arXiv:2305.01202v1 [cs.IR] 2 May 2023

安全在线学习重排序中未排序项目的探索

椎野宏明 CyberAgent, Inc. 日本 shiino_hiroaki@cyberagent.co.jp

Kaito Ariu CyberAgent, Inc. 日本 kaitoariu@gmail.com

阿部健志 CyberAgent, Inc. 日本 abe_kenshi@cyberagent.co.jp

Riku Togashi CyberAgent, Inc. 日本 rtogashi@acm.org

摘要

用于在线学习排序（OLTR）问题的老虎机算法通常旨在利用用户反馈来最大化长期收益。然而，从实践角度来看，这类算法由于探索过于激进，存在损害用户体验的高风险。因此，近年来对安全探索的需求日益增长。安全探索的一种方法是逐步提升原始排序的质量，而该排序已确保达到可接受的质量。本文提出一种安全的在线学习排序算法，通过将当前排序中的一个项目与排序之外的项目（即未排序项目）高效交换来进行探索。我们基于 Kullback-Leibler 上置信界（KL-UCB），以乐观策略选择一个未排序项目进行探索，并对包括所选项目在内的项目安全地重新排序。实验表明，所提出的算法在没有任何安全违规的情况下，相较于基线算法降低了长期遗憾。

CCS 概念

• **信息系统** → **学习排序**； • **计算方法** **在线学习设置**； **从隐式反馈中学习**。

关键词

安全性；在线学习排序；隐式反馈

ACM参考文献格式：

Hiroaki Shiino、Kaito Ariu、Kenshi Abe 和 Riku Togashi。2023。《安全在线重新排序学习中的未排序项目探索》。收录于第*46*届 *ACM* 信息检索研究与发展国际会议论文集（*SIGIR '23*），*2023*年*7*月*23*日至*27*日，中国台湾台北。ACM，纽约州纽约市，美国，5页。https://doi.org/10.1145/3539618.3591985

1 引言

学习排序（LTR）方法在为生活于信息过载时代的用户提供有吸引力的内容方面发挥着关键作用，在这个时代，每天都有新内容涌入数据库。系统必须快速、准确地向可能对此感兴趣的用户提供这些新内容。然而，我们经常缺乏有关新内容的信息，因此探索至关重要，尽管这不可避免地意味着在初始收集信息时进行某种程度上缺乏依据的预测，而这可能会损害用户满意度。因此，系统陷入了两难境地——探索是必要的，但其安全性也不可或缺。

在线学习排序（OLTR） [5, 19] 是一种有前景的方法，可通过预测—观察循环立即反映收集到的新鲜用户反馈，从而解决新项目的信息不足问题。若干传统研究利用物品特征 [15], 开发了 OLTR 方法，这种方法同样适用于处理新项目。另一方面，在实际应用中，新项目的特征可能并不可靠，尤其是在某个新项目意外走红时；在这种情况下，先前有效的特征将无法发挥良好作用。近期的 OLTR 研究探索了基于排序老虎机 [2, 8–12, 16, 21]的方法。尽管这种方法无需依赖物品特征即可学习对新项目进行排序，但通常需要假设一个点击模型，以便从有偏的点击反馈中学习项目的吸引力。然而，准确指定用户反馈背后的点击模型通常很有挑战性，因此，这种特定于模型的方法可能并不安全，因为模型设定错误会导致预测不准确，并损害用户满意度。近期研究提出了点击模型无关算法 [4, 13, 14, 20], ，这类算法在模型设定错误的情况下仍然安全。Li 等人 [14] 提出了一种点击模型无关方法，*BubbleRank*，它在 OLTR 场景中明确考虑了探索的安全性，在此场景下，算法可以利用此前已部署于生产系统的方法生成的原始排序。根据 Li 等人 [14], 的定义，BubbleRank 是“安全的”，因为它以较高概率保证呈现给用户的排序不会显著劣于原始排序。尽管如此，BubbleRank 基本上是为对原始排序中的项目重新排序而设计的，无法高效处理未出现在原始排序中的未排序项目；尽管文中讨论了通过随机探索未排序项目来扩展该方法，但如本文所示，这种朴素策略在统计上效率低下。

本文通过扩展 BubbleRank [14]，开发了一种能够安全探索未排序项目的在线学习排序算法。为了在没有任何先验信息的情况下对未排序项目进行安全探索，我们采用 Kullback-Leibler 上置信界

允许为个人或课堂使用而免费制作本作品部分或全部内容的数字版或纸质副本，前提是这些副本不得用于或分发以获取利润或商业利益，且副本须包含本声明及首页的完整引文。必须尊重本作品中第三方组件的版权。其他用途请联系版权所有者/作者。*SIGIR '23，2023*年*7*月*23*日至*27*日，中国台湾台北© 2023 版权所有者/作者保留所有权利。ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591985

[1]（KL-UCB）作为衡量项目吸引力的乐观置信度度量。为了检验所提出算法在各种场景下的有效性，我们在真实世界数据集上开展了半模拟实验。

2 相关工作

传统的在线学习排序算法可分为点击模型特定方法 [2, 8–12, 16, 21] 和点击模型无关方法 [4, 13, 14, 20]。模型特定算法假设用户反馈数据背后存在某种特定的点击模型，以便在用户遵循该假设的点击模型时高效地学习最优排序，例如基于位置的模型（PBM）[8, 12] [3, 和 10, 11, 21] 级联模型（比较模型）。然而，当假设的点击模型与实际用户行为不匹配时，其理论保证并不成立，因此可能存在不安全 性 [4, 14]。相比之下，点击模型无关方法只需满足较弱的假设。UniRank [4] 是当前最先进的模型无关算法，在遗憾方面表现出色，但未考虑安全约束。为实现不依赖点击模型的安全重排序，Li 等人[14] 提出了 BubbleRank，但该方法在处理未排序项目方面存在严重局限，因为其提出的随机探索没有考虑统计效率。

OLTR 中安全性的概念与保守型老虎机算法中的安全性相关。在保守型老虎机中，安全性被定义为对累积奖励的约束 [6, 18]；值得注意的是，只要在整个轮次过程中始终满足该约束，算法就可以在每一轮选择可能导致高遗憾的臂。除了这种“粗粒度”的定义外，近期一些研究还考虑了逐阶段安全性，要求算法在每一轮都保持保守 [7, 17]。这种逐阶段定义与本研究关注的安全性相当相关。遗憾的是，现有算法是为线性老虎机环境 [7, 17] 设计的，因此无法高效地应用于 OLTR 环境。

本文提出了一种受 BubbleRank 和 UniRank 启发的模型无关算法，可在未排序项目下实现逐阶段安全的重新排序与探索。

3 问题形式化

一个随机点击老虎机实例是一个元组 $(K,\ L,\ P_\alpha,\ P_\chi)$，其中 $L \in \mathbb{N}$ 是所有项目集合 D 的大小，P_α 是定义在二元吸引力向量 $\{0,\ 1\}^L$ 上的一个分布，而 P_χ 是定义在二元检查矩阵 $\{0,\ 1\}^{\Pi_K(\mathcal{D})\times K}$ 上的一个分布，其中 $\Pi_K(\mathrm{D})$ 是由 D 中 $K(\le L)$ 个项目构成的所有排列的集合。

对于 $n \in \mathbb{Z}_+$（即非负整数集合），令 $[n] := \{1, \dots, n\}$。在每一轮 $t \in [T]$，算法向用户展示一个排序 $\boldsymbol{\sigma}_t \in \Pi_K(\mathrm{D})$，并观察用户在 $\boldsymbol{\sigma}_t$ 中的所有位置上的点击 $\{c_t(k)\}_{k=1}^K \in \{0,1\}^K$。注意，$\boldsymbol{\sigma}_t$ 取决于截至第 $t-1$ 轮的过往历史。当且仅当某个位置被检查且该位置上的项目具有吸引力时，该位置才会被点击；也就是说，对于任意 $k \in [K]$，$c_t(k) = X_t(\boldsymbol{\sigma}_t, k) A_t(\boldsymbol{\sigma}_t(k))$，其中 $X_t(\boldsymbol{\sigma}, k) \in \{0,1\}$ 是第 t 轮排序 $\boldsymbol{\sigma} \in \Pi_K(\mathrm{D})$ 中位置 k 的检查指示变量；$A_t(\boldsymbol{\sigma}(k)) \in \{0,1\}$ 是第 t 轮排序 $\boldsymbol{\sigma} \in \Pi_K(\mathrm{D})$ 中位置 k 上的项目 $\boldsymbol{\sigma}(k)$ 的吸引指示变量。为简化记号，对于 $k > K$，令 $X_t(\boldsymbol{\sigma}, k) = 0$，且 $c_t(k) = 0$。$\{X_t(\boldsymbol{\sigma}, k) :\ \boldsymbol{\sigma} \in \Pi_K(\mathrm{D})\ k \in [K]\}$ 和 $\{A_t(i) :\ i \in [L]\}$ 均为随机的，且从 $P_\chi \bigotimes P_\alpha$ 中独立同分布地抽取。

给定排序 $\boldsymbol{\sigma}$ 时，算法在第 t 轮的期望奖励是期望点击次数之和：

$$r(\boldsymbol{\sigma}, \alpha, \chi(\boldsymbol{\sigma})) := \sum_{k=1}^{L} \mathbb{E}[c_t(k) | \boldsymbol{\sigma}_t = \boldsymbol{\sigma}] = \sum_{k=1}^{L} \chi(\boldsymbol{\sigma}, k) \alpha(\boldsymbol{\sigma}(k)),$$

其中，$\chi(\boldsymbol{\sigma}, k) = \mathbb{E}[X_t(\boldsymbol{\sigma}_t, k) | \boldsymbol{\sigma}_t = \boldsymbol{\sigma}]$ 是位置 k 在 $\boldsymbol{\sigma}$ 中的检查概率，$\alpha(i) = \mathbb{E}[A_t(i)]$ 是项目 i 的吸引概率，$\chi(\boldsymbol{\sigma}) = \{\chi(\boldsymbol{\sigma}, k)\}_{k=1}^K$，且 $\alpha = \{\alpha(i)\}_{i=1}^L$。因此，该问题的目标是最小化累积期望遗憾

$$R(T) = T \cdot r^* - \mathbb{E}\left[\sum_{t=1}^{T} r(\boldsymbol{\sigma}_t, \alpha, \chi(\boldsymbol{\sigma}_t))\right], \tag{1}$$

其中 $r^* := \max_{\boldsymbol{\sigma} \in \Pi_K(\mathcal{D})} r(\boldsymbol{\sigma}, \alpha, \chi(\boldsymbol{\sigma}))$ 是最高期望奖励，期望是针对算法生成的排序和点击而言的。

我们的问题需要采用 Li 等人 [14] 引入的假设。这些假设在比较模型中成立；当检查概率随位置增加而降低时，在 PBM 中也成立。那么，最优排序 $\boldsymbol{\sigma}^* := \mathrm{argmax}_{\boldsymbol{\sigma} \in \Pi_K}$（D） $r(\boldsymbol{\sigma}, \alpha, \chi(\boldsymbol{\sigma}))$ 会在位置 k 包含吸引力排名第 k 高的项目。

4 提出的方法

我们的算法称为 KL-UCB-BR，详见算法 1。KL-UCB-BR 在每个轮次 t 中维护以下三种排序：主 排序 $\boldsymbol{\sigma}_{\mathrm{LDR}t}$、临时排序 $\boldsymbol{\sigma}_{\prime t}$ 和展示排序 $\boldsymbol{\sigma}_t$。主排序 $\boldsymbol{\sigma}_{\mathrm{LDR}t}$ 是截至轮次 t 估计出的中间最优排序；我们将其初始化为原始排序 $\boldsymbol{\sigma}_0$（如果 $t = 1$），若 $t > 1$，则初始化为上一临时排序 $\boldsymbol{\sigma}_{\prime t-1}$ 的前 K 项部分排序（$[K]$）。临时排序 $\boldsymbol{\sigma}_{\prime t}$ 用于交换项目，以便安全地重新排列项目对；我们将其初始化为当前主排序 $\boldsymbol{\sigma}_{\mathrm{LDR}t}$，设置一个未排序项目在第 $(K+1)$ 个位置进行探索，并交换连续位置上的项目以确保顺序正确。展示排序 $\boldsymbol{\sigma}_t$ 呈现给用户，且可以观察用户对其中项目的点击；我们将其初始化为当前临时排序 $\boldsymbol{\sigma}_{\prime t}$，若 KL-UCB-BR 无法确定它们吸引力的顺序，则通过随机交换连续位置上的项目来更新展示排序。

作为安全地重新排序项目对以确保正确性的标准，我们采用临时排序 [14] 中交换项目时的以下统计量：

$$s_t(i,j) := \sum_{s=1}^{t-1} O_s(i,j)(c_s(\boldsymbol{\sigma}_s^{-1}(i)) - c_s(\boldsymbol{\sigma}_s^{-1}(j))),$$
$$n_t(i,j) := \sum_{s=1}^{t-1} O_s(i,j),$$

其中，$\boldsymbol{\sigma}^{-1}(i)$ 是项目 i 在排序 $\boldsymbol{\sigma}$ 中的位置，$O_s(i,j) := 1\{(i,j) \in \mathrm{P}_s(\boldsymbol{\sigma}_s)\}\, 1\{c_s(\boldsymbol{\sigma}_s^{-1}(i)) \neq c_s(\boldsymbol{\sigma}_s^{-1}(j))\}$，其中 $\mathrm{P}_s(\boldsymbol{\sigma})$ 是排序 $\boldsymbol{\sigma}$ 中偶数/奇数轮次里处于奇偶/偶奇连续位置的项目对集合。根据 Li 等人的引理~~9 [14]，当 $s_t(i,j) > \sqrt{n_t(i,j)\log(1/\delta)}$~~ 且 δ 足够小时，项目 i 以高概率优于项目 j。

我们选择一个未排序项目进行探索，在第 t 轮次中根据以下相对于统计数据的乐观索引进行探索

算法 1KL-UCB-BR

输入： 项目 $[L]$，初始列表 $\sigma_0 \in [L]^K$, $\delta \in (0,1)$
  $s_0(i,j) \leftarrow 0, n_0(i,j) \leftarrow 0, \forall (i,j) \in [L]^2$
  $\sigma_1^{\mathrm{LDR}} \leftarrow \sigma_0$
对于 $t = 1, \ldots, T$ **do**
    $h \leftarrow t$ 模 2
    $\sigma'_t \leftarrow \sigma_t^{\mathrm{LDR}}$
    $\sigma' t(K+1)$ =取最大值$_{j \in [L] \setminus \sigma_t^{\mathrm{LDR}}} \bar{\bar{s}}_t(j, \sigma_t^{\mathrm{LDR}}(K))$
    $\sigma_t \leftarrow \sigma' t$
    对于 $k = 1, \ldots, \lceil (K-h)/2 \rceil$ **执行**
      $i \leftarrow \sigma_t(2k-1+h)$, $j \leftarrow \sigma_t(2k+h)$
      如果 $s_{t-1}(i,j) \le 2\sqrt{n_{t-1}(i,j)\log(1/\delta)}$ **那么**
        随机交换项目 $\sigma_t(2k-1+h)$
        以及 $\sigma_t(2k+h)$，在列表 σ_t 中
    显示 $\sigma_t([K])$ 并观察点击 $\{c_t(k)\}_{k=1}^K \in \{0,1\}^K$
    $s_t \leftarrow s_{t-1}, n_t \leftarrow n_{t-1}$
    对于 $k = 1, \ldots, \lceil (K-h)/2 \rceil$ **执行**
      $i \leftarrow \sigma_t(2k-1+h)$, $j \leftarrow \sigma_t(2k+h)$
      如果 $|c_t(2k-1+h) - c_t(2k+h)| = 1$，**则**
        $s_t(i,j) \leftarrow s_t(i,j) + c_t(2k-1+h) - c_t(2k+h)$
        $n_t(i,j) \leftarrow n_t(i,j) + 1$
        $s_t(j,i) \leftarrow s_t(j,i) - c_t(2k-1+h) + c_t(2k+h)$
        $n_t(j,i) \leftarrow n_t(j,i) + 1$
    对于 $k = 1, \ldots, K$ **执行**
      $i \leftarrow \sigma' t(k)$, $j \leftarrow \sigma' t(k+1)$
      如果 $s_t(j,i) > 2\sqrt{n_t(j,i)\log(1/\delta)}$ **那么**
        在列表 $\sigma' t$ 中交换项目 $\sigma' t(k)$ 和 $\sigma' t(k+1)$
    $\sigma_{t+1}^{\mathrm{LDR}} \leftarrow \sigma' t([K])$

$s_t(i,j)/n_t(i,j)$，对于一个未排序项目 i 和主排序底部的项目 jLDR σ，如下所示 [4]:

$$\bar{\bar{s}}_t(i,j) := 2 \times f\left(\frac{1 + s_t(i,j)/n_t(i,j)}{2}, n_t(i,j), \tilde{t}_{\sigma_t^{\mathrm{LDR}}}(t)\right) - 1,$$

其中，$\tilde{t}_\sigma(t)$ 是 σ 在第 t 轮次中曾作为主排序的轮次数，f 是统计量 $\hat{\ }\mu f \mu N t = \mu \in [\mu\ 1]\ N \times kl\ \mu\ \mu \le t +$ 的 KL-UCB 指数 [1]：$^{3}{}_{t}{}^{kl}(\hat{\ },\hat{\ }_{,})$: $\sup\{\hat{\ },: (\hat{\ },) \log(\,) \log(\log(\,))\}$，其中 $(\cdot,\cdot)$ 表示均值分别为对应值的两个伯努利分布之间的 Kullback-Leibler（KL）散度；当 $t = 0$、$N = 0$ 或 $\hat{\ }\mu = 1$ 时，$f(\hat{\ }\mu, N, t) = 1$。注意，当 $n_t(i,j) = 0$ 时，$\bar{\bar{s}}_t(i,j) = 1$。然后，我们可以高效地探索一个有潜力的未排序项目，它可能优于当前排序中位于末尾的项目。

5 实验

5.1 设置

我们将 KL-UCB-BR 与安全感知和安全无关的基线方法进行比较：BubbleRank [14]，、TopRank [13], 和 UniRank [4]，。这些方法均与点击模型无关，且不利用任何物品特征。为了考察原始排序带来的性能提升，我们还将原始排序作为一种非自适应基线方法；下文称之为 OriginalRank。由于我们关注的是包含未排序项目的在线学习排序问题，因此采用 Li 等人 [14], 描述的 BubbleRank 扩展版本，其中当前排序中位于末尾的项目会与一个随机的未排序项目随机交换，而该项目尚未被确定优于排名末尾的项目。

在我们的实验中，我们使用 *Yandex* 点击数据集 [1] 来模拟用户对算法生成的展示排序的点击。该数据集包含超过 3000 万个用户会话，其中包括从 Yandex 搜索日志中提取的超过 2000 万个不同搜索查询。我们基本遵循传统研究的实验方案 [14, 20]。为了模拟数据集中的用户行为，我们使用 PyClick 库 [2]，根据最常见的前 100 个查询对应的用户会话估计点击模型参数。在整个实验中，我们考察 PyClick 中实现的两种点击模型：基于位置的点击模型（PBM）和级联点击模型（比较模型）。在一次模拟的用户会话中，我们根据学得的点击模型生成用户对算法展示的排序的点击，然后评估该算法。

对于每个查询，我们使用包含 10 个项目的最常见排序，并将排序中的前 5 个项目视为原始排序，其余 5 个项目视为未排序项目。针对每个查询，模拟的目标是在这 10 个已排序/未排序项目中，按吸引力从高到低对最有吸引力的 5 个项目进行排序。我们通过计算式（1）中定义的累积期望遗憾以及展示排序的安全违规情况，来衡量每种算法的性能。安全违规是指算法违反定义 5.1 中安全约束 [3] 的轮次数。违反安全约束往往会导致用户流失，因为应用向用户展示的排序远逊于原始排序。值得注意的是，KL-UCB-BR 和 BubbleRank 可以保证以高概率在第 T 轮之前不违反安全约束。

定义 5.1（安全约束）。设 $\sigma^* = \{1, 2, \ldots, K\}$ 为最优排序；$\alpha(1) \ge \alpha(2) \ge \ldots \ge \alpha(L)$；并且

$$V(\sigma) := |\{(i,j) \in [L]^2 : i < j, \sigma^{-1}(j) < \sigma^{-1}(i), \sigma^{-1}(j) \le K\}|$$

表示在展示排序 σ 中，一个或两个项目都包含在内的错误排序项目对的数量。那么，轮次 t 中展示排序 σ_t 的安全约束为 $V(\sigma_t) \le V(\sigma_0) + L - K/2$，其中 σ_0 是原始排序。

5.2 结果

在实验中，我们在 PBM 和比较模型（CM）点击模型下，将 KL-UCB-BR 与 TopRank、Uni-Rank、BubbleRank 和 OriginalRank 进行比较。作为评估指标，我们通过对 100 次重复实验所得测量结果取平均值，计算式 (1) 中定义的累积期望遗憾，以及定义 5.1 中定义的安全违规；每次实验包含 $T = 10^5$ 轮次。阴影区域表示测量结果的标准误差。

图 1 展示了全部 100 个查询的结果。这里，各项评估指标是对总计 10,000 次运行取平均值得出的，其中每个查询进行了 100 次实验。我们可以看到，在上方两幅图中，在 PBM 和比较模型下，各种自适应算法的累积期望遗憾都比 OriginalRank（灰色）增长得更慢。这表明，对于大多数查询，原始排序仍有改进空间

[1] https://www.kaggle.com/c/yandex-personalized-web-search-challenge [2] https://github.com/markovi/PyClick [3] 我们的安全约束是在 Li 等人提出的安全约束基础上扩展而来，[14] 用于处理未排序项目。

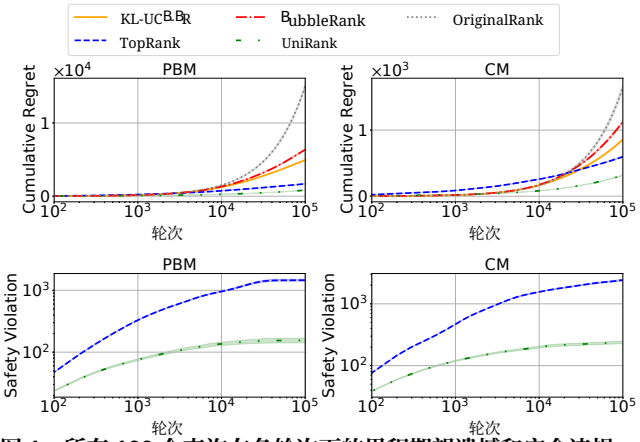

图 1：所有 100 个查询在各轮次下的累积期望遗憾和安全违规。

。上方两幅图还显示，在 PBM 和比较模型下，TopRank（蓝色）和 UniRank（绿色）的累积期望遗憾低于 BubbleRank（红色）和 KL-UCB-BR（橙色），尤其是在后期轮次；而下方两幅图则显示，由于采用无安全感知探索，它们在 PBM 和比较模型下违反了安全性。这意味着，这些算法在早期阶段的不安全行为可能导致用户流失，而且它们在长期遗憾方面的优势在实践中未必显著。另一方面，KL-UCB-BR 和 BubbleRank 通过逐步更新原始排序，没有发生安全违规，因此图中看不到它们的曲线。在安全算法中，KL-UCB-BR 的累积期望遗憾尤其最低，这表明它比 BubbleRank 更有效地探索未排序项目。

其次，我们报告每个特征查询的结果。图 2 展示了原始排序为最优的查询结果；也就是说，吸引力最高的 5 个项目按吸引力降序排列。请注意，在这种情况下，当算法探索已排序和未排序项目时，仍会产生遗憾。因此，在上方两幅图的所有轮次中，OriginalRank（灰色）都没有遗憾。与 TopRank（蓝色）和 UniRank（绿色）相比，KL-UCB-BR（橙色）和 BubbleRank（红色）在 PBM 和比较模型下的累积期望遗憾较低，因为它们会在安全约束下，根据原始排序更新展示排序（见下方两幅图），从而避免造成破坏性遗憾。在安全算法中，与 BubbleRank 相比，KL-UCB-BR 在后期轮次的累积期望遗憾更低，因为它能够减少由吸引力较低的未排序项目造成的损失。图 3 展示了原始排序未包含吸引力最高的 5 个项目时的查询结果。上方两幅图显示，在 PBM 和比较模型下，KL-UCB-BR 和 BubbleRank 的累积期望遗憾高于 TopRank 和 UniRank。这是因为安全感知方法避免大幅更新原始排序，因此会因原始排序未达到最优而遭受遗憾。尽管如此，我们仍可以利用

在实际应用中，安全感知方法在一定程度上改善了从 OriginalRank 开始的累积期望遗憾。特别是，就累积期望遗憾而言，KL-UCB-BR 优于 BubbleRank；KL-UCB-BR 的性能提升

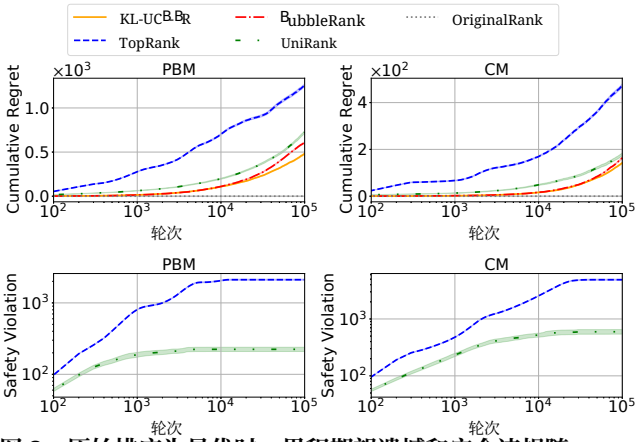

图 2：原始排序为最优时，累积期望遗憾和安全违规随 $T = 10^5$ 轮次的变化。

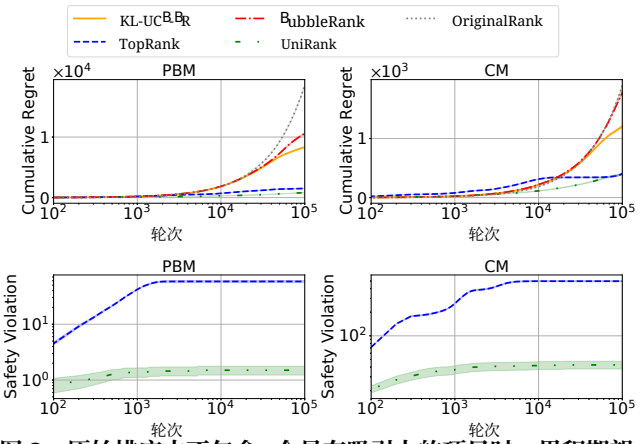

图 3：原始排序中不包含5 个最有吸引力的项目时，累积期望遗憾和安全违规随 $T = 10^5$ 轮次的变化。

在后期轮次中尤为明显。同样值得注意的是，在比较模型（CM）下，KL-UCB-BR 相较于 OriginalRank 有显著改进，而 BubbleRank 未能降低 OriginalRank 所产生的遗憾。这些结果表明，基于 KL-UCB 的探索能够从未排序项目中找到最具吸引力的 5 个项目中的一些，而随机探索则无法找到它们，因为在比较模型（CM）下，即使在展示排名中错误地排列项目，也不会产生遗憾。

6 结论

我们提出 KL-UCB-BR，这是一种安全的在线学习排序算法，能够在不违反安全约束的情况下探索已排序和未排序项目。与采用随机探索的原始 BubbleRank 相比，KL-UCB-BR 利用 KL-UCB 指数高效探索未排序项目 [14]。我们通过实验证明，在各种场景下，KL-UCB-BR 的表现优于 BubbleRank，且不会造成任何安全违规。

参考文献

[1] Olivier Cappé、Aurélien Garivier、Odalric-Ambrym Maillard、Rémi Munos 和 Gilles Stoltz。2013。用于最优序贯分配的 Kullback-Leibler 上置信界。统计年鉴41，3（2013），1516–1541。[2] Richard Combes、Stefan Magureanu、Alexandre Proutiere 和 Cyrille Laroche。2015。学习排序：遗憾下界与高效算法。载于*2015 年 ACM SIGMETRICS 计算机系统测量与建模国际会议论文集*。231–244。[3] Nick Craswell、Onno Zoeter、Michael Taylor 和 Bill Ramsey。2008。点击位置偏差模型的实验比较。载于*2008 年网络搜索与数据挖掘国际会议论文集*。87–94。[4] Camille-Sovanneary Gauthier、Romaric Gaudel 和 Elisa Fromont。2022。Unirank：用于在线排序的单峰老虎机算法。载于国际机器学习大会。PMLR，7279–7309。[5] Katja Hofmann、Shimon Whiteson 和 Maarten de Rijke。2013。在信息检索的列表式和成对式在线学习排序中平衡探索与利用。信息检索16（2013），63–90。[6] Abbas Kazerouni、Mohammad Ghavamzadeh、Yasin Abbasi Yadkori 和 Benjamin Van Roy。2017。保守的上下文线性老虎机。神经信息处理系统进展30（2017）。[7] Kia Khezeli 和 Eilyan Bitar。2020。安全线性随机老虎机。载于*AAAI 人工智能大会论文集*，第 34 卷。10202–10209。[8] Junpei Komiyama、Junya Honda 和 Akiko Takeda。2017。位置偏差未知的位置型多重选择老虎机问题。载于神经信息处理系统进展，第 30 卷。[9] Branislav Kveton、Ofer Meshi、Masrour Zoghi 和 Zhen Qin。2022。在线学习排序中先验的价值。载于人工智能与统计国际会议。PMLR，6880–6892。[10] Branislav Kveton、Csaba Szepesvari、Zheng Wen 和 Azin Ashkan。2015。级联老虎机：在级联模型中学习排序。载于国际机器学习大会。PMLR，767–776。[11] Branislav Kveton、Zheng Wen、Azin Ashkan 和 Csaba Szepesvari。2015。组合级联老虎机。神经信息处理系统进展28（2015）。[12] Paul Lagrée、Claire Vernade 和 Olivier Cappe。2016。基于位置模型的多重选择老虎机。神经信息处理系统进展29（2016）。[13] Tor Lattimore、Branislav Kveton、Shuai Li 和 Csaba Szepesvari。2018。Toprank：一种实用的在线随机排序算法。神经信息处理系统进展31（2018）。[14] Chang Li、Branislav Kveton、Tor Lattimore、Ilya Markov、Maarten de Rijke、Csaba Szepesvári 和 Masrour Zoghi。2020。BubbleRank：利用隐式点击反馈进行安全的在线重排序学习。载于人工智能的不确定性。PMLR，196–206。[15] Shuai Li、Tor Lattimore 和 Csaba Szepesvári。2019。基于特征的在线排序学习。载于国际机器学习大会。PMLR，3856–3865。[16] Stefan Magureanu、Alexandre Proutiere、Marcus Isaksson 和 Boxun Zhang。2017。在线学习最优多样化排序。*ACM 测量、分析与计算系统论文集*1，2（2017）。[17] Ahmadreza Moradipari、Christos Thrampoulidis 和 Mahnoosh Alizadeh。2020。分阶段保守线性老虎机。神经信息处理系统进展33（2020），11191–11201。[18] Yifan Wu、Roshan Shariff、Tor Lattimore 和 Csaba Szepesvári。2016。保守老虎机。载于国际机器学习大会。PMLR，1254–1262。[19] Yisong Yue 和 Thorsten Joachims。2009。将信息检索系统的交互式优化作为双人对抗老虎机问题。载于第*26*届国际机器学习年会论文集。1201–1208。[20] Masrour Zoghi、Tomas Tunys、Mohammad Ghavamzadeh、Branislav Kveton、Csaba Szepesvari 和 Zheng Wen。2017。随机点击模型中的在线排序学习。载于国际机器学习大会。PMLR，4199–4208。[21] Shi Zong、Hao Ni、Kenny Sung、Nan Rosemary Ke、Zheng Wen 和 Branislav Kveton。2016。用于大规模推荐问题的级联老虎机。*arXiv 预印本 arXiv:1603.05359*（2016）。