'下更偏好 β（因为 $c > -L$）。因此，存在一个稳定状态。对于意图 τ，用户策略是提交价格查询（例如，ORDER BY price ASC），并针对 τ'提交评分查询（例如，ORDER BY rating DESC）。根据效用函数，数据源会在前一种查询后选择β'，在后一种查询后选择 β，以响应这些查询。然而，如果佣金非常高 $c \geq L$，那么即使在 τ下，数据源也会弱偏好 β（因为 $c - L \geq 0$），并在 τ'下严格偏好 β。因此，数据源会返回 β；这可能会引发新一轮推理。

贝叶斯均衡。由于数据源不知道用户意图，我们采用博弈论文献中的方法，使用贝叶斯均衡 这一概念来刻画用户与数据源推理的稳定状态 [9]。数据源的信念 Φ（$\tau|q$）是从 Q 到 Q 的随机映射。数据源的信念将用户提交的每个查询q 映射为可能意图集合上的概率分布。用户对查询 q 的期望效用 为

$$\mathbb{E}[U^r(\tau,\beta) \mid q] = \sum_{\beta \in Q} U^r(\tau,\beta) P^s(\beta \mid q).$$

数据源针对查询的 q 期望效用为

$$\mathbb{E}[U^s(\tau,\beta') \mid q] = \sum_{\tau \in Q} U^s(\tau,\beta') \Phi(\tau \mid q).$$

其中 $\Phi(\tau \mid q) = \frac{P^r(q|\tau)\pi(\tau)}{\sum_{\gamma \in Q} P^r(q|\gamma)\pi(\gamma)}$。

定义 *2.5*。交互 (R τ，U^r，U^s) 的均衡是一对用户和数据源策略 ($P^r(q|\tau)$ $P^s(\beta|q)$)，满足

- 若 $P^r(q \mid \tau) > 0$，$q = \arg\max_{q'} \mathbb{E}[U^r(\tau,\ \beta) \mid q']$。
- 若 $P^s(\beta \mid q) > 0$，$\beta = \arg\max_{\beta'} \mathbb{E}[U^s(\tau,\ \beta') \mid q]$。

3 影响的可能性

数据源可能会严重偏向于针对给定查询返回特定顺序的元组，以至于任何用户策略都无法改变其决定；也就是说，无论输入查询表达了用户希望元组按何种顺序排列，数据源都会按照自己偏好的顺序返回元组。例如，如果某个购物网站明显偏向于返回价格较高的产品，那么无论用户提交什么查询（例如，要求按照产品的质量评分返回产品），该网站都会按照价格以固定顺序返回产品。本节将探讨这些情况，并提出用于检测它们的高效算法。

3.1 有影响力的交互

如果且唯有至少存在两个查询 $q, q' \in$Q，使得 $P^r(q|\tau) > 0$和 $P^r(q'|\tau) > 0$，且 $P^s(\beta|q) \neq P^s(\beta|q')$，均衡 (P^r, P^s) 才是交互 (R τ, U^r, U^s) 中的**有影响力的** 均衡。否则，该均衡是无影响力的。如果均衡没有影响力，数据源会根据其对该均衡中用户意图的先验 π，选择能最大化其期望效用函数的解释。也就是说，数据源认为，完全依据其先验信念返回答案是最优的。

根据均衡的定义可知，每种交互都存在一个无影响力的均衡，在该均衡中，用户无论意图为何，都会发送相同的查询。如果一种交互至少存在一个有影响力的均衡，则称其为**有影响力的**；否则，称其为无影响力的。例如，在例 2.4 中，稳定状态是一个有影响力的均衡，而另一个则是无影响力的均衡。给定交互设置后，在从数据源的结果中提取可靠信息或为用户寻找理想策略之前，我们应先检查这种交互是否有影响力。

问题 1。 给定交互设置 $(\mathcal{R}\ \tau, U^r, U^s)$，判断该交互是否有影响力。

例 2.4 展示了一种用户与数据的目标极为对立的设置，因此不可能存在有影响力的交互。以下结果将这一直觉形式化。为简化论述，我们假设意图、提交的查询以及对模式 R 的解释，在模式 R 的任何实例上都不会返回重复元组。我们的结果也适用于会返回重复元组的情况。模式 R 上的两个意图是**集合等价的**，当且仅当它们在 R 的所有实例上都返回相同的答案集合。

定理 3.1。 当且仅当存在集合等价的意图 $\tau \neq \tau'$ 和解释 $\beta \neq \beta'$，并且对于用户和数据源效用函数 $U^t \in \{U^r, U^s\}$，以下条件均成立时，交互 $(\mathcal{R}\ \tau, U^r, U^s)$ 才有影响力：

$$U^t(\tau,\beta) \geq U^t(\tau,\beta') \quad \textit{and} \quad U^t(\tau',\beta') \geq U^t(\tau',\beta).$$

or

$$U^t(\tau,\beta') \geq U^t(\tau,\beta) \quad \textit{and} \quad U^t(\tau',\beta) \geq U^t(\tau',\beta').$$

我们理论结果的证明见第8节。定理 3.1 给出了用户和数据源的效用函数满足何种充要条件时，才能产生有影响力的交互。我们可以利用这一结果，通过检查 Q 中的意图和解释来判断交互是否具有影响力。然而，对于表达性意图和查询语言而言，这种方法效率太低。

3.2 高效检测有影响力的交互

本节将研究一些情形，在这些情形下，我们可以高效地检测是否存在有影响力的均衡。我们首先提出一些

效用函数满足温和且自然的条件。给定这些条件，我们提供了高效算法，用于检测是否可能存在有影响力的均衡。

偏置函数。数据源的效用可能因元组而异。考虑一个销售来自各个品牌（包括网站所有者）的产品的购物网站。由于盈利能力更高，数据源可能更偏好包含其自身产品信息的元组。我们使用偏置函数来形式化这种异质性。给定关系模式 R，偏置函数 $b(e)$将 R 域中的每个元组 e 映射为实数值（$\mathbb{R}$）。如果数据源对元组 e没有偏差，则有 $b(e) = 0$。由于数据源偏向（或反对）某些元组，对于某些元组 e，$b(e) \neq 0$。偏置函数的具体实例化因应用而异。例如，可以将偏置函数定义为元组的一个或多个属性的映射，例如品牌。

加性效用。给定意图 τ 以及模式 R 的实例 I上的数据源解释β，我们假设数据源的效用函数定义为：

$$U^s(\tau, \beta) = \sum_{e \in \tau(I)} u^s\big(\mathrm{r}(e, \tau(I)), \mathrm{r}(e, \beta(I)), b(e)\big) \tag{1}$$

这里，$\mathrm{r}(e, q(I))$ 是一个返回元组 $e \in q(I)$ 位置的函数，而 $u^s(\cdot, \cdot, \cdot)$ 是一个将位置对和偏差值映射为实数的函数。对于用户效用，我们也采用相同的可加性假设，其中偏置函数将每个元组都映射为零，因为用户不会对任何特定元组有所偏好。由于数据源对某些元组存在偏差，可能偏向或排斥这些元组，因此 $\beta(I)$ 中可能存在不属于 $\tau(I)$ 的元组。如果数据源没有返回 $\tau(I)$ 中的某些元组，我们就将 $\mathrm{r}(e, \beta(I))$ 设为一个固定值，该值大于 $\tau(I)$ 中元组的数量。在本文余下部分，我们假设所有效用函数的形式均为公式1。由于用户通常不知道确切的实例 I，因此无法准确计算 I 的 U^s 值。不过，正如我们将要展示的那样，根据 I 的模式，他们仍然可以推理效用。

示例*3.2*。考虑示例2.3中的交互，并为其指定具体的效用函数。当结果与用户的意图完全匹配时，用户获得最大效用。我们用以下函数对此建模：

$$u^r\big(\mathrm{r}(e, \tau(I)), \mathrm{r}(e, \beta(I))\big) = -\big(\mathrm{r}(e, \beta(I)) - \mathrm{r}(e, \tau(I))\big)^2$$

当解释排名等于意图排名时，该函数在 0 处取得最大值，这解释了用户为何偏好解释 β_2。为了说明数据源为何偏好解释 β_1，我们基于品牌属性定义偏置函数 b：

$$b(e) = \begin{cases} 2 & \text{if } e.\text{brand} = \text{'Skullcandy'} \\ 0 & \text{otherwise} \end{cases}$$

因此，数据源对 Skullcandy 元组 e 的效用变为$u^s_e(\cdot) = -\Big(\mathrm{r}(e, \beta(I)) - \big(\mathrm{r}(e, \tau(I)) + 2\big)\Big)^2$。因此，数据源对 Skullcandy 元组的理想输出排名比意图排名低两个位置（数值更大），所以它更偏好 β_1 而非 β_2。

部分共同利益。尽管数据源可能会偏向于返回某些元组，而非其他元组，但数据源返回的结果不应与用户意图完全无关。具体而言，当偏差相同时，返回的结果应与用户意图相关。更确切地说，如果元组 e 在意图 τ 中的位置高于其在意图 τ'中的位置，那么数据源应将 e 置于对 τ 的回答中，并使其位置不低于 e 在 τ'中的位置。

定义 3.3。给定模式 R 的一个实例 I 上的意图 τ、解释 β 和 β'，以及数据源效用函数 U^s，若对于所有满足 $\mathrm{r}(\beta(I)\ e) < \mathrm{r}(\beta'(I)\ e)$ 的元组 e，都有 $u^s(\mathrm{r}(\tau(I)\ e), \mathrm{r}(\beta(I)\ e)\ b(e)) - u^s(\mathrm{r}(\tau(I)\ e), \mathrm{r}(\beta'(I)\ e)\ b(e))$ 随 $\mathrm{r}(\tau, I, e)$ 单调不增，则称 U^s 是超模的。

在超模性的定义中，我们比较不同意图中一个元组的位置，因为改变一个意图中元组的位置会产生一个新的意图。效用函数超模的数据源不 一定会遵循用户对意图中不同元组的相对排序偏好：它仍可能更偏向于将某些元组排在高于其他元组的位置。直观而言，用户偏好最符合其偏好的结果：用户越偏好元组 e，就越希望在返回结果中看到它处于更高的位置。因此，假设 U^r 也具有超模性（定义方式与定义3.3类似）是很自然的。在本文余下部分，我们假设数据源和用户的效用函数都是超模的。

示例*3.4*。考虑示例3.2中的超模效用函数。对于每个元组，数据源在解释中的最优位置为 $\mathrm{r}(e, \beta(I)) = \mathrm{r}(e, \tau(I)) + 2$，而用户偏好满足 $\mathrm{r}(e, \beta(I)) = \mathrm{r}(e, \tau(I))$ 的解释。因此，对于每个元组的位置，都存在一定的利益共性，因为双方代理人的最优解释位置都会随着 $\mathrm{r}(e, \tau(I))$ 增大；但也存在一定冲突，因为数据源的最优解释位置始终 +2 更低。

对称偏差。数据源可能同样偏向或排斥某个意图的元组集合。例如，意图可能要求根据产品质量，列出某个特定提供商生产的产品排序列表，而数据源却偏向于排斥该提供商生产的产品。在这种情况下，数据源可能会试图劝退用户：返回其他提供商的产品，并将其排在结果的较高位置；这些产品不属于当前数据库实例中该意图的答案。我们证明，在这些情况下，该交互具有影响力。

命题 3.5。给定交互设置 $(\mathcal{R}\ \tau, U^r, U^s)$，若对所有元组 e，$b(e)$ 均相同，则该交互具有有影响力的均衡。

定理 3.6。给定交互设置 $(\mathcal{R}, \tau, U^r, U^s)$，令 U^s 为

$$U^s(\tau, \beta) = -\sum_{e \in \tau(I)} L\big(\mathrm{r}(e, \tau(I)) - (\mathrm{r}(e, \beta(I)) + b(e))\big)$$

在实例 I 上的 $\mathcal{R}$，其中 $L: \mathbb{R} \to \mathbb{R}_+$ 是一个凸函数。假设 U^r 具有相同的结构。若存在一个解释 β'，使得对于每个 e $\beta' I = \beta(I\ L(\ e\ \tau\ I - e\ \beta\ I +$，都有 $()\arg\min_{b\ e})\mathrm{r}(,())(\mathrm{r}(,()))$，则该交互不存在任何有影响力的均衡。

直观而言，定理 3.6 表明，如果数据源偏差足够大，以至于存在一种解释，无论用户期望什么意图，都能为数据源提供最大效用，那么就不可能存在有影响力的均衡，即用户无法说服数据源返回其他结果。如果 L 可微，我们就能高效地检验定理 3.6 中的条件。

后续结果直接由定理 3.6 推出，并能高效地识别一大类效用函数下的一些非影响性交互，例如二次效用函数，其中L为$(\mathrm{r}(e,\tau(I)) - (\mathrm{r}(e,\beta(I)) + b(e)))^2$。当用户关注返回结果中的前-$k$项时，这一结果尤其有用。

推论 3.7。 给定交互 $(\mathcal{R}, \tau, U^r, U^s)$，令 U^s 为

$$U^s(\tau,\beta) = -\sum_{e\in\tau(I)} L\big(\mathrm{r}(e,\tau(I)) - (\mathrm{r}(e,\beta(I)) + b(e))\big)$$

例如，I属于 $\mathcal{R}$，其中 $L(v)$： $\mathbb{R}\to\mathbb{R}_+$是凸的、关于$|v|$非递减，并在 $v=0$处取得最小值。假设 U^r具有相同的结构。如果对于每个 $b(e)$，$|b(e)| \geq k - \frac{3}{2}$，该交互不存在有影响力的均衡，其中 k是 $\tau(I)$中的元组数量。

给定 $b(.)$的范围，我们可以在常数时间内检查推论 3.7 的条件。

4 检测可信答案

数据源可能会部分偏向于针对给定查询返回某种顺序的元组。例如，购物网站可能会返回一些自有产品，其中部分产品可能与用户查询无关，同时仍返回一些相关产品以保持用户参与度。本节将研究这些设置，并介绍用于检测数据源对用户查询的解释中可靠信息的算法。

可信信息。我们首先定义本研究场景中最基本的信息单元。

定义 *4.1*。 给定一个意图 τ、实例 I上模式 R 的解释 β，若存在一个元组 $e' \in \tau(I)\setminus\{e\}$，使得 $\mathrm{r}(e',\tau(I)) < \mathrm{r}(e,\tau(I))$，且满足以下任一条件：(i) $e' \notin \beta(I)$；或 (ii) $\mathrm{r}(e',\beta(I)) > \mathrm{r}(e,\beta(I))$，则返回的元组 $e \in \beta(I) \cap \tau(I)$ 是**不可信的**。否则，e是**可信的**。

直观而言，如果数据源的解释错误地排列了意图中某个元组的位置，那么该元组就是不可信的。正如第 3.2 节所述，数据源可能因偏差而错误地排列元组，也可能返回不完整结果，例如只返回使其效用最大化的前 k 个元组。具体而言，如果某个返回的元组 e可能挤掉了意图排名高于它的某个元组 e'，即 $\mathrm{r}(e',\tau(I)) < \mathrm{r}(e,\tau(I))$，那么该元组就是不可信的。这可能是因为数据源返回了 e'，却在 $\beta(I)$ 中将其排在 e之后（即对 e进行提升），也可能是因为数据源完全没有将 e' 列入返回列表（即因遗漏而降级）。

问题 2. 给定交互设置 $(\mathcal{R}, \tau, U^r, U^s)$和解释结果 $\beta(I)$，检测 $\beta(I)$ 中的所有不可信元组。

可能意图排名。 如第 2.5 节所述，数据源不知道用户的意图 τ，因此会维护关于用户意图的先验。具体而言，它会维护关于 $\tau(I)$ 中元组排名 的先验。具体来说，对于每个元组 $e \in \tau(I)$，我们假设排名 $\mathrm{r}(e, \tau(I))$ 是一个取值范围为 $[z] = \{1, 2, \ldots, z\}$ 的离散随机变量，并且在先验 $[z]$下服从均匀分布。

无差异边界。 如第 3.2 节所述，数据源的效用函数具有超模性，这在一定程度上促使数据源将意图中位置较高的元组放在解释中相对较高的位置。具体而言，当解释颠倒意图中元组的相对顺序时，如果这些元组的意图排名相距越远，数据源遭受的效用损失就越大。然而，数据源的偏差仍可能导致其选择能带来更高效用、但错误排序这些元组的解释。尤其是当两个元组在意图中的位置足够接近时，数据源因错误排序它们而承受的效用损失较小，可能会被偏差抵消。为了刻画偏差在何种情况下足以促使数据源错误排序元组，我们将说明数据源和用户如何比较不同解释所带来的效用。

定义 *4.2*. 给定意图 τ 以及模式 R 的实例 I上的解释 β, β'，若 $U^r(\tau,\beta) = U^r(\tau,\beta')$，则用户对 β和β'无差异。给定关于意图的后验信念 Φ，若 (,) (,)，则数据源对 β β' Φ $\mathbb{E}_{\tau\sim\Phi}[U^s\ \tau\ \beta] =$和 at 无差异。$\mathbb{E}_{\tau\sim\Phi}[U^s\ \tau\ \beta']$

直观地说，如果用户对任意两个不同的解释获得相同的效用，那么用户对它们无差异。然而，当用户并非无差异，即严格偏好其中一种解释时，他们会策略性地选择能够说服数据源基于偏好的解释返回结果的查询。因此，用户无差异的条件实际上定义了可能意图空间中的决策边界；等价地说，它也定义了意图中元组可能位置的决策边界，用户会据此提交不同的查询。数据源的无差异条件以类似方式定义，不同之处在于，数据源会在其后验信念 Φ 下评估效用，因为它无法观察到真实意图（第 2.5 节）。接下来，我们将介绍二次效用函数的无差异条件结构。

命题 4.3。 给定交互设置 $(\mathcal{R}, \tau, U^r, U^s)$，假设用户和数据源对每个元组 e具有以下效用函数：

$$u^r(\mathrm{r}(e,\tau(I)),\mathrm{r}(e,\beta(I))) = -(\mathrm{r}(e,\tau(I)) - \mathrm{r}(e,\beta(I)))^2,$$

$$u^s_e(\mathrm{r}(e,\tau(I)),\mathrm{r}(e,\beta(I))) = -(\mathrm{r}(e,\tau(I)) - (\mathrm{r}(e,\beta(I)) + b(e)))^2.$$

在 R 的实例 I 上。用户在解释β和 β'之间无差异，当且仅当 $\sum_{e\in E}\gamma_e r(e,\ \tau) = \alpha^r$，其中法向量为 $\gamma_e = 2(r(e,\ \beta(I)) - r(e,\ \beta'(I)))$。令 Φ为关于意图的信念，并令 $\Phi_e := \mathbb{E}_{\tau\sim\Phi}[r(e,\ \tau(I))]$。数据源在信念 Φ下无差异，当且仅当 $\sum_{e\in E}\gamma_e\Phi_e = \alpha^s$，$\alpha^s$: $=$ $\alpha^r + 2\sum_{e\in E} b\ e\ (\ e\ \beta\ I\ -\ e\ \beta'\ I\ )$. $(\ )\ r(\ ,\ (\ ))\ r(\ ,\ (\ ))$

证明方法是令 β 和 β' 的效用相等，然后化简所得表达式。命题 4.3 表明，代理人的无差异条件具有仿射形式。具体而言，对于用户，在可能意图排名的空间中存在一个超平面 $\{\mathrm{r}(e,\tau)\}_{e\in\tau(I)}$；对于数据源，在后验均值空间中存在一个超平面 $\{\Phi_e\}_{e\in\tau(I)}$。由于偏差，这两个超平面具有相同的法向量，但截距不同。结果可推广到任意可加的超模效用函数，其中差值 $u(\mathrm{r}(e,\tau(I))$, $-$ $u\ \ e\ \tau\ I\ \ \ e\ \ I\ \ \ \ \ e\ \tau\ I\ \ \ \ \

错误排名。 根据定义 4.1，当且仅当存在 e'，使得意图将 e 排在 e 之上（即，$\mathrm{r}(e', \tau(I)) < \mathrm{r}(e, \tau(I))$），但解释没有将 e' 排在 e 之上（ e' 要么被省略，要么被排在 e 之后）时，返回的元组 e 才是不可信的。这是一种成对 违规：其见证由成对元素 (e', e) 构成。因此，即使数据源执行了复杂的重新排序（例如一系列交换），分析成对交换也已足够，因为成对交换是标准的局部见证，可以隔离出一个可能被颠倒的单独元素对。

引理 4.4. 在命题 *4.3* 的设定中，假设 β、β' 仅在 $e \neq e'$ 上不同，并满足 $\mathrm{r}(e, \beta(I)) - \mathrm{r}(e, \beta'(I)) = -(\mathrm{r}(e', \beta(I)) - \mathrm{r}(e', \beta'(I))) \neq 0$。则用户的无差异条件取决于意图中两个元组排名之差：

$$\mathrm{r}(e, \tau(I)) - \mathrm{r}(e', \tau(I)) = \delta, \qquad \textit{where} \qquad \delta = \frac{\alpha^s}{\gamma_e} - (b(e) - b(e')) \tag{2}$$

直观而言，引理 4.4 表明，当两种解释 β和 β' 交换两个元组的位置时，用户对这两种解释的偏好仅取决于元组在意图中的相对位置以及数据源偏差。我们将公式 2 中的 δ称为**无差异阈值**。在以下结果中，我们使用该无差异阈值来检测交互中的可信信息。

命题 *4.5.* 在引理 *4.4* 的设定中，令 $\delta_0 = \frac{\alpha^s}{\gamma_e}$ 表示当 $b(e) = b(e')$ 时的无差异阈值，并令 $\delta = \delta_0 - (b(e) - b(e'))$ 表示偏差平移后的无差异阈值。若存在一个元组 $e' \in \tau(I)$，使得元组在意图中的排名差满足 $r(e, \tau(I)) - r(e', \tau(I)) \in \big(min\{\delta_0, \delta\}, max\{\delta_0, \delta\}\big]$，则对于 $e \in \beta(I)$，数据源解释是不可信的。

基线算法。 命题 4.5 给出了一种基线算法，用于检测交互中的不可信信息。对于每个返回的元组 $e \in \beta(I)$，以及模式域 R 上意图结果中的每个元组，只要满足 $e' \notin \beta(I)$ 或 $\mathrm{r}(e', \beta(I)) > \mathrm{r}(e, \beta(I))$，算法就会计算无偏和有偏的无差异阈值 $\delta_0 = \frac{\alpha^s}{\gamma_e}$ 和 $\delta = \delta_0 - (b(e) - b(e'))$。如果这些元组在意图中的相对位置满足命题中的条件，那么 $e \in \beta(I)$ 就是可信的。该算法效率低下，因为对于解释中的每个元组，都必须根据模式计算其在意图中的位置，而计算开销会随模式域的大小而增加。

定理 *4.6*。给定用户和数据源的二次效用函数、解释结果 $\beta(I)$ 和 $\delta \in \{1, \ldots, z-1\}$，若元组 $e \in \beta(I)$ 的偏差为 $b(e)$，则当且仅当存在任意元组e'，满足 $e' \notin \beta(I)$ 或 $r(e', \beta(I)) > r(e, \beta(I))$，且其偏差 $b(e')$ 满足：

$$b(e') \in \Big[b(e) - g(z, \delta),\ b(e) - \max\{g(z, \delta) - 1,\ s(z, \delta)\} \Big). \tag{3}$$

其中 $g(z, \delta) = \frac{-\delta^3 + 3\delta^2 z + \delta z^2 + \delta + z^2 - z}{d(z, \delta)}$，$s(z, \delta) = (z - \delta)(z - \delta + 1)$ $(2\delta + z - 1)\, d(z, \delta)$ 且 $d(z, \delta) = 3\big(z^2 - z + \delta(2z + 1) - \delta^2\big) > 0$

根据定理 4.6，当存在一个元组 e'（它要么不在 $\beta(I)$ 中，要么在 $\beta(I)$ 中的排名严格低于 e）且其偏差满足公式 3 中针对某个 δ 的条件时，元组 $e \in \beta(I)$ 就是不可信的。具体而言，该定理为 $b(e')$ 值提出了两个要求。首先，$g(z, \delta)$ 描述了哪些偏差差值会使 δ 成为一个

算法 1高效检测可信答案

```
输入：交互设置 (R, τ, U^r, U^s)，解释结果 β(I)
输出：集合 C 中的 可信 元组 β(I)
C ← ∅
S ← ∅
根据 b(.) 和 ℛ计算 [b_min, b_max]
对于 δ = 1 到 z − 1 执行
    计算 g ← g(z, δ) 和 s ← s(z, δ)                          ▷ 定理 4.6
    g − 1  s    b_δ^+ ← g : b_δ^− ← 最大{,} : 如果
    b^−
        δ < b_δ^+ then
                                                            ▷ 存储可行的 (δ, b δ̄, b_δ^+)
        δ, b_δ^+) 至 S
对于所有 e ∈ β(I) 执行
    可信 ←真
    对于所有 (δ, b_δ^−, b_δ^+) ∈ S 执行    ▷可能的区间 b(e′): b(e′) ∈ [a, c)
        a ← b(e) − b_δ^+;   c ← b(e) − b_δ^−
        如果最大值{a, b_min} < 最小值{c, b_max} 则
            可信 ← 假
            中断                                            ▷ 提前退出
    如果可信，则
        C ← C ∪ {e}
返回C
```

可行无差异阈值。其次，$s(z, \delta)$ 是两个元组 $b(e)$ 和 $-b(e')$ 之间的最小偏差差值，使得数据源将 e 排在 e' 之前，从而最大化其效用。

高效算法。算法 1 仅利用基于模式域的数据源偏差函数 $b(.)$ 的偏差范围 $[b_{\min}, b_{\max}]$，实现定理 4.6 中的条件。首先，对于所有 $\delta \in \{1, \ldots, z-1\}$，计算 $g(z, \delta)$ 和 $s(z, \delta)$，并仅保留定理区间非空的那些 δ，即 $b_\delta^- = \max\{g(z, \delta) - 1,\ s(z, \delta)\} < b_\delta^+ = g(z, \delta)$。对于返回的元组 $e \in \beta(I)$ 和固定的 δ，算法会检查公式 3。由于未知单个值 $b(e')$，只要该区间与可行偏差范围 $[b_{\min}, b_{\max}]$ 对上文保留的每个 δ 的交集为空，算法就会将 $e \in \beta(I)$判定为可信。如果所有保留的 δ对应的交集都为空，则没有任何可行偏差值能够满足定理中的条件，因此 e 可信。如果某个 δ对应的交集非空，则 e 可能不可信。

复杂度。 计算所有 $g(z, \delta)$ $s(z, \delta)$ 并构造 S 需要 $O(z)$ 时间。每个元组的检查最多执行 $|\mathrm{S}| \leq z - 1$ 次重叠测试，每次耗时 $O(1)$。因此，对于 $k = |\mathrm{elm}(\beta(I))|$ 个元组，总运行时间为 $O(z + k|\mathrm{S}|) \subseteq O(kz)$。

5 影响力策略

数据源可能会将结果导向赞助产品、高利润产品或其他有偏产品。因此，直接表达意图（例如搜索某一特定低价产品）可能会得到一份由推广替代项主导的排序列表。因此，用户可能会提交一个查询，策略性地调整向数据源展示的产品排序，以找回相关结果。然而，用户策略的有效性可能取决于数据源对其的解释以及数据源的偏差程度。在本节中，我们研究这些情形，并提出用于寻找策略的算法，这些策略能使交互达到均衡并提高用户效用。

5.1 影响力检测查询

我们首先刻画关于用户意图的哪些信息能够可信地传达给数据源。在此基础上，我们提出一种算法，用于寻找能使交互达到有影响力的均衡的查询。

> **问题3.** 给定交互设置 $(\mathcal{R}\ \tau, U^r, U^s)$，找出一个能够导向有影响力的均衡的查询。

在第4节中，我们识别了数据源的解释中的不可信排名。意识到这些错误排名的用户可以通过调整查询中元组的排序来抵消这种偏差。例如，如果购物网站有动机推广某个产品 e'（或贬低 e），那么只有当该产品 e 在意图中的排名比 e' 高足够多 时，用户的查询才会将它排在更高位置。我们使用相对排名约束来形式化这一概念。

相对排名约束。 给定一个模式 R 上实例 I 的意图 τ、元组 $e, e' \in \tau(I)$，以及整数 $\delta \geq 1$，相对排名约束是不等式 $\mathrm{r}(e', \tau(I)) - \mathrm{r}(e, \tau(I)) \geq \delta$. 直观地说，该不等式表示， e 在意图中的位置至少比 e' 高 δ 位。

定理*5.1*。考虑与定理 *4.6*相同的情形。对于有序对（e， e'） $\in \tau$（I），令 $b = b$（e） $- b$（e'），并令 g（z， δ）如定理 *4.6* 中所述。定义 $\delta^\star$（e， e'）为满足 g（z， δ） $-1 < b \leq g$（z， δ）的 $\delta \in \{1, \ldots, z-1\}$ 中的（最大）整数解；如果不存在这样的整数解，则设 $\delta^\star$（e， e'） $= \bot$。如果 $\delta^\star$（e， e'） $\neq \bot$，则相应的相对秩约束为：r（e'， τ（I）） $- r$（e， τ（I）） $\geq \delta^\star$（e， e'）

定理5.1给出了求解元组之间相对秩约束的闭式解。非正式地说，该定理表明，数据源越偏向于 e'，用户就越需要提供更有力的证据，证明 e 确实在意图中排名高于 e'，也就是说，需要更大的 $\delta^\star$。以下结果由该定理推出。

命题 5.2。 如果存在一对 e, e'，使得 $\delta^\star(e, e') \neq \bot$，那么该交互存在一个有影响力的均衡。

例5.3。假设在耳机实例 I 上，意图 τ 将 e_2 排在 e_3 之上很远，例如，$\mathrm{r}(e_2, \tau(I)) = 1$，且 $\mathrm{r}(e_3, \tau(I)) = 3$。考察有序对 (e_2, e_3)，并假设数据源偏差之差为 $b = b(e_2) - b(e_3)$，其中 $b < 0$；也就是说，该数据源相对更偏向于提升 e_3 e_2 e_3 $\bot$ $\delta^\star$ τ I $-$ τ I 的排名。假设定理5.1得出 $\delta^\star(\ ,\ )$。对应的排序约束为 $\mathrm{r}(e_3$ e_2 $\geq$ $2.$ τ I $-$ τ I $= 2$ τ $,(\ ))$ $\mathrm{r}(\ ,(\ ))$。由于 $\mathrm{r}(e_3, (\ ))$ $\mathrm{r}(e_2, (\ ))$，具有该意图的用户可以提交此相对排序信息，以影响数据源的解释。现在考虑同一实例上的另一种意图 τ'，其中 e_2 仅比 e_3 高一个位置，例如，$\mathrm{r}(e_2, \tau'(I)) = 2$，且 $\mathrm{r}(e_3, \tau'(I)) = 3$。于是 $\mathrm{r}(e_3, \tau'(I)) - \mathrm{r}(e_2, \tau'(I)) = 1 < 2$，因此上述约束在 τ' 下不成立，具有意图 τ' 的用户会改为提交互补信息（即 $\mathrm{r}(e_3, \tau'(I)) - \mathrm{r}(e_2, \tau'(I)) < 2$）。

定义 *5.4.* 给定模式 R 的实例 I 上的一个意图 τ， **δ-查询** q^δ 是由形如 $\mathrm{r}(e', \tau(I)) - \mathrm{r}(e, \tau(I)) \geq \delta$，的相对排名约束构成的任意合取，其中 $\delta \in \mathbb{Z}$。等价地，对于某个集合 $\mathrm{V} \subseteq \tau(I) \times \tau(I) \times \mathbb{Z}$，

$$q^\delta = \bigwedge_{(e,e',\delta) \in \mathcal{V}} \big(\mathrm{r}(e', \tau(I)) - \mathrm{r}(e, \tau(I)) \geq \delta\big).$$

引理 5 5 $\neg($ e' τ I $-$ e τ I $\geq$。。对于整数值 $\delta)$ e τ I $-$ e' τ I $\geq 1 - \delta.$ 排名，$\mathrm{r}(,())$ $\mathrm{r}(,())$ 当且仅当 $\mathrm{r}(,())$ $\mathrm{r}(,())$

算法。 算法 2 构造查询 q^δ，使输入意图 τ 能够提交该查询，以影响数据源的解释。如第 3.2 节所述，用户通常不知道确切的实例 I，因此算法使用基于模式 R 的域的信息。因此，我们在由模式和意图的排序属性所诱导的排名域 上运行算法。例如，如果 τ 具有排序属性 A（例如，ORDER BY rating），我们将排名域设为$\mathbf{dom}(A)$；如果 τ 按属性 A_1、……、 A_l排序，我们将排名域设为$\mathbf{dom}(A_1) \times \cdots \times \mathbf{dom}(A_l)$ 的乘积。我们将 τ 视为笛卡尔积域上的一种排序，并使用该域上由其诱导的排名函数 $\mathrm{r}(\cdot\ \tau)$。

该算法遍历此域中严格有序的有序对 $e \prec_\tau e'$，并调用子程序（算法 3），根据定理5.1计算 $\delta^\star(e, e')$。定理5.1为二次情形提供了闭式解，但这一方法也可推广到其他具有可加偏差的可加超模效用函数。对于其他超模效用函数，可以替换 $g(.)$ 的解。为避免物化所有 $g(.)$ 的值，算法 3 按需计算 $g(.)$，并通过在 $\delta \in \{1, \ldots, z-1\}$ 上进行二分搜索来找到 $\delta^\star$。输出查询 q^δ是相对排名约束的合取，其中 V 对每个被考察的有序对恰好包含一个约束。重要的是， q^δ的构造确保其在输入意图 τ 下为真。直观来说，对于每个满足 $\delta^\star(e, e') \neq \bot$ 的有序对 (e, e')，算法会添加正向约束 $\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) \geq \delta^\star$（如果该约束成立）；否则，添加逻辑上等价的补充约束 $\mathrm{r}(e, \tau) - \mathrm{r}(e', \tau) \geq 1 - \delta^\star$（引理 5.5）。

示例 *5.6*。 假设给定一个关于耳机实例I的查询，算法 2 输出查询 $q^\delta = (\mathrm{r}(e_2, \tau(I)) - \mathrm{r}(e_1, \tau(I)) \geq 1) \wedge (\mathrm{r}(e_4, \tau(I)) - \mathrm{r}(e_3, \tau(I)) \geq 1)$，该合取式要求将 e_1 排在 e_2 之前，并将 e_3 排在 e_4 之前，但不对成对的 $\{e_1, e_2\}$ 和 $\{e_3, e_4\}$ 之间的关系施加约束。因此， q^δ与多个排序列表一致。例如， q_1^δ I $e_1 \prec e_2 \prec (\ )$：，$(\ )$：两 $e_3 \prec e_4$ q_2^δ I $e_3 \prec e_4 \prec e_1 \prec e_2$ q^δ 者都满足 。

q^δ的语义。算法 2 的输出是一个查询 q^δ，它由相对排名约束合取而成。直观而言， q^δ涵盖了一组满足其约束的排名。根据 q^δ 中的约束，这组排名可能是：(1) 空集，(2) 单元素集合，或 (3) 包含多个排名。为了帮助理解，请注意，排名取值于有限集合 $[z]$。因此，每个不等式都可以展开为针对排名差值的有限析取。具体而言，对于任意有序对 (e, e') 以及任意 $\delta \in \{1, \ldots, z-1\}$，$\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) \geq \delta \equiv \bigvee_{t=\delta}^{z-1} \big(\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) = t\big)$. 等价地，由于 $\mathrm{r}(e', \tau) \in [z]$，该约束可以写成一个析取，列出在 e的排名给定时 e'可能的排名，即 $\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) \geq \delta \equiv \bigvee_{j=1}^{z} (\mathrm{r}(e, \tau) = j \wedge \mathrm{r}(e', \tau) \geq j + \delta)$，其中含有 $j + \delta > z$的项不可满足。例如，若 $\delta = 2$且 $\mathrm{r}(e, \tau) = 4$，那么任意满足 $\mathrm{r}(e', \tau) \in \{6, 7, \ldots, z\}$的值都满足该不等式。这解释了上述三种情况。包含 $\delta > 1$ 的约束仅施加最小间隔，因此通常会留有余量，允许多种排名。如果足够多的约束使用 $\delta^\star = 1$，它们就能确定一个全序，从而得到单元素集合。如果约束相互冲突（例如，强制 $j + \delta > z$），该集合则为空。

算法 2检测有影响力的查询

输入： 交互设置 (R τ, U^r, U^s)
输出： 查询 q^δ
V $\leftarrow \emptyset$ ▹ 相对排名约束集合
对于每个有序对$(e, e') \in \tau$**执行** ▹ 按 τ顺序迭代
$\ell \leftarrow$ 约束二次算法$(e, e', \tau, b(.))$
  如果 $\ell \neq \perp$**则**
    V $\leftarrow$ V $\cup \{\ell\}$
$q^\delta \leftarrow \bigwedge_{\ell \in \mathrm{V}} \ell$ ▹ 定义 5.4
返回 q^δ

算法 3约束二次算法

输入： e, e', τ, $b(.)$
输出： 相对秩约束（或 $\perp$）
$\Delta b \leftarrow b(e) - b(e')$ ▹ 定理5.1
$\delta^\star \leftarrow$ 对 $\delta \in \{1, \ldots, z - 1\}$ 进行二分搜索，使用 $g(z, \delta)$
如果 $\delta^\star = \perp$ **则**
  返回 $\perp$
如果 $\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) \geq \delta^\star$ **则**
  返回 $(\mathrm{r}(e', \tau) - \mathrm{r}(e, \tau) \geq \delta^\star)$
否则
返回 $(\mathrm{r}(e, \tau) - \mathrm{r}(e', \tau) \geq 1 - \delta^\star)$ ▹ 引理 5.5

复杂度。算法 2 遍历意图在模式域上的结果中的元组有序对。在最坏情况下（当 τ为全序时），搜索空间大小为 $\binom{m}{2} = O(m^2)$，其中 $m = \prod_{i=1}^{l} |\mathrm{dom}(A_i)|$是意图 τ中排序属性 $A_i..A_l$的域大小。对于每一对，算法会定位满足 $\Delta b \in (g(z, \delta) - 1, g(z, \delta)]$的 $\delta^\star$。如果通过扫描 $\delta \in \{1, \ldots, z - 1\}$找到 $\delta^\star$，则每对的开销为 $O(z)$，总运行时间为 $O(m^2 z)$。如果 $g(z, \delta)$关于 δ单调不减，则二分搜索每对需要 $O(\log z)$次 $g(.)$ 求值，总时间为 $O(m^2 \log z)$。输出查询 q^δ至多包含 $O(m^2)$个约束。

分桶。 算法 2 的复杂度取决于意图的排序属性所决定的排名域大小；当属性具有高基数域时，排名域可能很大。遵循标准做法 [3, 16], ，我们因此将较大的属性域分桶为少量组。例如，评分等连续属性可以离散化为 3–4 个桶。此选择也有实践依据。在许多数据源中，偏差自然是以元组组 为单位表达的，例如所有者品牌、赞助卖家或较粗略的价格区间，而非以单个元组为单位 [16]。我们在第 6.2 节中介绍分桶的具体细节，并通过实验研究这一选择如何影响算法的可扩展性和用户效用。

查询语言的局限性。支持类似 q^δ 的查询，需要一种能够表达基于约束的排序规范的语言，或者一个接受排序查询并集的数据源。如上所述， q^δ 表示与其约束一致的排名集合。该集合决定了 q^δ 能否影响数据源的解释。如果该集合为空， q^δ 就没有影响力。如果该集合涵盖了整个查询空间，例如用户意图中元组的所有排列，那么 q^δ 同样没有影响力。只有当 q^δ排除部分排名但并非全部时，它才能传达有意义的信息，并影响数据源的后验。此时，用户可以提交 q^δ，从而使交互达到有影响力的均衡（命题 5.2）。然而，我们当前的查询语言（第 2.1 节）将用户限制为单条 SQL 查询，其排名由 ORDER BY子句决定。这类查询必然只生成一个排序列表，无法直接编码相对排名约束。为弥补这一差距，我们可以从一致排名的非平凡集合中选取一个排名 q^δ，并提交其 SQL 编码。鉴于 SQL 的普及，以下各节将重点讨论使用 SQL 查询进行的交互，并将扩展到其他查询语言留作未来工作。

5.2 提升用户效用

在上一节中，我们介绍了一种查询，它能够说服数据源尽管存在数据源偏差，仍返回一些相关结果。然而，它可能仍然只返回与用户意图相关的元组中的一小部分。在实践中，用户可以通过进一步调整排序来细化此查询。本节将探讨这种细化，并提供一种算法来寻找能够提升用户期望效用的查询。

例5.7。考虑表1中的耳机实例，并令用户意图 τ 为检索评分最高的耳机。假设数据源返回排名前3的元组，其逐元组效用函数为 $u_e^s = (\mathrm{r}(e, \tau, I), \mathrm{r}(e, \beta(I)) - b(e)$，偏差为 $b(e)$。给定查询 q，令 $\Phi_e = \mathbb{E}[\mathrm{r}(e, \tau, I) \mid q]$ 表示元组 e的后验均值排名。根据后验均值，数据源会最大化 $\mathbb{E}[u_e^s \mid q] = (\Phi_e - 1.9)\,\mathrm{r}(e, \beta(I))$。因此，为了最大化效用，若 $\Phi_e < 1.9$，数据源会将 e分配到位置 $\mathrm{r}(e, \beta(I)) = 1$；若 $\Phi_e > 1.9$，则分配到位置$\mathrm{r}(e, \beta(I)) = 4$（不返回）。现在考虑与用户意图 q 等价的查询：ORDER BY rating DESC，它会产生排序 $e_3 \prec e_2 \prec e_1 \prec e_4$。根据次序统计量理论 [10], ，在给定 q 的条件下，$\Phi_{e_3} = 1$, $\Phi_{e_2} = 2$, $\Phi_{e_1} = 3$ 对4!种可能的意图排序采用均匀先验，得到后验：，，，$\Phi_{e_4} = 4$。因此，只有 e_3 会被返回。现在考虑另一个查询 q'，它会产生 $q'(I)$：$e_3 \sim e_2 \prec e_1 \prec e_4$。在 q'条件下，后验为 $\Phi'_{e_3} = \Phi'_{e_2} = \frac{1+2}{2} = 1.5$, $\Phi'_{e_1} = 3$, $\Phi'_{e_4} = 4$.。因此， e_3 和 e_2 都会被返回。

—

延续上述示例，我们首先引入一类有助于改进所返回结果的查询优化。

定义 *5.8.* 给定模式 R 的实例 I 上的查询 q, q'，若 q' 是 q 的 **超排名查询**：

(1) $\mathrm{elm}(q(I)) \subseteq \mathrm{elm}(q'(I))$。(2) 对于所有 e、 $e' \in q(I)$，如果 $\mathrm{r}(e, q(I)) = \mathrm{r}(e', q(I))$，则 $\mathrm{r}(e, q'(I)) = \mathrm{r}(e', q'(I))$。(3) 对于每个 $e \in q'(I) \setminus q(I)$ 以及每个 $e' \in q(I)$，我们有 $\mathrm{r}(e, q'(I)) > \mathrm{r}(e', q'(I))$。

我们将 q 的所有超排名查询的集合记为 $\mathrm{Q}^{\mathrm{super}}(q)$。

直观地说，在某个实例上执行超排名查询（$q' \in \mathrm{Q}^{\mathrm{super}}$）可能会：(i) 将带排名的元组的某个子列表变为具有相同位置的一组元组，或 (ii) 保留部分顺序，(iii) 在列表末尾追加额外的（带排名的）元组。每个查询都是其自身的超排名。

示例 *5.9.* 给定一个查询 q，它在耳机实例 I上返回 $e_1 \sim e_3 \prec e_4 \prec e_2$。某些超排名查询的结果

q 分别为:

(a) set : $e_1 \sim e_3 \sim e_4 \sim e_2$
(b) preserve order : $e_1 \sim e_3 \prec e_4 \prec e_2$
(c) append : $e_1 \sim e_3 \prec e_4 \prec e_2 \prec e_5$

超排名策略。 如第 5.1 节所述，在我们的查询语言中，用户必须提交一个能够生成单个排序列表的查询。因此，给定一个意图 τ，我们运行算法 2，得到查询 q^{δ}（定义 5.4），然后通过选择任意一个与 q^{δ}一致的可执行查询来确定基础查询 q^{base}。

定义 *5.10*。 用户策略 $P^r(\cdot \mid \cdot)$ 是一种**超排名策略**，当且仅当对于每个意图 τ和查询 q，都有 $P^r(q \mid \tau) > 0$ 仅当 $q \in \text{Q}^{\text{super}}(q^{\text{base}})$。

接下来我们将证明，当用户采用超排名策略时，用户与数据源之间的交互总是会达到均衡。

定理 5.11。 给定交互设置（$\mathcal{R}$、τ、 U^r、U^s），假设用户和数据源对每个元组 e具有以下效用函数:

$$u^r(\text{r}(e,\tau(I)),\text{r}(e,\beta(I))) = -(\text{r}(e,\tau(I)) - \text{r}(e,\beta(I)))^2,$$
$$u^s_e(\text{r}(e,\tau(I)),\text{r}(e,\beta(I))) = -(\text{r}(e,\tau(I)) - (\text{r}(e,\beta(I)) + b(e)))^2.$$

在实例 I 上$\mathcal{R}$ 然后，. 如果用户采用超排名策略，则该交互具有有影响力的均衡。

例 *5.12.* 考虑来自例 5.9 的 q 的两个超排名。 q_1^{sup} （I）：$e_1 \sim e_3 \sim e_4 \sim e_2$ 以及 q_2^{sup} （I）： $e_1 \sim e_3 \sim e_4 \prec e_2$同等偏好三个耳机品牌的查询（$q_1^{sup}$）和对特定耳机品牌具有更高偏好的查询（$q_2^{sup}$）都是超排名查询，并且都会在交互中导致均衡。

上述结果表明，为了使推理达到某个均衡，用户选择某些类型的策略就已足够。尽管如此，用户可能更偏好某些均衡而非其他均衡。 如例 5.12 所示，通过使用某些超排名查询，用户可以夸大其对某些元组的偏好，而这些元组原本可能不会被数据源返回。然而，若一种超排名策略将意图映射为这样的查询：把过多元组置于高于其在意图中真实位置的位置，或向意图中添加过多元组，则可能降低所有元组的期望后验。例如，在例 5.7 中，再次假设数据源仅有四副耳机。如果用户提交的查询会返回数据源中的所有耳机，则每个元组的期望后验将为 $_{5\ 2}$。如果数据源仅返回后验均值高于 2.5 的元组，则不会返回任何元组。因此，用户策略必须找到对意图进行适当程度的修改，以抵消偏差并最大化所提取的信息。

完全有影响力的策略。 在完全有影响力的均衡中，数据源会返回与所提交用户查询背后的意图相关的所有元组，这对用户来说是理想的:

定义 *5.13*。 如果对于每个查询 q和意图 τ，只要 $P^r(q|\tau) > 0$， τ就是对所有满足 $P^s(\beta|q) > 0$的 β而言的 β的超排名，那么均衡 $(P^r(.|.)\ P^s(.|.))$ 就是**完全有影响力的**。用户策略P^r 如果能导向一个完全有影响力的均衡，就称为**完全有影响力的策略**。

要实现完全有影响力的均衡，用户必须让数据源相信，所有其期望的元组都与用户的意图高度相关。但是，正如我们在前文所解释的，这并非总是可行。因此，我们可以追求一种均衡，使其尽可能多地从数据源中提取有用信息。

定义 *5.14*。 如果对于每个 $E = (P^r(\cdot \mid \cdot)\ P^s(\cdot \mid \cdot))$，只要其中的 P^r 是超排名策略，对于每个查询 q，以及每个满足 $P^s_{\max}(\beta_{\max} \mid q) > 0$的$\beta_{\max}$，都有 $\beta_{\max}$ 至少与每个满足 $P^s(\beta \mid q) > 0$ 的 β 一样有影响力，那么均衡 $E_{\max} = (P^r_{\max}(\cdot \mid \cdot)\ P^s_{\max}(\cdot \mid \cdot))$ 就是**最大影响力**的。若用户策略 $P^r_{\max}$ 能导出最大影响力的均衡，则称其为**最具影响力的策略**。

接下来，我们将介绍用于找出导出最大影响力均衡的用户策略的算法（定义 5.14）。

问题 4。 给定交互设置 $(\mathcal{R}\ \tau, U^r, U^s)$，找出一个可导出最大影响力均衡的查询 $q^\star$。

基线算法。一种暴力方法是先获取由 τ 推导出的 getq^{base}，然后枚举所有可能的超排名查询 $\text{Q}^{\text{super}}(q^{\text{base}})$。接着，对于每个超排名查询 (q^{sup})，获取数据源的解释 (β)，并计算用户效用 $U^r(q^{sup},\ \beta)$，最后返回使用户效用最大化的查询。然而，查询集合 Q^{super} 的大小可能呈超指数增长（命题 5.15），尤其是因为超排名查询可以追加任意数量的额外元组。一种简单的优化方法是将超排名查询集分为两个子集：（i）追加额外元组的查询；（ii）不追加额外元组的查询。我们可以先找出第二个子集中的最佳超排名，然后向其中追加额外元组。这样可以显著缩小搜索空间。然而，这个超排名子集的规模仍然是指数级的，因此该问题仍然是计算困难的。我们证明，在一般情况下，寻找最大影响力查询是 NP难的。

命题 5.15。 设 m 为模式 $\mathcal{R}$ 的实例 I 上q^{base} 的查询结果中的元组数，r为可追加的元组数。该算法会输出超指数数量的均衡，并且运行时间关于 r呈超指数增长。

定理 5.16。 一般而言，寻找最具影响力的策略是 *NP*难问题

合并查询。 尽管一般情况下该问题是 NP难的，但对于可加效用函数，该问题具有适用于动态规划（DP）的最优子结构。通过逐步将带排名元组的一个子列表更改为一组位置相同的元组，可得到 q^{base}的一个超排名。

定义 *5.17*。 给定实例 I 的模式 R 上的查询 q，q'，令 m 表示 $q(I)$ 中不同排名位置的数量。若满足以下条件，我们称 q' 为 q 的**合并查询**：

(1) $\text{elm}(q(I)) = \text{elm}(q'(I))$

(2) 存在整数 $1 \le i \le j \le m$，使得对于每个 $e \in q(I)$，

$$\mathrm{r}(e, q'(I)) = \begin{cases} \mathrm{r}(e, q(I)) & \text{if } \mathrm{r}(e, q(I)) < i, \\ i & \text{if } i \le \mathrm{r}(e, q(I)) \le j, \\ \mathrm{r}(e, q(I)) - (j - i) & \text{if } \mathrm{r}(e, q(I)) > j. \end{cases}$$

直观地，q' 是通过将连续的排名位置 i、$i+1$、……、j 合并为排名 i 处的一个并列位置而得到的。我们将 q 的合并查询集合记为 $\mathrm{Q}^{\mathrm{merge}}(q)$。

引理 *5.18*。给定一个查询 q，令 m 表示 q 在模式 R 上的结果中的排名位置数。对于每个区间 $1 \le i \le j \le m$，令 $q_{i:j} \in Q^{\mathrm{merge}}(q)$ 表示通过恰好合并 q 的位置 i、……、j 而得到的合并查询。令 u 表示在 $q_{i:j}$ 下的每元组预期用户效用。对于任意合并查询 $q' \in Q^{\mathrm{merge}}(q)$，令 $\Pi(q') = \{[i_1, j_1], \ldots, [i_t, j_t]\}$ 表示 $\{1, \ldots, m\}$ 的唯一划分，该划分由连续区间组成，且 q' 会将这些区间合并为并列位置。用户提交 q' 所获得的期望效用可表示为 $\sum_{\ell=1}^{t} n_{i_\ell:j_\ell}\, u(\tau, j_\ell) + C$，其中 $n_{i:j} = |\{e \in q : i \le r(e, q) \le j\}|$，且 C 与 q' 无关。

直观而言，引理 5.18 表明，合并任意连续位置区间 $[i, j]$ 所贡献的预期用户效用仅取决于该区间，而与其他位置如何合并无关。

高效算法。算法 4 计算一个影响力最大的查询。给定一个意图 τ，我们首先运行算法 2 得到约束查询 q^{δ}，然后通过选择任意一个其诱导的排序列表与 q^{δ} 一致的查询，确定一个可执行基础查询 q^{base}（定义 5.10）。与算法 2 一样，我们不假设可以访问实际实现；相反，我们基于排序属性 A_1、……、A_l 在 τ 中诱导的 m 排名位置进行处理，其中 $m = \prod_{i=1}^{l} |\mathbf{dom}(A_i)|$。

合并查询 $q' \in \mathrm{Q}^{\mathrm{merge}}(q)$ 通过选择连续的位置区间，并将原始排名落在每个所选区间内的所有元组设为并列，从而使该排序变得更粗略。等价地，每个 q' 都对应于将 $\{1, \ldots, m\}$ 划分为连续区间的一种唯一划分。根据引理 5.18，预期用户效用可分解为区间得分之和。因此，对 $\mathrm{Q}^{\mathrm{merge}}(q)$ 进行优化，归结为选择总得分最高的划分。对于每个区间 $[i, j]$，算法计算 $Score(i, j) = n_{i:j} \cdot u$，$n_{i:j} = \left|\{e \in q : i \le \mathrm{r}(e, q) \le j\}\right|$，其中 u 表示恰好合并位置 i、……、j（而其他位置保持在 q 中的状态）时的每元组预期用户效用。为计算 u，我们使用其效用函数 $\beta_{i:j} \in \arg\max_{\beta'} \mathbb{E}\left[U^s(\tau, \beta') \mid q_{i:j}\right]$，求得数据源针对合并查询 $q_{i:j} \in \mathrm{Q}^{\mathrm{merge}}(q)$ 的解释。然而，某些效用函数可能有多个最大值，因此 arg max 可能不唯一。也就是说，数据源可能对多个提供相同最大效用的最优解释（β）无偏好差异。对于此类效用函数，我们假设采用一致的平局处理规则：如果多个解释 $\{\beta_1, \beta_2, \ldots\}$ 产生相同的最大效用，数据源会选择对用户最有利的那个。由于数据源不知道用户的真实意图 τ，它会选择最接近用户查询的解释，即最大化 $U^r(q, \cdot)$ 的 β。这是一个合理的假设，因为数据源希望让用户持续参与。

最后，给定 $Score(i, j)$，算法应用命题 5.19 中的动态规划递推式，计算 $\{1, \ldots, m\}$ 的最优划分为连续区间，然后通过回溯构造相应的合并查询 $q^{\star} \in \mathrm{Q}^{\mathrm{merge}}(q)$。

命题 5.19。令 $\mathrm{OPT}(j)$ 表示合并位置前缀1、……、j所能达到的最大总效用。则

$$\mathrm{OPT}(0) = 0, \qquad \mathrm{OPT}(j) = \max_{1 \le i \le j} \left\{\mathrm{OPT}(i-1) + Score(i, j)\right\}.$$

算法 4最大影响力查询

输入： 交互设置 $(\mathrm{R}, \tau, U^r, U^s)$
输出： $q^{\star} \in \mathrm{Q}^{\mathrm{merge}}(q^{\mathrm{base}})$
$q^{\mathrm{base}} \leftarrow$ 算法 2 $(\mathcal{R}, \tau, U^r, U^s)$
函数 评分(i, j) ▷ 合并位置的评分 $i..j$ (引理 *5.18*)
$q_{i:j} \leftarrow$ 合并$[i, j]$ 查询于 $\mathrm{Q}^{\mathrm{merge}}(q)$
$\beta \in \arg\max_{\beta'} \mathbb{E}[U^s(\tau, \beta') \mid q_{i:j}]$
$n \leftarrow \left|\{e \in q : i \le \mathrm{r}(e, q) \le j\}\right|$
$u \leftarrow$ 在 $(q_{i:j}, \beta)$ 下的每元组预期用户效用
返回 $n \cdot u$
动态规划$[0] \leftarrow 0$ ▷ 前缀上的动态规划（命题 *5.19*）
对于 $j = 1$ 到 m **执行**
　　(动态规划$[j]$, 父项$[j]$) $\leftarrow$ 最大值$_{1 \le i \le j}$ {动态规划$[i-1]$ + 评分(i, j)}
Π⋆ ←从父项回溯区间[·]
$q^{\star} \leftarrow$ 在$Q^{\mathrm{merge}}(q)$ 中合并查询，恰好合并 Π⋆ 中的区间
返回 $q^{\star}$

复杂度。与算法 2 一样，我们在由意图的排序属性 $A_1, \ldots, A_l$ 所诱导的排名域上进行操作，其大小为 $m = \prod_{i=1}^{l} |\mathbf{dom}(A_i)|$。算法 4 在 m 个诱导的排名位置上执行动态规划。动态规划会考虑所有满足 $1 \le i \le j \le m$ 的区间 (i, j)，产生 $O(m^2)$ 次状态转移；回溯需要 $O(m)$ 时间。区间得分 $Score(i, j)$ 的制表也涉及 $O(m^2)$ 个区间。如果每个 $Score(i, j)$ 都能在 $O(1)$ 时间内计算（例如，使用每个区间缓存的后验/效用汇总），则总运行时间为 $O(m^2)$；否则为 $O(m^2)$，再加上计算所有 $Score(i, j)$ 值的开销。

6 实证评估

我们通过实证评估所提出的解决方案，以展示其用途和有效性。我们介绍实验设置，然后开展定量研究，评估算法的效率，并考察不同的数据集、查询、属性以及可能域的大小。

6.1 实验设置

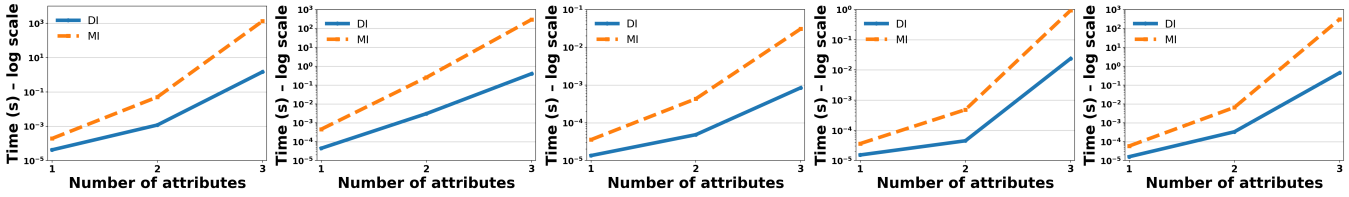
平台。所有实验均在一台搭载 Apple M1 Pro 芯片、配备 16GB 内存的 macOS Sequoia 15.6 设备上进行。

数据集。我们在五个规模、域和利益冲突场景各异的真实世界数据集上评估了我们的方法，相关内容总结于表 2。

亚马逊： 该数据集包含亚马逊 [11] 上超过 1400 万条商品列表。数据集包含产品类别、价格、评分、评论、上月销售额和畅销商品标记等属性。由于亚马逊可能有动机通过清理库存来处理评分较低且销量较少的产品，因此我们根据评分、上月销售额和价格来测试是否存在利益冲突。

PriceRunner： 该数据集采集自 PriceRunner[1]，这是一个颇受欢迎的商品比价平台 [1]。其中包含 35,311 条产品

[1]https://www.pricerunner.com/



(a) 亚马逊 (b) PriceRunner (c) 航班 (d) 人口普查 (e) COMPAS

图1：属性数量对影响力检测（DI）和最大影响力（MI）执行时间的影响

表 2：实验中使用的数据集概述。

数据集	元组	属性	排序属性（|A|）
亚马逊	14M	11	评分 (41)，上月销售额 (30)，价格 (29961)
PriceRunner	35K	7	卖家 (306)，产品型号 (12849)，类别 (10)
航班	300K	12	航空公司 (6)，距行程天数 (49)，价格 (12157)
人口普查	49K	12	种族 (5)，性别 (2)，教育程度 (16)
COMPAS	6.8K	12	年龄（65）、既往次数（37）、暴力评分（10）

来自 10 个类别（例如手机）的报价，由不同卖家提供。我们根据以下属性测试利益冲突：卖家、产品型号和产品类别。例如，系统可能会倾向于推荐某些卖家的产品。

航班： 该数据集是从 EaseMyTrip[2]，这家航班预订服务提供商 [6]抓取的。数据集包含 2022年2月11日至3月31日期间的 300,261 笔不同航班预订。我们根据以下属性检验利益冲突：航空公司、距行程天数和价格。例如，预订服务可能会偏向某些航空公司的高价航班。
我们还考虑了以往偏差研究中使用的热门数据集 [16, 18]。

人口普查： 美国人口普查数据集包含 49,000 个元组 [5]。我们根据种族、性别和教育程度检验利益冲突。例如，系统可能会偏袒教育程度较高的个人（例如，博士）。

COMPAS： COMPAS 数据集由 ProPublica 收集并发布[3]。该数据集包含 6,889 名个人的人口统计信息、COMPAS软件生成的暴力评分以及刑事犯罪信息。我们根据年龄、既往犯罪次数和暴力十分位评分检验利益冲突。

查询。 对于每个数据集，我们生成了一组多样化的 SQL 查询，形式为 SELECT A_1, ..., A_l WHERE Sel(q) ORDER BY A_1, ..., A_l，其中 Sel(q) 是由最多三个非键谓词构成的随机合取式，而 $l \in \{1, 2, 3\}$ 表示排序属性的数量。我们随机选择不同的筛选属性子集作为 WHERE子句。在查询投影中，为简单起见，我们纳入相同的一组排序属性，因为我们关注的是基于排序属性的冲突。为保持一致性，

[2]https://www.easymytrip.us/ [3] https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

我们假设意图中的排序顺序对于数值属性为降序（例如，ORDER BY Rating DESC），对于类别属性则采用原始数据集中的固有顺序。

效用函数。 在所有实验中，我们都使用二次效用函数。如第 3.2 节所述，数据源的效用函数可能因元组或元组集合而异。为了在不同数据集之间保持一致，同时改变公式1中的函数 $u_e^s(.,.)$ 和 $u_{e'}^s(.,.)$，我们随机分配偏差值 $[0, 1]$，并根据不同数据集的域对其进行缩放。所有报告的结果均为 5 次运行的平均值。

分桶如第 3.2 节所述，用户通常不知道确切的实例 I，因此我们的算法会基于模式 R 的域进行推理。因此，具有高基数域的属性（例如价格或产品型号）会影响算法的可扩展性。此外，数据源偏差通常适用于项目组（例如昂贵与便宜的商品），而非特定值，这促使我们将域值分组到较少数量的分桶中。我们研究了一系列具有不同域大小的排序属性（表 2）。按照惯例，[16],连续属性（如年龄）会根据其域和值划分为 3 到 4 组。对于高基数类别属性，我们采用了另一种常见做法：保留最常见的前 k 个值，并将其余值映射到一个“其他”分桶中 [3]。例如，在 PriceRunner 数据集中，我们设置 $k = 1$，并针对每个类别中卖家和型号的前 10 个可能域值定义分桶。对于属性域较小的数据集（如人口普查），则无需分桶。我们注意到，分桶的选择可能会影响查询可信度和算法的运行时间；因此，我们在第 6.2 节研究这一影响。

6.2 实验结果

在本节中，我们首先介绍第4节中算法的结果，该算法旨在从数据源结果中检测可信答案。然后介绍第 5.2 节中用于查找影响力查询的算法的结果。

6.2.1 检测可信答案。我们展示了算法 1 在可扩展性方面的结果。该算法的复杂度取决于与用户意图（z）相关的元组数量，以及数据源返回的元组数量（k）。例如，在本文撰写时，亚马逊每页显示前 16 条结果；因此，我们将 $k = 80$ 设置为模拟用户查看前五页结果的情形。为了在不同规模的数据集之间保持一致，我们将 z 设为各数据集总规模的百分比。如

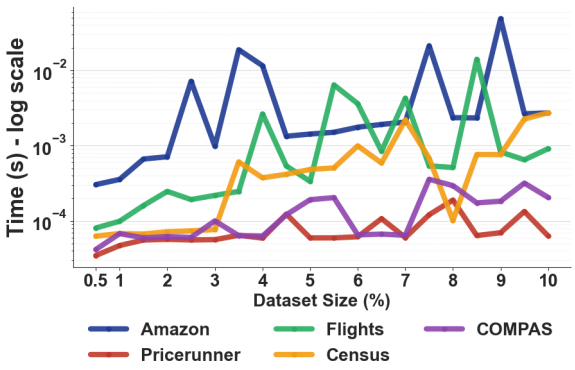


图2： z **对检测可信答案所需时间的影响**

图2显示，运行时间通常随 z线性增长。然而，正如第4节所述，由于算法往往会提前终止，我们观察到其性能呈现一定的次线性特征。

6.2.2 影响力策略。如第5节所述，由于可能均衡集合的复杂度呈超指数增长，寻找最具影响力策略的蛮力方法是NP难问题。因此，我们仅评估高效的最大影响力算法（MI；算法 4）和用于检测有影响力查询的算法（DI；算法 2）。

属性数量：第一组实验研究可能域的大小对算法运行时间的影响。为此，我们将查询中的属性数量从1变为3。属性数量（及其基数）决定了搜索空间。图1展示了随着 ORDER BY 属性数量增加，运行时间的变化。由于搜索空间会随着属性域的笛卡尔积而扩大，两个算法在数据集规模增大时都会变慢。亚马逊数据集的域规模最大，因此耗时最高；不过，最大影响力算法（MI）在3个属性的情况下仍能在15分钟上限内完成。寻找最具影响力策略的算法包含一个调用算法 2 的子程序，因此其复杂度高于检测单个有影响力查询（DI）的算法。

分桶的影响：为了研究分桶的影响，我们重点关注分桶前域最大的属性（表 2）：亚马逊和航班数据集中的连续属性价格、PriceRunner 中的分类属性产品型号，以及 COMPAS 中的年龄。人口普查是以往偏差相关研究中常用的数据集 [16, 18], ，不需要进行分桶。

我们考虑数据源偏差的两个极端情况：(i) 在粗粒度属性区间上存在组级 偏差（例如，最昂贵的分桶），以及 (ii) 集中于特定取值（例如 99.99 美元）的点级 偏差。为了在控制运行时间的同时对这两个极端情况进行插值，我们对有序域采用二叉树（二进制）细化方法。具体而言，我们从 $B = 2$ 个相等的连续分桶开始，并反复拆分每个分桶，得到 $B \in \{2、4、8、.....\}$。对于每个分桶级别 B，我们运行算法 2 和算法 4，同时在完整 （未分桶）域上评估所得查询。我们为每个分桶级别设置 10分钟时间预算。

图3总结了四个数据集之间的权衡。对于每个数据集，我们报告执行时间和用户效用。时间图

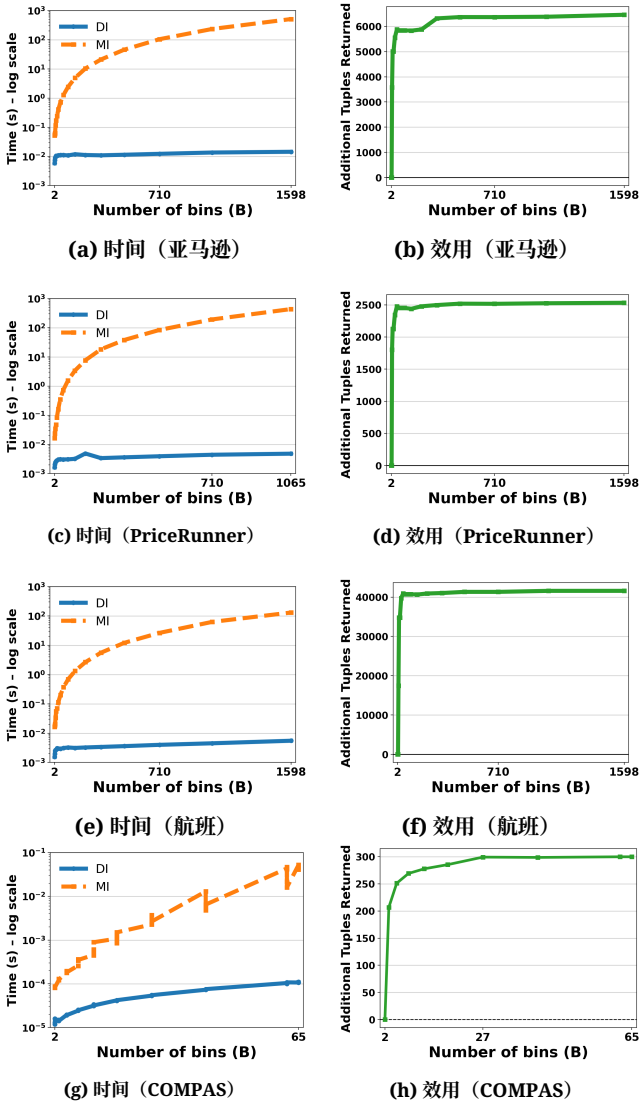


图3：分桶对亚马逊、PriceRunner、航班和 COMPAS 上运行时间及用户效用的影响。

将检测影响力查询的成本与计算最大影响力查询的端到端成本分开。效用图衡量最终查询的有效性 $q^\star$：我们报告相对于原始查询（基线）的用户效用增益，并将其表示为所恢复的额外相关元组数量的函数。正如预期，运行时间随着分桶数量 B的增加而增长，因为这同时增加了成对检查的数量和动态规划搜索空间。与此同时，更细的分桶能够识别出更可信的排名信息，从而生成可恢复更多相关元组的查询。从经验上看，所达到的效用是全域最优值的一个单调非递减下界。

7 相关工作

负责任的数据管理。 如第1节所述，我们的工作与负责任的数据管理和科学领域的相关工作具有相同目标 [32]。具体而言，研究人员已经提出了若干公平排序方案 [17, 21, 33–35]。我们的目标并非设计公平排序方案，而是研究在存在利益冲突的情况下能否进行信息丰富的查询，并帮助用户从有偏数据源中提取可靠信息。此外，与我们的设定不同，这些算法通常要求完整了解底层数据集。研究人员研究了代理（例如学校申请者）的付出与其在基于排名的奖励机制（例如学校录取）中的排名之间的权衡，并提出了设计稳健排序系统的方法 [20]。然而，我们关注的是查询数据源时所面临的挑战，即用户与数据源之间存在利益冲突。不同于我们的设定，[20] 中的代理所付出的努力可能代价高昂，例如投入课外活动的时间。

策略性沟通。 经济学和博弈论中已有大量关于利益不完全一致的代理之间策略性沟通的研究 [4, 7–9, 19]。我们的问题和方法与这些领域中策略性沟通的设定存在重要差异。首先，我们提出计算方法，以寻找排序查询的有影响力的均衡和策略。其次，我们应对在大规模（结构化）数据上进行存在利益冲突的查询所面临的挑战。

8 附录

本节包含我们理论结果的证明。

8.1 第 3.2 节的证明

定理 3.1 的证明。证明很简单，但我们提供一个较长的证明来阐释相关概念。

固定两个集合等价意图 $\tau \neq \tau'$ 和两个不同的解释 $\beta \neq \beta'$。考虑由此诱导的 2×2 博弈，其中用户可以发送两个查询 q, q'，而数据源在观察到查询后，从β, β' 中选择。

(⇐) 首先假设对于两个 $t \in \{r, s\}$,

$$U^t(\tau, \beta) \geq U^t(\tau, \beta') \qquad \text{and} \qquad U^t(\tau', \beta') \geq U^t(\tau', \beta).$$

定义策略

$$P^r : \tau \mapsto q,\ \tau' \mapsto q', \qquad P^s : q \mapsto \beta,\ q' \mapsto \beta'.$$

由于这两个意图发送不同的均衡路径上的查询（以正概率发送），贝叶斯法则意味着

$$\Pr(\tau \mid q) = 1, \qquad \Pr(\tau' \mid q') = 1.$$

因此，在观察到 q后，数据源选择 β所获得的期望效用为 $U^s(\tau, \beta)$，而选择 β' 所获得的期望效用为 $U^s(\tau, \beta')$。根据假设，

$$U^s(\tau, \beta) \geq U^s(\tau, \beta'),$$

因此，在 q之后， β是最佳响应。同样，在观察到 q'之后，

$$U^s(\tau', \beta') \geq U^s(\tau', \beta),$$

因此 β' 是 q'之后的最佳响应。

现在考虑用户的激励。如果意图 τ 遵循 P^r 并发送 q，则获得效用 U^r（τ, β）。如果它偏离并发送 q'，数据源以 β'响应，产生效用 U^r（τ, β'）。由于

$$U^r(\tau, \beta) \geq U^r(\tau, \beta'),$$

意图 τ 没有有利偏离（定义 2.5 中的第一项）。同样，类型 τ' 通过发送 q' 获得 $U^r(\tau', \beta')$，若偏离至 q则会获得 $U^r(\tau', \beta)$，并且

$$U^r(\tau', \beta') \geq U^r(\tau', \beta),$$

因此，意图 τ' 也没有有利偏离。因此，(P^r, P^s) 是一个均衡，并且它具有影响力，因为这两个意图引发了不同的解释。

反之，若对于两者都有 $t \in \{r, s\}$,

$$U^t(\tau, \beta') \geq U^t(\tau, \beta) \qquad \text{and} \qquad U^t(\tau', \beta) \geq U^t(\tau', \beta'),$$

交换 β和 β'的标签后，同样的论证依然成立。因此，这种交互具有影响力。

(⇏)假设这种交互具有影响力。那么，存在一个均衡，其中两个集合等价意图 $\tau \neq \tau'$ 引发了两个不同的解释 $\beta \neq \beta'$。相应的均衡路径上的查询为 q和 q'。由于该均衡具有影响力，用户在均衡路径上的行为会区分这两个意图。不失一般性，假设 τ 发送 q，而 τ' 发送 q'，并且数据源在 q 之后回复 β，在 q'之后回复 β'。

由于这些查询是由不同意图在均衡路径上发出的，贝叶斯法则意味着

$$\Pr(\tau \mid q) = 1, \qquad \Pr(\tau' \mid q') = 1.$$

因此，数据源的序贯理性（定义 2.5 的第二项）要求

$$U^s(\tau, \beta) \geq U^s(\tau, \beta') \qquad \text{and} \qquad U^s(\tau', \beta') \geq U^s(\tau', \beta),$$

因为 β在 q之后必须是最佳响应（最优），而 β'在 q'之后必须是最佳响应。

类似地，用户的激励相容性意味着

$$U^r(\tau, \beta) \geq U^r(\tau, \beta') \qquad \text{and} \qquad U^r(\tau', \beta') \geq U^r(\tau', \beta),$$

因为意图 τ可以从 q偏离至 q'，从而诱发 β'；而意图 τ'可以从 q'偏离至 q，从而诱发 β。

因此，对于 U^r和 U^s，陈述中的第一组不等式均成立。如果均衡转而将 τ与 β'匹配，并将 τ'与 β匹配，那么第二组不等式成立。因此，当且仅当所述两个条件之一成立时，这种交互才具有影响力。□

命题 3.5 的证明。这可由 [8] 中的定理 1 推出。□

定理 3.6 的证明。固定任意查询 q和任意元组 $e \in E$。令 $X_e := r_e^\tau + b_e$。由于 $r_e^\tau \in [z]$，在任何信念下， $X_e \in [1 + b_e, z + b_e]$几乎必然成立。对于任意解释排名 $\tilde{r}_e^\beta \in [z]$，每个条件期望效用项为

$$\mathbb{E}\left[L\left(r_e^\tau - (\tilde{r}_e^\beta + b_e)\right) \mid q\right] = \mathbb{E}\left[L\left(\tilde{r}_e^\beta - X_e\right) \mid q\right].$$

根据假设，对于每个 $r \in [z]$ 和每个 $\tilde{r}_e^\beta \in [z]$,

$$L\left(r - (r_e^{\beta\star} + b_e)\right) \leq L\left(r - (\tilde{r}_e^\beta + b_e)\right).$$

代入 $r = r_e^{\tau}$ 得

$$L(r_e^{\beta^\star} - X_e) \le L(\tilde{r}_e^{\beta} - X_e)$$

取给定 q 的条件期望可得

$$\mathbb{E}\Big[L(r_e^{\beta^\star} - X_e) \mid q\Big] \le \mathbb{E}\Big[L(\tilde{r}_e^{\beta} - X_e) \mid q\Big] \qquad \text{for all } \tilde{r}_e^{\beta} \in [z].$$

因此，对于元组 e，在每次查询 q之后，相同的解释排名 $r_e^{\beta^\star}$ 都是最优的。由于 U^s在元组之间具有可加性，数据源存在一种最优解释，为每个 e 分配相同的排名$r_e^{\beta^\star}$，且与 q无关。因此，数据源的响应不能取决于查询，任何均衡都不可能具有影响力。 □

推论 3.7 的证明。固定任意查询 q和 $e \in \tau(I)$。由于 $r(e, \tau(I)) \in [k]$，令 $X_e := \mathrm{r}(e, \tau(I)) + b(e) \in [1 + b(e)\ \ k + b(e)]$。如果 $b(e) \ge k - \frac{3}{2}$，则 $X_e \ge k - \frac{1}{2}$，因此 k是距离 X_e最近的可行排名；又由于 L关于 $|\cdot|$是非递减的，$r_e^{\beta} = k$在每次查询 q之后都是最优的。如果 $b(e) \le \frac{3}{2} - k$，则 $X_e \le \frac{3}{2}$，因此 1 是最近的排名，且 $r_e^{\beta} = 1$在每次查询 q之后都是最优的。因此，每个元组都有一个与查询无关的最优解释排名；根据可加性，整个解释与 q 无关。因此，任何均衡都不可能具有影响力。 □

8.2 第 5.1 节的证明

命题 4.3 的证明。证明通过代数化简得出。

对于 (i)，令 $r_e := \mathrm{r}(e, \tau)$, $\beta_e := \mathrm{r}(e, \beta)$, $\beta'_e := \mathrm{r}(e, \beta')$。那么

$$u^r(r_e, \beta_e) - u^r(r_e, \beta'_e) = -(r_e - \beta_e)^2 + (r_e - \beta'_e)^2 = 2r_e(\beta_e - \beta'_e) - (\beta_e^2 - \beta_e'^2).$$

对 e 求和可得

$$U^r(\tau, \beta) - U^r(\tau, \beta') = \sum_{e \in E} \gamma_e r_e - \sum_{e \in E} (\beta_e^2 - \beta_e'^2),$$

因此，无差异等价于 $\sum_{e \in E} \gamma_e \mathrm{r}(e, \tau) = \alpha^r$。

对于 (ii),

$$u_e^s(r_e, \beta_e) - u_e^s(r_e, \beta'_e) = \gamma_e r_e - \Big((\beta_e + b_e)^2 - (\beta'_e + b_e)^2\Big).$$

求和并在 Φ 下取期望，得到

$$\mathbb{E}_{\tau \sim \Phi}[U^s(\tau, \beta) - U^s(\tau, \beta')] = \sum_{e \in E} \gamma_e \Phi_e - \alpha^s,$$

其中 $\alpha^s = \sum_e \left((\beta_e + b_e)^2 - (\beta'_e + b_e)^2\right) = \alpha^r + 2\sum_e b_e(\beta_e - \beta'_e)$。因此，当且仅当 $\sum_{e \in E} \gamma_e \Phi_e = \alpha^s$ 时，数据源无差异。 □

引理的证明

4 4 $\gamma_{\bar{e}} = 。。$ 在推论的 0 $\bar{e} \notin e\ e'$ $\gamma_{e'} = -\gamma_e \neq 0$ 假设下，对于 {,}，有，因此命题 4.3 中的用户无差异条件简化为 $\mathrm{r}(e, \tau) - \mathrm{r}(e', \tau) = \frac{\alpha^r}{\gamma_e}$。此外，只有$e, e'$ 会影响截距，并且利用 $\mathrm{r}(e', \beta) - \mathrm{r}(e', \beta') = -(\mathrm{r}(e, \beta) - \mathrm{r}(e, \beta'))$ 得到

$$\alpha^s = \alpha^r + 2\big(b(e) - b(e')\big)\big(\mathrm{r}(e, \beta) - \mathrm{r}(e, \beta')\big) = \alpha^r + \gamma_e\big(b(e) - b(e')\big).$$

因此 $\alpha^r = \alpha^s - \gamma_e(b(e) - b(e'))$，代入后得到

$$\mathrm{r}(e, \tau) - \mathrm{r}(e', \tau) = \frac{\alpha^s}{\gamma_e} - (b(e) - b(e')).$$

这就确立了所声称的阈值形式 (2)。 □

命题 4.5 的证明。对于意图 τ，定义 $d(\tau) = \mathrm{r}(e, \tau(I)) - \mathrm{r}(e', \tau(I))$。根据引理 4.4，用户对 β 和 β' 的无差异仅取决于 $d(\tau)$ 高于还是低于阈值。无偏情形使用阈值 δ_0，有偏情形使用阈值 δ。若 $d(\tau) \le \min\{\delta_0, \delta\}$，则 $d(\tau) \le \delta_0$ 且 $d(\tau) \le \delta$，因此 τ 位于两个边界的同一侧（或恰好位于边界上），并在有偏和无偏情形下诱发相同的解释。若 $d(\tau) > \max\{\delta_0, \delta\}$，则 $d(\tau) > \delta_0$ 且 $d(\tau) > \delta$，因此 τ 同样位于两个边界的同一侧，并在两种情形下诱发相同的解释。因此，只有当元组的相对位置落在两个阈值之间时，诱发的解释才会改变。 □

定理 4.6 的证明。令 $r = \mathrm{r}(e, \tau(I))$ 和 $r' = \mathrm{r}(e', \tau)$，其中排名越小越好，并固定 $\delta \in \{1, \dots, z-1\}$。每个 δ划分两个元组所有可能排名的空间 $(r, r\prime) \in [z]^2$为两个集合 H 和 H′。直观而言，H 对应于排名对之差 $r\prime - r$至少为 δ 的情形，而 H′是其补集。我们考虑偏向 e的 δ-阈值划分：

$$\mathcal{H}(\delta) = \{(r, r') \in [z]^2 : r' - r \ge \delta\}, \qquad \mathcal{H}'(\delta) = [z]^2 \setminus \mathcal{H}(\delta).$$

因此，在 H (δ) 上，元组 e至少比 e' 高出 δ 个意图排名。

将（均匀）先验限制在 H 和 H′上，可得到后验平均排名。令 $\Phi_e(\mathrm{H}) = \mathbb{E}[r \mid (r, r\prime) \in \mathrm{H}]$，且 $\Phi_{e'}(\mathrm{H}) = \mathbb{E}[r\prime \mid (r, r\prime) \in \mathrm{H}]$。对离散三角形 $\{r\prime - r \ge \delta\}$ 直接计数可得

$$\Phi_e(\mathcal{H}) = \frac{z - \delta + 2}{3}, \qquad \Phi_{e'}(\mathcal{H}) = \frac{2z + \delta + 1}{3}.$$

对于补集均值 $\Phi_e(\mathrm{H}') = \mathbb{E}[r \mid (r, r\prime) \in \mathrm{H}\prime\ ]$和$\Phi_{e'}(\mathrm{H}\prime) = \mathbb{E}[r\prime \mid (r, r\prime) \in \mathrm{H}']$，对 $[z]^2$使用总和并化简可得

$$\Phi_e(\mathcal{H}') = \frac{\delta^3 - 3(z+1)\delta^2 + (3z^2 + 6z + 2)\delta + 2z(z^2 - 1)}{d(z, \delta)},$$

$$\Phi_{e'}(\mathcal{H}') = \frac{-\delta^3 + 3\delta z^2 + (3z^2 + 1)\delta + z(z^2 - 1)}{d(z, \delta)}.$$

特别地，补区域上的差异可准确化简为：

$$\Phi_e(\mathcal{H}') - \Phi_{e'}(\mathcal{H}') = s(z, \delta). \tag{4}$$

为简单起见，假设数据源解释 $\mathrm{r}(\cdot\ \beta) \in \mathbb{R}$（不进行舍入或截断）不受约束且取实值，并采用二次效用 $u_x^s(r_x^{\tau}, r_x^{\beta}) = -(r_x^{\tau} - (r_x^{\beta} + b_x))^2$。观察到 $X \in \{\mathrm{H}, \mathrm{H}'\}$ 后，唯一的最优后验解释为：

$$\mathrm{r}(x, \beta_X) = \Phi_x(X) - b_x, \qquad x \in \{e, e'\}. \tag{5}$$

尽管这些评分取实值，解释结果 $\beta(I)$ 通过按 $\mathrm{r}(\cdot\ \beta_X)$ 对元组进行排序（并返回前 k 个）来产生离散排序。因此，在 H′上，当且仅当 $\mathrm{r}(e, \beta_{\mathcal{H}'}) < \mathrm{r}(e', \beta_{\mathcal{H}'})$ 时，元组 e排名在 e'之前。

在 $u^r(r_x^{\tau}, r_x^{\beta}) = -(r_x^{\tau} - r_x^{\beta})^2$,中，用户对两种后验解释 $\beta_{\mathcal{H}}$和 $\beta_{\mathcal{H}'}$ 的无差异在 $(r, r\prime)$ 中形成一条仿射边界，该边界可以表示为 $(r' - r)$ 上的阈值。在离散整数网格 $[z]^2$ 上的阈值均衡中，任何实数无差异阈值 $t \in \mathbb{R}$ 都会诱导出完全相同的划分

H (δ)，只要 $t \in (\delta-1,\ \delta]$。评估用户的无差异条件可得

$$t = \frac{(\mathrm{r}(e', \beta_{\mathcal{H}}) + \mathrm{r}(e', \beta_{\mathcal{H}'})) - (\mathrm{r}(e, \beta_{\mathcal{H}}) + \mathrm{r}(e, \beta_{\mathcal{H}'}))}{2}$$
$$= \delta - g(z,\delta) + (b(e) - b(e')),$$

其中，将后验均值和最优数据源的后验解释（公式 5）代入，即得 $g(z,\ \delta)$。因此，所需的分离条件 $t \in (\delta-1,\ \delta]$ 成立，当且仅当

$$b(e) - b(e') \in \big(g(z,\delta) - 1,\ g(z,\delta)\big]. \tag{6}$$

若 $\delta > 1$，补区域 $\mathrm{H}'(\delta)$ 包含非空带状区域 $\mathrm{S}(\delta) = \{(r,\ r') \in [z]^2 :\ 1 \le r_\prime - r \le \delta - 1\} \subseteq \mathrm{H}_\prime(\delta)$，在该区域上，$r < r_\prime$（即 e 实际上排名高于 e'），并且还包含所有满足 $r > r_\prime$ 的意图（其中 e' 实际上排名高于 e）。若在由 $\mathrm{H}'(\delta)$ 诱导的后验解释下，数据源将 e 排在 e' 之前，则元组 e 由 e' 见证为不可信：

$$\mathrm{r}(e, \beta_{\mathcal{H}'}) < \mathrm{r}(e', \beta_{\mathcal{H}'}) \iff b(e) - b(e') > \Phi_e(\mathcal{H}') - \Phi_{e'}(\mathcal{H}')$$

将公式 (4) 中补区域上的差值代入此不等式，得到：$b(e) - b(e') > s(z,\ \delta)$。将其与公式 6 求交，得到偏差区间：

$$b(e) - b(e') \in \Big(\max\{g(z,\delta) - 1,\ s(z,\delta)\},\ g(z,\delta)\Big]$$

□

8.3 第 5.2 节的证明

定理 5.11 的证明。证明采用构造法，扩展命题 5.2 的结果。固定一个有序对 (e, e')，其中 $\delta := \delta^\star(e, e') \in \{1, \ldots, z-1\}$，并考虑可能排名差的空间中的两个区域。

$$\mathcal{H} = \{(r, r') \in [z]^2 :\ r' - r \ge \delta\}, \qquad \mathcal{H}' = [z]^2 \setminus \mathcal{H}\ .$$

根据命题 5.2，存在一个有影响力的均衡，该均衡由两条能够区分 H 与 H′ 的查询支撑。在我们的查询语言下，用户必须提交一个排序列表。由于 H 和 H′ 均非空，我们可以选择两个可执行查询 $q_{\mathcal{H}}$ 和 $q_{\mathcal{H}'}$，使其产生的排名分别与 H 和 H′一致。我们将数据源对均衡路径上的查询 $q_{\mathcal{H}}$ 和 $q_{\mathcal{H}'}$ 的响应定义为与命题 5.2 所保证的均衡响应相同。这保留了最佳响应和激励约束（定义 2.5 中的两个条件），因此在可执行查询下产生一个有影响力的均衡。最后，允许用户提交 q^{base} 的任意超排名并不会消除这个均衡。我们构造均衡策略：用户策略可以将全部概率分配给均衡路径上的两条查询 $q_{\mathcal{H}}$ 和 $q_{\mathcal{H}'}$，而其他任何超排名查询的概率均设为 0，数据源对这些查询的解释则在均衡路径外指定。因此，在超排名策略下存在一个有影响力的均衡。□

命题 5.15 的证明。令 m 为 q^{base} 中的元组数量，令 $r = |R(I)|$ 为可追加的元组数量。该算法恰好输出 $2^{m-1}\, a(r)$ 个超排名，其中 $a(r) = \sum_{k=1}^{r} k!\ S(r,\ k)$ 为有序贝尔数（Fubini 数）。因此，运行时间为 $\Theta\big(2^{m-1}\, a(r)\,(m+r)\big)$，并且关于 r 呈现超指数级（因为 $a(r) \ge r!$）。我们分别分析该算法的两个阶段，然后将其合并。

（仅合并计数。）q^{base} 中的元组构成一个具有 $m-1$ 个相邻边界的全预序。每种仅合并粗化都对应于选择要删除的边界；共有 2^{m-1} 种选择，每种选择都会得到唯一的粗化。

(追加计数。) 将 r 个带标签的项作为排序/并列后缀追加，与在一个含 r 个元素的集合上定义弱序（全预序）一一对应。此类弱序的数量为 $a(r) = \sum_{k=1}^{r} k!\ S(r,\ k)$，其中 $S(r,\ k)$ 表示将 r 个项划分为 k 个无标签块的方法数，而因子 $k!$ 表示将这些块排列成各个排名的方法数。 [28]

(乘积。) 后缀与所选的仅合并粗化无关，因此超排名总数为乘积 $2^{m-1}\, a(r)$。

(运行时间。) 算法会生成每种粗化一次（对相邻合并进行广度优先搜索），并针对每种粗化枚举可追加项的全部 $a(r)$ 种弱序。生成一个输出需要 $O(m+r)$ 时间（复制/合并 m 个元组并写入含 r 个项的后缀）。因此，运行时间为 $\Theta(2^{m-1}\, a(r)\,(m+r))$。

（超指数增长于 r。）由于 $S(r,\ r) = 1$，$k = r$ 项给出 $a(r) \ge r!$ 而 $a(r) \le \sum_{k=1}^{r} k^r \le r^{r+1}$。因此，对于固定的 $c > 1$，$a(r)$（进而总量）的增长速度快于任何 c^r，即关于 r 呈超指数增长。□

定理 5.16 的证明。我们通过从子集和进行卡普归约来证明 NP 困难性。

子集和。给定正整数 $w_1, \ldots, w_n$ 和目标值 W，判断是否存在 $S \subseteq \{1, \ldots, n\}$ 使得 $\sum_{j \in S} w_j = W$。

判定变体（MaxInf）。给定一个基础排序 $q^{\min} = e_1 \prec e_2 \prec \cdots \prec e_m$、效用 $U^s\ U^r$，（如下所述）以及阈值 $T \in \mathbb{Z}$，判断是否存在一个仅合并型超排名查询 $q \in \mathrm{Q}^*_{\text{merge}}(q^{\min})$，使得 $U^r(q,\ \beta^*(q)) \ge T$。此处，$\mathrm{Q}^*_{\text{merge}}(q^{\min})$ 表示由 $q^{\min}$ 通过合并任意子集的相邻边界（即将 $e_1, \ldots, e_m$ 划分为连续的并列块）所得到的超排名查询集合，而 $\beta^*(q) = \arg\max_\beta U^s(q,\ \beta)$。

由于 $\mathrm{Q}^*_{\text{merge}}(q^{\min}) \subseteq \mathrm{Q}^{\text{super}}(q^{\min})$，MaxInf 在 $\mathrm{Q}^*_{\text{merge}}$ 上的 NP 困难性意味着完整超排名问题是 NP 困难的。

归约。给定一个实例 $(w_1, \ldots, w_n, W)$，令 $m := n+1$ 并构造

$$q^{\min} = e_1 \prec e_2 \prec \cdots \prec e_m.$$

任何 $q \in \mathrm{Q}^*_{\text{merge}}(q^{\min})$ 都可以唯一地编码为一个二进制向量 $b = (b_1, \ldots, b_n) \in \{0,1\}^n$，其中 $b_j = 0$ 表示排名 j 和 j+1 之间的边界被合并（并列），而 $b_j = 1$ 表示该边界保留为切分。

固定一个任意的解释 $\beta^\dagger$（例如，规范解释）。定义数据源效用，使得 $\beta^\dagger$ 成为唯一的最佳响应：

$$U^s(q, \beta) := \begin{cases} 1 & \text{if } \beta = \beta^\dagger, \\ 0 & \text{otherwise.} \end{cases}$$

因此，对于所有 q，$\beta^*(q) = \beta^\dagger$。

定义用户效用（可根据 b 在多项式时间内计算）如下：

$$U^r(q, \beta^\dagger) := -\left|\sum_{j:\, b_j = 0} w_j - W\right|.$$

最后设置阈值 $T := 0$.

正确性。对于由b编码的任意仅合并超排名查询 q，

$$U^r\big(q,\beta^*(q)\big)\ge 0 \iff -\left|\sum_{j:\,b_j=0} w_j - W\right| \ge 0 \iff \sum_{j:\,b_j=0} w_j = W.$$

因此，令 $S=\{\,j:\,b_j=0\,\}$，当且仅当子集和实例有解时，存在 q 使得 $U^r(q,\ \beta^*(q))\ge 0$。

复杂度。构造 $q^{\min}$ 并显式给出效用需要 $\mathrm{poly}(n)$ 时间。此外，给定 b，计算 U^r 需要计算 $\sum_{j:\ b_j=0} w_j$，耗时 $\mathrm{poly}(n)$。因此，子集和可在多项式时间内归约到 MaxInf，从而证明其 NP 困难性。□

引理 5.18 的证明。固定 $q'\in \mathrm{Q}^{\mathrm{merge}}(q)$，并将其合并区间划分写为 $\Pi(q')=\{[i_1,j_1],\ \ldots\ldots,\ [i_t,j_t]\}$。由 U^r 的可加性，期望效用可分解为对排名落在每个区间 $[i_\ell,j_\ell]$ 内的元组集合求和。在独立同分布先验下，同一合并区间内的元组在条件上可交换，因此每个元组的期望逐元组效用都相同，为 $u_{q'}(i_\ell,\ j_\ell)$，该区间的贡献为 $n_{i_\ell:\ j_\ell}u_{q'}(i_\ell,\ \ j_\ell)$。最后，合并与 $[i,j]$ 不相交的区间只会改变这些不相交区间内部的并列情况，且不会对原排名位于 $[i,j]$ 的元组施加额外的序数约束。因此，对于 $[i,j]$ 中的元组，在 q' 下的后验（以及数据源对元组的逐一后验解释）与单区间合并查询 $q_{i:j}$ 下相同。因此，对于每个 $[i,j]\ \in\ \Pi(q')$，都有 $u_{q'}(i,\ \ j)=u_{ij}$；对各区间求和即可得到所述形式。□

命题 5.19 的证明。任何关于前缀 $[1..j]$ 的最优解都以某个最后的连续段 $[i..j]$ 结束。根据引理 5.18，该段的贡献为 $\mathit{Score}(i,\ \ j)$，而它之前所能取得的最佳贡献为 $\mathrm{OPT}(i-1)$。对 i 的选择取最大值，即得递推关系。反过来，对每个 j 选取最大的 i，即可构造出一个最优划分。□

参考文献

[1] Leonidas Akritidis. 2020年。产品分类与聚类。UCI 机器学习资料库。数字对象标识符: https://doi.org/10.24432/C5M91Z。[2] 亚马逊。2025年。亚马逊销售费用。https://sell.amazon.com/pricing 访问日期：2025-01-21。[3] Abolfazl Asudeh、H. V. Jagadish、Gerome Miklau 和 Julia Stoyanovich。2018年。关于获得稳定排名。*CoRR* abs/1804.10990（2018）。arXiv：1804.10990 http://arxiv.org/abs/1804.10990 。Matthew Backus、Thomas Blake 和 Steven Tadelis。2019年。廉价言谈信号传递的经验内容：议价中的应用。《政治经济学杂志》127, 4 2019 1599 1628 10 1086 701699（无）。–。[5] https://doi.org/ 。/ Barry Becker 和 Ronny Kohavi。。成人。UCI 机器学习资料库。数字对象标识符：https://doi.org/10.24432/C5XW20。[6] Shubham Bhatwal。2021年。航班价格预测。Kaggle。https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction。[7] Andreas Blume、Ernest K. Lai 和 Wooyoung Lim。2020年。第13章 战略信息传递：实验综述与理论基础。载于《实验博弈论手册》。爱德华·埃尔加出版社，英国切尔滕纳姆。https://doi.org/ 10 4337 9781785363337 00022 。/。Archishman Chakraborty 和 Rick Harbaugh。2007年。比较性廉价谈话。《经济理论杂志》132，1（2007），70–94。https://doi.org/10.1016/j.jet. 2005.07.003[9] Vincent P. Crawford 和 Joel Sobel。1982年。战略信息传递。《计量经济学》50, 6 1982 1431 1451 1913390（），–。[10] http://www.jstor.org/stable/ H. A. David 和 H. N. Nagaraja。。次序统计量（第 rd 版）。约翰·威立父子公司，纽约。https://doi.org/ 10 1002 0471722162 。/ Aaron Frias。。亚马逊产品数据集。Kaggle。https://www.kaggle.com/datasets/aaronfriasr/amazon-products-dataset/data。[12] Sidney Fussell。2019年。亚马逊的搜索引擎有偏见吗？很难判断。《大西洋月刊》（2019年9月18日）。[13] Ulrich Güntzer、Wolf-Tilo Balke 和 Werner Kießling。2000年。优化图像数据库的多特征查询。载于第26届超大型数据库国际会议论文集（VLDB '00）。Morgan Kaufmann 出版公司，美国加利福尼亚州旧金山，419–428。[14] Ihab F. Ilyas、George Beskales 和 Mohamed A. Soliman。2008年。关系数据库系统中 Top-k 查询处理技术综述。《ACM计算调查》40, 4, 11 2008 58 10 1145 1391729 1391730 文章（10月），页码。[15] https://doi.org/ 。/ 。Werner Kießling 和 Gerhard Köstler。。章节——偏好 SQL——设计、实现与实践。载于 VLDB '02：第28届超大型数据库国际会议论文集，Philip A. Bernstein、Yannis E. Ioannidis、Raghu Ramakrishnan 和 Dimitris Papadias（编）。Morgan Kaufmann，旧金山，990 1001 10 1016 978 155860869 6 50098 6 – 。https://doi.org/ 。/B - - /[16] - Jinyang Li 、 Yuval Moskovitch 和 H. V. Jagadish。。排序中有偏代表性群体的检测。载于 2023 年 IEEE第39届数据工程国际会议（ICDE）。IEEE计算机学会，美国加利福尼亚州洛斯阿拉米托斯，2167 2179 10 1109 55515 2023 00168 – 。https://doi.org/ 。/ICDE 。。Jinyang Li、Yuval Moskovitch 和 H. V. Jagadish。。排序中有偏代表性群体的检测。arXiv:2301.00719 [cs.LG] https://arxiv.org/abs/ 2301 00719 。Jinyang Li、Yuval Moskovitch、Julia Stoyanovich 和 H. V. Jagadish。2023年。满足多样性约束的查询优化。《VLDB基金会会刊》17卷2期（10月2023 106 118 10 14778 3626292 3626295），–。https://doi.org/ 。/ 。Lingxiao Li 和 Jia Xie。。比较性廉价谈话：来自住房市场的证据。《国际房地产评论》 28 卷 2 期（ 2025 年 None ），117–148。https://doi.org/None [20] Lydia T. Liu、Nikhil Garg 和 Christian Borgs。2022年。策略性排序。《机器学习研究论文集》151卷（2022年），2489–2518。出版方版权：版权所有 © 2022 作者；第25届人工智能与统计国际会议（AISTATS 2022）；会议日期：2022年3月28日至2022年3月30日。[21] Michael Mathioudakis、Carlos Castillo、Giorgio Barnabò 和 Sergio Celis。2020年。前 k 名候选人选择的平权行动政策：及其在大学招生政策设计中的应用。载于第35届ACM应用计算研讨会年会论文集。美国计算机协会，美国纽约州纽约市，1640–1649。https://doi.org/10.1145/3341105. 3373878[22] Dana Mattioli。2019年。亚马逊更改搜索算法，以提升自家产品。《华尔街日报》（2019年9月16日）。https://www.wsj.com/articles/amazon-changed-search-algorithm-in-ways-that-boost-its-own-products-11568645345 [23] Niccolò Meneghetti、Denis Mindolin、Paolo Ciaccia 和 Jan Chomicki。2015年。优先级天际线查询的输出敏感评估。载于2015年ACM SIGMOD数据管理国际会议论文集（澳大利亚维多利亚州墨尔本）（SIGMOD '15）。美国计算机协会，美国纽约州纽约市， 1955 1967 10 1145 2723372 2723736 – 。https://doi.org/ 。/ 。[24] Tripp Mickle。。苹果主导 App Store 搜索结果，阻碍竞争对手。《华尔街日报》（2019年7月23日）。https://www.wsj.com/articles/apple-dominates-app-store-search-results-thwarting-competitors-11563897221 [25] [ ] 8 4 MySQL。无日期。。MySQL。https://dev.mysql.com/doc/refman/ 。/en/case.html。访问日期：2025-02-10。[26] Jack Nicas 和 Keith Collins。2019年。苹果如何让自家应用在其掌控的 App Store 中超越竞争对手。《纽约时报》（2019年9月9日）。https://www.nytimes.com/interactive/2019/09/09/technology/apple-app-store-competition.html [27] Oracle。[无日期]。Oracle。https://docs.oracle.com/cd/B13789_01/appdev.101/b10807/ 13elems004.htm。访问日期：2025-02-10。[28] B.C. Rennie 和 A.J. Dobson。1969年。论第二类斯特林数。《组合理论杂志》7卷2期（1969年），116–121。https://doi.org/10.1016/S0021 9800 69 80045 1 - ( ) - D. Kossmann、S. Borzsony 和 K. Stocker。2001年。天际线算子。载于第17届数据工程国际会议论文集（ICDE '01）。IEEE计算机学会，美国，421。[30] C. Spearman。1961年。两种事物之间关联性的证明与测量。载于《个体差异研究：智力探索》。Appleton-Century-Crofts，美国康涅狄格州东诺沃克，45 58 10 1037 11491 005 – 。https://doi.org/ 。/ -[31] [ ] SQLServer。无日期。。SQLServer。https://learn.microsoft.com/en-us/sql/t-sql/language-elements/case-transact-sql?view=sql-server-ver16。访问日期：2025-02-10。[32] Julia Stoyanovich、Serge Abiteboul、Bill Howe、H. V. Jagadish 和 Sebastian Schelter。2022年。负责任的数据管理。《美国计算机协会通讯》65卷6期（2022年5月），64–74。https://doi.org/10.1145/3488717[33] Behzad Tabibian、Vicenç Gomez、Abir De、Bernhard Schölkopf 和 Manuel Gomez-Rodriguez。2020年。论后果性排序算法的设计，Jonas Peters 和 David Sontag（编），第124卷。PMLR，171–180。http://proceedings.mlr.press [34] Meike Zehlike、Francesco Bonchi、Carlos Castillo、Sara Hajian、Mohamed Megahed 和 Ricardo Baeza-Yates。2017年。FA*IR：公平 Top-k 排序算法。载于

*2017*年美国计算机协会信息与知识管理会议论文集（新加坡，新加坡）（*CIKM '17*）。美国计算机协会，美国纽约州纽约市，1569 1578 10 1145 3132847 3132938 – 。https://doi.org/ 。/ 。

[35] Meike Zehlike 和 Carlos Castillo。 。减少排序中的差异化曝光：一种学习排序方法。载于*2020*年万维网大会论文集（台北，台湾）（*WWW '20*）。美国计算机协会，美国纽约州纽约市，2849 2855 10 1145 3366424 3380048 – 。https://doi.org/ 。/ 。2020

[36] Meike Zehlike、Ke Yang 和 Julia Stoyanovich。 。排序公平性（第一部分）：基于分数的排序。《*ACM*计算调查》55卷6期，文章118（2022年12月），36页。2022

https://doi.org/ 10 1145 3533379 。/

[37] Meike Zehlike、Ke Yang 和 Julia Stoyanovich。2022年。《排序公平性》第二部分：学习排序与推荐系统。《*ACM*计算调查》55卷，6期，文章编号 117 2022 41 10 1145 3533380（12月），页码。https://doi.org/。/

[38] Karen Zraik。《农民因气候数据从政府网站删除而提起诉讼》。《纽约时报》（2024年2月25日）。https://www.nytimes.com/2025/02/24/climate/agriculture-farmer-website-data-lawsuit.html 2024