arXiv:2301.06787v1 [cs.DS] 17 Jan 2023

即时使用子模函数进行排序*

Guangyi Zhang† Nikolaj Tatti‡ Aristides Gionis§

摘要

针对广泛的子集选择问题，最大化子模函数已得到深入研究。然而，子模性在序列选择和排序问题中的作用却鲜少受到关注。近期提出的一个名为最大子模排序（MSR）的框架，旨在解决一类自然产生的排序问题：多个需求共享资源，且各自预算不同。例如，MSR 框架可用于针对多种用户意图对网页进行排序。本文将 MSR 框架扩展至流式设置。具体而言，我们考虑两种不同的流式模型，并提出实用的近似算法。在第一种流式模型中，即函数到达模型，我们假设子模函数（需求）持续以流的形式到达；在第二种模型中，即项目到达模型，我们假设项目（资源）持续到达。此外，我们还研究了带有输出序列附加约束的 MSR 问题，例如拟阵约束，该约束可确保不同组别项目获得公平曝光。这些扩展显著拓宽了 MSR 框架所能涵盖的问题范围。在实际应用方面，我们基于 MSR 公式设计了若干新颖应用，并通过实证评估了所提方法的性能。

1 引言

次模集合函数刻画了许多现实世界现象中存在的“收益递减”特性 [16]。次模函数广受欢迎，因为它们允许使用文献中发展出的丰富优化技术工具。实际问题中使用的次模函数包括文档摘要[19],、社交网络中的病毒式营销 [12], 、社会福利最大化 [27], 等。现有的大多数基于子模性的問題表述都局限于选择项目的子集，而完全忽略了项目顺序的影响。然而，在许多应用中，项目的顺序都起着重要作用。本文研究了一种在子模性框架内对项目进行排序的通用方法。

在广泛的应用中，都需要创建一个资源序列，供多个需求共享。例如，在响应用户查询对网页进行排序时，我们希望满足多种用户意图；在为一群人创建共享音乐内容的直播时，我们希望满足所有听众的喜好；在公共显示屏上选择广告时，我们希望广告对所有乘客都相关。通常，每项需求都有各自的最大预算。例如，在前一种情形中，预算代表用户预期浏览的网页数量。这类问题的主要挑战在于，找到资源的一个（部分）排序，使其在不同预算下尽可能满足多项需求。

具体而言，最大子模排名（MSR）问题 [28]涉及资源和需求。资源在全集 V 中称为物品。需求需要资源，而某个需求对一组资源的效用由一个非递减子模集合函数 $f\colon\ 2^V \to \mathrm{R}_+$来刻画。需求的预算由基数约束表示，即其对应函数允许选取的物品最大数量。目标是找出一个（部分）物品序列，使总效用最大化，也就是函数值之和。第 3 节将给出问题的正式定义。

为涵盖更广泛的问题并提升框架的通用性，本文在两种自然的流式模型中扩展了 MSR 问题，并将其命名为函数到达和物品到达。在函数到达模型中，我们假设需求以在线方式持续到达，且事先不知道未来需求的类型和/或数量。这一模型在实践中十分常见；例如，新的观众可能随时加入直播。在物品到达模型中，我们假设物品以流的形式到达，并且在看到流中的所有物品后，需要使用单遍扫描和有限内存输出一个物品序列。换句话说，我们需要在每个物品到达后立即对其进行处理。这一模型提供了一种

*本研究由芬兰科学院资助，资助项目包括 MALSOME (343045)、AIDA (317085) 和 MLDB (325117)，ERC高级研究资助项目REBOUND (834862)、EC H2020 RIA 项目 SoBigData++ (871042)，以及由克努特与爱丽丝·瓦伦堡基金会资助的瓦伦堡人工智能、自主系统与软件计划 (WASP)。

†瑞典皇家理工学院。guaz@kth.se
‡赫尔辛基信息技术研究所，赫尔辛基大学。
nikolaj.tatti@helsinki.fi§瑞典皇家理工学院。argioni@kth.se

版权所有 © 2023 SIAM。未经授权，禁止转载本文。

在物品持续到达或数量过多而无法全部加载到内存时处理 MSR 问题的方法。

对于我们考虑的两种流式模型，我们都提出了实用的近似算法。此外，我们还研究了带有输出序列附加约束的 MSR 问题，例如拟阵约束（见第 3 节），它可以确保不同群组中的物品获得公平曝光。

在实际应用方面，我们基于 MSR 表述提出了新颖的应用。这里重点介绍一种渐进多样化个性化推荐的应用，而其他许多有趣的应用将在第 6 节讨论，包括直播和目录式病毒营销。个性化推荐中一种流行的方法是选择一个物品的简洁子集，使两个子模项——相关性和多样性——的加权和（带有一个权衡参数）最大化。请注意，对于给定的参数值，一个子集所呈现的相关性与多样性之间的权衡是固定的。另一方面，以自适应方式满足用户对多样性的需求或许效果更好。例如，目标用户可能会更喜欢这样的推荐列表：最相关的物品排在前面，列表越往后则逐渐越多样化。通过 MSR 表述，可以最大化多个此类函数的加权和来满足这一需求，其中每个函数的权衡值依次递增。正式定义见第 3 节。

本文的贡献总结如下。

- 我们研究两种不同流式模型下的最大子模排序（MSR）问题，并针对每种模型设计近似算法。

- 在函数到达流式模型中，我们证明，如果允许重复使用物品，一个简单的贪心算法便可为 MSR 问题提供 2 近似。对于贪心算法而言，这一近似比是紧的。我们还证明，如果每件物品至多使用一次，该问题无法近似。

- 我们提出一种新颖的归约，将排序问题映射为一个受二分图匹配约束的子集选择问题。由此可直接得到一个结论：针对输出序列上的一般 p-拟阵约束，存在适用于 MSR 问题的近似算法。另一个结论是，我们可以为物品到达流式模型中的 MSR 问题设计高效的近似算法。

- 我们将增强型 MSR 框架应用于若干新颖的现实应用，并通过实证评估所提出算法的性能。

本文其余部分安排如下。第 2 节讨论相关工作。第 3 节给出正式的问题定义。第 4 节和第 5 节分别介绍函数到达型 MSR 问题和物品到达型 MSR 问题的近似算法。第 6 节进行实证评估，第 7 节总结全文。缺失的证明及更多实验细节见补充材料。我们的实现已公开发布。[1]

2 相关工作

序列次模性。 MSR 问题由 Zhang 等人提出 [28], ，后来被证明是有序子模性的一个特例；有序子模性由 Kleinberg 等人提出 [14]。特别是，这两种表述都避免了不自然的后缀单调性 性质，而先前的表述则要求这一性质 [1, 26, 29]。后缀单调性要求：在序列前端添加任意物品后，函数值仍然单调非减。此外，MSR 问题也可视为子模排序问题的对偶问题 [2], ；后者旨在没有单独预算限制时，最小化所有函数的总覆盖时间。目前尚无针对 MSR 问题的流式扩展。

二分图匹配。 二分图匹配及其众多变体已得到广泛研究 [21]。离线加权二分图匹配可以通过最大流建模等方法精确求解，而在线变体目前已知的最佳近似比由 Fahrbach 等人实现 [6]。众多常见变体已知可视为子模社会福利问题的特例[17]。与我们在第 5 节中的归约类似，一些分配或调度问题也可以归约为一个受二分图匹配约束的子集选择问题 [24, 27]。

约束次模最大化。 在离线设置下，对于 a p-拟阵约束，众所周知，贪心算法对模函数可保证 a p-近似，对次模函数可保证 $(p+1)$-近似 [10, 15]。Feldman 等人 [7] 在 p-Exchange 约束下实现了目前已知的次模最大化最佳 $(p+\epsilon)$- 近似；这种约束比 p- 拟阵更一般。对于 p- 拟阵上的模函数最大化问题（ p 维匹配是其特例），已知其近似下界为 $\Omega(p/\ln p)$ [11]。

[1] https://github.com/Guangyi-Zhang/subm-ranking-on-the-fly

版权所有 © 2023 SIAM 未经授权，禁止转载本文。

关于流式设置，Levin 和 Wajc [18] 提出了一种满足匹配约束的 $3 + 2\sqrt{2} \approx 5.828$ 近似算法。在 p-拟阵约束下，Chakrabarti 和 Kale [5], 受 Badanidiyuru [3] 中模变体的启发，得到了一个 $4p$-近似算法。就下界而言，Badanidiyuru [3] 的分析已被证明对于任何在线算法都是最优的（即任何时刻仅维护一个可行解的流式算法）。此外，即使满足二分图匹配约束，也不可能实现 2.692-近似；并且有证据表明，下界可能高达 3-近似 [9]。对于 p-拟阵约束，已证明任何使用亚线性内存的流式算法都存在 p 的下界；此外，要将目前已知的 $4p$-近似实现任何对数级改进，所需内存都必须是超多项式级别的 p [9]。

3 问题定义

本节将在两种不同的流式模型中介绍最大子模排名（MSR）问题。随后，我们将介绍一种渐进式多样化个性化推荐的形式化表述。在此之前，我们先简要回顾子模性和拟阵的概念。

子模性。给定一个集合 V，函数 $f\colon 2^V \to \mathrm{R}+$ 称为子模函数，如果对于任意 $X \subseteq Y \subseteq V$ 和 $v \in V \setminus Y$，都有 $f(v \mid Y) \leq f(v \mid X)$，其中 $f(v \mid Y) = f(Y + v) - f(Y)$ 是 v 相对于集合 Y 的边际增益。函数 f 称为模函数，如果对于任意 $X \cap Y = \emptyset$，都有 $f(X) + f(Y) = f(X \cup Y)$。函数 f 称为单调非减函数，如果对于任意 $X \subseteq Y \subseteq V$，都有 $f(Y) \geq f(X)$。不失一般性，我们可以假设函数 f 是归一化的，即 $f(\emptyset) = 0$。

拟阵。对于一个集合 V，如果子集族 $\mathcal{M} \subseteq 2^V$ 满足以下两个条件，则称其为拟阵：(1) 向下封闭性：如果 $X \subseteq Y$ 和 $Y \in \mathcal{M}$，那么 $X \in \mathcal{M}$；(2) 增广性：如果 $X, Y \in \mathcal{M}$ 和 $|X| < |Y|$，那么对于某个 $v \in Y \setminus X$，有 $X + v \in \mathcal{M}$。两个有用的特例是均匀拟阵和分划拟阵。前者就是一个 k基数约束，即 $\mathcal{M} = \{S \subseteq V : |S| \leq k\}$；后者由多个基数约束组成，每个约束分别施加于 $V = \cup_\ell G_\ell$ 的一个互不相交的子集 G_ℓ 上，即 $\mathcal{M} = \{S \subseteq V : |S \cap G_\ell| \leq k_\ell$，对所有 $\ell\}$ 均成立。给定 p 个拟阵 $\{\mathcal{M}_j\}_{j\in[p]}$，它们的交称为 p-拟阵。

我们用 $\sigma(V)$ 表示由 V 中的物品组成的所有序列的集合。给定一个序列 $\pi \in \sigma(V)$，我们用 π_i表示 π 中第 i 个物品，用 $\pi + v$ 表示将物品 v 追加到 π 后得到的新序列。序列 π 的长度记作 $|\pi|$。π 中的物品集合记作 $V(\pi) \subseteq V$。序列 π 是另一个序列 π' 的子序列，记作 $\pi \preceq \pi'$。给定一个区间 $w = [s, e]$，其中 s, e 是整数，我们记作 $\pi[w] = \pi[s : e] = \{\pi_s, \ldots, \pi_e\}$。更一般地，给定物品的一个子集 $R \subseteq V$，我们记作 $\pi[R] = \{\pi_i \mid i \in R\}$。

现在，我们可以定义 MSR 问题 [28]及其流式变体了。

问题 3.1。（最大子模排序（MSR））给定一个包含 n 个物品的集合 V，以及一组 m 单调非减的次模函数 $F = \{f_i\}_{i\in[m]}$，每个函数都关联一个整数 k_i，目标是找到一个序列，使其满足

$$\arg\max_{\pi\in\sigma(V)} \sum_{f_i\in F} f_i(\pi[1 : k_i]). \tag{3.1}$$

如果对可行序列 π 中的物品施加额外的 p-拟阵约束 $\mathcal{M} \subseteq 2^V$，即 $V(\pi) \in \mathcal{M}$，则我们将该问题称为 MSRp。例如，这种 p- 拟阵约束有助于避免返回的序列中过度代表某些物品组。如果 F 中的函数是模函数，我们则将该问题称为最大模排序 （MMR）。

在函数到达流式模型中，我们会在时间步 t 观察到一组新函数 F_t，目标是实时生成一个序列 π，也就是说，在每个时间步都不可撤销地决定将一个物品放入 π。请注意，先前物品步骤中已放入的物品不能再使用。我们假设在步骤 t 决定第 t 个物品之前，会观察到新函数 F_t；函数到达后会在后续步骤中持续生效，直到其预算耗尽。在某个步骤中也可以不放入任何物品（通过在 V 中引入虚拟项）。更正式地说，函数到达模型中的 MSR 问题定义如下。

问题 3.2。（函数到达型 MSR（MSR-F））给定一个由 V 个 n 组成的集合，以及一组单调不减的次模函数 F_t，它们在步骤 t 开始时到达，到达时间为 $\tau(f) = t$；并给定每个 $f \in F_t$ 对应的整数 $k(f)$，目标是找到一个序列，使其满足

$$\arg\max_{\pi\in\sigma(V)} \sum_t \sum_{f\in F_t} f(\pi[t : k(f)]), \tag{3.2}$$

通过在步骤 t 不可撤销地决定第 t 个物品 π_t。

相比之下，在物品到达流式模型中，我们完全了解所使用的函数。实际上，我们还允许每个函数 f_i 在序列中任意“预留” k_i 个时段 $R_i \subseteq [n]$，而不是仅仅预留 MSR 中最前面的 k_i 个时段 $[k_i]$。给定一个序列时，函数 f_i 只会从时段 R_i接收物品。例如，在电影院安排放映时间时，用户（f_i）可能只在周末或一天中的特定时间有空，或者在

版权所有 © 2023 SIAM。未经授权，禁止转载本文。

某个特定时间有空。物品到达型MSR问题的目标是，在单遍扫描所有到达的物品后，使用“小”的内存大小生成一个序列 π 。换句话说，从内存中丢弃的物品之后无法再使用。如果序列中某个时段没有函数可用，则可以不放置任何物品。形式化地说，物品到达流式模型中的MSR 问题定义如下。

问题 3.3。（物品到达型MSR（MSR-I））
给定一组单调非减的次模函数 $F = \{f_i\}_{i\in[m]}$，每个函数都与由 $R_i \subseteq [n]$ 指定的 k_i 可用时段相关联，并且 V 中的物品以流的形式到达，目标是在

$$\arg\max_{\pi\in\sigma(V)} \sum_{f_i\in F} f_i(\pi[R_i]). \tag{3.3}$$

此外，任意时刻能够存储的物品数量仅取决于 $\{k_i\}$ ，而非 n。

在所有物品均已就位的离线设置中，我们将这一变体称为带可用性约束的*MSR* （MSR-A）问题。注意，当 $R_i = [k_i] = \{1, \ldots, k_i\}$ 时，MSR-A问题等价于原始MSR问题。我们还注意到，当 F中只有一个函数时，MSR-I推广了流式次模最大化问题；对于该问题，在 n 中没有任何流式算法能以次线性内存实现优于 2 [8]的近似比。

逐步增加多样性的个性化推荐。一种流行的个性化推荐方法 [22] 是选择一个简洁子集 S ，其中包含的物品可最大化以下形式的次模函数

$$f_{\lambda,k}(S) = (1-\lambda)\sum_{v\in S} \mathrm{rel}(v) + \frac{\lambda k}{|V|}\sum_{u\in V}\max_{v\in S}\mathrm{sim}(u,v),$$
$$\text{such that } |S| \le k. \tag{3.4}$$

其中，rel(v)衡量物品 v 对目标用户的相关性，sim(u, v)则表示两个物品 u、 v之间的相似度。第二项体现了一种特定的多样性概念（也称为代表性或全局覆盖）：对于每个未选中的物品 $u \in V$，都存在某个与之足够相似的物品 $v \in S$ u。为了生成能够自适应满足用户多样性需求的推荐列表，可以最大化多个函数 $\{f_{\lambda,k}\}$ 的加权和，这些函数的权衡值 $\lambda \in [0,1]$ 和基数 k逐渐增大，使得列表后面的部分由具有较大 λ的函数主导。

4 函数到达型 MSR

本节讨论函数到达型 MSR（MSR-F）问题的两种情形，具体取决于

算法 1： 函数到达型的贪心算法
到达式 MSR（MSR-F）

```
初始化一个空序列 π
F ← ∅
对于 t = 1, ... 执行
    F ← F ∪ F_t        // 接收函数 F_t
    A ← {f ∈ F : k(f) ≥ t}        // 活跃函数在第 t 步
    v* ← arg max_{v∈V} Σ_{f∈A} f(v | π[τ(f) : t−1])
    π ← π + v*
返回 π
```

V 中的项目最多只能使用一次，还是可以使用多次。我们证明，前一种情形下该问题无法近似；对于后一种情形，我们提出了一种 2-近似算法。

作为分析的开始，如果要求输出序列不包含重复项目，可以证明，即使所有函数都是模函数，MSR-F 问题也是不可近似的。下面给出的这一结果源于在线选择问题的不可近似性；该问题旨在从一个无法回溯的对手序列中选出最大值 [13]。

定理 4.1。 如果每个物品最多只能使用一次，则 *MMR-F*问题是在线选择问题的推广。因此，没有任何随机算法能够为 *MMR-F* 问题保证 $o(n)$-近似。

鉴于在输出序列不应包含重复项目的情况下，MMR-F 难以近似，我们接下来研究 V 中的项目可以重复使用时的问题。在许多应用中，这一假设是合理的，例如，一首歌曲可以多次添加到播放列表中——并且请注意，对于已经包含这些项目的函数而言，不必要的重复项的边际增益为零，因此会被隐式抑制。

当输出序列允许包含重复项目时，我们证明，一个简单的贪心算法能够返回近似保证为 2 的解。该贪心算法见算法 1：它在每个步骤中，针对当前“活跃”的函数，选择最有益的项目。如果某个函数 f 尚未耗尽其项目预算 $k(f)$，则称其为活跃。

定理 4.2。如果 V 中的项目可以在输出序列中重复使用，则算法 *1* 可为 *MSR-F* 问题提供一个近似比为 *2* 的解。

贪心算法的近似保证可以证明是紧的，因为 Zhang

版权所有 © 2023 SIAM。未经授权，禁止复制本文。

等人 [28] 给出的下界同样适用于我们的情况。具体而言，由于 MSR-F 问题是 MSR 问题的推广，只需令所有函数在开始时到达，即可得到以下结果。

注4.1。（Zhang 等人 [28]）如果一个物品可以在输出序列中重复使用，那么算法 *1* 得到的 *2*-近似解对于 *MSR-F* 问题是紧的。

在本节余下部分，我们证明定理 4.2，而定理 4.1 的证明推迟到附录 A.1 节。

证明。 [定理 4.2] 的证明。我们记 $A_t = \{f \in \bigcup_{t' \le t} F_{t'} : \ k(f) \ge t\}$ 为步骤 t 中活跃函数的集合。我们将算法 1 生成的序列记为 π，其目标函数值为 ALG；将最优序列记为 π^*，其目标函数值为 OPT。

根据贪心选择准则，对于任意物品 $v \in V$，我们知道

$$\sum_{f \in A_t} f(\pi_t \mid \pi[\tau(f) : t-1]) \ge \sum_{f \in A_t} f(v \mid \pi[\tau(f) : t-1]). \tag{4.5}$$

为简化记号，我们记 $\pi[f]$ 表示 $\pi[\tau(f) : k(f)]$。

容易看出，ALG 等于t 对式 (4.5) 左侧求和，因此

$$\begin{aligned}
\mathrm{ALG} &= \sum_t \sum_{f \in A_t} f(\pi_t \mid \pi[\tau(f) : t-1]) \\
&\overset{(a)}{\ge} \sum_t \sum_{f \in A_t} f(\pi^*_t \mid \pi[\tau(f) : t-1]) \\
&= \sum_{f \in F} \sum_{t=\tau(f)}^{k(f)} f(\pi^*_t \mid \pi[\tau(f) : t-1]) \\
&\overset{(b)}{\ge} \sum_{f \in F} \sum_{t=\tau(f)}^{k(f)} f(\pi^*_t \mid \pi[f]) \\
&\overset{(c)}{\ge} \sum_{f \in F} f(\pi^*[f] \mid \pi[f]) \\
&= \sum_{f \in F} f(\pi[f] \cup \pi^*[f]) - f(\pi[f]) \\
&\ge \sum_{f \in F} f(\pi^*[f]) - f(\pi[f]) = \mathrm{OPT} - \mathrm{ALG},
\end{aligned}$$

其中，不等式 (a) 由式 (4.5) 得出，不等式 (b) 和 (c) 则源于子模性。 (b) □

5 物品到达型MSR

本节将介绍如何通过归约将排序问题转化为一个受二分图匹配约束的子集选择问题，以及由此带来的丰富结果。表1汇总了不同设定下的近似比，所用算法均来自引文。

定理 5.1. 对于任意整数 $p \ge 1$，*MSR-A*p问题是一个实例：在 $(p+1)$-拟阵约束下最大化单调非减次模函数。

由定理 5.1 可直接得出，物品到达型MSR（MSR-I）可以通过约束次模最大化的流式算法求解。

推论 5.1. 对于任意整数 $p \ge 1$，*MSR-I*p问题是一个实例：通过单遍扫描，在 $(p+1)$-拟阵约束下最大化单调非减次模函数，并在任意时刻记住 $\mathcal{O}(\sum_i k_i)$ 个物品。

此外，当函数为模函数时，我们可以得到一个更强的结果。

推论 5.2。 对于任意整数 $p \ge 1$，*MMR-A*p问题都是一个在 $(p+1)$-拟阵约束下最大化模函数的实例。特别地，*MMR-A* 问题是最大权二分图匹配的一个实例。

证明。 [定理 5.1] 的证明。归约的主要思路是构造一个扩展全集 V'，即

$$V' = \{(v,t) : v \in V, t \in [n]\}, \tag{5.6}$$

它可以看作是连接物品 $L = V$ 与排名 $R = [n]$ 的完全二分图中的边。定义$V(S') = \{v : (v, \ t) \in S'\}$ 为 S' 到 V的投影。我们还记 $X_v = \{(v, \ t) : \ t \in [n]\}$ 和 $Y_t = \{(v, \ t) : \ v \in V\}$。

假设给定一个 p-拟阵 $\mathcal{M} \subseteq 2^V$，定义在 V 上。定义 $\mathcal{M}' = \mathcal{A} \cap \mathcal{B} \cap \mathcal{C}$，其中

$$\begin{aligned}
\mathcal{A} &= \{S' \subseteq V' : V(S') \in \mathcal{M}\}, \\
\mathcal{B} &= \{S' \subseteq V' : |S' \cap X_v| \le 1, \text{ for all } v \in V\}, \\
\mathcal{C} &= \{S' \subseteq V' : |S' \cap Y_t| \le 1, \text{ for all } t \in [n]\}.
\end{aligned}$$

这里， $\mathcal{B}$ 强制每个物品只出现一次， $\mathcal{C}$强制每个时刻 t 只出现一个物品。因此，可行序列 π 可以写成满足 $\mathcal{M}'$ 的 V' 的子集。

另一方面，可根据物品在 S' 中对应的排名，对 $V(S')$ 中的物品排序，从而将可行子集 $S' \subseteq V'$ 转换为一个序列。如果 S' 由不连续的排名组成，可以插入虚拟项（或在 MSR 中将物品向前移动）来构造一个序列，且不会降低目标函数值。

我们声称 $\mathcal{M}'$ 是一个 $(p+1)$-拟阵。我们将通过论证 $\mathcal{A} \cap \mathcal{B}$ 是一个 p-拟阵来证明这一断言

版权所有 © 2023 SIAM。未经授权，禁止复制本文。

表1：MSR-A 和 MSR-I 的近似比总结

	无约束	p-拟阵
MMR-A	精确	$p+1$ （Korte 和 Hausmann [15]）
MSR-A	$2+\epsilon$ （Feldman 等人 [7]）	$p+1+\epsilon$ （Feldman 等人 [7]）
MMR-I	$1/0.5086$ （Fahrbach 等人 [6]）	$2(p+1+\sqrt{(p+1)p})-1$ (Badanidiyuru [3])
MSR-I	5.828（Levin 和 Wajc [18]）	4（$p+1$） （Chakrabarti 和 Kale [5]）

（根据 A.2 节中的引理 A.1）。因此，$\mathcal{A}\cap\mathcal{B}\cap\mathcal{C}$ 是一个 ($p+1$)-拟阵，因为 $\mathcal{C}$ 是一个拟阵。

接下来要证明的是，MSR-A 的目标函数（公式 3.3）相对于新的全集 V' 是单调非减且具有子模性。单调非减性显而易见，因此我们只详细说明子模性。回想一下，$R_i\subset[n]$ 由函数 f_i 可用的时段组成。定义 $R_i'=\{(v,\ t)\colon\ v\in V,\ t\in R_i\}$。现在，目标函数可写为 $g(S)=\sum_{f_i\in F}f_i(V(S\cap R_i'))$。对于任意子集 $S\subseteq W\subseteq V'$，将一个物品（$v,\ t$）加入 S 所带来的边际增益 $g(\cdot\mid S)$ 为

$$g((v,t)\mid S)=\sum_{f_i\in F:t\in R_i}f_i(v\mid V(S\cap R_i'))$$
$$\geq\sum_{f_i\in F:t\in R_i}f_i(v\mid V(W\cap R_i'))=g((v,t)\mid W),$$

由于 $V(S\cap R_i')\subseteq V(W\cap R_i')$，且每个 f_i 都具有子模性。因此，MSR-Ap 问题可以表述为在一个（$p+1$）-拟阵约束下的单调非减次模最大化实例。 □

6 实验

我们在推动 MSR-F 和 MSR-I 问题产生的新颖用例上评估了我们的方法。对于每个用例，我们使用现实世界的数据模拟一项具体任务，并通过实证评估所提出算法的性能。数据集摘要见表 2。关于运行时间的考察留待 A.4 节。我们的实现已公开发布。[2]

6.1 函数到达型 MSR 我们针对 MSR-F 问题介绍了两个用例：直播和目录式病毒营销，并在相关数据集上评估我们的方法。

算法。算法 1 中提出的贪心算法称为贪心算法。其他基线算法包括随机，它在每个步骤随机选择一个物品；以及TopK，它反复选择 Top-k 项目——在

表 2：数据集统计信息

数据集	$n=\lvert V\rvert$	$m=\lvert F\rvert$
音乐 [4]	61 415	10 000
GitHub 社交网络 [25]	37 700	100
搜狗网页 [20]	725	1 017
Twitter 词语 [23]	10 000	8

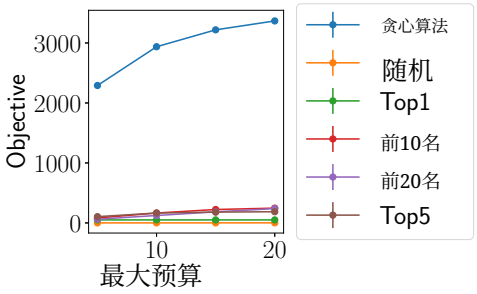
我们的使用场景中，这相当于循环播放最受欢迎的歌曲。请注意，TopK 是全知的，因为它需要提前获知物品的受欢迎程度。

直播。互联网上一种日益流行的应用是直播，主播会持续进行各种表演，而观众可以随时加入或离开。更具体地说，我们考虑音乐直播，其中主播会持续播放歌曲。为了模拟这一应用，我们使用百万歌曲数据集 [4], ，该数据集由表示用户、歌曲和播放次数的三元组组成。我们假设，如果一首歌曲被播放不止一次，用户就喜欢这首歌。我们将用户效用定义为对用户喜欢的歌曲的分数覆盖，这是一种次模函数。我们为每位用户设置一个介于 1 和最大预算参数之间的随机预算，即用户将收听的歌曲数量。此外，每位用户在一个较长的时间范围内获得一个随机到达时间。我们的目标是实时确定歌曲序列，以最大化用户效用总和。

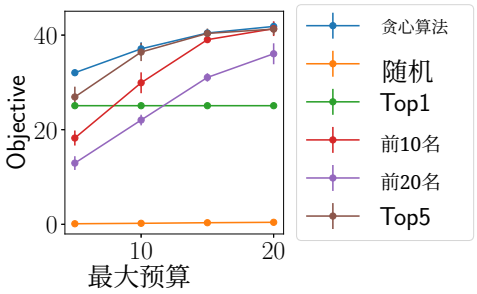
目录式病毒营销。对于社交网络中的病毒式营销应用，目标是找出一小组种子节点，使其能够影响许多其他用户。对于常见的扩散模型，受影响节点的数量是种子集的次模函数 [12]。在此，我们将这一经典结果推广，以同时应对多个营销需求；每项需求只关注触达特定的一组用户。例如，一位广告商可能希望影响女性用户，而另一位则可能希望影响特定城市附近的用户。每项需求都会向种子节点提供一定数量的产品样本，希望通过口碑传播为其产品做广告。

[2]https://github.com/Guangyi-Zhang/subm-ranking-on-the-fly

版权所有 © 2023 SIAM。未经授权，禁止转载本文。



（a）音乐直播

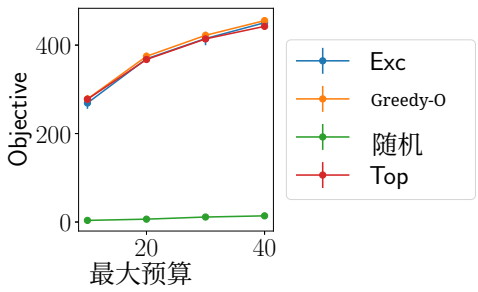


(b) GitHub 上的目录式病毒营销

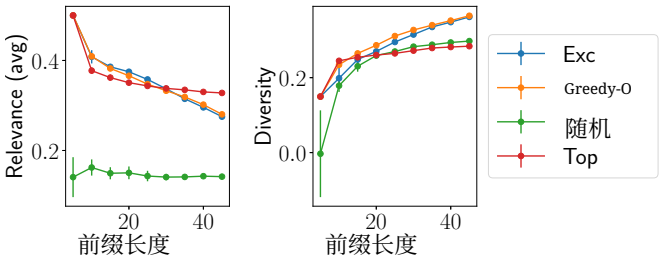
图 1：MSR-F。每项需求都会被赋予一个随机到达时间，以及一个介于 1 和最大预算参数之间的随机预算。(a) 用户需求的效用为 $f(S) = |S \cap S_{\text{like}}|/|S_{\text{like}}|$，其中 S_{like} 是用户喜欢的歌曲集合。(b) 营销需求的效用为 $f(S) = |N(S) \cap S_g|/|S_g|$，其中 $N(S)$ 是 S 的邻域，而 S_g 是网络中的目标节点群体。

作为实验评估的一个应用案例，我们考虑一个旨在满足这些营销需求的平台，并假设每个时间步都可以向一个种子节点发送一包不同产品的样品。随着营销需求实时到达，我们的目标是通过找出一个能使所有需求都受益的种子节点，将尚未完成的需求编入目录。请注意，每项需求的样品都应在该需求到达后尽快加入待发包裹。为模拟这一应用案例，我们使用 GitHub 社交网络 [25]。我们考虑 100 项需求，每项需求针对网络的一个随机子集，并附带随机数量的样品（数量从 1 到预算上限），以及一个随机到达时间。我们的目标是在每个时间步向一个精心挑选的种子节点发送一包样品，从而最大化需求的总效用。

结果。 模拟结果如图 1 所示，其中每个数据点表示三次运行的平均值



(a) 搜狗网页排名



(b) 在 Twitter 上查找“特朗普”的同义词

图2：MSR-I。(a) 用户意图的效用为 $f(S) = |S \cap S_{\text{rel}}|/|S_{\text{rel}}|$，其中 S_{rel} 是相关页面的集合。每项意图都会被赋予一个介于 1 和最大预算参数之间的随机预算。(b) 给定一个序列，其相关性和多样性（公式 3.4）会在每个前缀处进行测量。

每次运行使用不同的随机种子。对于音乐流媒体任务（图 1(a)），贪心算法 的表现远胜于所有其他基线算法。这表明，对于歌曲偏好多样化的用户，要取得良好表现，就需要使用像贪心算法这样的算法，以适应当前活跃用户的需求。在已分类的病毒式营销任务中（图 1(b)），贪心算法 仍然取得了最佳表现。不过，随着需求预算增加，TopK 中的几种基线算法逐渐接近贪心算法 。这说明，存在一群影响者，他们能够在 GitHub 网络中合力触达最多用户。

6.2 物品到达式 MSR-I。接下来，我们将介绍 MSR-I 以及渐进式多样化个性化推荐的实验。

算法。 我们采用 Chakrabarti 和 Kale [5] 提出的最先进流式算法作为我们的算法：该算法从排名 1 开始，尽可能将每个到达的物品贪心地分配到序列中的某个排名。我们将此算法称为 Exchange（Exc）。如果某个排名已被先前的物品占据，Exc 就会替换

版权所有 © 2023 SIAM 未经授权，禁止复制本文

当当前物品的价值是现有物品的两倍时，替换该物品。其他基线算法包括随机，它会生成一个随机序列；以及Top，它按单元素效用非递增的顺序排列物品。我们还纳入了一种离线基线算法——全知贪心算法（Greedy-O）[28], ，它能为最优值提供更精确的估计。Greedy-O会根据当前活跃函数集，依次贪心地选择一个物品。

多意图重排序。 在没有明确给定查询的用户意图时，搜索引擎需要在返回网页列表时考虑所有可能的意图。每种意图只与部分网页相关。我们用用户失去耐心前浏览到的相关页面的分数覆盖来表示该意图的效用。目标是生成一个网页列表，使所有意图的效用最大化。

在此应用场景中，我们从搜狗Q点击日志数据集 [20] 中提取用户意图。我们考虑所有与“电影”相关的查询，并针对每个此类查询收集所有发出该查询的用户。我们还收集他们点击的页面。我们将每位用户视为一种意图，并将其点击的页面视为相关页面集合。对于每种意图，我们生成一个介于 1 和最大预算参数之间的随机数，作为用户的“耐心度”，即用户将浏览的页面数量。

渐进式多样化个性化推荐。 接下来，我们考虑第 3 节介绍的推荐任务。为模拟一个具体任务，我们考虑针对给定关键词寻找代表性同义词。我们选择“特朗普”作为关键词。我们使用在 Twitter 语料库 [23] 上预训练的词嵌入 GloVe。由于采用稠密向量，任意两个词之间的相似度或相关性通过余弦相似度减去 0.5 来衡量。排名前 10 000 的相关词构成候选集 V，其中随机选取 100 个词作为基准，用于计算多样性项（公式 3.4）。给定推荐列表的最大长度 ($k = 40$)，我们为 $i \leq 8$ 创建多个函数 $\{f_i\}$ （公式 3.4），其中第 i 个函数与权重 $(1/2)^i$、权衡值 $\lambda = (i-1)/k$ 以及预算 $5i$ 相关。因此，函数 f_i 在第 i 个长度为 5 的子序列中占主导地位，而具有较大 i 的 f_i 更倾向于选择多样性不断增加的物品。

结果。 结果如图2所示。每个数据点表示三次运行的平均值，每次运行使用不同的随机种子。对于网页排名任务（图2(a)），除随机 算法外，所有算法的表现几乎相同。这表明不同用户意图之间的相关页面高度重叠。对于寻找多样化同义词的任务（图 2(b)），按目标函数排序的结果为Greedy-O $\approx$Exc $>$Top $>$ 随机($7.139 \approx 7.136 > 6.652 > 3.527$)。我们在图 2(b) 中分别展示了目标函数的两个组成部分：相关性和多样性。请注意，随机 算法是寻找多样化代表项的经典方法，而当目标函数退化为单一的模相关性项时，Top 算法是最优的。Exc 算法返回的词表确实越来越多样，同时也能在开头找到相关的同义词。Exc 实际返回的词表见 A.3 节。

7 结论

本文研究了两种流式模型下 MSR 问题的扩展。在第一种需求到达模型中，我们证明，如果物品可以重复使用，贪心算法可保证达到 2-近似。在第二种物品到达模型中，我们发现了一种归约，可将排序问题转化为受约束的子集选择问题，并继承标准次模最大化的近似保证。该归约还使我们能够在额外的 p-拟阵约束下近似求解 MSR 问题。最后，我们介绍了 MSR 问题的若干新颖应用，并考察了所提出算法的实证表现。

MSR 问题的一项局限是，在某些应用中，个体预算并不总是已知。另一项局限是，当需求和物品的数量都很大时，计算成本可能很高，尤其是对于物品到达型 MSR 问题。

就本研究的伦理考量而言，我们的算法是一个基于子模性的通用排序框架，并非针对特定应用。我们无法发现明显的负面社会影响。许多更广泛的机器学习问题，例如技术滥用、数据偏见、自动化对社会的影响等，同样与本研究相关，但其相关程度并不高于整个机器学习领域。

参考文献

[1] S. Alaei、A. Makhdoumi 和 A. Malekian。最大化序列次模函数及其在在线广告中的应用。《管理科学》，2021 年。[2] Y. Azar 和 I. Gamzu。基于次模价值的排序。载于第二十二届 *ACM-SIAM* 离散算法年会论文集，第 1070–1079 页。SIAM，2011 年。[3] A. Badanidiyuru。回购问题——具有取消成本的近似拟阵交。载于自动机与语言国际研讨会，

版权所有 © 2023 SIAM。未经授权，禁止复制本文。

与编程，第 379–390 页。Springer，2011 年。[4] T. Bertin-Mahieux、D. P. Ellis、B. Whitman 和 P. Lamere。百万歌曲数据集。载于第十二届国际音乐信息检索会议（*ISMIR 2011*）论文集，2011 年。[5] A. Chakrabarti 和 S. Kale。次模最大化遇上流式处理：匹配、拟阵及更多内容。数学规划，154(1)：225–247，2015 年。[6]M. Fahrbach、Z. Huang、R. Tao 和 M. Zadi-moghaddam。边加权在线二分图匹配。载于*2020* 年 *IEEE* 第 *61* 届计算机科学基础年会（*FOCS*），第 412–423 页。IEEE，2020 年。[7] M. Feldman、J. S. Naor、R. Schwartz 和 J. Ward。k-交换系统的改进近似算法。载于欧洲算法研讨会，第 784–798 页。Springer，2011 年。[8] M. Feldman、A. Norouzi-Fard、O. Svensson 和 R. Zenklusen。次模最大化的一向通信复杂度及其在流式处理和鲁棒性中的应用。载于第 *52* 届 *ACM SIGACT* 计算理论研讨会论文集，第 1363–1374 页，2020 年。[9] M. Feldman、A. Norouzi-Fard、O. Svensson 和 R. Zenklusen。在线最大化受拟阵交约束的次模函数。*arXiv* 预印本 *arXiv:2204.05154*，2022 年。[10] M. L. Fisher、G. L. Nemhauser 和 L. A. Wolsey。最大化次模集合函数的近似算法分析——II。载于多面体组合学，第 73–87 页。Springer，1978 年。[11] E. Hazan、S. Safra 和 O. Schwartz。关于近似 k-集合打包的复杂度。计算复杂性，15(1)：20–39，2006 年。[12] D. Kempe、J. Kleinberg 和 É. Tardos。通过社交网络最大化影响力传播。计算理论，11(4)：105–147，2015 年。[13] T. Kesselheim。关于姚氏原理和秘书问题的讲义，2016 年。网址：https://www.mpi-inf.mpg.de/fileadmin/inf/d1/teaching/summer16/random/yaosprinciple.pdf。[14] J. Kleinberg、E. Ryu 和 É. Tardos。有序子模性及其在多样化推荐中的应用。*arXiv* 预印本 *arXiv:2203.00233*，2022 年。[15] B. Korte 和 D. Hausmann。独立系统贪心启发式的分析。载于离散数学年刊，第 2 卷，第 65–74 页。爱思唯尔，1978 年。[16] A. Krause 和 D. Golovin。次模函数最大化。可处理性，3：71–104，2014 年。[17] B. Lehmann、D. Lehmann 和 N. Nisan。具有边际效用递减的组合拍卖。《博弈与经济行为》，55(2):270–296，2006 年。[18]R. Levin 和 D. Wajc。流式子模匹配与原始-对偶方法相结合。载于*2021* 年 *ACM-SIAM* 离散算法研讨会（*SODA*）论文集，第 1914–1933 页。SIAM，2021 年。[19] H. Lin 和 J. Bilmes。用于文档摘要的一类子模函数。载于计算语言学协会第 *49* 届年会：人类语言技术论文集，第 510–520 页，2011 年。[20] Y. Liu、J. Miao、M. Zhang、S. Ma 和 L. Ru。用户如何描述其信息需求：基于摘要片段点击模型的查询推荐。《应用专家系统》，38(11): 13847–13856，2011 年。[21]A. Mehta。在线匹配与广告分配。理论计算机科学基础与趋势®，8(4):265–368，2013 年。[22] S. Mitrović、I. Bogunovic、A. Norouzi-Fard、J. Tarnawski 和 V. Cevher。流式鲁棒子模最大化：一种分区阈值法。*arXiv* 预印本 *arXiv:1711.02598*，2017 年。[23]J. Pennington、R. Socher 和 C. D. Manning。GloVe：用于词表示的全局向量。载于*2014* 年自然语言处理实证方法会议（*EMNLP*）论文集，第 1532–1543 页，2014 年。[24] M. L. Pinedo。调度，第 29 卷。Springer，2012 年。[25] B. Rozemberczki、C. Allen 和 R. Sarkar。多尺度属性节点嵌入，2019 年。[26] M. Streeter 和 D. Golovin。一种用于最大化子模函数的在线算法。载于第 *21* 届神经信息处理系统国际会议论文集，第 1577–1584 页，2008 年。[27] J. Vondrák。值预言机模型中子模福利问题的最优近似。载于第四十届 *ACM*计算理论研讨会论文集，第 67–74 页，2008 年。[28] G. Zhang、N. Tatti 和 A. Gionis。预算约束下基于子模函数的排序。《数据挖掘与知识发现》，36(3):1197–1218，2022 年。[29] Z. Zhang、E. K. Chong、A. Pezeshki、W. Moran 和 S. D. Howard。平衡二叉中继树中融合规则的子模性与最优性。载于*2012* 年 *IEEE* 第 *51* 届 *IEEE*决策与控制会议（*CDC*），第 3802–3807 页。IEEE，2012 年。

版权所有 © 2023 SIAM。未经授权，禁止复制本文。

附录 A

A.1 定理 4.1 的证明

证明。 [定理 4 的证明。1] 对于在线选择问题，输入是一个由 n 整数a_1，……，a_n组成的序列，这些整数依次揭示。问题要求在每个数揭示后立即且恰好选择一个数，目标是最大化所选数的值。众所周知，该问题不存在 $o(n)$ 近似算法 [13]。

我们观察到，在线选择问题是 MMR-F 问题的一个特例。为说明这一点，考虑以下实例：令全集 V 由一个物品 v 和 $n-1$ 个其他虚拟项组成。每个到达的模函数 f 都具有相同的形式，其中$f(v) = a$ 且 $f(u) = 0$，对于任意物品 $u \neq v$均如此。此外，每个函数 f 都关联一个预算 $k(f) = 1$。在第 t 个步骤，一个满足上述定义且具有 $f(v) = a_t$ 的函数到达。容易看出，MMR-F 的最优目标函数值等于最大的数 a_t。因此，确定物品 v 在 MMR-F 问题的输出序列中的排名，等价于确定在 $\{a_t\}$ 中选择哪个数来解决在线选择问题。因此，MMR-F 推广了在线选择问题。

□

A.2 定理 5.1 中缺失的证明

引理 A.1。 $\mathcal{A} \cap \mathcal{B}$ 是一个 p-拟阵。

证明。 由于 $\mathcal{M}$ 是定义在 V上的一个 p-拟阵，因此我们可以写成$\mathcal{M} = \mathcal{M}_1 \cap \cdots \cap \mathcal{M}_p$，其中 $\mathcal{M}_i$ 是一个拟阵。对于每个拟阵 $\mathcal{M}_i$，我们在 V'上构造另一个系统，

$$\mathcal{M}'_i = \{S' \subseteq V' : V(S') \in \mathcal{M}_i\} \cap \mathcal{B}$$
$$= \{S' \subseteq V' : V(S') \in \mathcal{M}_i, |S' \cap X_v| \leq 1, \text{ for all } v \in V\}.$$

注意，$\mathcal{A} = \bigcap_i \{S' \subseteq V' : V(S') \in \mathcal{M}_i\}$，且 $\mathcal{A} \cap \mathcal{B} = \bigcap_i \mathcal{M}'_i$。我们通过验证每个 $\mathcal{M}'_i$ 都是一个拟阵，证明 $\mathcal{A} \cap \mathcal{B}$ 是一个 p-拟阵。

要证明 $\mathcal{M}'_i$ 是拟阵，向下封闭性显然成立，我们只需验证增广性。任取满足 $|T| < |U|$ 的 $T, U \in \mathcal{M}'_i$，有$|V(T)| = |T| \leq |V(U)| = |U|$。由于 $V(T)$ $V(U) \in \mathcal{M}_i$，存在 $v \in V(U)$，使得 $v+V(T) \in \mathcal{M}_i$。假设对于该特定的 v，$(v,t) \in U$，显然 $(v,t) + T \in \mathcal{M}'_i$。因此，$\mathcal{M}'_i$ 是拟阵，这意味着$\mathcal{A} \cap \mathcal{B} = \bigcap_i \mathcal{M}'_i$ 是一个 p-拟阵。□

A.3 Twitter 上“特朗普”的同义词特朗普 班克斯 沃伦 克林顿 盖茨

纽曼 巴菲特 创始人 里根 卡森 首席执行官 任命 屠夫 达菲 卡尔森 洛威 traviscostello 加入 爱彼迎 公司 特斯拉 桑福德 克劳斯邓拉普 卡西迪 从事造船业 枪手 雇用 卢旺达 阿斯麦哈特曼 父系祖父 里格 交换 劳氏 瓦雷拉 拉蒙塔涅

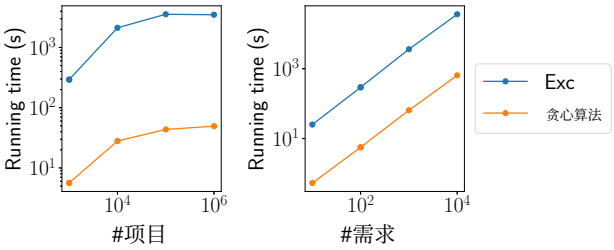


图3：运行时间

A.4 运行时间 为考察所提出算法的运行时间，我们生成了项目数或需求数逐渐增加的合成数据。具体而言，我们要么固定需求数为100，同时增加项目数；要么固定项目数为1000，同时增加需求数。每项需求由一个大小为100的随机子集的覆盖函数表示，并分配一个介于1到100之间的随机预算。对于MSR-F模型（Exc），我们为每项需求指定一个随机到达时间。

结果如图3所示。Exc 和贪心算法的运行时间都会随着需求数线性增长。随着项目数增加，两种算法的运行时间似乎呈次线性增长，而非线性增长，这是因为许多项目无法命中任何需求子集。

A.5 其他实验细节 所有实验均在一台服务器上进行，该服务器配备24个AMD皓龙(tm)处理器6172（2.1 GHz），62GB随机存取存储器，运行Linux 2.6.32-754.35.1.el6.x86_64。我们使用Python 3.8.5。

版权 © 2023 SIAM。未经授权，禁止复制本文。