PPAR: A Privacy-Preserving Adaptive Ranking Algorithm for Multi-Armed-Bandit Crowdsourcing

Shuzhen Chen⋆, Dongxiao Yu⋆, Feng Li⋆, Zongrui Zou⋆, Weifa Liang† and Xiuzhen Cheng⋆
⋆School of Computer Science and Technology, Shandong University, China
Email:{szchen, zou.zongrui}@mail.sdu.edu.cn, {dxyu, fli, xzcheng}@sdu.edu.cn
†School of Computer Science, City University of Hong Kong, Hong Kong
Email:weifa.liang@cityu.edu.hk

2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS) | 978-1-6654-6824-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWQoS54832.2022.9812914

***Abstract*—This paper studies the privacy-preserving adaptive ranking problem for multi-armed-bandit crowdsourcing, where according to the crowdsourced data, the arms are required to be ranked with a tunable granularity by the untrustworthy third-party platform. Any online worker can provide its data by arm pulls but requires its privacy preserved, which will increase the ranking cost greatly. To improve the quality of the ranking service, we propose a Privacy-Preserving Adaptive Ranking algorithm called PPAR, which can solve the problem with a high probability while differential privacy can be ensured. The total cost of the proposed algorithm is $\mathcal{O}(K \ln K)$, which is near optimal compared with the trivial lower bound $\Omega(K)$, where K is the number of arms. Our proposed algorithm can also be used to solve the well-studied fully ranking problem and the best arm identification problem, by proper setting the granularity parameter. For the fully ranking problem, PPAR attains the same order of computation complexity with the best-known results without privacy preservation. The efficacy of our algorithm is also verified by extensive experiments on public datasets.**

***Index Terms*—Adaptive ranking for QoS, multi-armed bandits, crowdsourcing, differential privacy**

I. INTRODUCTION

Ranking is one of key applications of crowdsourcing which is an emerging paradigm of making full use of available workers to perform target tasks. For example, given a set of unknown target items (e.g., online courses [1], books and movies [2], etc.), a crowdsourcing platform (broker) could recruit a group of workers to "sample" the qualities of these (unknown) items (e.g., the service or food qualities of the restaurants, the relevance of the documents to the given baseline topic and the perceptual qualities of the images). The true qualities are then learned based on the crowdsourced samples, according to which, the items can be ranked, e.g., for the recommendation purpose.

Although *Multi-Armed Bandits* (MAB) mechanism has been applied to leverage the trade-off between exploitation and exploration in making sampling decisions, by letting items correspond to arms [3], it may not always be necessitated to achieve a full ranking result such that each of the items is categorized with a unique rank. For example, some of movies (or books) have very similar qualities with each other such that i) distinguishing them may require a large number of samples, which thus results in considerable cost for recruiting workers, and ii) a coarse ranking is still able to serve the recommendation purpose reasonably. To improve the quality of ranking service, we are interested in ranking items in a flexible manner, possibly with a tunable granularity, so that the arms in the same rank have no big differences in terms of quality. In other words, for each rank, the quality difference between its best arm and its worst item is in a *desired range*. Some recent proposals, e.g., [4], [5], investigated the coarse ranking problem, but their algorithms rank the items by specifying the maximum number of arms in each rank which cannot serve our goal. *How to achieve an adaptive range-based ranking thus is still an open problem.*

Additionally, due to the proliferation of the crowdsourcing paradigm in growing numbers of applications, a crowdsourcing system has to be of scalability to handle diverse tasks. Hence, it usually sends its crowdsourced data to a powerful third-party platform (e.g., cloud computing platforms) for data storage and processing [6], [7], whereas the third-party platform may not be trusted and the crowdsourcing service provider (broker) is usually not fully trusted. Since the data provided by the crowd of workers may contain or imply the workers' private information, another crucial concern is the privacy preservation issue. Hence, the crowdsourced data has to be perturbed by injecting noise to protect the privacy of workers, which however may greatly increase the cost to obtain an acceptable ranking. Extensive studies have been conducted to strive for the trade-off between privacy preservation and sampling cost, such as [8]–[11], but these results just focus on preventing the privacy leakage induced by arm pulls for identifying best arms. It is unknown how to keep privacy protected in solving the adaptive ranking problem when we still require a minimized sampling cost.

In this work, we propose a *Privacy-Preserving Adaptive Ranking* (PPAR) algorithm for MAB crowdsourcing. With granularity parameter α specifying the maximum difference between the quality of the best arm and the worst arm in each rank category, the algorithm can achieve an adaptive ranking with minimizing the number of required data samples

This work is partially supported by NSFC (No. 62122042, 61971269, 620722278) and Shandong University multidisciplinary research and innovation team of young scholars (No. 2020QNQT017).

Corresponding authors: Dongxiao Yu; Feng Li

978-1-6654-6824-4/22/$31.00 ©2022 IEEE

(and thus the induced cost to recruit workers). Furthermore, the PPAR algorithm leverages the notion of *differential privacy* [12], [13] to ensure that sharing the data with an untrusted third-party platform would not result in privacy leakage. Specifically, we propose to adapt a hybrid mechanism [14] that combines the logarithmic mechanism with the binary mechanism to perturb the data delivered to an untrusted broker and a third-party platform. We perform rigorous theoretical analysis to quantify the trade-off between the cost for ranking and the budget for privacy preservation, given the specified ranking granularity. We also perform extensive experiments on a public dataset so as to verify the efficacy of the PPAR algorithm. Our main contributions can be summarized as follows:

- We investigate an adaptive ranking problem under an MAB framework, with the aim to rank arms (corresponding to the items) with a range-based granularity.
- We propose the PPAR algorithm such that the arms can be adaptively ranked while the required cost for pulling the arms (or recruiting the workers to sample the arms' quality distribution) is minimized and the privacy preservation guaranteed.
- We reveal the trade-off between the sampling cost and the privacy budget through theoretical analysis. We also evaluate the PPAR algorithm by performing extensive experiments on a public dataset.

It deserves to note that when setting the granularity parameter $\alpha = 0$, our proposed algorithm can also be used to solve the fully ranking problem [3] and the best arm identification problem [15]–[17]. For the full ranking problem, our algorithm attains the same order of cost with the best known result [3], [18] without privacy preservation.

The remainder of this paper is organized as follows. We first summarize the related work in Sec. II. We then introduce the system model and present our privacy preserving adaptive ranking algorithm in Sec. III and Sec. IV, respectively. The performance of our proposed algorithm is theoretically analyzed in Sec. V. We also perform extensive experiments to verify the efficacy of our algorithm in Sec. VI. We finally conclude this paper in Sec. VII.

II. Related Work

In recent years, crowdsourcing has gradually evolved into a distributed problem-solving model. Human intelligence is pooled on crowdsourcing platforms to solve many problems that computers alone cannot solve. The crowdsourcing is applied in many application scenarios, e.g., information systems [19], voting systems [20], and monitoring systems [21]. In [22], Yuen et al. proposed a common setting about crowdsourcing systems. Crowdsourcing shows a pre-submitted task list given by a requester, and each arriving online worker selects a task to complete, submitting his response. At the same time, online workers can receive relevant rewards. Crowdsourcing provides a platform for tapping into the wisdom of the crowd to explore new data, which is then exploited to solve problems. This is consistent with the essence of exploration and exploitation in the classical Multi-Armed Bandits (MAB) problem [23]–[26]. In our work, we make use of the traditional crowdsourcing platform for efficient data collection to complete the adaptive arm ranking.

The ranking problem is extensively studied in the multi-armed bandits setting, e.g., fully ranking problem [3], best arm identification [15]–[17], $top\text{-}K$ arm identification [27]–[29]. As mentioned, the adaptive ranking problem in this paper is a generalization of the above three problems in the multi-armed bandits setting. Conveniently, these three problems can be well solved by adjusting the rank span input by our designed algorithm. In many practical applications, the adaptive ranking problem naturally arises, including peer-evaluation in large-scale, online public courses (MOOC) [1], recommendation systems for books and movies [2]. In these applications, entities (courses, books, movies, etc.) are divided into different credits/ratings. Katariya et al. studied the MaxGap Bandit problem in [3], focusing on achieving approximate ranking. However, it was a centralized implementation. In our work, we take the advantage of distributively problem solving crowdsourcing to achieve the adaptive ranking. The coarse ranking in the multi-armed bandits model was studied in [4], [5]. In [4], Katariya et al. considered the active coarse ranking. The goal was to rank items based on their mean values into predetermined sizes through adaptively sampling based on their reward distributions. In [5], Karpov et al. initiated the study of coarse ranking in a batched model under fixed confidence and fixed budget variants, respectively. But the coarse ranking considered in [4], [5] was from pairwise comparisons in the multi-armed bandits setting. Thus, they are not applicable to our setting where only the reward about the arm can be observed.

In order to rank similar products, we need to collect a large number of online workers' evaluation data in our framework, where the evaluation data can leak some private information. A lot of work has recently been done to address privacy leakage in the different multi-armed bandits setting, e.g., federated bandits [8], collaborative bandits [9], combinatorial semi-bandits [10] and contextual linear bandits [11]. To solve the privacy issue in our adaptive ranking problem, we design the privacy-preserving algorithm under the requirement of Differential Privacy. Differential Privacy [12], [13] is a rigorous definition and standard for quantifying privacy loss. In our work, we modify the Hybrid Mechanism [14] to achieve ϵ-differential privacy. The noise added is logarithmic in the number of rewards, such that the deviation from the real value does not affect the efficiency of classification significantly. As far as we know, we propose the first privacy-preserving adaptive arm ranking algorithm with a small required total cost with crowdsourcing.

III. System Model and Problem

A. System Overview

Consider a crowdsourcing system consisting of a platform (broker) and online workers. A requester wants to have a knowledge of the ranks of some items with limited cost. These

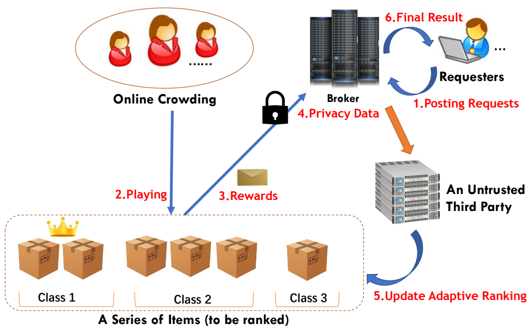


Fig. 1. Illustration of adaptive ranking via crowdsourcing.

items can be referred to a series of products that need to be evaluated before launching. The requester hands over the items to be ranked to a crowdsourcing platform and proposes the span of the desired adaptive ranking. Then all the items are publicized to online workers by the crowdsourcing platform. The arriving worker will choose one product to experience and generate a corresponding score. For each worker, he can get paid for each response. Over time, the platform can collect a large amount of information about the items. The crowdsourcing platform provides privacy protection for workers. The crowdsourcing platform will not obtain the raw score data reported by workers directly, thus lessening workers' distrust of the platform. On the whole, the crowdsourcing platform takes advantage of data collection in a distributed manner, then hands the preserved data to a third-party with powerful computing and storage capabilities in a steady stream. The adaptive ranking request can be efficiently implemented by the third-party while the detailed information about workers is protected. The main process is illustrated in Fig. 1.

For the sake of generality, each item can be evaluated by more than one worker, and each worker can also choose one from multiple items to experience at each time. We assume that the online workers are unknown and each worker has no knowledge of the items. Because workers arrive online at different time, the crowdsourcing platform should release the scoring protocol in advance. With the crowdsourcing platform, the adaptive ranking task for requesters is aimed to be solved. The items are finally expected to be ranked into the same rank result according to descending qualities, and the quality gap of items belonging to the same rank is upper bounded by the adaptive ranking span that is provided by the requester.

B. Model

Let $\mathcal{K} = \{A_1, A_2, \ldots, A_K\}$ denote K items. For each item A_i, its quality obeys a probability distribution Q_i and has an unknown expectation, denoted by β_i. Without loss of generality, we assume that the qualities are normalized into a range of $[0, 1]$. We also suppose that time can be divided into a sequence of epochs $t = 1, 2, \cdots$ and a new epoch starts if there is an available worker arriving at the crowdsourcing system. For each worker, we assign one of the target items to the worker, and the worker then "senses" the quality of the assigned item and reports the protected results to the crowdsourcing system. We denote by $\omega_{i,t} \in [0, 1]$ the quality of A_i in epoch t. $\omega_{i,t}$ is an i.i.d. random variable for fixed i and the mean of $\omega_{i,t}$ is β_i for $\forall A_i \in \mathcal{K}$. For example, if a worker arriving at epoch t is assigned with A_i, he/she could collect a sample $\omega_{i,t}$ independently from the probability distribution Q_i and then generates an information token $(A_i, \omega_{i,t})$. The tokens are protected by the encapsulated program provided by the crowdsourcing platform. The crowdsourcing platform then acts as a broker by submitting the protected data to the third-party.

C. Multi-Armed Bandit Adaptive Ranking Problem

The adaptive ranking problem can be cast to a *Multi-Armed Bandits Adaptive Ranking* (MABAR) problem where items correspond to arms. We receive a reward feedback $\omega_{i,t}$ and have to pay a unit of cost when having a worker "pulling" arm A_i in epoch t. Our aim is to categorize the arms into a set of classes under a granularity of α according to the reward feedbacks. We formally define the MABAR problem as follows.

Definition 3.1 (Multi-Armed Bandit Adaptive Ranking): Let $\mathcal{P}(\mathcal{K})$ denote the power set of $\mathcal{K}$. The α-ranking of $\mathcal{K}$ is defined as $\mathcal{C} \subseteq \mathcal{P}(\mathcal{K})$ (with $L = |\mathcal{C}|$ and $C_l \in \mathcal{C}$) such that

(a) For $\forall l = 1, 2, \cdots, L$, $C_l \neq \emptyset$;
(b) For $\forall C_l, C_{l'} \in \mathcal{C}$ with $l \neq l'$, $C_l \cap C_{l'} = \emptyset$;
(c) $\bigcup_{1 \leq l \leq L} C_l = \mathcal{K}$;
(d) For any $1 \leq l \leq L-1$, it holds that $\max\{\beta_i | A_i \in C_l\} - \min\{\beta_i | A_i \in C_l\} \leq \alpha$ and $\max\{\beta_i | A_i \in C_l\} - \alpha > \max\{\beta_i | A_i \in C_{l+1}\}$.

Conditions (a), (b) and (c) imply that $\mathcal{C}$ is a disjoint partition of $\mathcal{K}$, while condition (d) indicates that the quality difference of the arms in the same class is at most α and the one between the best arms in adjacent classes is at least α such that the classes are ranked with granularity α.

Furthermore, as mentioned in Sec. III-A, the samples reported by workers will be delivered to an untrusted third-party platform for data storage and processing, which may result in privacy leakage for the workers. In this paper, we leverage the notion of differential privacy for the purpose of privacy preservation.

Definition 3.2 (Differential Privacy): An adaptive ranking algorithm is said to be ϵ-differential privacy, if for any two possible reward sequences of K arms denoted by $\sigma_{1:K}$ and $\sigma'_{1:K}$ that differ at most one record related to one worker, it holds that for all $\mathcal{S} \subset \mathcal{P}^{\mathbb{N}}(\mathcal{K})$:

$$P(\mathcal{C} \in \mathcal{S} | \sigma_{1:K}) \leq P(\mathcal{C} \in \mathcal{S} | \sigma'_{1:K}) e^{\epsilon},$$

where $\mathcal{C}$ is the ranking from the output of algorithm π.

The notations used in this paper are summarized in Table I.

IV. Algorithm

In this section, we present the Privacy-Preserving Adaptive Ranking (PPAR) algorithm for the MABAR problem with a cost of $\mathcal{O}(K \ln K)$.

TABLE I
FREQUENT NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
$\mathcal{A}$	The crowdsourcing platform
$\mathcal{T}$	The total cost
K	The number of items
$\mathcal{K}$	Set of items $\{1, 2, \cdots K\}$
$\omega_{i,t}$	The quality of A_i in epoch t
β_i	Quality of item A_i
μ_i	Empirical mean related to item A_i
$\hat{\mu}_i$	Perturbed empirical mean related to item A_i
$\overline{\mu}_i$	The average of new τ scores
α	Rank span
$\mathcal{S}$	An active set
l	The serial number of the class
$\mathcal{C}_l$	Set of items belong to the g-th rank
$\mathcal{H}_i$	The hybrid mechanism relative to item A_i
σ_i	A stream of scores of the item A_i
σ_i^w	The w-th element of a stream related to the item A_i
ϵ	A privacy parameter
τ	The cost of each reward collection
$\hat{\mu}_{max}$	The maximum perturbed empirical mean
$\sum[i,j]$	A p-sum(a adaptive sum)
(A_i, ω_i)	The token related to the item A_i

The algorithm is given in Algorithm 1, which proceeds as follows. With each iteration, the set of arms that are still not ranked is denoted as an active set $\mathcal{S}$. The platform iteratively collects τ results with respect to arms that are still not ranked from workers, till all arms in $\mathcal{S}$ are identified as either belonging to or not belonging to the class defined in the current iteration. Here τ is the cost parameter whose value will be given in the analysis. Every average of the τ rewards for arm A_i is denoted as $\overline{\mu}_i$. Then $\overline{\mu}_i$ is perturbed by a hybrid differential privacy mechanism given in Algorithm 2, and the perturbed value is denoted as $\hat{\mu}_i$. The perturbed data is sent to a third-party through the crowdsourcing platform (broker), such that the arms with high expected rewards can be ranked. The above process continues utill all arms are ranked.

In the l-th iteration, the algorithm consists of the following three steps.

- **Arms Announcement:** The platform announces arms to asynchronously arrived workers. When a worker arrives, the platform makes the worker pull an arm that is still not pulled for τ times. And then the worker generates the token $(A_i, \omega_{i,t})$.
- **Data Perturbation:** Once an arm has been pulled for τ times, the average reward $\overline{\mu}_i$ is computed and protected by the hybrid mechanism given in Algorithm 2. The perturbed value then is sent to the third-party computation platform for the adaptive ranking task, while keeping the privacy preserved.
- **Arm Ranking:** After receiving the perturbed data, the computation platform compares the perturbed average reward of every arm with $\hat{\mu}_{max}$, which is the maximum value of the perturbed average rewards of arms. Only if the difference between the perturbed average reward of the current best arm $\hat{\mu}_{max}$ and that of the current arm $\hat{\mu}_i$ is smaller than a specified threshold $\alpha - 2\sqrt{\frac{\ln(4k/\delta)}{2T_i}}$ (Line 28), the current arm will be put in class C_l, and the other arms need more pulls for adaptive ranking.

Algorithm 1: PPAR: Privacy-Preserving Adaptive Ranking

Input: $\mathcal{K}$, τ, α, error δ, privacy budget ϵ.
Output: $C_1, C_2, \cdots, C_{\lceil \frac{1}{\alpha} \rceil}$.

1 **Initialize:** $\mathcal{S} \leftarrow \mathcal{K}$, $l \leftarrow 1$,
2 Initialize perturbed empirical qualities $\hat{\boldsymbol{\mu}} \in \mathcal{R}^{\mathbb{N}}$.
`/* Initialize parameters related to privacy preserving */`
3 **for** *arm* $A_i \in \mathcal{K}$ **do**
4   Initialize $\boldsymbol{\theta}_i, \tilde{\boldsymbol{\theta}}_i \in \mathcal{R}^{\mathbb{N}}$;
5     $t_i, N_i, \varpi_i, \omega_{i,t} \leftarrow 0$.
6 **for** $j = 1$ *to* $\lceil \frac{1}{\alpha} \rceil$ **do**
7     $C_j \leftarrow \emptyset$;
8 **while** $\mathcal{S} \neq \emptyset$ **do**
9   **while** $\mathcal{S} \neq C_l$ **do**
10   `Arms Announcement()`
        `/* data perturbtion */`
11   **foreach** *arm* $A_i \in \mathcal{S}$ **do**
12   Collect τ tokens;
13   Compute $\overline{\mu}_i$ for arm A_i by selected tokens;
14             $\hat{\mu}_i, t_i, \boldsymbol{\theta}_i, \tilde{\boldsymbol{\theta}}_i, N_i, \omega_{i,t}, \varpi_i \leftarrow$
15                 $\mathcal{H}(\epsilon, \overline{\mu}_i, t_i, \boldsymbol{\theta}_i, \tilde{\boldsymbol{\theta}}_i, N_i, \omega_{i,t}, \varpi_i)$;
16   Insert $\hat{\mu}_i$ to the perturbed empirical
17   qualities $\hat{\boldsymbol{\mu}}$.
18   Platform $\mathcal{A}$ submits $\hat{\boldsymbol{\mu}}$ to the third-party.
19   `Arm Ranking`($\mathcal{S}$, $\hat{\boldsymbol{\mu}}$)
20     $\mathcal{S} \leftarrow C_{l+1}$, $C_{l+1} \leftarrow \emptyset$, $l \leftarrow l + 1$.

21 **Procedure:** `Arms Announcement()`
  foreach *arm* $A_i \in \mathcal{S}$ **do**
22   **for** $t = 1, 2, 3, \cdots, \tau$ **do**
23   Assign A_i to an arrived worker;
24   Generate a token $(A_i, \omega_{i,t})$.

25 **Procedure:** `Arm Ranking`($\mathcal{S}$, $\hat{\boldsymbol{\mu}}$)
Input: active set $\mathcal{S}$ and perturbed empirical qualities $\hat{\boldsymbol{\mu}}$.
26 $\hat{\mu}_{max} \leftarrow max\{\hat{\mu}_i | A_i \in \mathcal{S}\}$;
27 **foreach** *arm* $A_i \in \mathcal{S}$ **do**
28   **if** $\hat{\mu}_i \geq \hat{\mu}_{max} - \alpha + 2\sqrt{\frac{\ln(4k/\delta)}{2T_i}}$ **then**
29         $C_l \leftarrow C_l \cup \{A_i\}$;
30   **if** $\hat{\mu}_i < \hat{\mu}_{max} - \alpha - 2\sqrt{\frac{\ln(4k/\delta)}{2T_i}}$ **then**
31         $C_{l+1} \leftarrow C_{l+1} \cup \{A_i\}$;
32         $\mathcal{S} \leftarrow \mathcal{S} - \{A_i\}$;

In order to add as little noise as possible while obtaining the maximum privacy protection for each arm, we face the trade-

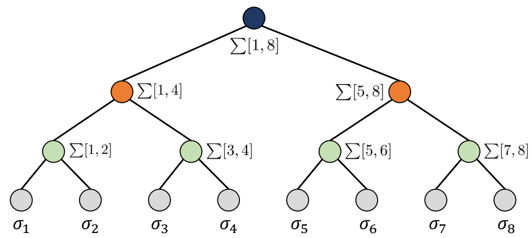


Fig. 2. Each circle corresponds to a p-sum.

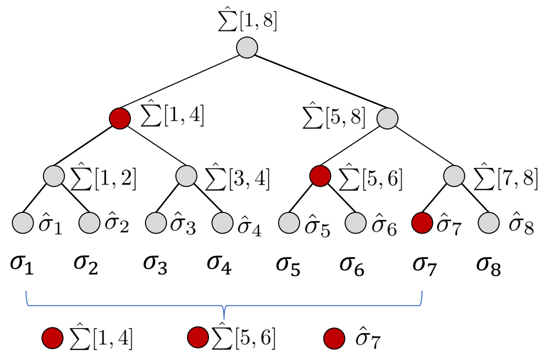


Fig. 3. The sum of $T = 7$ items can be obtained by 3 noisy p-sums corresponding to the red nodes.

off challenge of data utility against data protection. To protect the rewards, a common way to obtain privacy protection is to directly add noise to every item of the stream of rewards by the Laplace mechanism. However by doing so, the added noise linearly increases with the number of rewards, which incurs an unaffordable cost and results in that the average reward obtained by the perturbed value heavily deviates from the expected value, making it impossible to obtain correct adaptive ranking for arms efficiently.

Hence, we here adopt a hybrid mechanism [14] to implement privacy preservation, where the noise added is logarithmic with the growth of the number of rewards, such that the deviation from the real value does not affect the efficiency of adaptive ranking significantly. Essentially, our adapted hybrid mechanism in Algorithm 2 leverages the p-sum framework. The p-sum shown in Fig. 2 can be viewed as the intermediate result of each step of counting. For example, $\sum[a,b] = \sum_{t=a}^{b} \sigma_i^t, 1 \le a \le b$. $\sum[a,b]$ is a p-sum as well as an adaptive sum. The noisy p-sum is represented by $\hat{\sum}[a,b] = \sum_{t=a}^{b} \sigma_i^t + noise$. In our algorithm, the average reward $\overline{\mu}_i$ generated in the t-th round on arm A_i can be viewed as σ_i^t. The hybrid mechanism releases the union of noisy p-sums.

The adapted hybrid mechanism in Algorithm 2 is mainly composed of two components, namely the Logarithmic Mechanism and the Binary Mechanism, respectively, where the Logarithmic Mechanism and the Binary Mechanism are shown in Algorithm 3 and Algorithm 4, respectively.

The perturbed value delivered by Algorithm 2 can be distinguished into two cases. When the length of the stream

Algorithm 2: Hybrid mechanism $\mathcal{H}$

Input: A privacy parameter ϵ and a real number $r \in [0,1]$, counting variable c, auxiliary parameter $\boldsymbol{\theta}$, $\tilde{\boldsymbol{\theta}}$, N, Ω, ϖ.
Output: F, c, $\boldsymbol{\theta}$, $\tilde{\boldsymbol{\theta}}$, N, Ω, ϖ.

```
Initialize: F ← 0.
c ← c + 1;
Ω ← Ω + r;
if c = 2^m for some m ∈ Z then
    ϖ ← L(τε/2, Ω);
    N ← c;
    for k = 0, 1, 2, ··· , log_2 N − 1 do
        θ_k ← 0;
        θ̃_k ← 0;
    F ← L(τε/2, Ω)/c.
else
    v ← c − N;
    θ, θ̃ ← B(N, v, τε/2, θ, θ̃, r);
    Express v in binary form v = Σ_j Bin_j(v) · 2^j;
    F ← [ϖ + (Σ_{j:Bin_j(v)=1} θ̃_j)]/c.
```

Algorithm 3: Logarithmic mechanism $\mathcal{L}$

Input: A privacy parameter ϵ, a real number Ω.
Output: Ω.

```
/* Logarithmic mechanism will add
   noise to the output only when
   counting variable in Hybrid
   mechanism is an integer power of
   2.                                  */
Ω ← Ω + Lap(1/ε).
```

Algorithm 4: Binary Mechanism $\mathcal{B}$

Input: A time upper bound N, a counting variable v, a privacy parameter ϵ, a vector of p-sums $\boldsymbol{\theta}$, a vector of perturbed p-sums $\tilde{\boldsymbol{\theta}}$ and a real number $r \in [0,1]$.
Output: $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}$.

```
Initialize: ε' ← ε/ log_2 N.
Express v in binary form v = Σ_j Bin_j(v) · 2^j;
/* p-sum                               */
Let k = min{j : Bin_j(v) ≠ 0}; θ_k ← Σ_{j<k} θ_j + r;
for j = 0 to k − 1 do
    θ_j ← 0, θ̃_j ← 0.
θ̃_k ← θ_k + Lap(1/ε').
```

input to Algorithm 2 is a power of 2, only Algorithm 3 is invoked to output the perturbed value $\hat{\mu}_i$ through the Laplace distribution with scale $\frac{2}{\tau\epsilon}$. Otherwise, Algorithm 4 is invoked to output a noisy p-sum for the c-th sequential $\overline{\mu}_i$, where

$c \in (\varphi_1, \varphi_2)$, and φ_1, φ_2 are two adjacent numbers that are both power of 2. Then the hybrid mechanism outputs $\hat{\mu}_i$ by combining the last output of Algorithm 3 and the new noisy p-sum obtained from Algorithm 4. Every sum of T consecutive items can be represented by $\log T$ nodes in the tree. As shown in Fig. 3, the sum of $T = 7$ items can be obtained by 3 noisy p-sums corresponding to the red nodes.

V. Algorithm Analysis

In this section, we analyze the performance of the proposed adaptive ranking algorithm, Algorithm 1. We first show that with a high probability, our algorithm can rank the arms correctly with small cost, and then illustrate the privacy preservation of our algorithm.

A. Adaptive Ranking Analysis

We first bound the difference between real qualities and perturbed empirical rewards of all arms by the union bound of Chernoff-Hoeffding inequality and hybrid mechanism's error bound. We use a tight error bound of the hybrid mechanism derived in [30], and by observing a vital fact that the sensitivity of every item in the input stream of the hybrid mechanism can be bounded by $\frac{1}{\tau}$, the error dramatically decreases by noise.

The gap between the real qualities of arms and segmentation points is critical. To bound the cost, we subtlety place a "virtual arm" A_0 on the segmentation point in each iteration. With A_0, we can treat real qualities of arms and segmentation points equally, which greatly simplifies the proof. Apart from that, it is shown that by elaborately setting the threshold of adaptive ranking (i.e. $\pm 2\sqrt{\frac{\ln(4k/\delta)}{2T_i}}$ for arm A_i), a high success probability of adaptive ranking is guaranteed.

The following Chernoff-Hoeffding Bound wil be used in subsequent analysis, which can be found in the most advanced mathematical textbook on statistics.

Lemma 5.1 (Chernoff-Hoeffding Bound): Let $\eta_1, \eta_2, \cdots, \eta_n \in [0, 1]$ be independent random variables, and $\eta = \sum_{i=1}^{n} \eta_i$. For every $z \geq 0$, it holds with additive error that

$$\mathbb{P}(|\eta - \mathbb{E}[\eta]| \geq z) \leq 2e^{-\frac{2z^2}{n}}.$$

We consider the n-th reward collection for an arm A_i. At first, we give the following result to bound the difference between the average reward of an arm and its perturbed value. Let μ_i denote the empirical mean of rewards on an arm A_i after the n-th reward collection, and recall that $\hat{\mu}_i$ is the perturbed average reward output by the privacy-preserving mechanism after the n-th reward collection.

Lemma 5.2: For any arm A_i, it holds in the l-th iteration that

$$\mathbb{P}(|\mu_i - \hat{\mu}_i| \geq z') \leq 2e^{(-\frac{z' T_i \epsilon}{2\sqrt{2}(\ln \frac{T_i}{\tau}+1)})},$$

where T_i $(= n\tau)$ is the number of rewards received for A_i till the n-th reward collection and ϵ is the required privacy budget.

Proof. Denote by X the sum of rewards received by the platform on arm A_i till the n-th reward collection, and denote by X_n the perturbed value of X, using the hybrid mechanism. In [30], it has been shown that

$$\mathbb{P}\left(|X - X_n| \geq \frac{2\sqrt{2}}{\epsilon\tau} \ln \frac{2}{\gamma} (\ln n + 1)\right) \leq \gamma. \tag{1}$$

Let $z' = \dfrac{2\sqrt{2}}{\epsilon\tau} \ln \dfrac{2}{\gamma}(\ln n + 1)$, then

$$\gamma = 2e^{(-\frac{z'\epsilon\tau}{2\sqrt{2}(\ln n+1)})}.$$

Eqt. (1) can be rewritten as

$$\mathbb{P}(|X - X_n| \geq z') \leq 2e^{(-\frac{z'\epsilon\tau}{2\sqrt{2}(\ln n+1)})}. \tag{2}$$

By dividing $\frac{T_i}{\tau}$ in the above equation, we can then get that for $z' \geq 0$

$$\mathbb{P}(|\mu_i - \hat{\mu}_i| \geq z') \leq 2e^{(-\frac{z' T_i \epsilon}{2\sqrt{2}(\ln \frac{T_i}{\tau}+1)})},$$

which completes the proof. □

Lemma 5.2 shows the relationship between the empirical reward and the perturbed empirical reward. We next establish the relationship between the expected reward and the perturbed empirical reward through the empirical reward.

Lemma 5.3: For any arm A_i, if $\epsilon \geq 8\sqrt{\dfrac{\ln(4K/\delta)}{T_i}} \ln(\dfrac{T_i}{\tau} + 1)$, with probability at least $(1 - \delta)$, we have

$$|\beta_i - \hat{\mu}_i| \leq \sqrt{\frac{2\ln(4K/\delta)}{T_i}}.$$

Proof. By Algorithm 2, the added noise is independent of the input value. For any $t' \geq 0$, according to the independence of events, we have

$$\begin{aligned}
&\mathbb{P}\left(|\beta_i - \hat{\mu}_i| \leq 2t'\right) \\
&\geq \left(1 - \mathbb{P}\left(|\beta_i - \mu_i| \geq t'\right)\right)\left(1 - \mathbb{P}(|\mu_i - \hat{\mu}_i| \geq t')\right).
\end{aligned}$$

Therefore,

$$\mathbb{P}(|\beta_i - \hat{\mu}_i| \geq 2t') \leq \mathbb{P}(|\beta_i - \mu_i| \geq t') + \mathbb{P}(|\mu_i - \hat{\mu}_i| \geq t').$$

By Lemma 5.1 and Lemma 5.2, it holds that

$$\mathbb{P}(|\beta_i - \hat{\mu}_i| \geq 2t') \leq 2e^{(-\frac{t' T_i \epsilon}{2\sqrt{2}(\ln \frac{T_i}{\tau}+1)})} + 2e^{-2t'^2 T_i}. \tag{3}$$

If $\epsilon \geq 4\sqrt{2}t' \ln(\frac{T_i}{\tau} + 1)$, then by Eqt. (3),

$$\mathbb{P}(|\beta_i - \hat{\mu}_i| \geq 2t') \leq 4e^{-2t'^2 T_i}$$

and

$$\mathbb{P}(|\beta_i - \hat{\mu}_i| \geq t') \leq 4e^{-\frac{t'^2}{2} T_i}.$$

When $t' = \sqrt{\dfrac{2\ln(4K/\delta)}{T_i}}$, we have

$\epsilon \geq 8\sqrt{\dfrac{\ln(4K/\delta)}{T_i}} \ln\left(\dfrac{T_i}{\tau} + 1\right)$, it can be obtained that

$$\mathbb{P}\left(|\beta_i - \hat{\mu}_i| \leq \sqrt{\frac{2\ln(4K/\delta)}{T_i}}\right) \geq \left(1 - \frac{\delta}{K}\right)^K \geq 1 - \delta.$$

The lemma follows. $\square$

As discussed before, we will add a "virtual arm" to facilitate the analysis. The following Lemma will help set this arm. Denote by β_x the largest expected reward of arms that have not been ranked.

Lemma 5.4: For any iteration, Let $\mathcal{S} \subset \mathcal{K}$ be the active set, assume arm $A_x \in \mathcal{S}$ has the largest expected reward β_x and arm $A_y \in \mathcal{S}$ has the largest perturbed empirical reward $\hat{\mu}_y$. It holds that

$$\mathbb{P}\left(|\beta_x - \hat{\mu}_y| \geq \sqrt{\frac{2\ln(4K/\delta)}{min\{T_x, T_y\}}}\right) \leq \frac{\delta}{K}$$

Proof. We show the claim by the following two cases.

Case 1: $\beta_x \geq \hat{\mu}_y$.
According to Lemma 5.3, with a probability at least $(1 - \frac{\delta}{K})$, it holds that

$$|\beta_x - \hat{\mu}_x| \leq \sqrt{\frac{2\ln(4K/\delta)}{T_x}}. \tag{4}$$

Given that arm $A_y \in \mathcal{S}$ has the largest perturbed empirical reward $\hat{\mu}_y$, we have

$$\hat{\mu}_x \leq \hat{\mu}_y. \tag{5}$$

From Eq. (4) and (5), it can be obtained that

$$0 \leq \beta_x - \hat{\mu}_y \leq \beta_x - \hat{\mu}_x \leq \sqrt{\frac{2\ln(4K/\delta)}{T_x}}.$$

Case 2: $\beta_x \leq \hat{\mu}_y$**.**
It can be obtained that

$$0 \leq \hat{\mu}_y - \beta_x \leq \hat{\mu}_y - \beta_y \leq \sqrt{\frac{2\ln(4K/\delta)}{T_y}}.$$

Combining both cases together, it holds that

$$\mathbb{P}\left(|\beta_x - \hat{\mu}_y| \geq \sqrt{\frac{2\ln(4K/\delta)}{min\{T_x, T_y\}}}\right) \leq \frac{\delta}{k}.$$

$\square$

We define an arm A_0 with an expected reward of $(\beta_x - \alpha)$ and a perturbed empirical reward of $\hat{\mu}_0 = \hat{\mu}_y - \alpha$. And the total cost of A_0 is $min\{T_x, T_y\}$. Our definition is reasonable since by Lemma 5.4, it holds that

$$\mathbb{P}\left(|\mu_0 - \hat{\mu}_0| \geq \sqrt{\frac{2\ln(4K/\delta)}{T_0}}\right) \leq \frac{\delta}{K}.$$

The defined arm A_0 satisfies Lemma 5.3 as well as all other arms.

With the facilitation of arm A_0, we can get the following result. Denote by Δ_{min} the minimum gap of qualities between all arms and all segmentation points.

We give the following Lemma 5.5, using arm A_0 skillfully.

Lemma 5.5: In any iteration, let $\mathcal{K}'$ be any subset of $\mathcal{K} \cup \{A_0\}$ and $T' = min\{T_i | A_i \in \mathcal{K}'\}$. For any $a, b \in [0, 1]$, if $b > a$ and $b - a < n\Delta_{min} - 2\sqrt{\frac{2\ln(4K/\delta)}{T'}}$, with probability at least $(1 - \delta)$, there are no more than n arms $\in \mathcal{K}'$ whose perturbed empirical rewards fall in $[a, b]$.

Proof. We prove the claim by contradiction. Assume that there exist more than n arms whose perturbed empirical rewards fall in $[a, b]$. We denote the set of these arms as $\mathcal{S}'$, i.e., $\mathcal{S}' = \{A_i | A_i \in \mathcal{K}' \wedge a \leq \hat{\mu}_i \leq b\}$.

Notice that $|\mathcal{S}'| \geq n + 1$ and $\mathcal{S}' \subset \mathcal{K} \cup \{A_0\}$. Assume A_x and $A_{x'}$ have the largest and smallest expected rewards among all arms in $\mathcal{S}'$. By Lemma 5.3, with probability at least $(1-\delta)$, we have

$$\begin{aligned}\hat{\mu}_x - \hat{\mu}_{x'} &\geq \beta_x - \sqrt{\frac{2\ln(4K/\delta)}{T_x}} - \left(\beta_{x'} + \sqrt{\frac{2\ln(4k/\delta)}{T_{x'}}}\right) \\ &\geq n\Delta_{min} - 2\sqrt{\frac{2\ln(4K/\delta)}{T'}},\end{aligned}$$

which contradicts the assumption that $b - a < n\Delta_{min} - 2\sqrt{\frac{2\ln(4K/\delta)}{T'}}$. Hence, in $\mathcal{K}'$, there are no more than n arms whose perturbed empirical rewards fall in $[a, b]$. $\square$

Now, we are ready to prove the final result.

Theorem 5.1: Let $R = \lceil \frac{1}{\alpha} \rceil$ and $r = \lceil \frac{72\ln(4K/\delta)}{\Delta_{min}^2 \tau} \rceil$. With probability at least $(1 - \delta)^{Rr}$, algorithm PPAR can deliver correct adaptive ranking with a total cost of $\mathcal{O}(K\ln(K))$ when $\epsilon \geq 8\sqrt{\frac{\ln(4K/\delta)}{T_i}} \ln(\frac{T_i}{\tau} + 1)$ for any arm A_i.

Proof. When $T' \geq \frac{72\ln(4K/\delta)}{\Delta_{min}^2}$, we have

$$4\sqrt{\frac{2\ln(4K/\delta)}{T'}} \leq \Delta_{min} - 2\sqrt{\frac{2\ln(4K/\delta)}{T'}}. \tag{6}$$

Then, by Lemma 5.5, in any iteration, with probability at least $(1 - \delta)$, there is at most one arm in $\mathcal{S} \cup \{A_0\} - C_l$ whose perturbed empirical reward falls in the interval

$$\left[\hat{\mu}_{max} - \alpha - 2\sqrt{\frac{2\ln(4K/\delta)}{T'}}, \hat{\mu}_{max} - \alpha + 2\sqrt{\frac{2\ln(4K/\delta)}{T'}}\right].$$

The last arm in $\mathcal{S} \cup \{A_0\} - C_l$ whose perturbed empirical reward stays in the interval must be A_0, since the perturbed empirical reward is $\hat{\mu}_{max} - \alpha$. Therefore, all arms has been ranked (recall line 28-32 in Algorithm 1) in any iteration when $T' \geq \frac{72\ln(4K/\delta)}{\Delta_{min}^2}$. In other words, the maximum cost for an arm to be ranked is at most $\frac{72\ln(4K/\delta)}{\Delta_{min}^2}$. Hence, the total cost for algorithm PPAR is at most $\frac{72K\ln(4K/\delta)}{\Delta_{min}^2}$.

Furthermore, considering the cumulative error for all iterations, the success probability is at least $(1-\delta)^{Rr}$, as there are at most Rr iterations. The theorem then follows. $\square$

From Theorem 5.1, we can assign a proper value to δ to improve the algorithm's accuracy, and the impact on the total cost is logarithmic meanwhile. For example, by setting $\delta \leq$

$\frac{0.01}{Rr}$, the success probability can be at least $(1-\delta)^{Rr} \geq 1 - \delta Rr \geq 0.99$.

B. Privacy Preserving

The rest is to show that the hybrid mechanism in Algorithm 2 can satisfy differential privacy defined in Definition 3.2.

We first give the following Lemma 5.6 to illustrate the properties of the Laplace Mechanism.

Lemma 5.6: Let $\mathcal{M}$ be a summing function on a stream. Assume σ and σ' are two streams that differ at one item only t and $|\sigma(t) - \sigma'(t)| \leq \Delta$. Let $Y \backsim \text{Lap}(\frac{\Delta}{\epsilon})$ be a random variable drawn from the Laplace distribution. Then, for any subset S of $\mathbb{R}$, we have

$$\mathbb{P}(\mathcal{M}(\sigma) + Y \in S) \leq \exp(\epsilon) \cdot \mathbb{P}(\mathcal{M}(\sigma') + Y \in S).$$

The proof of Lemma 5.6 can be found in [31].

Theorem 5.2: The *Privacy-Preserving Adaptive Ranking (PPAR)* preserves ϵ-differential privacy for each reward generated by any arm in $\mathcal{K}$.

Proof. For each arm A_i, the empirical mean of every τ rewards is the input to the hybrid mechanism. We denote σ_i as the input stream for arm A_i. Therefore, for arm A_i, the change of any one empirical reward only impacts one item of σ_i. The maximum change is bounded by $\frac{1}{\tau}$.

For the Logarithmic Mechanism used in Algorithm 2, $\frac{\epsilon}{2}$-differential privacy is preserved by Lemma 5.6. For Binary mechanism used in Algorithm 2, the change of any one empirical reward for arm A_i impacts at most $\log_2 N$ p-sums because it can participate in at most $\log_2 N$ p-sums from Fig. 2. The Binary mechanism can also preserve $\frac{\epsilon}{2}$-differential privacy, since each perturbed p-sum keeps $\frac{\epsilon}{2\log_2 N}$-differential privacy. Thus we can obtain the final result by applying a composition theorem [31], since Algorithm 1 performs a randomized mapping from the output of hybrid mechanisms to a sorted sequence of subsets of $\mathcal{K}$. □

The main parameters of algorithm PPAR are the error rate δ and the private budget ϵ. We must be careful to assign values to these two variables to achieve the desired cost. By Lemma 5.3 and Theorem 5.1, it can be shown that algorithm PPAR succeeds with a high probability under a total sample cost of $O(K \ln K)$, as long as ϵ and δ satisfies that $\epsilon \geq 8\sqrt{\frac{\ln(4K/\delta)}{T_i}} \ln(\frac{T_i}{\tau} + 1)$ while the ϵ-differential privacy is ensured. Furthermore, by Theorem 5.1, it can be observed that the upper bound on the total cost logarithmically increases with the decease on the value of δ.

VI. Experiment

In this section, we evaluate the performance of our ranking algorithm PPAR in real settings. Specifically, we first evaluate the impact of different parameters on the accuracy of the adaptive ranking, including the number of arms K, the iteration cost τ, and the privacy level ϵ. We then study the variation tendency of accuracy, by varying the distribution of qualities of arms. We finally test PPAR by real-world dataset, the *Open Bandit Dataset* [32]. The experiments are conducted on a laptop with Intel Core i7-7600U and 8GB of RAM. Each result reported is the average of 150 times of experiments.

In the default setting, we set the qualities of arms drawn from the Gaussian distribution with a mean of 0.5 and a variance of 10, and all sampled qualities are clipped within $[0, 1]$. Then, we set $(K, \tau, \epsilon) = (20, 6 \times 10^3, 0.25)$. Based on the default setup, we change each parameter respectively and show how they affect the accuracy of the solution, where the accuracy of each class is defined as follow. For the c-th class, let $\mathcal{U}$ denote the arms ranked into c by the standard adaptive ranking, while $\mathcal{V}$ denotes the arms ranked into c by PPAR. Then the accuracy of the c-th class defined as

(a): $|\mathcal{U} \cap \mathcal{V}|/|\mathcal{U}|$ if $|\mathcal{U}| > 0$;
(b): 1 if $|\mathcal{U}| = |\mathcal{V}| = 0$;
(c): 0 if $|\mathcal{U}| = 0$ and $|\mathcal{V}| \neq 0$.

We first evaluate the influence of class span α on the accuracy with $\alpha = 0.1, 0.2, 0.3$ and 0.4 respectively. As shown in Fig. 4, the accuracy of all classes in these four different cases maintains at a very high level, with a minimum accuracy higher than 0.98. Furthermore, the average accuracy tends to be higher when α increases. In this case, the number of classes is small, which means the granularity of the division is large, and hence the chance of incorrect ranking of arms decreases.

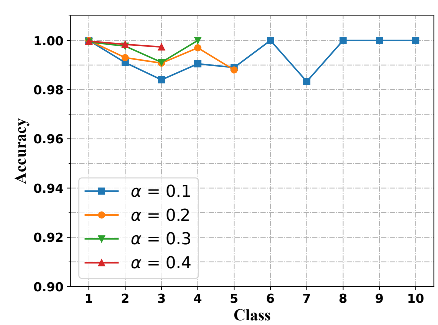
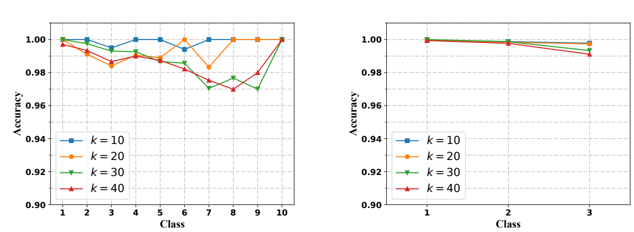


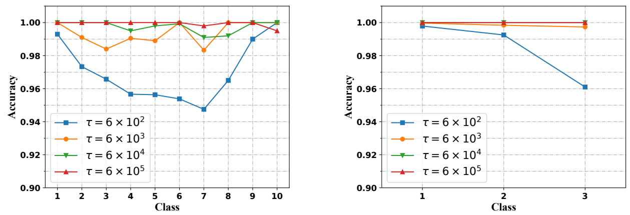
Fig. 4. Accuracy of algorithm PPAR by varying α.

Fig. 5 illustrates the accuracy of the adaptive ranking process with different numbers of arms. The results are shown in two different class spans $\alpha = 0.1$ and $\alpha = 0.4$, respectively. The number of arms is set as $K = 10, 20, 30$ and 40 respectively. From Fig. 5, it can be seen that as the number of arms increases, the average accuracy of ranking has a



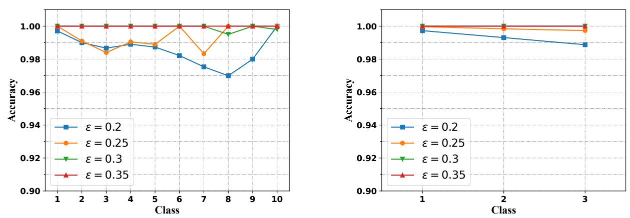
(a) $\alpha = 0.1$ (b) $\alpha = 0.4$

Fig. 5. Accuracy of algorithm PPAR by varying K.



(a) $\alpha = 0.1$ (b) $\alpha = 0.4$

Fig. 6. Accuracy of algorithm PPAR by varying τ.



(a) $\alpha = 0.1$ (b) $\alpha = 0.4$

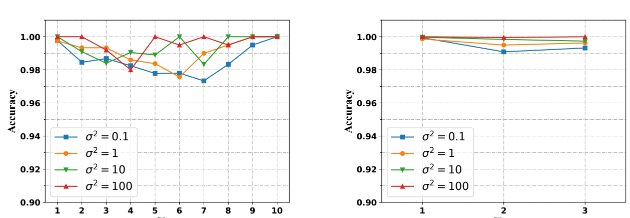
Fig. 7. Accuracy of algorithm PPAR by varying ϵ.

decreasing tendency. For example, when $\alpha = 0.1$, the average accuracy for all groups when $K = 40$ is 0.986, while this value is 0.999 when $K = 10$. With fewer arms, the ranking task is simpler and more accurate. This result is consistent with our theoretical analysis.

In Fig. 6, we evaluate the impact of iteration cost τ. The iteration cost is set as $\tau = 6 \times 10^2, 6 \times 10^3, 6 \times 10^4$ and 6×10^5 respectively. Other parameters are set as default values. From Fig. 6, we can observe that a larger iteration cost can increase the accuracy. The reason is that when τ is large, more information about arms can be explored, which improves the probability of success in the whole ranking process.

We then illustrate the impact of privacy budget ϵ in Fig. 7. The privacy budget is set as $\epsilon = 0.2, 0.25, 0.3$ and 0.35 respectively. It can be seen from in Fig. 7 that the accuracy increases when the privacy budget gets larger. This is because that the larger the privacy budget is, the data is more useful. Then, a more accurate result can be obtained.

Finally, we evaluate the influence of the distribution of arms' qualities. The results are shown in Fig. 8. We consider the Gaussian distribution with different variance (denoted as σ^2), and the distribution with a variance of 10 is our default setup. The quality of each arm is sampled obeying the Gaussian distribution. We set the mean of all distributions as 0.5, and $\sigma = 0.1, 1, 10$ and 100 respectively. From Fig. 8, we can see that the average accuracy under Gaussian distribution increases when σ gets higher. This result is reasonable because a bigger standard deviation means the distribution of qualities of arms is less intensive, which is favorable for ranking.



(a) $\alpha = 0.1$ (b) $\alpha = 0.4$

Fig. 8. Accuracy of algorithm PPAR under different distributions of quality.

We finally test the performance of PPAR when changing the value of α by a real-world dataset, the *Open Bandit Dataset* [32]. The Open Bandit Dataset is a public real-world logged multi-bandits feedback dataset, provided by the largest Japanese fashion e-commerce company ZOZO, Inc. This dataset contains 80 commercial items(considered as 80 arms) and each arm is associated with some row data as pulling records(26M records in total). We use the propensity score of each record as the reward of pulling the corresponding arm. The qualities of arms in the Open Bandit Dataset are distributed as shown in Fig. 9.

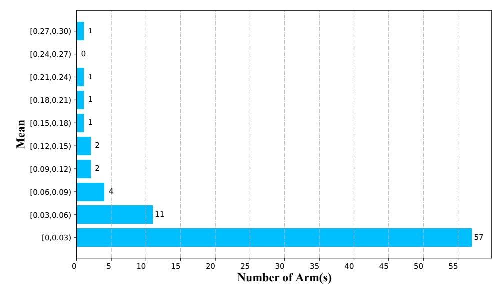


Fig. 9. Distributions of arms' qualities.

The initialization of K, τ and ϵ are same as default setup. When pulling arm i in algorithm PPAR, we randomly sample a reward from the corresponding row data without putting it back. The average performance of algorithm PPAR in the Open Bandit Dataset is shown in Fig. 10. It can be seen that the average accuracy for all groups decrease when reducing α, while the accuracy of all groups maintain a high value over 0.94 when $\alpha \geq 0.1$.

VII. Conclusion

In this paper, we presented a privacy preserving arm ranking algorithm in crowdsourcing systems. We formalized by the multi-armed bandits model, and our proposed algorithm can make every arm ranked into correct classes with a high probability of $(1-\delta)^{Rr}$, using $\mathcal{O}(K \log K)$ cost for data collection even under the consideration of privacy. A stringent privacy protection is provided for workers in crowdsourcing systems by a hybrid mechanism satisfying ϵ-differential privacy. We also evaluated the proposed algorithm through experiments. Extensive experimental results showed that our proposed algorithm has good performance in practice.

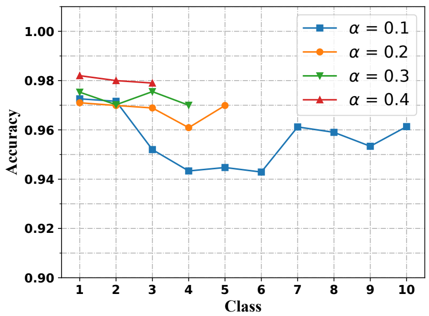


Fig. 10. Accuracy of algorithm PPAR based on the Open Bandit Dataset.

REFERENCES

[1] M. J. Gomez, M. Calderón, V. Sánchez, F. J. G. Clemente, and J. A. R. Valiente, "Large scale analysis of open MOOC reviews to support learners' course selection," *CoRR*, vol. abs/2201.06967, 2022.

[2] Richa and P. Bedi, "Trust and distrust based cross-domain recommender system," *Appl. Artif. Intell.*, vol. 35, no. 4, pp. 326–351, 2021.

[3] S. Katariya, A. Tripathy, and R. Nowak, "MaxGap Bandit: Adaptive Algorithms for Approximate Ranking," in *Proc. of the 33rd NIPS*, 2019, pp. 11 045–11 055.

[4] S. Katariya, L. Jain, N. Sengupta, J. Evans, and R. Nowak, "Adaptive Sampling for Coarse Ranking," in *Proc. of the 21th AISTATS*, vol. 84, 2018, pp. 1839–1848.

[5] N. Karpov and Q. Zhang, "Batched Coarse Ranking in Multi-Armed Bandits," in *Proc. of the 34th NIPS*, vol. 33, 2020.

[6] S. Egelman, R. Kannavara, and R. Chow, "Is This Thing On?: Crowdsourcing Privacy Indicators for Ubiquitous Sensing Platforms," in *Proc. of the 33rd ACM CHI*, 2015, pp. 1669–1678.

[7] B. Bergvall-Kåreborn and D. Howcroft, "The apple business model: Crowdsourcing mobile applications," in *Accounting Forum*, vol. 37, no. 4, 2013, pp. 280–289.

[8] A. Dubey and A. Pentland, "Differentially-Private Federated Linear Bandits," in *Proc. of 34th NIPS*, 2020.

[9] H. Wang, Q. Zhao, Q. Wu, S. Chopra, A. Khaitan, and H. Wang, "Global and Local Differential Privacy for Collaborative Bandits," in *Proc. of the 20th ACM RecSys*, 2020, pp. 150–159.

[10] X. Chen, K. Zheng, Z. Zhou, Y. Yang, W. Chen, and L. Wang, "Locally Differentially Private Combinatorial Semi-Bandits," in *Proc. of the 37th ICML*, 2020, pp. 1757–1767.

[11] R. Shariff and O. Sheffet, "Differentially Private Contextual Linear Bandits," in *Proc. of the 32rd NIPS*, 2018, pp. 4301–4311.

[12] C. Dwork, "Differential privacy: A survey of results," in *TAMC*, ser. Lecture Notes in Computer Science, vol. 4978, 2008, pp. 1–19.

[13] C. Dwork and A. Smith, "Differential privacy for statistics: What we know and what we want to learn," *J. Priv. Confidentiality*, vol. 1, no. 2, 2010.

[14] T. Chan, E. Shi, and D. Song, "Private and continual release of statistics," in *Proc. of ICALP*, 2010, pp. 405–417.

[15] L. Chen, J. Li, and M. Qiao, "Towards Instance Optimal Bounds for Best Arm Identification," in *Proc. of 30th COLT*, vol. 65, 2017, pp. 535–592.

[16] Z. Karnin, T. Koren, and O. Somekh, "Almost Optimal Exploration in Multi-Armed Bandits," in *Proc. of the 30th ICML*, 2013, pp. 1238–1246.

[17] S. Chen, Y. Tao, D. Yu, F. Li, and B. Gong, "Distributed learning dynamics of multi-armed bandits for edge intelligence," *J. Syst. Archit.*, vol. 114, p. 101919, 2021.

[18] V. Bengs, R. Busa-Fekete, A. E. Mesaoudi-Paul, and E. Hüllermeier, "Preference-based online learning with dueling bandits: A survey," *J. Mach. Learn. Res.*, vol. 22, pp. 7:1–7:108, 2021.

[19] S. He and K. G. Shin, "Information fusion for (re)configuring bike station networks with crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 736–752, 2022.

[20] F. Tao, L. Jiang, and C. Li, "Differential evolution-based weighted soft majority voting for crowdsourcing," *Eng. Appl. Artif. Intell.*, vol. 106, p. 104474, 2021.

[21] A. Pimpinella, M. Repossi, and A. E. C. Redondi, "Unsatisfied today, satisfied tomorrow: A simulation framework for performance evaluation of crowdsourcing-based network monitoring," *Comput. Commun.*, vol. 182, pp. 184–197, 2022.

[22] M. Yuen, I. King, and K. Leung, "A Survey of Crowdsourcing Systems," in *Proc. of IEEE SocialCom*, 2011, pp. 766–773.

[23] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[24] S. Chen, Y. Tao, D. Yu, F. Li, B. Gong, and X. Cheng, "Privacy-preserving collaborative learning for multiarmed bandits in iot," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3276–3286, 2021.

[25] Y. Yuan, F. Li, D. Yu, J. Zhao, J. Yu, and X. Cheng, "Distributed social learning with imperfect information," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 841–852, 2021.

[26] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: A survey," *IEEE Wirel. Commun.*, vol. 27, no. 1, pp. 24–30, 2020.

[27] L. Chen, J. Li, and M. Qiao, "Nearly Instance Optimal Sample Complexity Bounds for Top-k Arm Selection," in *Proc. of the 20th AISTATS*, 2017, pp. 101–110.

[28] W. Ren, J. Liu, and N. Shroff, "Exploring k out of Top ρ Fraction of Arms in Stochastic Bandits," in *Proc. of the 22nd AISTATS*, 2019, pp. 2820–2828.

[29] R. Sen, A. Rakhlin, L. Ying, R. Kidambi, D. Foster, D. Hill, and I. Dhillon, "Top-k extreme contextual bandits with arm hierarchy," *CoRR*, vol. abs/2102.07800, 2021.

[30] A. Tossou and C. Dimitrakakis, "Algorithms for differentially private multi-armed bandits," in *Proc. of the 30th AAAI*, vol. 30, no. 1, 2016.

[31] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[32] Y. Saito, A. Shunsuke, M. Megumi, and N. Yusuke, "Large-scale open dataset, pipeline, and benchmark for bandit algorithms," *arXiv preprint arXiv:2008.07146*, 2020.