# 参数在排序函数中的重要性

**Christoph Standke** 亚琛工业大学，
德国亚琛
**Nikolaos Tziavelis** 加州大学圣克鲁兹分校，美国加利福尼亚州圣克鲁兹
**Wolfgang Gatterbauer** 东北大学，美国马萨诸塞州波士顿
**Benny Kimelfeld**
以色列海法的以色列理工学院，以及美国加利福尼亚州伯克利的 RelationalAI公司

arXiv:2601.06001v1 [cs.DB] 9 Jan 2026

**摘要**

给定列的权重对于确定表中元组的排序有多重要？为了解答这一类关于排序函数的解释问题，我们采用 Grohe 等人近期提出的框架 [ICDT'24]，研究列权重的 SHAP 分数计算。该分数的精确定义取决于三个关键组成部分：(1) 所使用的排序函数；(2) 用于量化采用其他权重对排序影响的效应函数；以及 (3) 基础权重分布。针对一系列基本排序函数和效应函数，我们分析了该框架不同实例化形式的计算复杂性，重点关注各列单个元组的概率独立有限分布。

对于排序函数，我们考察字典序以及由求和、最小值和最大值函数定义的基于分数的排序。对于效应函数，我们考虑全局、*top-k*和局部 视角：全局度量量化扰动后的排序与原始排序之间的差异；top-$k$ 度量考察 top-$k$ 答案集合的变化；局部度量则刻画对某个特定元组的影响。尽管所有情形都存在加性全多项式时间随机近似方案（全多项式时间随机近似方案），我们仍确定了精确计算的复杂度，并指出哪些情形可在多项式时间内求解，哪些情形是 #P-困难的。我们还证明，所有复杂度结果、下界和上界均可推广到一项相关任务：计算整列（不论其权重如何）的 Shapley 值。

**2012年ACM主题分类** 信息系统 →检索模型与排序

**关键词与短语** 排序、解释、Shapley 值、SHAP 分数

**资助** *Christoph Standke*：欧洲研究理事会资助项目 101054974（SymSim）和 德国研究联合会资助项目 GRK 2236（UnRAVeL）。然而，文中所表达的观点和意见仅代表作者本人，并不一定反映欧盟或欧洲研究理事会的观点和意见。欧盟和资助机构均不对此承担责任。*Wolfgang Gatterbauer*：美国国家科学基金会职业奖 IIS-1762268*Benny Kimelfeld*：德国研究联合会资助项目 KI 2348/1-1

**致谢** 本研究始于Dagstuhl研讨会24032： 数据库理论与逻辑中的表示、来源追溯与解释。

## 1 引言

一个排序函数以一组记录作为输入，并根据记录的条目值（通常称为特征）对这组记录进行排列。排序在决策中十分常见，因此自然会想要了解给定数据集排序结果的解释。近年来，信息检索领域对排序函数的解释给予了广泛关注 [26,27,34,39]此前也是如此，

## 2 参数在排序函数中的重要性

正如关于“可解释信息检索”的综述中所述 [2, 32]。排序解释大致可分为两类：一类是基于值的解释，旨在阐明某条记录的值如何使其排在某个位置（或者为何将其纳入或排除在前$k$个答案之外）；通常通过量化每个条目值对该记录状态的贡献来实现 [1, 10, 27]。另一方面，基于函数的解释旨在从特定记录或整个排列的角度，探究排序函数中的哪些方面导致了这一结果；通常通过量化函数组件对结果的贡献来实现 [16, 21, 34, 38]。本研究属于后一类。

通常，排序函数涉及参数。例如，如果排序由一个评分函数决定，而该函数是记录各属性的线性组合，那么参数就是权重，它们与记录的特征相关联。数据库领域此前的研究探讨了如何选择参数，使排序函数满足特定准则 [4, 9]。在这项工作中，我们研究参数选择对排序结果的影响。为此，我们采用 Grohe 等人 [20] 提出的框架，以衡量参数在数据库查询中的重要性；参数的贡献由其 SHAP 分数决定 [25]。与此前关于排序函数解释的研究不同，我们关注计算这种贡献的计算复杂性。

**SHAP 分数与 Shapley 值。** SHAP 分数 [25] 是 Shapley 值的一种具体形式；而 Shapley 值用于在合作博弈中为每位参与者分配份额，其中每个联盟（参与者集合）都能获得一定效用 [33]。Shapley 值以其发明者的名字命名，并且在满足某些理性公理的条件下是唯一的（例如，各份额之和等于整个集合的效用） [30, 33] 。除了在众多领域中的应用外，它还被用于机器学习 [12, 25, 31]、信息检索 [10, 21], 和数据库 [7, 11, 23]等以数据为中心的领域中的解释。SHAP 分数最初是为了解释机器学习模型而提出的，尤其用于将模型结果的责任归因于特定实例 [22, 25]的特征值。

具体而言，SHAP 是合作博弈中的 Shapley 值，其中特征值是玩家，而联盟 $C$ 的效用是这样一个实例的期望效应（模型输出）：该实例采用了 $C$ 的特征值，其余特征值则随机选取。在 Grohe 等人 [20], 的框架中，参数值发挥着与特征值相同的作用：无论是在将 SHAP 应用于机器学习解释时 [22, 25] ，还是在将 SHAP 应用于排序的基于值的解释时 [10, 28]，特征值都发挥着这一作用。此外，近期也有研究在基于函数的解释中使用 Shapley 值 [21]。

联系这项工作，考虑一个特定的参数化排序函数，以及其参数的一组特定取值。它们共同定义了给定记录集上的一种排序。要应用 SHAP 分数，还需要确定两个额外要素。第一个是一个分布，参数选择被假定从该分布中抽取。第二个是一个效应函数，用于确定由某个特定（随机）参数选择得到的排序与采用原始参数选择的基准排序之间的差异。效应函数体现了 Heuss 等人 [21]所说的“列表级特征归因”的不同含义。与 Grohe 等人 [20], 的研究类似，我们将讨论限制在简单的参数分布上，即有限且概率独立的参数（也就是“完全因子化”的分布 [37]）。

从计算角度来看，SHAP 分数涉及对参与者随机选择取期望（相对于该随机选择），再对随机参数选择取期望。然而，Grohe 等人 [20] 证明，SHAP 分数也可以视为在一个可高效采样的空间上取单次

期望；因此，在效应函数可在多项式时间内计算这一温和假设下，只要接受随机加法近似（全多项式时间随机近似方案），就能通过采样在多项式时间内计算参数的 SHAP 分数。这一技术也能为更一般的概率分布类别提供加法近似，包括可高效采样的完全因子化连续分布。此外，如果我们作出固定参数数量的数据复杂度假设，那么就可以通过显式枚举概率空间，在多项式时间内计算 SHAP 分数。我们研究对于给定的（非固定数量的）参数，能否在多项式时间内精确（且确定性地）计算 SHAP 分数。Van den Broeck 等人 [37] 已经证明，计算 SHAP 分数与计算期望效应之间存在双向多项式时间归约，而后者可以说是一个更简单的概念。因此，遵循他们的结果，本文主要关注期望效应。

**研究排序函数与效应函数**。我们的研究考察了多种基本排序函数和效应函数。对于排序，我们考察以下求和、最小值和最大值评分函数，以及字典序排序（降序或升序）。在所有这些函数中，参数都是记录属性上的系数（权重）。我们将数据集简单地视为一个矩阵（每行对应一条记录），并将参数视为按列施加的乘法权重。对于效应，我们考察三种类型：全局度量用于确定整个排序结果上的效应，前$k$ 度量用于确定对前$k$个答案集合的影响，而局部度量用于确定对单条记录的影响。我们考察的全局度量是排列之间的标准距离 [6, 29], ，具体包括随机排序（由参数的随机选择产生）与基准排序之间的肯德尔 tau、最大位移和汉明距离。前$k$ 度量包括（基准排序和随机排序的）前$k$ 个元素集合之间的对称差，以及用于指示前个元素集合是否存在任何差异的$k$ 二元指示函数。局部度量包括记录位置的变化，以及该记录是否属于前$k$ 个答案集合的状态变化。详细示例见第3节末尾。

**贡献**。 我们首先研究一个更简单任务的算法：给定两条记录 $r_1$ 和 $r_2$，计算（对于随机选择的参数） $r_1$ 在排序中位于 $r_2$ 之前的概率。对于我们考虑的所有排序函数而言，这个问题都是可处理的；而对于求和 ，我们作出必要的假设，即这些数采用一元表示（因为对于二进制表示，该问题是 $\mathrm{FP}^{\#\mathrm{P}}$-困难的[1] ）。我们的结果（总结于表1中）表明，一般而言，这种成对情形足以保证某些效应函数的可处理性，即肯德尔 tau、元组的位置以及小 $k$ 时某行的前部成员资格 $k$；对于其余情形，我们证明了困难性结果。我们还考虑记录数有界的情形，即竞争者数量较少的情形。在这种情况下，我们证明所有问题都变得可处理，但采用二进制数值表示的求和 除外。

最后，我们考虑排序的基于函数的解释的另一种变体，此时不含参数。目标是计算整列（属性）对排序的贡献。更准确地说，我们考虑合作博弈的 Shapley 值，其中

[1] 回顾一下，$\mathrm{FP}^{\#\mathrm{P}}$ 是这样一个函数类：其中的函数可以借助针对某个 #P 函数的预言机，在多项式时间内计算得到。若存在从 FP 中的每个函数到函数 $F$ 的多项式时间图灵归约，则函数 $^{\#\mathrm{P}}$是 $\mathrm{FP}^{\#\mathrm{P}}$ -困难的。此类问题至少与多项式层次中的所有问题一样困难 [35]。

**表1** 复杂度结果总结。汉明距离的复杂度与 MD（最大位移）相同。前$k$ 视角的效应的复杂度与前$k$ 成员资格相同。当属性数量固定时，所有问题都可在多项式时间内求解。当记录数量固定时，它们同样可在多项式时间内求解，但使用二进制数值表示的 求和 除外。所有 FP$^{\#P}$困难性结果实际上都是 FP$^{\#P}$完备性，这是由于在 FP 中进行概率计算的标准技术$^{\#P}$ [14, 19]。

| | 全局 | | 局部 | | |
|---|---|---|---|---|---|
| | | | | Top-$k$ 成员资格 | |
| 效应 / 排序 | 肯德尔的 $\tau$ | MD | 位置 | 固定 $k$ | 给定 $k$ |
| 一元表示的求和 | P | FP$^{\#P}$-h。 | P | FP$^{\#P}$-h. | FP$^{\#P}$-h。 |
| 二进制求和 | FP$^{\#P}$-h. | | FP$^{\#P}$-h. | FP$^{\#P}$-h. | |
| 最大值升序 / 最小值降序 | P | | P | FP$^{\#P}$-h。 | |
| 最大降序/最小升序 | P | | P | P | |
| 字典序 | P | | P | FP$^{\#P}$-h。 | |

参与者是各列，而列集合的效用是仅考虑该集合中的列、忽略其余列时所得排序的效应。Heuss 等人提出了这项任务。 [21] 他们将忽略某一列的操作称为“屏蔽特征向量”，但未进行复杂度分析。我们证明，该问题可在多项式时间内归约为本文重点研究的问题，即计算参数选择的 SHAP 分数。因此，我们用于计算参数 SHAP 分数的算法也可用于计算列的 Shapley 值。此外，我们证明，计算 SHAP 分数的困难情形，对于列的 Shapley 值而言也同样困难。

**组织结构**。 本文其余部分安排如下。下一节将介绍预备定义。在第3节中，我们将描述本文研究的形式化框架和计算问题，即参数的 SHAP 分数、期望效应以及列的 Shapley 值。我们将在第4节中给出计算期望的算法，并在第5节中建立下界。最后，我们将在第6节中讨论将结果扩展到列的 Shapley 值，并在第7节中作出总结。由于篇幅有限，部分证明见附录。

## 2 预备知识

我们首先介绍本文中始终使用的预备概念和术语。

矩阵与排列。对于自然数 $n$，我们用 $[n]$ 表示集合 $\{1, \ldots, n\}$。我们还用 $S_n$ 表示 $[n]$ 上排列的集合，其中 $S_n$ 中的排列是双射函数 $\pi : [n] \to [n]$。我们用 $\pi_{\mathrm{id}}$ 表示由 $\pi_{\mathrm{id}}(i) = i$ 定义的恒等排列，对所有 $i \in [n]$ 均成立。我们用 S 表示所有排列的集合，即 $S := \cup_n S_n$。我们用 $M^{n\times m}$ 表示有理数上的所有 $n \times m$ 矩阵的集合，这些矩阵有 $n$ 行和 $m$ 列；用 M 表示所有维度的矩阵的集合（即 $M := \cup_{n,m} M^{n\times m}$）。如果 $M \in M^{n\times m}$ 且 $C \subseteq \{1, \ldots, m\}$ 表示列集合，那么我们用 $M|_C$ 表示从 M 中移除除 $C$ 中列以外的所有列后得到的 $n \times |C|$ 矩阵。换言之，$M|_C$ 是 M 在 $C$ 上的包语义投影。我们可以将矩阵的一行称为

一个 元组。我们用粗体表示这样的行（以及每个数值向量）$\mathbf{v}$，并用 $\mathbf{v}[j]$ 表示其第 $j$ 个分量。

就计算而言，我们假设每个数都表示为一对 $(a, b)$，代表有理数 $a/b$，其中 $a$ 和 $b$ 是以二进制表示的整数。当我们说一个数以一元编码表示时，指的是将整数 $a$ 编码为长度为 $a$的字符串。

**排序函数**。 我们将函数 $r : \mathcal{M} \to \mathcal{S}$ 称为排序函数，它将每个矩阵 $\mathbf{M} \in \mathcal{M}^{n\times m}$ 映射为 $\mathcal{S}_n$中的一个排列；因此， $r$ 对其输入矩阵$\mathbf{M}$的各行进行排序。具体而言，我们将关注若干种排序函数 $r$:

- 根据一行 $(a_1, \ldots, a_m)$ 的得分递减或递增排序，其中得分可以是 $a_j$之和（记为 Sum）、 $a_j$中的最大值（记为 Max），或 $a_j$中的最小值（记为 Min）。对于每个评分函数 $s$，我们分别用$r_s^{\text{dsc}}$和 $r_s^{\text{asc}}$表示按 $s$递减和递增的排序结果。例如， $r_s^{\text{dsc}}(i) < r_s^{\text{dsc}}(j)$ 表示，只要$s(\mathbf{t}_i) > s(\mathbf{t}j)$成立，第 $i$行 $\mathbf{t}_i$ 就排在第 $j$行 $\mathbf{t}j$之前。按各行从左到右的字典序排序，记为 $r_{\text{Lex}}$。
-

在我们的分析中，对于按 最小值 和 最大值排序而言，得分的排序方向（升序/降序）很重要，如表1所示。对于按 求和 排序而言，方向并不重要，因为我们可以通过将各项乘以负数（或用矩阵中的最大值减去每个值）来切换方向；因此，我们简单地将 $r_{\text{Sum}}$ 视为 $r_{\text{Sum}}^{\text{asc}}$（因为 $r_{\text{Sum}}^{\text{dsc}}$ 与之等价）。在字典序中，出于同样的原因，我们假设每一列的顺序都是升序（即较小的数排在较大的数之前）。同样出于这个原因，我们不再显式考虑按 最小值排序，因为这等同于对取负后的矩阵按反向顺序进行 最大值 排序。

为了使框架定义明确，我们需要处理平局。为此，我们将按行索引排序；也就是说，如果按排序规则排列时，行 $i$ 和 $i'$ 与$i < i'$并列，那么我们优先排列第 $i$ 行。

**Shapley 值**。合作博弈是一个二元组 $(P, \nu)$，其中 $P$ 是一个有限的参与者集合，$\nu : 2^P \to \mathbb{R}$ 是一个效用函数，它为每个联盟 $C \subseteq P$ 关联一个值 $\nu(C)$，使得 $\nu(\emptyset) = 0$。参与者 $p \in P$ 的 Shapley 值由以下公式定义： [30,33]。

$$\text{Shapley}(p, \nu) = \sum_{C \subseteq P \setminus \{p\}} \frac{|C|!(|P| - |C| - 1)!}{|P|!} \cdot (\nu(C \cup \{p\}) - \nu(C)) \tag{1}$$

直观地说，我们考虑这样一种情形：从空集开始，依次选择参与者且不重复；参与者 $p \in P$ 的 Shapley 值是加入 $p$时效用增量的平均值。

**SHAP 分数**。 设 $f$： $D^m \to \mathbb{R}$ 是定义在某个定义域 $D$上的 $m$元函数。设 $\Pi$ 是 $D^m$上的离散分布，且 $\mathbf{w} = (w_1, \ldots, w_m)$ 是 $\Pi$ 中概率非零的一个元组。对于 $j = 1, \ldots, m$， $j$相对于（w.r.t.）$\mathbf{w}$的 *SHAP* 分数 记作 Shap$(j, f, \mathbf{w})$，其值为合作博弈 $(P, \nu)$ 的 Shapley$(j, \nu)$，该博弈定义如下 [25]。

- $P := \{1, \ldots, m\}$； $\nu(C) := \mathbb{E}_{\mathbf{u}\sim\Pi}[f(\mathbf{u})$
- $\mid u_\ell = w_\ell$ 对所有 $\ell \in C]$均成立。

也就是说，参数子集 $C$ 的效用是 $f$ 在分布 Π下的期望，条件是 $C$ 中的每个参数都取自w的特定值。

## 3 框架与计算问题

我们的目标是计算列及列权重对给定矩阵各行排序的贡献。为此，我们需要分析：如果删除某些列或更改其权重，排序会发生多大变化。因此，我们需要采用一种衡量排列差异的方法。我们将这类度量称为效应函数，因为它们决定了列变动所产生的效应。

**效应函数。** 给定一个矩阵M 和一个具有原始顺序$\pi_0 := r$(M)的排序函数 $r$ 。对各列（或仅对部分列）应用具有不同权重的排序函数后，我们会得到一个不同的排列 $\pi$。我们将重点讨论三类效应函数：

- 全局视角： 与 $\pi$ 原始排列相差多远？ 为此，我们可以采用若干种衡量排列之间距离的概念。这里介绍三种常用的距离，其中前两种曾用于 RankSHAP 长篇版本中的排序解释 [10]。
  - 肯德尔 tau： $e_{k\tau}(\pi) := \sum_{1\le i<j\le n} \mathbb{1}_{\pi_0(i)<\pi_0(j)\wedge\pi(i)>\pi(j)}$ 确定 $\pi$ 与原始顺序之间成对不一致（交换）的数量。该效应函数的取值范围为 $[\binom{n(n-1)}{2}]$。最大位移（MD）距离：$e_{\text{md}}(\pi) := \max_{1\le i\le n}|\pi(i) - \pi_0(i)|$ 是两个排列中某项位置差的最大值。该差值属于 $[n-1]$中的一个数。汉明距离： $e_{\text{Ham}}(\pi) := \sum_{1\le i\le n} \mathbb{1}_{\pi(i)\neq\pi_0(i)}$ 确定元组在原始顺序与排列后的顺序中不同的位置数量。其取值可以是 $[n]$。
- Top-$k$ 视角：对前$k$ 个答案集合有何影响？为此，我们将考虑两个函数：$T = \{\pi^{-1}(1)\ldots, \pi^{-1}(k)\}$ 和 $T_0 = \{\pi_0^{-1}(1)\ldots, \pi_0^{-1}(k)\}$
  - 前$k$ 差异： $e^k_{\Delta}(\pi)$: $= |T \cup T_0| - |T \cap T_0|$ 是前$k$ 个元素集合之间对称差的大小。前$k$ 任意变化： $e^k_{\text{any}}(\pi)$: $= \mathbb{1}_{T\neq T_0}$ 用于确定前$k$ 个元素是否有任何 变化。
- 局部视角：对特定行的影响是什么 $i$？为此，我们将考虑两个函数。
  - 位置： $e_{\text{pos}}(i,\ \pi)$: $= \pi(i) - \pi_0(i)$表示第 $i$行的位置变化，取值范围为$\{-n+1,\ \ldots\ldots,\ n-1\}$。前$k$成员资格： $e^k_{\text{top}}(i,\ \pi)$: $= \mathbb{1}_{\pi(i)\le k} - \mathbb{1}_{\pi_0(i)\le k}$用于确定第 $i$行在前$k$个元组中的成员资格如何变化。其取值范围为$\{-1,\ 0\}$或$\{0,\ 1\}$。

在本文余下部分的分析中，我们重点关注四种效应：肯德尔 tau、MD、位置和 top-$k$ 成员资格。汉明距离的复杂度结果与 MD 的相同（见D节）。top-$k$ 视角下影响的复杂度结果与 top-$k$ 成员资格的相同（见 D 节）。

**计算 SHAP 分数**。接下来，我们假设输入矩阵 $\mathbf{M}$ 的列带有权重，目标是计算给定权重的 SHAP 分数。具体而言，假设我们的数据是一个二元组 $(\mathbf{M}, \mathbf{w})$，其中 $\mathbf{M} \in \mathcal{M}^{n\times m}$ 且 $\mathbf{w} = (w_1, \ldots\ldots, w_m) \in \mathbb{Q}^m$是列权重序列。二元组 $(\mathbf{M}, \mathbf{w})$ 表示矩阵 $\mathbf{M} \circ \mathbf{w}$，该矩阵由将第 $j$ 列乘以 $w_i$ 得到，其中 $i = 1, \ldots\ldots, m$；也就是说，对 $\mathbf{M}$ 的每一行应用 $\mathbf{w}$ 进行逐元素乘法，换言之，$\mathbf{M}' = \text{diag}(\mathbf{w}) \cdot \mathbf{M}$，其中 $\text{diag}(\mathbf{w})$ 是对角元素由 $\mathbf{w}$ 给出的对角矩阵。

我们还假设，除了 $\mathbf{M}$ 和 $\mathbf{w}$之外，我们还给定一个有限概率分布 $\Pi$，其定义于 $\mathbb{Q}^m$。我们的复杂度研究仅限于列间独立分布（即 完全因子分解的 分布 [37]），因此，$\Pi$由有限分布 $\Pi_1$、……、$\Pi_m$ 表示，每个权重参数对应一个定义于 $\mathbb{Q}$ 上的分布；每个分布均以值—概率对列表的形式给出；此类值—概率对的总数记为 $|\Pi|$。

令 $r$ 为一个排序函数，$e$ 为一个效应函数。给定数对 $(\mathbf{M},\mathbf{w})$ 和列编号 $j$，我们的目标是计算 $\text{Shap}(j, f, \mathbf{w})$，其中 $f(\mathbf{u}) := -e(r(\mathbf{M} \circ \mathbf{u}))$，也就是说，计算将$\mathbf{w}$ 的权重替换为$\mathbf{u}$的权重对排序产生的效应的相反数。我们将此值记为 $\text{Shap}\langle r, e\rangle(j,\mathbf{M}, \mathbf{w}\ \Pi)$。使用相反数（即减号）的原因是，我们越接近原始排序顺序 $\pi_0$（即效应越小），就越认为其对实际在位者 $\pi_0$ 的贡献更大。

| |
|---|
| **问题：** SHAP$\langle r,\ e\rangle$：SHAP 分数计算**固定：** 排序函数 $r$ 和效应函数 $e$**输入：** 矩阵 $\mathbf{M}$、权重向量 $\mathbf{w}$、分布 $\Pi$以及$\mathbf{M}$的列号 $j$ **目标：** 计算 Shap$\langle r,\ e\rangle(j,\ \mathbf{M},\ \mathbf{w}\ \Pi)$ |

在本问题及我们考虑的其他问题中，如果效应函数 $e$ 是局部的，则输入还包括该效应的行号 $i$。此外，如果 $e$ 是 top-$k$ 成员资格，则输入还包括 $k$（除非我们明确说明假设 $k$ 为固定值）。

Van den Broeck 等人 [37] 已证明，对于有限且相互独立的参数，计算 SHAP 分数在计算上等价于计算 $f$的期望值，而后者可以说更容易处理。

▶ **定理1**。[37, 定理 2] 设 $f$ 为一个可接受任意数量数值参数且能在多项式时间内计算的函数。以下问题在多项式时间（图灵）归约下彼此等价。

1. 计算 $\text{SHAP}(j,\ f,\ \mathbf{w})$，给定 $\mathbf{w}$，$j$和 $\Pi$。
2. 计算 $\mathbb{E}_{\mathbf{u}\sim\Pi}[f(\mathbf{u})]$，，给定 $m$ 和 $\Pi$。

因此，我们还将研究以下问题。

| | |
|---|---|
| **问题：** | EXP$\langle r,\ e\rangle$：期望计算 |
| **固定：** | 排序函数 $r$ 和效应函数 $e$ |
| **输入：** | 矩阵 $\mathbf{M}$ 和分布 $\Pi$ |
| **目标：** | 计算 $\mathbb{E}_{\mathbf{u}\sim\Pi}[e(r(\mathbf{M}\circ\mathbf{u}))]$ |

**计算 Shapley 值**。设 $r$ 为排序函数，$e$ 为效应函数。给定矩阵 $\mathbf{M} \in \mathcal{M}^{n\times m}$ 和列编号 $j \in \{1, \ldots, m\}$，第 $j$ 列的 Shapley 值定义为博弈 $(P, \nu)$ 中的 $\text{Shapley}(i, \nu)$，该博弈定义如下：

- $P := \{1, \ldots, m\}$.

- $\nu(C) := -e(r(\mathbf{M}|_C))$，也就是说，这是仅对 $C$ 的列应用 $r$ 后得到的排序效应的倒数（倒数的理由与 SHAP 相同）。[2]

我们将这个值记为 Shapley$\langle r, e\rangle(j, \mathbf{M})$。

**问题：** Shapley$\langle r, e\rangle$：Shapley 值计算**固定：** 排序函数 $r$ 和效应函数 $e$**输入：** 矩阵 $\mathbf{M}$ 及其列号 $j$ $\mathbf{M}$**目标：** 计算 Shapley$\langle r, e\rangle(j, \mathbf{M})$

近似。在转向精确算法之前，我们想强调的是，只要 $r$ 和 $e$ 能在多项式时间内计算，且 $e$ 的取值范围关于行数为多项式，那么这三个问题 EXP$r, e$、SHAP$r, e$ 和 Shapley$r, e$ 都可以高效地进行加性近似。本节介绍的所有排序函数和效应函数都满足这一条件。正如我们将在第5节 5.2 中展示的那样，对于乘法近似，情况并非总是如此。由于这三个问题都可以表示为概率空间上随机变量的期望，且该概率空间支持高效采样，因此可以通过基本的蒙特卡洛采样实现加性近似：对于 EXP$r, e$，使用 $\Pi$；对于 Shapley$r, e$，使用 $\mathsf{S}_m$；对于 SHAP$r, e$，使用 $\Pi \times \mathsf{S}_m$。在非常温和的假设下（见 [20]），即使参数分布不一定相互独立且为离散分布，或者参数分布相互独立、为连续分布且可高效采样，我们也可以利用这一思路进行高效的加性近似。[3] 这一简单思路后来得到进一步改进，从而实现更快的加性近似（例如 [8]）。因此，即使在我们证明精确计算很困难的情况下，我们的框架也能应用于实践。

**一个详细示例。**在开始复杂度分析之前，我们先来看一个小例子，详细说明该框架。考虑以下矩阵 $\mathbf{M}$：

| id | $a_1$ | $a_2$ |
|---|---|---|
| 1 | 20 | 26 |
| 2 | 30 | 13 |
| 3 | 40 | 0 |
| 4 | 0 | 39 |

假设我们使用求和并按降序对行进行排序，评分参数为$\mathbf{u} \in \{1,2\} \times \{1,2\}$。我们希望衡量在$\mathbf{u}$ 的所有可能取值中（假设其服从均匀分布），选择权重向量 $\mathbf{w} = (1, 1)$ 所产生的影响。

为衡量这一点，我们首先确定每个$\mathbf{u}$对应的排序结果：对于$\mathbf{u} = \mathbf{w} = (1,1)$，得分为 $(46, 43, 40, 39)$，因此得到排序结果 $(1, 2, 3, 4)$。同理，对于$\mathbf{u} = (1, 2)$，得到 $(4, 1, 2, 3)$；对于$\mathbf{u} = (1, 2)$，得到 $(3, 2, 1, 4)$；对于$\mathbf{u} = (2, 2)$，再次得到 $(1, 2, 3, 4)$。

[2] 对空列集应用排序函数 $r$ 总会得到矩阵序 $\pi_{\mathsf{id}}$，因为 $\mathbf{M}|_\emptyset$ 的所有元组都相同。特别是，这一定义不取决于评分函数在空列集上的取值。[3] 对于连续分布，通过条件表示定义 SHAP 分数并不明确，因为作为条件的集合测度为 0。对于独立分布，我们可以通过解体推导出一个兼容的定义：将 $C$ 中的参数固定为其参考值，并对其他参数随机取值。

| $\mathbf{u}$ | 分数 | 排列 $\pi$ | $e_{k\tau} = d_{k\tau}(\pi, \pi_0)$ | $e_{\text{pos}}(4, \pi)$ |
|---|---|---|---|---|
| $\mathbf{w} = (1,1)$ | (46,43,40,39) | $\pi_0 = (1, 2, 3, 4)$ | 0 | 4 |
| $(1, 2)$ | (72,56,40,78) | $(4, 1, 2, 3)$ | 3 | 1 |
| $(2, 1)$ | (66,73,80,39) | $(3, 2, 1, 4)$ | 3 | 4 |
| $(2, 2)$ | (92,86,80,78) | $(1, 2, 3, 4)$ | 0 | 4 |

现在，我们需要选择一个效应函数 $e$ 来衡量两个排序结果之间的距离。我们将考察肯德尔τ距离 $e_{k\tau}$ 以及第四行的位置 $e_{\text{pos}}(4, \pi)$。我们观察到 $d_{k\tau}((1,2,3,4),(4,1,2,3)) = d_{k\tau}((1,2,3,4),(3,2,1,4)) = 3$，并且第四行在除 (4, 1, 2,3) 之外的所有排序结果中都处于最后一个位置；在该排序结果中，它位于第一位。

对于 SHAP 分数的计算，列集合 $C$ 的估值是：将 $C$ 中各列的参数值固定为$\mathbf{w}$ 的值，并随机选择其余列时，负效应函数的期望值。例如，对于 $C = \{2\}$，我们得到 $e = e_{k\tau}$

$$\begin{aligned}\nu_{k\tau}(\{2\}) &= -\frac{e_{k\tau}\big(r_{\text{Sum}}(\mathbf{M} \circ (1,1))\big) + e_{k\tau}\big(r_{\text{Sum}}(\mathbf{M} \circ (2,1))\big)}{2} \\ &= -\frac{d_{k\tau}\big((1,2,3,4),(1,2,3,4)\big) + d_{k\tau}\big((1,2,3,4),(3,2,1,4)\big)}{2} = -\frac{3}{2},\end{aligned}$$

对于 $e = e_{\text{pos}}(4, \pi)$，我们得到

$$\nu_{\text{pos}(4)}(\{2\}) = -\frac{(4-4)+(4-4)}{2} = 0.$$

计算每个 $C \subseteq \{1, 2\}$ 的估值 $\nu$，可得如下结果。对于 $e = e_{k\tau}$，我们得到

$$\begin{aligned}\textsc{Shap}(1) &= \frac{1}{2}\big(\nu_{k\tau}(\{1,2\}) - \nu_{k\tau}(\{2\})\big) + \frac{1}{2}\big(\nu_{k\tau}(\{2\}) - \nu_{k\tau}(\emptyset)\big) \\ &= \frac{1}{2}\big(0 - (-\frac{3}{2})\big) + \frac{1}{2}\big(-\frac{3}{2} - (-\frac{3}{2})\big) = \frac{3}{4}\end{aligned}$$

Shap(2) $= \frac{3}{4}$ 也如此，因此这两种参数选择对于依据 $d_{k\tau}$ 衡量的排序全局变化具有相同的影响。这反映了上表第 4 列的对称性。相比之下，对于 $e = e_{\text{pos}}(4, \pi)$，我们得到 Shap(1) $= \frac{3}{8}$和 Shap(2) $= -\frac{9}{8}$。这符合我们的直觉：将第二个权重固定为 1 不利于第 4 行的排序，因为第 4 行唯一合适的权重向量是 $\mathbf{u} = (1,2)$。

在本文余下部分中，我们将研究精确求解本节所定义计算问题的复杂度。

## 4 期望（及 SHAP 分数）的精确算法

本节将展示计算期望效应的问题 表1 中 $\text{EXP}\langle r, e\rangle$ 的可处理性结果。结合 定理1，这些结果给出了表中 SHAP 分数问题的可处理性结果。我们先从一个更基础的问题开始，稍后将使用它进行期望计算。

### 4.1 成对情形的算法

我们首先给出以下问题的算法：给定一个元组$x$ 在排序中排在另一个元组 $y$ 之前的概率是多少？由于我们只关心它们的相对顺序，可以忽略表中的所有其他元组，并假设 $n = 2$。尽管这种成对情形看起来限制很大，但我们将在下一节看到，它构成了所有算法的基础。

## 10 排序函数中参数的重要性

**问题：** PREC$\langle r\rangle$：优先级概率计算**固定项：** 排序函数 $r$**输入：** 具有两行的矩阵 $\mathbf{M}$，分布 $\Pi$**目标：** 计算 $\mathbb{P}_{\mathbf{u}\sim\Pi}[s(1) < s(2)]$，其中 $s$ 表示 $r(\mathbf{M}\circ\mathbf{u}$

计算优先级概率的一种简单方法，是遍历指数级数量的权重 $\mathbf{u}$ 取值，并检查由此得到的顺序是否满足元组优先关系。我们的高效算法利用了这样一个事实：许多取值可以归为一组，因此只需考虑多项式数量的可能性。本文讨论的所有排序函数都能采用这种方法，唯独求和例外；对于求和，我们的算法是伪多项式时间的，也就是说，它所考虑的情况数量相对于给定数值的大小是多项式的。

▶**定理 2**。 问题 PREC$\langle r^{\mathsf{asc}}_{\mathsf{Max}}\rangle$、PREC$\langle r^{\mathsf{dsc}}_{\mathsf{Max}}\rangle$ 和 PREC$\langle r_{\mathsf{Lex}}\rangle$ 均可在多项式时间内求解。若矩阵 $\mathbf{M}$以及 $\Pi$ 中的权重均为采用一元编码的整数，则问题PREC$\langle r_{\mathsf{Sum}}\rangle$可在多项式时间内求解。

下面我们针对 定理 2 中的每种排序函数给出相应算法。为便于阅读，我们用 $\mathbf{x}$ 和 $\mathbf{y}$ 表示矩阵中的两个元组$\mathbf{t}_1$ 和$\mathbf{t}_2$ 。我们的目标是计算 $\mathbf{x}$ 优先于$\mathbf{y}$ 的概率；不失一般性，我们假设平局时这一关系不成立。

**求和排序**。 对于求和排序函数，我们的算法依赖动态规划，并采用与计数背包问题 [18] 的标准伪多项式算法类似的方法。具体而言，我们考虑一个子问题：将矩阵 $\mathbf{M}$ 限制在列 $j,\ldots,m$，并计算元组 $\mathbf{y}$ 的总分严格超过元组 $\mathbf{x}$ 超过 $s$ 的概率。我们可以递归地表达这一过程：对于权重 $u_j$ 的每种赋值，考察移除第 $j$ 列后得分差如何变化：

$$R[j,s] = \sum_{v_j\in\mathrm{supp}(\Pi_j)} \mathbb{P}[u_j = v_j]\cdot R[j+1, s - v_j(\mathbf{y}[j]-\mathbf{x}[j])] \tag{2}$$

$$R[m+1,s] = 1 \text{ if } s>0, \text{ and } R[m+1,s] = 0 \text{ if } s\le 0 \tag{3}$$

最终答案为 $R[1,0]$。运行时间为 $\mathcal{O}(|\Pi|\cdot||s||)$，其中 $||s||$ 是动态程序中 $s$ 的可能取值数量。如果输入由一元编码的整数构成，那么 $||s||$ 相对于输入规模是多项式的。事实上，假设矩阵 $\mathbf{M}$ 中的整数范围为 $[-c,c]$，且 $\Pi$ 中的权重范围为 $[-b,b]$。那么 $||s||$ 的上界为 $\sum_{j=1}^{m} b\cdot 2c$，属于 $\mathcal{O}(mbc)$。

**最大值排序**。对于最大值排序函数，关键在于：比较两个元组时，哪个元组是“获胜者”由单个列决定——也就是取值最大的那一列。因此，我们可以将问题分解为若干不同的情况，每种情况都对应一个决定结果的特定列。

▶**示例 3.** 设 $\mathbf{x}=(3,5,2)$ 和 $\mathbf{y}=(4,1,6)$ 是两个元组，并假设权重$\mathbf{u}$ 从$\{0,1\}$中均匀抽取，排序函数为 $r^{\mathsf{asc}}_{\mathsf{Max}}$。要使 $\mathbf{x}$ 排在 $\mathbf{y}$之前，需要满足 $\max(u_1\cdot 3, u_2\cdot 5, u_3\cdot 2) < \max(u_1\cdot 4, u_2\cdot 1, u_3\cdot 6)$。我们考虑该事件发生的三种不同情况：（情况 1）第一个$\mathbf{y}$坐标 $u_1\cdot 3$ 是整体最大值

**算法 1** 按 Max 升序排序的优先级计算。

**输入**：元组$\mathbf{x}$ 和$\mathbf{y}$，分布 $\Pi$ //假设$\mathbf{y}$在$\mathbf{x}$ 之前，按平局判定规则排列
**输出**：在 $r^{\text{asc}}_{\text{Max}}$ 排序下$\mathbf{x}$ 排在$\mathbf{y}$ 之前的概率
结果：$= 0$
**对**列 $j$ 中 $[m]$ **执行**
  **对于**权重 $v_j \in$supp$(\Pi_j)$ **执行**
    //查找 $v_j \cdot \mathbf{y}[j]$ 为获胜者的情形
    $U := \{\emptyset, \ldots, \emptyset\}$ //每列包含一个满足条件的权重列表的数据结构
    **对于**列 $k$ 在 $[m]$ **执行**
      //为列 $k$ 构建权重列表
      **若** $k = j$ **且** $v_j \cdot \mathbf{y}[j] > v_j \cdot \mathbf{x}[j]$ **则** $U[j] := \{v_j\}$
      **如果** $k \neq j$ **则**
        $U_1 := \{v_k \in \text{supp}(\Pi_k) \mid v_j \cdot \mathbf{y}[j] > v_k \cdot \mathbf{x}[k]\}$ //严格优于 $\mathbf{x}$ 在列 $k$
        **如果** $k < j$ **则**
          $U_2 := \{v_k \in \text{supp}(\Pi_k) \mid v_j \cdot \mathbf{y}[j] > v_k \cdot \mathbf{y}[k]\}$ //严格优于 $\mathbf{y}$ 在列 $k$
        **else**
          $U_2 := \{v_k \in \text{supp}(\Pi_k) \mid v_j \cdot \mathbf{y}[j] \geq v_k \cdot \mathbf{y}[k]\}$ //优于 $y$ 在列 $k$
        $U[j] := U_1 \cap U_2$
    结果 $+= \prod_{k \in [m]} \sum_{v_k \in U[k]} \mathbb{P}[u_k = v_k]$
**返回**结果

。我们遍历 $u_1$的所有可能取值，并针对每个取值分别确定 $u_2$和 $u_3$的有效权重。如果 $u_1 = 0$，那么 $u_1 \cdot 3$不可能是获胜者。如果 $u_1 = 1$，则有效权重集合为：$u_1 \in \{1\}$、$u_2 \in \{0\}$、$u_3 \in \{0\}$。其发生的概率为 $1/8$。（情况 2）第二个$\mathbf{y}$坐标 $u_2 \cdot 1$ 是整体最大值。这种情况不可能发生。（情况 3）第三个$\mathbf{y}$坐标 $u_3 \cdot 6$ 是整体最大值。此时，我们得到集合 $u_1 \in \{0,1\}$、$u_2 \in \{0,1\}$、$u_3 \in \{1\}$。这些权重的所有组合都有效，因此概率为$4/8$。将有效情况的概率相加，得到最终结果 $5/8$。◀

算法 1基于我们的示例中的思路，将两个最大值谓词的比较分解为 $m$ 个互不相同的情况，并高效地计算其概率。对于 $\mathbf{y}$ 的各列之间出现并列的情况，需要特别处理；在此类情况下，我们将最先出现的最大值选为获胜者，以确保 $m$ 个情况彼此不同。该分解可表示如下：

$$(\max(\mathbf{y} \circ \mathbf{u}) > \max(\mathbf{x} \circ \mathbf{u})) \equiv \bigvee_{j \in [m]} \bigvee_{v_j \in \text{supp}(\Pi_j)} W_{j,v_j} \tag{4}$$

$$W_{j,v_j} = (v_j \cdot \mathbf{y}[j] > v_j \cdot \mathbf{x}[j]) \wedge \Big( \bigwedge_{k \notin [j]} v_j \cdot \mathbf{y}[j] > u_k \cdot \mathbf{x}[k] \Big) \wedge \Big( \bigwedge_{k \in [j-1]} v_j \cdot \mathbf{y}[j] > u_k \cdot \mathbf{y}[k] \Big) \wedge \Big( \bigwedge_{k \in [j+1,m]} v_j \cdot \mathbf{y}[j] \geq u_k \cdot \mathbf{y}[k] \Big) \tag{5}$$

该算法构造这些互不相同的事件$W_{j,\ v_j}$ 并针对每个事件，确定 $\mathbf{u}$ 的有效权重集合，但位置 $j$ 除外，因为该位置的权重已固定为 $v_j$。最后，由于分布 $\Pi_j$ 相互独立，可以高效地计算这些事件的概率。该算法的运行时间为 $\mathcal{O}(|\Pi|^2)$。

**字典序排序**。现在我们继续讨论按字典序排序。我们的算法同样将问题分解为互不重叠的事件，每个事件都对应一个特定的

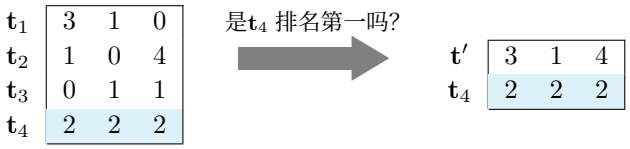



**图1** 例6：将 $r^{\mathsf{dsc}}_{\mathsf{Max}}$ 转换为成对情形以实现前$k$ 成员资格。

列，其权重决定排序结果。要实现这一点，两个元组在所有前置列上的值都必须相同。分解过程如下：

$$((\mathbf{y} \circ \mathbf{u}) >_{\mathsf{Lex}} (\mathbf{x} \circ \mathbf{u}) \equiv \bigvee_{j \in [m]} \bigvee_{v_j \in \mathrm{supp}(\Pi_j)} W'_{j,v_j} \tag{6}$$

$$W'_{j,v_j} = (v_j \cdot \mathbf{y}[j] > v_j \cdot \mathbf{x}[j]) \wedge (\bigwedge_{k \in [j-1]} v_j \cdot \mathbf{y}[j] = u_k \cdot \mathbf{x}[k]) \tag{7}$$

与最大值情形类似，我们构造这些事件，确定每个事件对应的有效权重赋值$\mathbf{u}$，并计算其概率。运行时间为$\mathcal{O}(|\Pi|^2)$。

**恢复满足条件的权重。** 在某些情况下，我们不仅需要知道元组 $\mathbf{x}$ 排在元组 $\mathbf{y}$之前的概率，还需要简洁地表示满足这一条件的权重。幸运的是，我们针对最大值 和 字典序 的基于分解的算法可以直接提供这种表示。我们只需将第 1 行 的求和替换为并集，并将乘法替换为集合乘积，即可应用于算法 1。

▶**观察 4。** $r^{\mathsf{asc}}_{\mathsf{Max}}$、$r^{\mathsf{dsc}}_{\mathsf{Max}}$ 和 $r_{\mathsf{Lex}}$ 的PREC、PREC及PREC算法还可以返回使优先关系成立的权重$\mathbf{u}$，并将其表示为不相交列表，其中每个元素都由每列的有效权重集合组成，即 $\bigtimes_{j \in m} \bigcup (u_j = v_j)$。

## 4.2 从成对影响到排名影响

在为成对情形建立了高效算法之后，我们现在将其扩展为计算对整体排名的期望效应。当效应函数衡量某个元组的位置或整个排列的肯德尔 tau 距离时，我们可以将该问题归约为对 PREC$\langle r \rangle$进行多项式次数调用。对于这两种度量，我们都可以将期望效应表示为元组优先关系的指示变量之和的期望，然后利用期望的线性性。

▶ **定理 5。** 对于排序函数 $r \in \{r^{\mathsf{asc}}_{\mathsf{Max}}, r^{\mathsf{dsc}}_{\mathsf{Max}}, r_{\mathsf{Lex}}\}$ 和效应函数 $e \in \{e_{\mathsf{KT}}, e_{\mathsf{pos}}\}$，问题 EXP$\langle r, e \rangle$ 可在多项式时间内求解。对于排序函数 $r = r_{\mathsf{Sum}}$ 和效应函数 $e \in \{e_{\mathsf{KT}}, e_{\mathsf{pos}}\}$，如果矩阵 $\mathbf{M}$以及 $\Pi$ 中的权重都是以一元编码表示的整数，则问题 EXP$\langle r, e \rangle$ 可在多项式时间内求解。

对于固定的 $k$ 并按 $r^{\mathsf{dsc}}_{\mathsf{Max}}$进行排序，最后一种可以采用多项式时间算法的情形是期望的前$k$ 成员资格。有趣的是，我们的方法仅适用于降序；事实上，我们将在下一节证明，当顺序为升序时，该问题是困难的。该算法的核心思想是：对于这一特定的排序函数，我们可以计算出所关注的元组胜过其他少量元组的子集的概率。

▶ 例6。假设矩阵 M 包含元组 $t_1$,$t_2$,$t_3$,$t_4$，如图1所示，排序函数为 $r^{\text{dsc}}_{\text{Max}}$，我们的目标是计算 $t_4$ 排在前1的概率。我们可以将竞争元组 $t_1, t_2$,$t_3$ 合并为一个竞争元组 t′，并保留每列中的最大值。此时，$t_4$ 排在前1，当且仅当它在排序中排在 t′ 之前。这种合并实际上将问题简化为成对问题 PREC$r^{\text{dsc}}_{\text{Max}}$，而我们已经证明该问题可以高效求解。◀

该示例说明了当 $k = 1$ 时算法如何运行。对于 $k > 1$，我们应用容斥原理，将我们的元组位于前 $k$ 名这一事件表示为若干事件的交集，其中每个事件表示该元组的排名高于其他元组的特定子集。

▶ **定理7.** 当 $k$ 是固定参数时，问题 EXP$\langle r^{\text{dsc}}_{\text{Max}}, e^{k}_{\text{top}}\rangle$ 可在多项式时间内求解。

稍后我们将证明，固定 $k$ 这一假设是必要的，因为否则该问题将变得难以处理（定理 12 ）。

### 4.3 限定矩阵维度

本节最后，我们将探讨将矩阵的两个维度之一限定为固定常数所带来的影响。这有助于我们更清楚地了解使问题变难的参数。我们发现，这种限制会显著降低问题的难度；在大多数情况下（任意取值的求和问题除外），只有两个维度都不是固定的，问题才会变难。

▶**定理 8.** 以下结论成立。

*1.* 如果矩阵 $M$ 的列数 $m$ 由一个常数界定，那么对于任何可在多项式时间内计算的排序函数 $r$ 和效应函数 $e$，问题 $EXP\langle r, e\rangle$ 都可以在多项式时间内求解。*2.* 如果矩阵 $M$ 的行数 $n$ 由一个常数界定，那么对于任何可在多项式时间内计算的效应函数 $e$，且满足以下条件之一：*a.* 排序函数 $r$ 属于 $\{r^{\text{asc}}_{\text{Max}}, r^{\text{dsc}}_{\text{Max}}, r_{\text{Lex}}\}$；或 *b.* $r$ 是排序函数 $r_{\text{Sum}}$，并且矩阵 $M$ 以及 Π 中的权重均为一元编码的整数，则问题 $EXP\langle r, e\rangle$ 可以在多项式时间内求解。

请注意，当 $m$ （列数）固定时，证明是显而易见的，因为此时我们可以用显式表示构造出整个概率空间。对于固定行数 $n$ 时的可处理性证明则更为复杂。

## 5 难处理情形下的期望（及 SHAP 分数）

本节将展示，第4节中的可处理性结果是完备的，即对于FP$^{\#\text{P}}$-难的所有其余 $r$ 和 $e$组合，期望计算（及 SHAP 分数计算）都是。

### 5.1 基于求和的排序

我们从这样一个观察出发：对于通过求和进行排序而言，优先概率是困难的。

## 14 参数在排序函数中的重要性

▶**定理 9.** 问题 PREC$\langle r_{\mathsf{Sum}}\rangle$ 是 FP$^{\#\mathsf{P}}$-困难的，前提是输入矩阵 $\mathbf{M}$ 以二进制编码。

该定理的证明见附录 B。我们通过将计数背包问题归约到该问题，利用以下引理证明这一结论。计数背包问题给定自然数$b_1, \ldots, b_\ell$ 以及一个数 $d$（均以二进制编码），要求计算满足 $S \subseteq [\ell]$ 的$\sum_{i\in S} b_i \leq d$子集数量。该问题已知是#P-难的（参见例如 [13, 18]）。

▶ **引理10**。 设 $b_1, \ldots, b_\ell$和 $d$是计数背包问题的一个输入。那么，存在一个矩阵 $\mathbf{M} \in M^{2\times(\ell+1)}$，满足以下性质：对于每个列集合 $C$，当且仅当 $C$ 包含 $\ell+1$ 且 $\sum_{i\in C\setminus\{\ell+1\}} b_i \leq d$时，有 $r_{\mathsf{Sum}}(\mathbf{M}|_C)(2) \leq r_{\mathsf{Sum}}(\mathbf{M}|_C)(1)$。

**证明**。 考虑以下矩阵 $\mathbf{M} \in \mathcal{M}^{2\times(\ell+1)}$:

| id | $a_1$ | $\ldots$ | $a_n$ | $a_{n+1}$ |
|---|---|---|---|---|
| 1 | 0 | $\ldots$ | 0 | $d+1$ |
| 2 | $b_1$ | $\ldots$ | $b_n$ | 0 |

不难验证，这具有所声称的性质。 ◀

我们可以利用这个定理证明，只要按求和进行排序，期望计算（因此也包括 SHAP 分数计算）就是困难的。

▶ **推论 11** (由 定理 9得出。 对于每个效应函数 $e \in \{e_{\mathsf{k}\tau}, e_{\mathsf{Ham}}, e_{\mathsf{pos}}, e^k_{\mathsf{top}}\}$，问题 EXP$\langle r_{\mathsf{Sum}}, e\rangle$ 在 FP$^{\#\mathsf{P}}$-困难，当输入矩阵 $\mathbf{M}$ 以二进制编码时。

**证明**。 我们注意到，对于 $n = 2$，效应函数的期望计算 $e \in \{e_{\mathsf{k}\tau}, e_{\mathsf{Ham}}, e_{\mathsf{pos}}\}$ 等价于优先级概率计算，因为它们都具有以下形式 $a\cdot \mathbb{P}(\pi(1) > \pi(2)) + b\cdot\mathbb{P}(\pi(2) > \pi(1))$ 其中 $a \neq b$。

对于 $\mathbf{M} \in \mathcal{M}^{2\times m}$, $k \geq 1$ 和 $e = e^k_{\mathsf{top}}$，我们考虑一个矩阵 $\mathbf{M}' \in \mathcal{M}^{(k+1)\times m}$，其中$\mathbf{t'_1} = \mathbf{t}_1$和$\mathbf{t'_2} = \ldots = \mathbf{t'_{k+1}} = \mathbf{t}_2$,，因此所有元组$\mathbf{t'_2}$直到$\mathbf{t'_{k+1}}$都相同。现在，$\mathbf{t'_1}$位于$\mathbf{M}'$的前-$k$位，等价于$\mathbf{t}_1$在$\mathbf{M}$中排名高于$\mathbf{t}_2$。由此得证。◀

## 5.2 top-k 成员资格

在第4节中，我们看到了一个用于EXP$\langle r^{\mathsf{dsc}}_{\mathsf{Max}}, e^k_{\mathsf{top}}\rangle$ 的多项式时间算法，其中 $k$ 是一个参数。在本小节中，我们将展示，对于我们所考虑的所有其他排序函数而言，这个问题都是困难的。

▶ **定理 12**。 对于每个排序函数 $r \in \{r^{\mathsf{asc}}_{\mathsf{Max}}, r_{\mathsf{Sum}}, r_{\mathsf{Lex}}\}$，问题 EXP$\langle r, e^k_{\mathsf{top}}\rangle$ 是 FP$^{\#\mathsf{P}}$-困难的；当 $k$ 是一个参数时，以及当 $k$ 属于输入时，结论均成立。此外，如果 $k$ 属于输入，则 EXP$\langle r^{\mathsf{dsc}}_{\mathsf{Max}}$, $e^k_{\mathsf{top}}\rangle$ 是 FP$^{\#\mathsf{P}}$-困难的。如果将 $\mathbf{M}$ 的条目限制为 $\{0, 1\}$，并且分布 $\Pi$ 是定义在 $\{0, 1\}^m$上的均匀分布，这些结论仍然成立。

我们在附录 B 中通过将计数正合取范式公式的满足赋值归约到一个问题，来证明这一定理。这个问题已知是 #P-困难 [36]。继续之前，我们先为本节余下部分固定一些记号。令 $X_1$、.....、 $X_m$ 为布尔变量。向量 $\mathbf{u} \in \{0, 1\}^m$ 定义一个赋值 $\alpha[\mathbf{u}]$: $[m] \to \{\text{true, false}\}$, 其中将 1 解释为 true，将 0 解释为 false。集合 $C \subseteq [m]$ 定义赋值 $\alpha[C] = \alpha[\mathbb{1}_C]$，其中 $\mathbb{1}_C \in \{0, 1\}^m$ 是 $C$ 的关联向量。

| u | 1 | 0 | 0 | 1 |
|---|---|---|---|---|
| $\mathbf{t}_4$ | 0 | 0 | 0 | 0 |
| $\mathbf{t}_1$ | 1 | 1 | 0 | 1 |
| $\mathbf{t}_2$ | 1 | 0 | 1 | 0 |
| $\mathbf{t}_3$ | 0 | 1 | 1 | 1 |

| u | 0 | 0 | 0 | 1 |
|---|---|---|---|---|
| $\mathbf{t}_2$ | 1 | 0 | 1 | 0 |
| $\mathbf{t}_4$ | 0 | 0 | 0 | 0 |
| $\mathbf{t}_1$ | 1 | 1 | 0 | 1 |
| $\mathbf{t}_3$ | 0 | 1 | 1 | 1 |

**图2** 例6：元组 $\mathbf{t}_4$ 在 $r^{\text{asc}}_{\text{Max}}$ 下是 top-1，当且仅当 $\mathbf{u}$ 为合取范式公式 $(X_1 \vee X_2 \vee X_4) \wedge (X_1 \vee X_3) \wedge (X_2 \vee X_3 \vee X_4)$ 的满足赋值。左边是满足赋值，右边则不是。

▶ **示例13**。 我们将说明如何使用 top-1 成员资格问题来建模 CNF 公式的可满足性。设我们的公式为 $(X_1 \vee X_2 \vee X_4) \wedge (X_1 \vee X_3) \wedge (X_2 \vee X_3 \vee X_4)$。我们构造图中所示的矩阵 图2，其列对应公式中的变量，元组 $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ 对应子句。值为 1 表示相应变量出现在该子句中。现在，设列权重为 $\{0,1\}$，分别表示变量为 真或 假，并设排序函数为 $r^{\text{asc}}_{\text{Max}}$。根据我们的构造， $\mathbf{t}_4$ 是 top-1，当且仅当所有子句都得到满足。◀

现在，我们可以陈述证明 定理 12 的主要工具了。

▶ **引理14.** 设 $\phi = \bigwedge_{i=1}^{\ell} D_i$ 是一个正合取范式公式，其变量集为 $X_1$、.....、$X_m$，子句为 $D_i = \bigvee_{j=1}^{r_i} X_{j_i}$。此外，设 $k \in \mathbb{N}$。那么，存在一个矩阵 $\mathbf{M} \in M^{(\ell+k)\times m}$，其元素取自 $\{0,1\}$，并具有以下性质：对于每个列集合 $C$ 以及每个排序函数 $r \in \{r^{\text{asc}}_{\text{Max}}, r_{\text{Sum}}, r_{\text{Lex}}\}$，当且仅当赋值$\alpha[C]$满足 $\phi$时，有 $r(\mathbf{M}|_C)(k+\ell) \leq k$。

▶ 注 15。 直接运用 [3, 推论 10], 前一个引理即可证明，除非 $r \in \{r^{\text{asc}}_{\text{Max}}, r_{\text{Sum}}, r_{\text{Lex}}\}$ 且 $e = e^k_{\text{top}}$ 满足 RP $=$ NP，否则我们无法近似计算 SHAP 分数。

## 5.3 最大位移距离

本节将证明，若使用最大位移距离 $d_{\text{md}}(\pi_1, \pi_2) =$ 最大值$_{1\leq i\leq n}|\pi_1(i) - \pi_2(i)|$来衡量效应，那么期望计算是一个困难问题。

▶ **定理 16**。 对于 $e = e_{\text{md}}$ 和 $r \in \{r^{\text{asc}}_{\text{Max}}, r^{\text{dsc}}_{\text{Max}}, r_{\text{Sum}}, r_{\text{Lex}}\}$，问题 EXP$\langle r, e\rangle$ 是 FP$^{\#\text{P}}$-困难的。即使将 $\mathbf{M}$ 的元素限制为 $\{0,1\}$，且分布 $\Pi$ 是 $\{0,1\}^m$上的均匀分布，这一结论仍然成立。

我们在 附录 B 中通过从计数正合取范式公式的满足赋值进行归约来证明该定理。证明与 定理 12的证明类似，但略有不同。由于 $d_{\text{md}}$ 是一个全局度量，我们无法定义一个矩阵 $\mathbf{M}$ 来区分满足赋值和不满足赋值。相反，我们将定义两个矩阵$\mathbf{M}_1$和 $\mathbf{M}_2$，使效应函数之差能够实现这种区分。

▶ **引理17**。 设 $\phi = \bigwedge_{i=1}^{\ell} D_i$ 是一个变量集为 $X_1, \ldots, X_m$、包含 $D_i = \bigvee_{j=1}^{r_i} X_{j_i}$个子句的正合取范式公式。那么，存在两个矩阵 $\mathbf{M}_1 \in M^{(2\ell+1)\times m}$ 和 $\mathbf{M}_2 \in M^{(2\ell+2)\times m}$，其元素属于 $\{0,1\}$，并具有以下性质：对于每个由 $C$ 列组成的集合，以及每个排序函数 $r \in \{r^{\text{asc}}_{\text{Max}}, r_{\text{Sum}}, r_{\text{Lex}}\}$，我们有

$$d_{\text{md}}\big(r(\mathbf{M}_2), r(\mathbf{M}_2|_C)\big) - d_{\text{md}}\big(r(\mathbf{M}_1), r(\mathbf{M}_1|_C)\big) = \begin{cases} 0, & \text{if } \alpha[C] \models \phi \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

**引理17的证明概要** **引理17**。 矩阵的构造与例6和引理14类似，只是增加了更多全零元组的副本；当公式成立时，这些副本占据顶部位置，从而产生零位移。矩阵 $\mathbf{M}_2$ 比$\mathbf{M}_1$, 多包含一个全零元组的副本，因此两个矩阵的表现相似，但当公式不成立时，$\mathbf{M}_2$ 中的最大位移恰好比$\mathbf{M}_1$ 中的最大位移高1。 ◀

## 6 Shapley 值的精确计算

本节中，我们将关注点转向通过 Shapley 值衡量列的重要性，而非通过 SHAP 分数衡量评分参数的重要性。我们首先将说明，第4节中用于计算 SHAP 分数的所有算法都可以用来确定 Shapley 值，随后通过扩展第5节中的结果来补全整个图景。

▶**定理18**。对于我们在 $r$ 中介绍的排序函数 $e$ 和效应函数第3节，Shapley$\langle r,\ e\rangle$ 的计算复杂性与SHAP$\langle r,\ e\rangle$ 在表1中给出的计算复杂性相同。

### 6.1 算法

Gilad等人 [17]给出了一种基于插值的简单算法；在能够访问一个针对 SHAP 分数计算特殊实例（称为 binarySHAP）的预言机时，该算法可在多项式时间内计算 Shapley 值。将他们的结果转化到我们的设定中，可得以下结论：

▶ 定理*19*（ [17, 定理*6*.15]）。设 $r$ 为一个排序函数， $e$ 为一个效应函数，并且满足：对于每个 $C$ 列集合，都有 $e(r(M|_C)) = e(r(M \circ \mathbb{1}_C))$。那么，我们可以通过 $m$ 次调用*SHAP*$\langle r, e\rangle$的预言机，在多项式时间内求解*Shapley*$\langle r, e\rangle$。

尽管对于按求和和字典序进行排序，该定理的条件总是成立，但当$\mathbf{M}$ 包含负数时，对于最大值排序，该条件不一定成立。幸运的是，我们可以将$\mathbf{M}$ 中的最小值加到每个条目上，使$\mathbf{M}$ 变为非负，同时不改变每个列集合的排序结果。因此，我们可以使用前一个定理来证明，我们的所有算法也适用于Shapley 值计算。

### 6.2 难处理情形

为了从第5节中得到困难性结果，我们将计数背包问题（引理10）以及计数正合取范式公式的满足赋值问题（引理14 和17）归约为SHAP 分数计算，这基于其与期望计算的等价性（定理1）。由于这种等价性通常不适用于Shapley 值计算，我们需要采用另一种方法来扩展我们的结果。具体而言，我们将定义两个辅助博弈，并分别证明计算它们的Shapley 值是困难的。证明见附录 C。它们采用了大多数Shapley 值计算困难性证明所使用的技术 [20, 24]：对于一个给定的博弈实例，定义一组相关实例，并通过插值利用它们的Shapley 值分别计算有效背包集合的数量或满足赋值的数量。

第一个博弈是背包博弈：对于以二进制给出的自然数 $b_1, \ldots, b_\ell$和 $b_{\ell+1} = d$，考虑玩家集合为 $[\ell + 1]$ 且估值函数为

$$\nu_{\mathsf{knap}}(C) := \begin{cases} 1 & \text{if } \ell+1 \in C \text{ and } \sum_{i \in C \setminus \{\ell\}} b_i \leq d; \\ 0 & \text{otherwise.} \end{cases}$$

▶**引理20.** 计算背包博弈中各玩家的 *Shapley* 值是$\mathsf{FP}^{\#\mathsf{P}}$-困难的。

第二个博弈是正 *CNF* 博弈：给定一个正合取范式公式 $\phi$，其变量集为 $X_1$、……、$X_m$，考虑玩家集合为 $[m]$ 且估值函数为

$$\nu_{\mathsf{CNF}}(C) \coloneqq \begin{cases} 1 & \text{if } \alpha[C] \models \phi; \\ 0 & \text{otherwise.} \end{cases}$$

▶**引理 21**。 正 *CNF* 博弈中参与者的 *Shapley*值计算是$\mathsf{FP}^{\#\mathsf{P}}$-难的。

利用这些引理以及第5节中的构造，我们可以轻松证明定理18。详细内容见附录 C。

## 7 结论

我们研究了衡量排序函数中参数选择重要性的计算复杂性。具体而言，我们考察了在一组特定（但基础的）排序函数和效应函数下参数的 SHAP 分数。为此，我们研究了计算随机参数值下期望效应的复杂性，重点关注有限分布和概率独立分布。我们还研究了计算列的 Shapley 值这一相关问题，以衡量列对排序的贡献。

我们将这项工作视为对排序的基于函数的解释之计算复杂性进行严格分析的第一步。因此，仍有许多方向有待未来探索。最重要的是，我们希望将研究结果推广到更广泛的类别的排序函数和效应函数，使我们能够为其设计高效算法，而不是分别考虑具体情形。具体而言，我们希望能够分析热门排序方案中不同选择所带来的影响；例如，WBSC 等体育排名[4] 和 FIFA/可口可乐世界排名，[5] 其结构类似于工作流图，内部包含特征的线性组合；以及计算机科学排名[6] ，它基于预先确定的发表场所选择。我们还希望将分析扩展到其他类别的概率分布，包括区间上的均匀分布，并刻画哪些情况下可以高效地对 SHAP 分数进行乘法近似。此外，我们希望理解一般类型效应函数的复杂性，这些函数还包括本文未研究的其他方案，例如 Spearman 足尺距离和 Cayley 距离。

最后，未来研究的一个重要方向是考察该框架在现实场景中的实用性，尤其要了解 SHAP 和 Shapley 的归因函数在多大程度上符合人们对排序函数各组成部分贡献的直觉。

### 参考资料

1 Hadis Anahideh 和 Nasrin Mohabbati-Kalejahi。全局排序的局部解释：对竞争性排序结果的洞见。 *IEEE Access*，10:30676–30693，2022年。 doi:10.1109/ACCESS.2022.3159245。

[4] https://www.wbsc.org/en/rankings
[5] https://inside.fifa.com/fifa-world-ranking
[6] https://csrankings.org/

2 Avishek Anand、Lijun Lyu、Maximilian Idahl、Yumeng Wang、Jonas Wallat 和 Zijian Zhang。《可解释信息检索：综述》，2022年。 arXiv:2211.02405。3 Marcelo Arenas、Pablo Barcelo、Leopoldo Bertossi 和 Mikael Monet。《基于 SHAP 分数的解释的复杂性：通过知识编译实现可处理性及不可近似性结果》。《机器学习研究杂志》，24(63):1–58，2023年。URL: http://jmlr.org/papers/v24/21-0389.html。4 Abolfazl Asudeh、H. V. Jagadish、Gerome Miklau 和 Julia Stoyanovich。《关于获得稳定排序》。《PVLDB》，12(3):237–250，2018年。 doi:10.14778/3291264.3291269。5 Roland Bacher。《与帕斯卡三角形相关的矩阵的行列式》。《J. Theorie Nombres Bordeaux》，14(1):9–41，2002年。URL: https://www.jstor.org/stable/43972640。6 Alexander Barg 和 Arya Mazumdar。《置换码与秩调制纠错》。《IEEE Trans. Inf. Theory》，56(7):3158–3165，2010年。 doi:10.1109/TIT.2010.20484。7 Leopoldo E. Bertossi、Benny Kimelfeld、Ester Livshits 和 Mikaël Monet。《数据库管理中的 Shapley 值》。《SIGMOD Record》，52(2):6–17，2023年。 doi:10.1145/3615952.361595。8 Javier Castro、Daniel Gómez、Elisenda Molina 和 Juan Tejada。《通过采用最优分配的分层随机抽样改进 Shapley 值的多项式估计》。《Computers & Operations Research》，82:180–188，2017年。 doi:10.1016/j.cor.2017.01.019。9 Zixuan Chen、Panagiotis Manolios 和 Mirek Riedewald。《为何尚未实现：修正对个人不公平的前 k 名排序》。《PVLDB》，16(9):2377–2390，2023年。 doi:10.14778/3598581.3598606。10 Tanya Chowdhury、Yair Zick 和 James Allan。《RankSHAP：用于学习排序的基于 Shapley 值的特征归因》。《ICLR》，2025年。URL: https://openreview.net/forum?id=4011PUI9vm。11 Daniel Deutch、Nave Frost、Benny Kimelfeld 和 Mikaël Monet。《计算查询回答中事实的 Shapley 值》。《SIGMOD》，第 1570–1583 页。ACM，2022年。 doi:10.1145/3514221.3517912。12 Alexandre Duval 和 Fragkiskos D. Malliaros。《GraphSVX：图神经网络的 Shapley 值解释》。《ECML/PKDD (2)》，LNCS 第 12976 卷，第 302–318 页。Springer，2021年。 doi:10.1007/978-3-030-86520-7_19。13 Martin E. Dyer、Alan M. Frieze、Ravi Kannan、Ajai Kapoor、Ljubomir Perkovic 和 Umesh V. Vazirani。《一种用于近似多维背包问题解的数量的亚指数时间算法》。《Comb. Probab. Comput.》，2:271–284，1993年。 doi:10.1017/S0963548300000675。14 Ronald Fagin、Benny Kimelfeld 和 Phokion G. Kolaitis。《概率数据交换》。《J. ACM》，58(4):15:1–15:55，2011年。 doi:10.1145/1989727.1989729。15 William Feller。《概率论及其应用导论》，第 1 卷。Wiley，第 3 版，1968年。URL: https://archive.org/details/dli.ernet.5666/。16 Abraham Gale 和 Amélie Marian。《解释单调排序函数》。《PVLDB》，14(4):640–652，2020年。 doi:10.14778/3436905.3436922。17 Amir Gilad、Martin Grohe、Benny Kimelfeld、Peter Lindner 和 Christoph Standke。《数据库查询中参数的重要性》，2024年。URL: https://arxiv.org/abs/2401.04606, arXiv:2401.04606。18 Parikshit Gopalan、Adam Klivans、Raghu Meka、Daniel Štefankovic、Santosh Vempala 和 Eric Vigoda。《背包计数问题及相关计数问题的完全多项式时间近似方案》。《FOCS》，第 817–826 页，2011年。 doi:10.1109/FOCS.2011.32。19 Erich Grädel、Yuri Gurevich 和 Colin Hirsch。《查询可靠性的复杂性》。《PODS》，第 227–234 页。ACM Press，1998年。 doi:10.1145/275487.29512。20 Martin Grohe、Benny Kimelfeld、Peter Lindner 和 Christoph Standke。《数据库查询中参数的重要性》。《ICDT》，LIPIcs 第 290 卷，第 14:1–14:17 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2024年。 doi:10.4230/LIPIcs.ICDT.2024.14。21 Maria Heuss、Maarten de Rijke 和 Avishek Anand。《RankingSHAP——用于排序模型的列表级特征归因解释》，2024年。 arXiv:2403.16085。

22 Uday Kamath 和 John Liu。《可解释人工智能：可解释机器学习导论》。Springer，2021年。 doi:10.1007/978-3-030-83356-5。23 Ahmet Kara、Dan Olteanu 和 Dan Suciu。从 Shapley 值到模型计数，再回到 Shapley 值。PACMMOD，2(2):79，2024年。 doi:10.1145/3651142。24 Ester Livshits、Leopoldo E. Bertossi、Benny Kimelfeld 和 Moshe Sebag。查询回答中元组的 Shapley 值。Log. Methods Comput. Sci.，17(3)，2021年。URL： https://doi.org/10.46298/lmcs-17(3:22)2021， doi:10.46298/LMCS-17(3:22)2021。25 Scott M. Lundberg 和 Su-In Lee。解释模型预测的统一方法。载于 NIPS，第4765–4774页，2017年。URL： https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html。26 Lijun Lyu 和 Avishek Anand。使用多个解释器为排序模型生成列表式解释。载于 ECIR，LNCS 第13980卷，第653–668页。Springer，2023年。 doi:10.1007/978-3-031-28244-7_41。27 Gustavo Penha、Eyal Krikon 和 Vanessa Murdock。面向语音商品搜索的基于成对评论的解释。载于 CHIIR，第300–304页。ACM，2022年。 doi:10.1145/3498366.350582。28 Venetia Pliatsika、Joao Fonseca、Kateryna Akhynko、Ivan Shevchenko 和 Julia Stoyanovich。ShaRP：一种用于排序的新型特征重要性框架，2024年。 arXiv:2401.16744。29 Ariel D. Procaccia、Nisarg Shah 和 Yair Zick。作为纠错码的投票规则。《人工智能》，231：1–16，2016年。 doi:10.1016/j.artint.2015.10.003。30 Alvin E. Roth，编。《Shapley 值：献给 Lloyd S. Shapley 的论文集》。剑桥大学出版社，1988年。URL： http://www.loc.gov/catdir/samples/cam031/88002983.html。31 Benedek Rozemberczki、Lauren Watson、Péter Bayer、Hao-Tsung Yang、Olivér Kiss、Sebastian Nilsson 和 Rik Sarkar。机器学习中的 Shapley 值。载于 IJCAI，第5572–5579页，2022年。 doi:10.24963/ijcai.2022/778。32 Sourav Saha、Debapriyo Majumdar 和 Mandar Mitra。文本处理和检索方法的可解释性：批判性综述，2022年。 arXiv:2212.07126。33 Lloyd S Shapley。n人博弈的一个值。载于 Harold W. Kuhn 和 Albert W. Tucker 编，《博弈论贡献》第二卷，第307–317页。普林斯顿大学出版社，普林斯顿，1953年。 doi:10.1515/9781400829156-012。34 Jaspreet Singh 和 Avishek Anand。通过意图建模实现排序器的模型无关可解释性。载于 FAT*，第618–628页。ACM，2020年。 doi:10.1145/3351095.33752。35 Seinosuke Toda 和 Mitsunori Ogiwara。计数复杂性类至少与多项式时间层次一样难。SIAM J. Comput.，21(2)：316–328，1992年。 doi:10.1137/0221023。36 Leslie G Valiant。枚举与可靠性问题的复杂性。SIAM计算杂志，8(3)：410–421，1979年。 doi:10.1137/0208032。37 Guy Van den Broeck、Anton Lykov、Maximilian Schleich 和 Dan Suciu。关于 SHAP解释的可处理性。人工智能研究杂志，74：851–886，2022年6月。doi:10.1613/jair.1.13283。38 Ke Yang、Julia Stoyanovich、Abolfazl Asudeh、Bill Howe、HV Jagadish 和 Gerome Miklau。排序结果的营养标签。载于 SIGMOD，第1773–1776页，2018年。 doi:10.1145/3183713.319356。39 Puxuan Yu、Razieh Rahimi 和 James Allan。迈向可解释的搜索结果：列表式解释生成器。载于 SIGIR，第669–680页。ACM，2022年。 doi:10.1145/3477495.3532067。

## 第4节中省略的证明第4节

本节中，我们给出此前在第4节中省略的证明。

▶ **定理 5.** 对于排序函数 $r \in \{r^{\mathsf{asc}}_{\mathsf{Max}}, r^{\mathsf{dsc}}_{\mathsf{Max}}, r_{\mathsf{Lex}}\}$ 和效应函数 $e \in \{e_{\mathsf{k}\tau}, e_{\mathsf{pos}}\}$，问题 $\mathsf{EXP}\langle r, e\rangle$ 可在多项式时间内求解。对于排序函数 $r = r_{\mathsf{Sum}}$ 和效应函数 $e \in \{e_{\mathsf{k}\tau}, e_{\mathsf{pos}}\}$，如果矩阵 $\mathbf{M}$以及 $\Pi$ 中的权重均为一元编码的整数，则问题 $\mathsf{EXP}\langle r, e\rangle$ 可在多项式时间内求解。

**定理 定理 5的证明。** 对于这两种度量，我们将期望效应写成元组优先关系对应的一组指示变量之和的期望，然后利用期望的线性性。

令 $\pi$ 为 $r(\mathbf{M} \circ \mathbf{u})$，则对于行索引为 $i$ 的元组，其期望排名 $e_{\mathsf{pos}}$ 为：

$$\mathbb{E}_{\mathbf{u}\sim\Pi}[e_{\mathsf{pos}}(i,\pi)] = \mathbb{E}_{\mathbf{u}\sim\Pi}\Big[\sum_{j\in[n]} \mathbb{1}_{\pi(i)>\pi(j)}\Big] = \sum_{j\in[n]} \mathbb{E}_{\mathbf{u}\sim\Pi}[\mathbb{1}_{\pi(i)>\pi(j)}] = \sum_{j\in[n]} \mathbb{P}_{\mathbf{u}\sim\Pi}[\pi(i) > \pi(j)].$$

因此，对于所有根据 $r$ 的 $\mathsf{PREC}\langle r\rangle$问题可在多项式时间内求解的排序函数，定理 2，问题 $\mathsf{EXP}\langle r, e_{\mathsf{pos}}\rangle$ 也可通过求解前者 $\mathcal{O}(n)$ 次，在多项式时间内求解。

同样，对于期望的肯德尔 tau 距离 $e_{\mathsf{k}\tau}$，我们有：

$$\begin{aligned}\mathbb{E}_{\mathbf{u}\sim\Pi}[e_{\mathsf{k}\tau}(\pi)] &= \mathbb{E}_{\mathbf{u}\sim\Pi}\Big[\sum_{1\le i<j\le n} \mathbb{1}_{\pi(i)>\pi(j)}\Big] \\ &= \sum_{1\le i<j\le n} \mathbb{E}_{\mathbf{u}\sim\Pi}[\mathbb{1}_{\pi(i)>\pi(j)}] = \sum_{1\le i<j\le n} \mathbb{P}_{\mathbf{u}\sim\Pi}[\pi(i) > \pi(j)].\end{aligned}$$

因此，对于 $\mathsf{EXP}\langle r,\ e_{\mathsf{k}\tau}\rangle$，我们求解 $\mathsf{PREC}\langle r\rangle$ 问题$\mathcal{O}(n^2)$ 次，这仍然是多项式时间。◀

▶ **定理7**。问题$\mathsf{EXP}\langle r^{\mathsf{dsc}}_{\mathsf{Max}}, e^{k}_{\mathsf{top}}\rangle$在将 $k$ 作为固定参数时，可在多项式时间内求解。

**定理7的证明。** 令所关注的元组为$\mathbf{t}_i$，其行索引为 $i$。任务是计算以下事件发生的概率

$T_{n-k}$ ≔ 元组$\mathbf{t}_i$ 在排序中击败（即排在其前面）至少 $n-k$ 个元组。

利用容斥原理 [15]，可将其表示如下：

$$\mathbb{P}[T_{n-k}] = \mathbb{P}[W_{n-k}] - \binom{n-k}{1}\mathbb{P}[W_{n-k+1}] + \binom{n-k}{2}\mathbb{P}[W_{n-k+2}] - \ldots \pm \binom{n-2}{n-k-1}\mathbb{P}[W_{n-1}]$$

与 $\mathbb{P}[W_d] = \sum_{S\subseteq n,\ |S|=d} \mathbb{P}[\bigwedge_{\ell\in S}(\pi(i) < \pi(\ell))]$。

子集的数量 $S$ 为 $\binom{n}{n-k}$，这是多项式级的，因为 $k$ 是常数。因此，只需在多项式时间内计算每个事件 $B_S \coloneqq \bigwedge_{\ell\in S}(\pi(i) < \pi(\ell))$ 的概率；该事件对应于元组 $\mathbf{t}_i$ 击败所有行索引位于 $S$ 中的元组。

首先，假设元组 $\mathbf{t}_i$ 在平局决胜机制中排在 $S$ 的元组之后。根据 $S$ 中的元组，我们构造一个单独的竞争者 $\mathbf{t}'$，其每一列都包含最大值。我们证明，$\mathbf{t}_i$ 在排序中位于 $\mathbf{t}'$ 之前，当且仅当它位于 $\mathbf{t}'$ 之前。事实上：

$$\text{最大值}_{j\in m}(\mathbf{t}_i[j]) > \text{最大值}_{\ell\in S}(\text{最大值}_{j\in m}(\mathbf{t}_\ell[j])) \Leftrightarrow \text{最大值}_{j\in m}(\mathbf{t}_i[j]) > (\text{最大值}_{j\in m}\text{最大值}_{\ell\in S}(\mathbf{t}_\ell[j])) \Leftrightarrow \text{最大值}_{j\in m}(\mathbf{t}_i[j]) > \text{最大值}_{j\in m}(\mathbf{t}'[j])。$$

为处理 $S$ 中一些元组在平局处理时排在$\mathbf{t}_i$ 之上、另一些排在其下的情况，我们将它们分为两组，并创建两个竞争者 $\mathbf{t}'$ 和 $\mathbf{t}''$。我们分别调用

用于 PREC$\langle r^{\mathsf{dsc}}_{\mathsf{Max}}\rangle$ 的算法，分别处理 $(\mathbf{t}_i,\mathbf{t}')$ 和 $(\mathbf{t}_i,\mathbf{t}'')$，并对平局处理采用不同的假设。根据观察 4，我们可以得到每次调用对应的有效权重列表，再将它们求交得到最终列表；相应的概率则可直接计算。

总体而言，我们的算法需要求解 PREC$\langle r^{\mathsf{dsc}}_{\mathsf{Max}}\rangle$ 问题$\mathcal{O}(kn^k)$ 次。 ◀

## A.1 第 4.3 节中省略的证明

▶**定理 8**。 以下结论成立。

1. 如果列数 $m$ 矩阵 $\mathbf{M}$ 受到常数限制，那么问题EXP$\langle r, e\rangle$ 对于任何可在多项式时间内计算的排序函数 $r$ 和效应函数 $e$均可在多项式时间内求解。

2. 如果行数 $n$ 矩阵 $\mathbf{M}$ 受到常数限制，那么问题EXP$\langle r, e\rangle$ 对于任何可在多项式时间内计算且 $e$ 的效应函数

   a. 排序函数 $r$ 属于 $\{r^{\mathsf{asc}}_{\mathsf{Max}}, r^{\mathsf{dsc}}_{\mathsf{Max}}, r_{\mathsf{Lex}}\}$，或者b. $r$ 是排序函数 $r_{\mathsf{Sum}}$，并且矩阵$\mathbf{M}$ 和 $\Pi$ 中的权重均采用一元编码的整数表示。

**证明**。 首先，假设 $m$ 为$\mathcal{O}(1)$。可能的$\mathbf{u}$ 权重对应的概率空间大小为$\mathcal{O}(|\Pi|^m)$，此时该规模为多项式级，因此我们可以枚举所有可能的权重赋值。对于每种赋值，我们按 $r$ 对表进行排序，并计算效应 $e$。随后，通过对这些情况进行加权求和，计算效应的期望。

其次，假设 $n$ 为$\mathcal{O}(1)$。此时，尽管概率空间的规模呈指数级增长，但$\mathbf{M}$ 矩阵的行可能排列数为 $n!$，即$\mathcal{O}(1)$。我们构造所有可能的排列，并针对每种排列在多项式时间内确定其概率，从而计算期望效应。

设 $\pi$ 是其中一种排列。对于排序函数 $r^{\mathsf{asc}}_{\mathsf{Max}}, r^{\mathsf{dsc}}_{\mathsf{Max}}, r_{\mathsf{Lex}}$，我们考察连续元组 $\pi(i) < \pi(i+1)$ 之间的所有比较，其中 $i \in [n-1]$。这些比较足以描述排列 $\pi$。对于每次比较，我们根据定理 2 和观察 4 计算满足条件的权重，然后取交集，得到 $\pi$的概率。

现在，设我们的排序函数为 $r_{\mathsf{Sum}}$，并额外假设 $\Pi$ 中的矩阵值和权重均为以一元编码给出的整数。我们修改定理 2 中的动态程序，使其计算整个排列的概率。假设元组$\mathbf{t}_1, \ldots, \mathbf{t}_n$ 按照排列 $\pi$ 的顺序编了索引。此外，设 $S_1$ 为在平局判定规则中排在$\mathbf{t}_{i+1}$ 之前的元组$\mathbf{t}_i$，$i \in [n]$ 的集合，$S_2$ 为其余元组的集合。

$$R[j, s_1, \ldots, s_{n-1}] = \sum_{v_j \in \mathrm{supp}(\Pi_j)} \mathbb{P}[u_j = v_j] \cdot R[j+1, s_1 - v_j(\mathbf{t}_2[j] - \mathbf{t}_1[j]), \ldots, s_{n-1} - v_j(\mathbf{t}_n[j] - \mathbf{t}_{n-1}[j])] \tag{9}$$

$$R[m+1, s_1, \ldots, s_{n-1}] = \begin{cases} 1, & \text{if } \forall i : \mathbf{t}_i \in S_1 \Rightarrow s_i \geq 0 \text{ and } \forall i : \mathbf{t}_i \in S_2 \Rightarrow s_i > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

最终答案是 $R[1, 0, \ldots, 0]$。与之前的动态程序相比，运行时间增加了$\mathcal{O}(n)$ 倍，这是一个常数。

◀

## B 第5节中省略的证明第5节

本节中，我们给出在第5节中略去的证明。

### B.1 第5.1节中省略的证明第5.1节

▶**定理 9**。问题 PREC$\langle r_{\mathsf{Sum}}\rangle$ 是 FP$^{\#\mathsf{P}}$-难的，若输入矩阵 $\mathbf{M}$ 以二进制编码。

定理 9 的证明。令 $\mathbf{M}$ 为引理10中的矩阵，$\Pi$ 为 $\{0,1\}^{\ell+1}$ 上的均匀分布。根据引理10，权重向量 $\mathbf{u}=(u_1,\ldots,u_{\ell+1})\in\{0,1\}^{\ell+1}$ 满足 $\mathsf{Sum}(\mathbf{t}_2\circ\mathbf{u})\leq\mathsf{Sum}(\mathbf{t}_1\circ\mathbf{u})$，当且仅当 $u_{\ell+1}=1$ 且集合 $S=\{1\leq i\leq\ell\mid u_i=1\}$ 满足$\sum_{i\in S}b_i\leq d$。因此，计数背包问题的答案恰好是 $\mathbf{t}_2$ 和 $\mathbf{t}_1$ 的优先概率乘以 $2^{\ell+1}$。◀

### B.2 第5.2节中省略的证明第5.2节

▶ **定理 12**。对于每个排序函数 $r\in\{r^{\mathsf{asc}}_{\mathsf{Max}},r_{\mathsf{Sum}},r_{\mathsf{Lex}}\}$，问题 EXP$\langle r,e^k_{\mathsf{top}}\rangle$ 是 FP$^{\#\mathsf{P}}$-难的，无论 $k$ 是参数还是属于输入的一部分。此外，如果 $k$ 属于输入的一部分，EXP$\langle r^{\mathsf{dsc}}_{\mathsf{Max}},e^k_{\mathsf{top}}\rangle$ 是 FP$^{\#\mathsf{P}}$-难的。这些结论在将 $\mathbf{M}$ 的元素限制为 $\{0,1\}$，且分布 $\Pi$ 为 $\{0,1\}^m$ 上的均匀分布时仍然成立。

定理 12 的证明。我们首先注意到，关于 $r\in\{r^{\mathsf{asc}}_{\mathsf{Max}},r_{\mathsf{Sum}},r_{\mathsf{Lex}}\}$ 的结论可直接由引理14推出。设 $\mathbf{M}$ 为引理14中的矩阵，且 $\Pi$ 为 $\{0,1\}^m$ 上的均匀分布。由于对于每个 $\mathbf{u}\in\{0,1\}^m$，$\mathbf{t}_{\ell+k}$ 在 $r(\mathbf{M}\circ\mathbf{u})$ 中的秩为 $\leq k$ 当且仅当 $\alpha[\mathbf{u}]\models\phi$，因此 $\phi$ 的满足赋值数量等于 $2^m$ 乘以 $\mathbf{t}_{k+\ell}$ 的秩为 $\leq k$ 的概率。

对于 $r=r^{\mathsf{dsc}}_{\mathsf{Max}}$，我们注意到

$$\mathbb{P}(r^{\mathsf{dsc}}_{\mathsf{Max}}(\mathbf{t})\leq k)=\mathbb{P}(r^{\mathsf{asc}}_{\mathsf{Max}}(\mathbf{t})\geq n-k)=1-\mathbb{P}(r^{\mathsf{asc}}_{\mathsf{Max}}(\mathbf{t})\leq n-k-1),$$

因此，当 $k$ 是输入的一部分时，使用升序还是降序并不重要。◀

▶**引理14**。设 $\phi=\bigwedge_{i=1}^{\ell}D_i$ 是一个变量集为 $X_1$、……、$X_m$ 且具有子句 $D_i=\bigvee_{j=1}^{r_i}X_{j_i}$ 的正合取范式公式。此外，设 $k\in\mathbb{N}$。那么，存在一个矩阵 $\mathbf{M}\in M^{(\ell+k)\times m}$，其元素属于 $\{0,1\}$，并具有以下性质：对于每个 $C$ 列的集合以及每个排序函数 $r\in\{r^{\mathsf{asc}}_{\mathsf{Max}},r_{\mathsf{Sum}},r_{\mathsf{Lex}}\}$，当且仅当赋值$\alpha[C]$ 满足 $\phi$时，有 $r(\mathbf{M}|_C)\ (k+\ell)\ \leq k$。

**证明**。我们如下定义 $\mathbf{M}\in\mathcal{M}^{(\ell+k)\times m}$。前 $k-1$ 个元组的所有条目均等于0。接下来的 $\ell$ 个元组 $\mathbf{t}_k$、……、$\mathbf{t}_{k+\ell-1}$ 是 $D_1$、……、$D_\ell$的关联元组；也就是说，

$$\mathbf{t}_{k-1+i}[j]:=\begin{cases}1,&\text{if } X_j \text{ appears in } D_i\\0,&\text{otherwise.}\end{cases}$$

最后一个元组 $\mathbf{t}_{\ell+k}$ 同样所有条目都等于 0。

令 $C$ 为一个列集，并考虑 $\mathbf{M}|_C$。在 $\mathbf{M}|_C$中，元组分为两类：零元组，即所有条目均为 0 的元组；以及非零元组，即至少包含一个 1 的元组。在三种排序函数中，零元组两两不可比较，因此它们将保持各自的顺序。每个非零元组的秩都高于所有零元组，因为它的分数更高，而且字典序会根据其首个非零条目所在的列将其排得更高。

由于 $\mathbf{t}_{\ell+k}$ 的索引最大，且所有 $k$ 元组 $\mathbf{t}_1, \ldots, \mathbf{t}_{k-1}$ 和 $\mathbf{t}_{k+\ell}$ 相对于每个列集都是零元组，因此，$\mathbf{t}_{\ell+k}$ 的秩为 $\leq k$ 当且仅当元组 $\mathbf{t}_k, \ldots, \mathbf{t}_{k+\ell-1}$ 均不是相对于 $C$ 的零元组。最后，我们观察到，对于 $1 \leq i \leq \ell$，元组 $\mathbf{t}_{k-1+i}$ 是非零元组当且仅当 $\alpha[C] \models D_i$。综合以上可知，$\mathbf{t}_{\ell+k}$ 的秩为 $\leq k$ 当且仅当 $\alpha[C] \models \phi$。◀

## B.3 第5.3节中省略的证明Section5.3

▶ **引理17**。设 $\phi = \bigwedge_{i=1}^{\ell} D_i$ 是一个变量集为 $X_1$、……、$X_m$ 且子句为 $D_i = \bigvee_{j=1}^{r_i} X_{j_i}$ 的正合取范式公式。则存在两个矩阵 $\mathbf{M}_1 \in M^{(2\ell+1)\times m}$ 和 $\mathbf{M}_2 \in M^{(2\ell+2)\times m}$，其元素属于 $\{0, 1\}$，并具有以下性质：对于每个 $C$ 列的集合以及每个排序函数 $r \in \{r^{\text{asc}}_{\text{Max}}, r_{\text{Sum}}, r_{\text{Lex}}\}$，有

$$d_{\text{md}}\big(r(\mathbf{M}_2), r(\mathbf{M}_2|_C)\big) - d_{\text{md}}\big(r(\mathbf{M}_1), r(\mathbf{M}_1|_C)\big) = \begin{cases} 0, & \text{if } \alpha[C] \models \phi \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

证明。对于 $k \in \{1, 2\}$，令 $\mathbf{M}_k \in \mathbf{M}^{(2\ell+k)\times m}$ 为如下矩阵：前 $\ell$ 个元组 $\mathbf{t}1$，……，$\mathbf{t}_\ell$ 是子句 $D_1$，……，$D_\ell$ 的关联元组。为简单起见，我们假设这些元组按照 $r(\mathbf{M}_k)$ 中的顺序排列。对于最大值而言，这始终成立；但对于求和与字典序而言，这意味着 $D_i$ 中的子句 $\phi$ 分别按照其变量数或其关联向量的字典序排列。其余 $\ell + k$ 个元组的所有元素均为零。我们声称，$\mathbf{M}_2$ 和 $\mathbf{M}_1$ 满足公式（8）。

设 $C \subseteq [m]$ 为一个列集。类似于引理14的证明，我们观察到如下事实：如果 $\alpha[C] \models \phi$，那么全零元组 $\mathbf{t}_{\ell+1}$、……、$\mathbf{t}_{2\ell+k}$ 在 $r(\mathbf{M}_k|_C)$ 中的排序最靠前，而子句元组 $\mathbf{t}_1$、……$\mathbf{t}_\ell$ 在排序中出现得更晚。特别地，对于 $\mathbf{M}_k = \mathbf{M}_k|_{[m]}$，这一点成立。如果 $\alpha[C] \not\models \phi$，那么与未满足子句对应的元组排序最靠前，接着是全零元组，最后是与已满足子句对应的元组。

现在我们细化这一分析，以证明公式（8）。首先假设 $\alpha[C] \models \phi$。此时，全零元组在 $r(\mathbf{M}_k)$ 和 $r(\mathbf{M}_k|_C)$ 中的排序都最靠前，因此它们的位移为 0。子句元组的位移不取决于全零元组的数量，所以公式（8）在该情况下成立。现在假设 $\alpha[C] \not\models \phi$，并令 $\{i_1, \ldots, i_s\}$ 为未满足子句的索引。由于我们假设这些索引也表示这些元组在 $r(\mathbf{M}_k)$ 中的相对顺序，因此有 $r(\mathbf{M}_k)(i_j) = \ell + k + i_j$ 以及 $r(\mathbf{M}_k|_C)(i_j) = j$，所以每个 $i_j$ 的位移至少为 $\ell + k$，且 $\mathbf{M}_1$ 和 $\mathbf{M}_2$ 中的位移相差 1。从 $r(\mathbf{M})$ 到 $r(\mathbf{M}_k|_C)(i_j) = j$，每个全零元组的秩增加的数量等于未满足子句的数量，因此其位移至多为 $\ell$，并且在 $\mathbf{M}_1$ 和 $\mathbf{M}_2$ 中相同。与已满足子句对应的元组位于矩阵的最后 $\ell$ 个秩内，因此其位移也是 $\leq \ell$，并且同样不取决于全零元组的数量。因此，最大位移由与某个未满足子句对应的元组取得，并且其在 $\mathbf{M}_1$ 和 $\mathbf{M}_2$ 中的值相差 1。这就证明了该结论。◀

▶ **定理 16.** 对于 $e = e_{\text{md}}$ 和 $r \in \{r^{\text{asc}}_{\text{Max}}, r^{\text{dsc}}_{\text{Max}}, r_{\text{Sum}}, r_{\text{Lex}}\}$，问题 EXP$\langle r, e\rangle$ 是 FP$^{\#\text{P}}$-困难的。即使将 $\mathbf{M}$ 的元素限制为 $\{0,1\}$，且分布 $\Pi$ 为 $\{0,1\}^m$ 上的均匀分布，这一结论仍然成立。

证明。我们首先注意到，对于 $r \in \{r^{\text{asc}}_{\text{Max}}, r_{\text{Sum}}, r_{\text{Lex}}\}$，该断言可由引理14直接推出。令 $\mathbf{M}_1$ 和 $\mathbf{M}_2$ 为引理17中的矩阵，并令 $\Pi$ 为 $\{0, 1\}^m$ 上的均匀分布。由于对于每个 $\mathbf{u} \in \{0, 1\}^m$，两个矩阵上的效应函数之差在 $\mathbf{u} \models \phi$ 时为 0，否则为 1，因此 $\phi$ 的满足赋值数量等于 $2^m$ 减去 $2^m$ 乘以 . 的期望值之差。

对于 $r = r^{\mathsf{dsc}}_{\mathsf{Max}}$，我们注意到， $d_{\mathsf{md}}$ 在同时反转两个顺序后保持不变，因此使用升序还是使用反向平局处理的降序都无关紧要。

◀

## C 第6节中省略的证明

▶**定理18**。 对于我们在 $r$ 中引入的排序函数 $e$ 和效应函数 $e$ ，其第3节中给出的Shapley$\langle r, e\rangle$ 的计算复杂性与SHAP$\langle r, e\rangle$ 的计算复杂性相同，后者见表1。

**证明**。 根据定理19，所有用于计算SHAP$\langle r, e\rangle$ 的多项式时间算法，均可在多项式时间内用于计算Shapley$\langle r, e\rangle$ 。我们需要证明的困难性结果是，问题Shapley$\langle r, e\rangle$ 对于以下FP$^{\#\mathsf{P}}$-组合而言是困难的： $r$ 和 $e$:

(a) $r = r_{\mathsf{Sum}}$ 和 $e \in \{e_{\mathsf{kT}}, e_{\mathsf{Ham}}, e_{\mathsf{pos}}, e^k_{\mathsf{top}}\}$，如果输入矩阵$\mathbf{M}$ 以二进制编码；(b) $r \in \{r^{\mathsf{asc}}_{\mathsf{Max}}, r_{\mathsf{Sum}}, r_{\mathsf{Lex}}\}$ 和 $e = e^k_{\mathsf{top}}$，其中 $k$ 是问题的一个参数；(c) $r = r^{\mathsf{dsc}}_{\mathsf{Max}}$ 和 $e = e^k_{\mathsf{top}}$，当 $k$ 是输入的一部分时；(d) $r \in \{r^{\mathsf{asc}}_{\mathsf{Max}}, r^{\mathsf{dsc}}_{\mathsf{Max}}, r_{\mathsf{Sum}}, r_{\mathsf{Lex}}\}$ 和 $e = e_{\mathsf{md}}$。

断言(a) 可直接由引理20 和引理10 通过采用PREC$\langle r\rangle$ 中的推论 11得出。断言(b) 和(c) 可由引理 21 和引理14 得出，所依据的是我们在最大值升序 与Maxdsc 等价时 $k$ 属于输入这一事实；我们已在定理 12的证明中证明了这一点。最后，断言(d) 可由引理 21和引理得出，依据是以下事实：$d_{\mathsf{md}}$ 在同时反转两个排列时保持不变；并且，按降序排序并采用反向的平局处理顺序，所得排序与按升序排序所得顺序相反。◀

▶**引理20。背包博弈中参与者的 Shapley 值计算是**FP$^{\#\mathsf{P}}$**难的**

**证明**。我们通过从计数背包问题进行归约，证明背包博弈中的 Shapley 值计算是FP$^{\#\mathsf{P}}$难的。

设 $I = (b_1, \ldots\ldots, b_\ell; d)$ 是背包博弈的一个输入。我们关注玩家 $\ell + 1$ 的 Shapley 值，该玩家属于背包容量 $d$。对于集合 $C \subseteq [\ell]$，我们观察到 $\nu_{\mathsf{knap}}(C) = 0$ 且

$$\nu_{\mathsf{knap}}(C \cup \{\ell + 1\}) - \nu_{\mathsf{knap}}(C) = \begin{cases} 1, \text{ if } \sum_{i \in C} b_i \leq d \\ 0, \text{ otherwise.} \end{cases} \tag{11}$$

令 $s_k$ 为大小为 $k$ 的有效物品集合的数量；也就是说

$$s_k = \left| \left\{ C \subseteq [\ell] \mid |C| = k \wedge \sum_{i \in C} b_i \leq d \right\} \right|.$$

采用此记号以及 公式 (11)，我们可以将 $\ell + 1$ 的 Shapley 值表示为

$$\text{Shapley}(\ell + 1, I, \nu_{\mathsf{knap}}) = \sum_{k=0}^{\ell} \frac{k!(\ell - k)!}{(\ell + 1)!} s_k.$$

对于 $0 \leq j \leq \ell$，我们定义新的实例 $I_j$，其中添加了 $j$ 个权重为 $d + 1$ 的物品，因此 $I_j = (b_1、\ldots\ldots、b_\ell, \ d + 1、\ldots\ldots、\ d + 1; \ d)$。由于每个新增物品的权重都大于背包容量 $d$，因此它们都不可能属于有效的物品集合，所以

$$\text{Shapley}(\ell + j + 1, I_j, \nu_{\mathsf{knap}}) = \sum_{k=0}^{\ell} \frac{k!(\ell + j - k)!}{(\ell + j + 1)!} s_k.$$

由此得到以下线性方程组：

$$\begin{pmatrix} \ell! & (\ell-1)! & \cdots & 0! \\ (\ell+1)! & \ell! & \cdots & 1! \\ \vdots & \vdots & \ddots & \vdots \\ (2\ell)! & (2\ell-1)! & \cdots & \ell! \end{pmatrix} \cdot \begin{pmatrix} 0!s_0 \\ 1!s_1 \\ \vdots \\ \ell!s_\ell \end{pmatrix} = \begin{pmatrix} (\ell+1)!\text{Shapley}(\ell+1, I_0, \nu_{\text{knap}}) \\ (\ell+2)!\text{Shapley}(\ell+2, I_1, \nu_{\text{knap}}) \\ \vdots \\ (2\ell+1)!\text{Shapley}(2\ell+1, I_\ell, \nu_{\text{knap}}) \end{pmatrix}$$

该线性方程组的系数矩阵可逆，因此该方程组满秩，如 [5,定理 1.1 的证明]所示。因此，我们可以通过求解该方程组并返回 $s_0 + \ldots + s_\ell$。◀

**引理 21的证明**。我们通过从合取范式公式的满足解计数问题归约，证明正 CNF 博弈中的 Shapley值计算是$\text{FP}^{\#\text{P}}$-难的。

对于正合取范式公式 $\phi = \bigwedge_{i=1}^{\ell} D_i$，其变量集为 $X_1, \ldots, X_m$，子句为 $D_i = \bigvee_{j=1}^{r_i} X_{j_i}$。我们通过引入一个新变量 $X_{m+1}$，并将该变量添加到每个子句中，从 $\phi$ 构造一个新的合取范式公式 $\hat{\phi}$。这样，每个包含 $m+1$ 的联盟都满足该公式，因此对于联盟 $C \subseteq [m]$，我们得到：

$$\nu_{\text{CNF}}(C \cup \{X_{m+1}\}) - \nu_{\text{CNF}}(C) = \begin{cases} 0, & \text{if } \alpha[C] \models \phi \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

令 $\bar{s}_k$ 表示变量赋值的数量 $\alpha\colon\ [m] \to \{\text{false}, \text{true}\}$ 这些赋值不满足 $\phi$。采用此记号以及方程（12），我们可以将$m+1$ 的 Shapley 值表示为

$$\text{Shapley}(m+1, \hat{\phi}, \nu_{\text{CNF}}) = \sum_{k=0}^{m} \frac{k!(m-k)!}{(m+1)!} \bar{s}_k.$$

对于 $0 \le j \le m$，我们通过引入 $j$ 个新变量 $X_{m+1}$、……、$X_{m+j}$并将每个变量添加到所有子句 $D_i$ 中，来定义新的合取范式公式 $\phi_j = \bigwedge_{i=1}^{\ell} D_i^{(j)}$。现在，如果对于所有 $m+1 \le i \le m+j$，$\alpha'|_{[m]} \not\models \phi$ 且 $\alpha'(i) =$ 均为 false，那么变量赋值 $\alpha'\colon\ [m+j] \to \{\text{false}, \text{true}\}$ 不满足 $\phi_j$，因此

$$\text{Shapley}(m+j+1, \hat{\phi}_j, \nu_{\text{CNF}}) = \sum_{k=0}^{m} \frac{k!(m+j-k)!}{(m+j+1)!} \bar{s}_k$$

这得出以下线性方程组：

$$\begin{pmatrix} m! & (m-1)! & \cdots & 0! \\ (m+1)! & m! & \cdots & 1! \\ \vdots & \vdots & \ddots & \vdots \\ (2m)! & (2m-1)! & \cdots & m! \end{pmatrix} \cdot \begin{pmatrix} 0!\bar{s}_0 \\ 1!\bar{s}_1 \\ \vdots \\ m!\bar{s}_\ell \end{pmatrix} = \begin{pmatrix} (m+1)!\text{Shapley}(m+1, \hat{\phi}_0, \nu_{\text{CNF}}) \\ (m+2)!\text{Shapley}(m+2, \hat{\phi}_1, \nu_{\text{CNF}}) \\ \vdots \\ (2m+1)!\text{Shapley}(2m+1, \hat{\phi}_m, \nu_{\text{CNF}}) \end{pmatrix}$$

该线性方程组是满秩的，因为如 [5, 定理 1.1 的证明所示，左侧矩阵可逆。]因此，我们可以通过求解该方程组并返回 $2^m - \bar{s}_0 - \ldots - \bar{s}_m$，来确定 $\phi$ 的满足赋值数量。◀

## D 前 k 名视角

我们证明了针对顶层-$k$ 成员资格 $e^k_{\text{top}}$ 效应函数（见表1）所证明的结果，同样适用于顶层-$k$ （对称）差异 $e^k_{\Delta}$ 和任意变化 $e^k_{\text{any}}$ 效应函数。

▶ **定理 22**。 令 $r \in \{r_{\mathsf{Sum}}, r^{\mathsf{asc}}_{\mathsf{Max}}, r^{\mathsf{dsc}}_{\mathsf{Max}}, r_{\mathsf{Lex}}\}$ 和 $e \in \{e^k_{\Delta}, e^k_{\mathsf{any}}\}$。问题 $\mathsf{EXP}\langle r, e\rangle$ 可在多项式时间内求解，当 $r = r^{\mathsf{dsc}}_{\mathsf{Max}}$ 且 $k$ 是一个固定参数时。否则，它是 $\mathsf{FP}^{\#\mathsf{P}}$-困难的；即使 $k$ 是一个固定参数，对于 $r \in \{r_{\mathsf{Sum}}, r^{\mathsf{asc}}_{\mathsf{Max}}, r_{\mathsf{Lex}}\}$，它仍然是 $\mathsf{FP}^{\#\mathsf{P}}$-困难的。

**证明**。 设 $T_0$ 为整个矩阵$\mathbf{M}$ 的排序中的前 $k$ 个元素集合，并且 $T$ 为将排序函数应用于一组 $C$ 列所诱导的另一种排列中的前 $k$个元素集合，因此我们的两个效应函数定义为 $e^k_{\Delta}(\pi)$: $= |T \cup T_0| - |T \cap T_0|$ 以及 $e^k_{\mathsf{any}}(\pi)$: $= \mathbb{1}_{T \neq T_0}$。我们将证明期望交集大小的定理，而不是预期的对称差 $e^k_{\cap}(\pi)$: $= |T \cap T_0|$，因为 $|T \cup T_0| - |T \cap T_0| = 2k - 2|T \cap T_0|$，这意味着我们可以由其中一个推导出另一个。

从否定的一侧入手，我们注意到，$\mathbf{M}$在引理14中由正合取范式公式 $\phi$ 定义的矩阵具有以下性质：若 $\alpha[C] \models \phi$，则 $T_0 = T$；否则 $|T_0 \cap T| = k - 1$。根据定理 12证明中的论证，我们可以推出$\mathsf{FP}^{\#\mathsf{P}}$-困难性：对于定理中所述的 $\mathsf{EXP}\langle r,\ e^k_{\cap}\rangle$ 和 $\mathsf{EXP}\langle r,\ e^k_{\mathsf{any}}\rangle$，其参数组合为 $r$和 $e$。

接下来需要证明当 $r = r^{\mathsf{dsc}}_{\mathsf{Max}}$ 且 $k$ 固定时的正面结果。首先，考虑期望交集大小 $e^k_{\cap}(\pi)$ $:= |T \cap T_0|$。利用期望的线性性，我们只需调用 top-$k$ 成员资格算法 $k$ 次即可解决该问题：

$$\mathbb{E}_{\mathbf{u}\sim\Pi}[e^k_{\cap}(\pi)] = \mathbb{E}_{\mathbf{u}\sim\Pi}\Big[\sum_{i\in T_0} \mathbb{1}_{i\in T}\Big] = \sum_{i\in T_0} \mathbb{E}_{\mathbf{u}\sim\Pi}\big[\mathbb{1}_{i\in T}\big].$$

根据 定理7， $\mathbb{E}_{\mathbf{u}\sim\Pi}\big[\mathbb{1}_{i\in T}\big]$ 在 $k$ 固定时可在多项式时间内计算，因此 $e^k_{\cap}$也可在多项式时间内计算。

对于期望变化 $e^k_{\mathsf{any}}$，我们逐一考虑每个元组 $\mathsf{t} \in T_0$，并采用定理7证明中的相同方法，对 t 以及一个不包含 $k-1$ 元组 $T_0 \setminus \{\mathsf{t}\}$ 的矩阵，求解 top-1 成员资格问题 $\mathsf{EXP}r^{\mathsf{dsc}}_{\mathsf{Max}}, e^k_{\mathsf{top}}$。具体而言，我们将不在 $T_0$ 中的 $n-k$ 元组合并为两个竞争元组 t′ 和 t″，具体取决于它们在平局判定规则下是否胜过 t。在这两种情况下，元组 t 和 t′ 在各自集合的每一列中都包含最大值。然后，我们调用算法 $\mathsf{PREC}r^{\mathsf{dsc}}_{\mathsf{Max}}$，返回使 t 满足 top-1 成员资格的权重列表（回想观察 4）。最后，我们求不同 $\mathsf{t} \in T_0$ 对应的权重列表的交集，并计算其概率，从而得到所需的期望。◀

## E 汉明距离的困难性

在本节中，我们证明引理17 对汉明距离 $d_{\mathsf{Ham}}(\pi_1,\ \pi_2)$ $=\sum_{1\le i\le n} \mathbb{1}_{\pi_1(i)\neq\pi_2(i)}$ 同样成立。

▶ **引理23**。 设 $\phi = \bigwedge_{i=1}^{\ell} D_i$ 是一个变量集为 $X_1$、……、$X_m$且具有子句 $D_i = \bigvee_{j=1}^{r_i} X_{j_i}$ 的正合取范式公式。那么，存在两个矩阵 $\mathbf{M}_1 \in M^{(2\ell+1)\times m}$ 和$\mathbf{M}_2 \in M^{(2\ell+2)\times m}$，其元素属于 $\{0,\ 1\}$，并满足以下性质：对于每个 $C$ 列集以及每个排序函数 $r \in \{r^{\mathsf{asc}}_{\mathsf{Max}},\ r_{\mathsf{Sum}},\ r_{\mathsf{Lex}}\}$，都有

$$d_{\mathsf{Ham}}\big(r(\mathbf{M}_2), r(\mathbf{M}_2|_C)\big) - d_{\mathsf{Ham}}\big(r(\mathbf{M}_1), r(\mathbf{M}_1|_C)\big) = \begin{cases} 0, & \text{if } \alpha[C] \models \phi \\ 1, & \text{otherwise.} \end{cases} \tag{13}$$

**证明**。 我们证明矩阵 $\mathbf{M}_1$ 和 $\mathbf{M}_2$ 在引理17 中定义的矩阵也满足 公式（13）。我们采用引理17 证明中的记号。若 $\alpha[C] \models \phi$，则我们已经证明，全零元组在 $r(\mathbf{M}_i)$ 和 $r(\mathbf{M}_i|_C)$ 中的秩相同，而其他元组具有

相同的位移，因此，对于 $\mathbf{M}_1$ 和 $\mathbf{M}_2$, ，位移的数量相等，且 公式（13）在此情形下成立。

如果 $\alpha[C] \nvDash \phi$，那么，与未满足子句对应的所有元组以及全零元组都会改变其位置，而与已满足子句对应的元组在$\mathbf{M}_1$ 中的位置发生变化，当且仅当它们在$\mathbf{M}_2$中的位置发生变化。由于$\mathbf{M}_2$ 比$\mathbf{M}_1$,多一个全零元组，这就证明了该结论。◀