Optimal PAC Multiple Arm Identification with Applications to Crowdsourcing

Yuan Zhou, Carnegie Mellon U YUANZHOU@CS.CMU.EDU
Xi Chen, UC Berkeley XICHEN@CS.CMU.EDU
Jian Li, Tsinghua U LIJIAN83@MAIL.TSINGHUA.EDU.CN

Abstract

We study the problem of selecting K arms with the highest expected rewards in a stochastic n-armed bandit game. Instead of using existing evaluation metrics (e.g., misidentification probability (Bubeck et al., 2013) or the metric in EXPLORE-K (Kalyanakrishnan & Stone, 2010)), we propose to use *the aggregate regret*, which is defined as the gap between the average reward of the optimal solution and that of our solution. Besides being a natural metric by itself, we argue that in many applications, such as our motivating example from crowdsourcing, the aggregate regret bound is more suitable. We propose a new PAC algorithm, which, with probability at least $1-\delta$, identifies a set of K arms with regret at most ϵ. We provide the sample complexity bound of our algorithm. To complement, we establish the lower bound and show that the sample complexity of our algorithm matches the lower bound. Finally, we report experimental results on both synthetic and real data sets, which demonstrates the superior performance of the proposed algorithm.

1. Introduction

We study the multiple arm identification problem in a stochastic multi-armed bandit game. More formally, assume that we are facing a bandit with n alternative arms, where the i-th arm is associated with an unknown reward distribution supported on $[0,1]$ with mean θ_i. Upon each sample (or "pull") of a particular arm, the reward is an *i.i.d.* sample from the underlying reward distribution. We sequentially decide which arm to pull next and then collect the reward by sampling that arm. The goal of our "top-K arm identification" problem is to identify a subset of K arms with the maximum total mean. The problem finds applications in a variety of areas, such as in industrial engineering (Koenig & Law, 1985), evolutionary computation (Schmidt et al., 2006) and medical domains (Thompson, 1933). Here, we highlight another application in *crowdsourcing*. In recent years, crowdsourcing services become increasingly popular for collecting labels of the data for many data analytical tasks. The readers may refer to (Raykar et al., 2010; Karger et al., 2012; Zhou et al., 2012; Ho et al., 2013; Chen et al., 2013b) and references therein for recent work on machine learning in crowdsourcing. In a typical crowdsourced labeling task, the requestor submits a batch of microtasks (e.g., unlabeled data) and the workers from the crowd are asked to complete the tasks. Upon each task completion, a worker receives a small monetary reward. Since some workers from the crowd can be highly noisy and unreliable, it is important to first exclude those unreliable workers in order to obtain high quality labels. An effective strategy for this purpose is to test each worker by a few gold samples, i.e., data with the known labels usually labeled by domain experts. We note that workers will not be informed that they are tested using gold samples. Since the requestor has to pay for each labeling of gold samples, it is desirable to select the best K workers with the minimum number of queries. This problem can be cast into our top-K arm identification problem, where each worker corresponds to a Bernoulli arm and the mean θ_i characterizes the i-th worker's underlying reliability/quality. In particular, an answer from the i-th worker is correct (which corresponds to obtain reward 1) with probability θ_i and is wrong (which corresponds to obtain reward 0) with probability $1-\theta_i$.

More formally, assume that the arms are ordered by their means: $\theta_1 > \theta_2 > \ldots > \theta_n$ and let T be the set of selected arms with size $|T| = K$. We define the *aggregate regret* (or *regret* for short) of T as:

$$\mathcal{L}_T = \frac{1}{K}\left(\sum_{i=1}^{K}\theta_i - \sum_{i\in T}\theta_i\right). \tag{1}$$

Our goal is to design an algorithm with low sample complexity and PAC (Probably Approximately Correct) style bounds. More specifically, given any fixed positive constants ϵ, δ, the algorithm should be able to identify a set T of K arms with $\mathcal{L}_T \le \epsilon$ (we call such a solution an

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

ϵ-optimal solution), with probability at least $1-\delta$.

We first note that our problem strictly generalizes the previous work by (Even-Dar et al., 2006; Mannor & Tsitsiklis, 2004) for $K = 1$ to arbitrary positive integer K and hence is referred to as multiple arm identification problem. Although the problem of choosing multiple arms has been studied in some existing work, e.g., (Bubeck et al., 2013; Audibert et al., 2013; Kalyanakrishnan & Stone, 2010; Kalyanakrishnan et al., 2012), our notion of aggregate regret is inherently different from previously studied evaluation metrics such as misidentification probability (MISID-PROB) (Bubeck et al., 2013) and EXPLORE-K (Kalyanakrishnan & Stone, 2010; Kalyanakrishnan et al., 2012). As we will explain in Section 2, our evaluation metric is a more suitable objective for many real applications, especially for the aforementioned crowdsourcing application.

We summarize our results in this paper as follows:

1. Section 3 & 4: We develop a new PAC algorithm with sample complexity $O\left(\frac{n}{\epsilon^2}\left(1+\frac{\ln(1/\delta)}{K}\right)\right)$ for any $\epsilon > 0, 0 < \delta < 1$, and $K \leq n/2$. For $K \geq n/2$, the sample complexity becomes $O\left(\left(\frac{n-K}{K}\cdot\frac{n}{\epsilon^2}\right)\left(\frac{n-K}{K}+\frac{\ln(1/\delta)}{K}\right)\right)$. It is interesting to compare this bound with the optimal bound $O(\frac{n}{\epsilon^2}\ln(1/\delta))$ for $K=1$ in (Even-Dar et al., 2006; Mannor & Tsitsiklis, 2004). For $K=1$ (i.e., selecting the best arm), our result matches theirs. Interestingly, when K is larger, our algorithm suggests that even less samples are needed. Intuitively, a larger K leads to a less stringent constraint for an ϵ-optimal solution and thus can tolerate more mistakes. Let us consider the following toy example. Assume all the arms have the same mean $1/2$, except for a random one with mean $1/2+2\epsilon$. If $K=1$, to obtain an ϵ-optimal solution, we essentially need to identify the special arm and thus need a lot of samples. However, if K is large, any subset of K arms would work fine since the regret is at most $2\epsilon/K$. Our algorithm bears some similarity with previous work, such as the halving technique in (Even-Dar et al., 2006; Kalyanakrishnan & Stone, 2010; Karnin et al., 2013) and idea of accept-reject in (Bubeck et al., 2013). However, the analysis is more involved than the case for $K=1$ and needs to be done more carefully in order to achieve the above sample complexity.

2. Section 5: To complement the upper bound, we further establish a matching lower bound for Bernoulli bandits: for $K \leq n/2$, any (deterministic or randomized) algorithm requires at least $\Omega\left(\frac{n}{\epsilon^2}\left(1+\frac{\ln(1/\delta)}{K}\right)\right)$ samples to obtain an ϵ-optimal solution with probability at least $1-\delta$; for $K \geq n/2$, the lower bound becomes $\Omega\left(\left(\frac{n-K}{K}\cdot\frac{n}{\epsilon^2}\right)\left(\frac{n-K}{K}+\frac{\ln 1/\delta}{K}\right)\right)$. This shows that our algorithm achieves the optimal sample complexity for Bernoulli bandits and for all values of ϵ, δ and K. To this end, we show two different lower bounds for $K \leq n/2$: $\Omega(\frac{n}{\epsilon^2})$ and $\Omega(\frac{n}{\epsilon^2}\frac{\ln(1/\delta)}{K})$. The first bound is established via an interesting reduction from our problem to the basic problem of distinguishing two similar Bernoulli arms (with means $1/2$ and $1/2+\epsilon$ respectively). The second one can be shown via a generalization of the argument in (Mannor & Tsitsiklis, 2004) for $K=1$. The lower bound for $K \geq n/2$ can be easily derived by a reduction to the case for $K \leq n/2$.

3. Section 6: Finally, we conduct experiments on both simulated and real data sets. The experimental results demonstrate that, using the same number of samples, our algorithm not only achieves lower regret but also higher precision than existing methods. Moreover, using our algorithm, the maximum number of samples taken from any individual arm is much smaller than that in the SAR algorithm (Bubeck et al., 2013). In fact, one can show that, if we fix the sample budget to be Q, the maximum number of samples for an arm is $Q/n^{\Omega(1)}$ for our algorithm (Theorem 4.3), while SAR might query an arm by $\Omega(Q/\log(n))$ times. This property is particularly desirable for crowdsourcing applications since it can be quite problematic, at least time-consuming, to test a single worker with too many samples.

2. Related Works

Multi-armed bandit problems have been extensively studied in the machine learning community over the past decade (see for example (Auer et al., 2002a;b; Beygelzimer et al., 2011; Bubeck & Cesa-Bianchi, 2012; Chen et al., 2013a) and the references therein). In recent years, multiple arm identification problem has received much attention and has been investigated under different setups. For example, the work (Even-Dar et al., 2006; Mannor & Tsitsiklis, 2004; Audibert et al., 2010; Karnin et al., 2013) studied the special case when $K = 1$. When $K > 1$, Bubeck et al. (2013) proposed the SAR (Successive Accepts and Rejects) algorithm which minimizes the *misidentification probability* (MISID-PROB), i.e., $\Pr(T \neq \{1,\ldots,K\})$, given a fixed budget (queries). Another line of research (Kalyanakrishnan et al., 2012; Kalyanakrishnan & Stone, 2010) proposed to select a subset T of arms, such that with high probability, for all arms $i \in T$, $\theta_i > \theta_K - \epsilon$, where θ_K is the mean of the K-th best arm. We refer this metric to as the EXPLORE-K metric.

Our notion of aggregate regret is inherently different from MISID-PROB and EXPLORE-K, and is a more suitable objective for many real applications. For example, MISID-PROB requires to identify the exact top-K arms, which is more stringent. When the gap of any consecutive pair θ_i and θ_{i+1} among the first $2K$ arms is extremely small (e.g., $o(\frac{1}{n})$), it requires a huge amount (e.g., $\omega(n^2)$) of samples to make the misidentification probability less than ϵ (Bubeck et al., 2013). While in our metric, any K arms among the first $2K$ arms constitute an ϵ-optimal solution. In crowdsourcing applications, our main goal is not to select the exact top-K workers, but a pool of good enough workers with a small number of samples. Another metric that is related to MISID-PROB is the expected regret $\frac{1}{K}\left(\sum_{i=1}^{K}\theta_i - \mathbf{E}[\sum_{i\in T}\theta_i]\right)$, which has also been considered in a number of prior works (Audibert et al., 2010; Bubeck et al., 2013; Audibert et al., 2013). In (Audibert et al., 2010; Bubeck et al., 2013), the expected regret was shown to be sandwiched by $\Delta\cdot$MISID-PROB and MISID-PROB (for $K=1$), where $\Delta=\theta_1-\theta_2$. However, Δ can be arbitrarily small, hence MISID-PROB can be an arbitrarily bad bound for the regret. It is worthwhile noting that it is possible to obtain an expected regret of ϵ with at most $O(n/\epsilon^2)$ samples, using the semi-bandit regret bound in (Audibert et al., 2013). In contrast, the goal of this paper is to develop an efficient algorithm to achieve an ϵ-regret with high probability.

To compare our aggregate regret with the EXPLORE-K metric, let us consider another example where $\theta_1,\ldots,\theta_{K-1}$ are much larger than θ_K and $\theta_{K+i}>\theta_K-\epsilon$ for $i=1,\ldots,K$. It is easy to see that the set $T=\{K+1,\ldots,2K\}$ also satisfies the requirement of EXPLORE-K However, the set T is far away from the optimal set with the aggregate regret much larger than ϵ. In crowdsourcing, the labeling performance can significantly drop if the best set of workers (e.g., $\theta_1,\ldots,\theta_{K-1}$ in the example) is left out of the solution.

3. Algorithm

In this section, we describe our algorithm for the multiple arm identification problem. Our algorithm OptMAI (Algorithm 1) takes three positive integers n,K,Q as the input, where n is the total number of arms, K is the number of arms we want to choose and Q is an upper bound on the total number of samples[1]. OptMAI consists of two stages, the *Quartile-Elimination (QE) stage* (line 4-6) and the *Accept-Reject (AR) stage* (line 8).

The QE stage proceeds in rounds. Each QE round calls the QE subroutine in Algorithm 2, which requires two parameters S and Q. Here, S is the set of arms which we still want to pull and Q is the total number of samples required in this round. We uniformly sample each arm in S for $Q/|S|$ times and then discard *a quarter* of arms with the minimum empirical mean, which is the average reward in this round. We note that in each call of the QE subroutine, we pass different Q values (exponentially decreasing). This is critical for keeping the total number of samples linear in n and achieving the optimal sample complexity. See Algorithm 1 for the setting of the parameters. The QE stage repeatedly calls the QE subroutine until the number of remaining arms is at most $4K$.

Now, we enter the AR stage, which also runs in rounds. Each AR round (Algorithm 3) requires four parameters, S,T,Q,K, where S,Q have the same meanings as in QE and T is the set of arms that we have decided to include in our final solution and thus will not be sampled any more. In each AR subroutine (Algorithm 3), we again sample each arm for $Q/|S|$ times. We define the *empirical gap* for the i-th arm to be the absolute difference between the empirical mean of the i-th arm and the K'-th (or $(K'+1)$-th) largest empirical mean, where $K'=K-|T|$ (see Eq.(2)). We remove a quarter of arms with the largest empirical gaps. There are two types of those removed arms: those with the largest empirical means, which are included in our final solution set T, and those with the smallest empirical means, which are discarded from further consideration.

Algorithm 1 Optimal Multiple Arm Identification (OptMAI)

1: **Input:** n,K,Q.
2: **Initialization:** Active set of arms $S_0=\{1,\ldots,n\}$; set of top arms $T_0=\emptyset$; $\beta=e^{0.2}\cdot\frac{3}{4}$. Let $r=0$.
3: **while** $|T_r|<K$ and $|S_r|>0$ **do**
4: **if** $|S_r|\geq 4K$ **then**
5: $S_{r+1}=\mathrm{QE}(S_r,\beta^r(1-\beta)Q)$
6: $T_{r+1}=\emptyset$
7: **else**
8: $(S_{r+1},T_{r+1})=\mathrm{AR}(S_r,T_r,\beta^r(1-\beta)Q,K)$
9: **end if**
10: $r=r+1$.
11: **end while**
12: **Output:** The set of the selected K-arms T_r.

Algorithm 2 Quartile-Elimination(QE) (S,Q)

1: **Input:** S,Q.
2: Sample each arm $i\in S$ for $Q_0=\frac{Q}{|S|}$ times and let $\widehat{\theta}_i$ be the empirical mean of the i-th arm.
3: Find the first quartile (lower quartile) of the empirical mean $\widehat{\theta}_a$, denoted by $\widehat{q}$.
4: **Output:** The set $V=S\backslash\{i\in S:\widehat{\theta}_i<\widehat{q}\}$.

[1] If Algorithm 1 stops at round $r=R$, the total number of samples is $(1-\beta^R)Q$, which is less than Q.

Algorithm 3 Accept-Reject(AR) (S, T, Q, K)

Input: S, T, Q, K and $s = |S|$.
Sample each arm $i \in S$ for $Q_0 = \frac{Q}{|S|}$ times and let $\widehat{\theta}_i$ be the empirical mean of the i-th arm.
Let $K' = K - |T|$. Let $\widehat{\theta}_{(K')}$ and $\widehat{\theta}_{(K'+1)}$ be the K'-th and $(K'+1)$-th largest empirical means, respectively. Define the empirical gap for each arm $i \in S$:

$$\widehat{\Delta}_i = \max(\widehat{\theta}_i - \widehat{\theta}_{(K'+1)}, \widehat{\theta}_{(K')} - \widehat{\theta}_i) \tag{2}$$

while $|T| < K$ and $|S| > 3s/4$ **do**
  Let $a \in \arg\max_{i \in S} \widehat{\Delta}_i$ and set $S = S \backslash \{a\}$.
  if $\widehat{\theta}_a \geq \widehat{\theta}_{(K'+1)}$ **then**
    Set $T = T \cup \{a\}$.
  end if
end while
Output: The set S and T.

4. Bounding the Regret and the Sample Complexity

We analyze the regret achieved by our algorithm. All the detailed proofs in this section are provided in the appendix due to space constraints. Firstly, let us introduce some necessary notations. For any positive integer C, we use $[C]$ to denote the set $\{1, 2, \ldots, C\}$. For any subset S of arms, let $\mathsf{ind}_i(S)$ be the arm in S with the i-th largest mean. We use $\mathsf{val}_C(S)$ to denote the average mean of the C best arms in S, i.e., $\mathsf{val}_C(S) \triangleq \frac{1}{C}\sum_{i=1}^{C} \theta_{\mathsf{ind}_i(S)}$. Let $\mathsf{tot}_C(S) = C \cdot \mathsf{val}_C(S)$ be the total sum of the means of the C best arms in S. We first consider one QE round. Suppose S is the set of input arms and V is the output set. We show that the average mean of the K best arms in V is at most ϵ-worse than that in S, for some appropriate ϵ (depending on Q and $|S|$).

Lemma 4.1 *Assume that $K \leq |S|/4$ and let V be the output of $\mathrm{QE}(S, Q)$ (Algorithm 2). For every $0 < \delta < 1$, with probability $1-\delta$, we have that $\mathsf{val}_K(V) \geq \mathsf{val}_K(S) - \epsilon$, where $\epsilon = \sqrt{\frac{|S|}{Q}\left(10 + \frac{4\ln(2/\delta)}{K}\right)}$.*

We further provide the regret bound for the AR algorithm in the following lemma.

Lemma 4.2 *Let (S', T') be the output of the algorithm $\mathrm{AR}(S, T, Q, K)$ (Algorithm 3). For every $0 < \delta < 1$, with probability $1 - \delta$, we have that*

$$\mathsf{tot}_{K-|T'|}(S') + \mathsf{tot}_{|T'|}(T') \geq \mathsf{tot}_{K-|T|}(S) + \mathsf{tot}_{|T|}(T) - \epsilon K,$$

where $\epsilon = \sqrt{\frac{|S|}{Q}\left(4 + \frac{\ln(2/\delta)}{K}\right)}$.

In each round of the AR-stage with $S = S_r$ and top arms $T = T_r$, the value $\frac{\mathsf{tot}_{K-|T|}(S)+\mathsf{tot}_{|T|}(T)}{K}$ is the best possible average mean of the set of K arms in S which contains T. Lemma 4.2 provides an upper bound for the gap between this value on the output (T', S') by AR and the best possible one. Applying this bound over all rounds would further imply that this value of the output of Algorithm 1 is not far away from that of the real top-K arms. With Lemma 4.1 and Lemma 4.2 in place, we prove the performance of Algorithm 1 in the next theorem.

Theorem 4.3 *For every $0 < \delta < 1$ and sample budget $Q > 0$, with probability at least $1-\delta$, the output of OptMAI algorithm T is an ϵ-optimal solution (i.e., $\mathsf{val}_K(T) \geq \mathsf{val}_K([n]) - \epsilon$) with $\epsilon = O\left(\sqrt{\frac{n}{Q}\left(1 + \frac{\ln(1/\delta)}{K}\right)}\right)$. Moreover, each arm is sampled by at most $O(Q/n^{0.3})$ times.*

Theorem 4.3 also provides us the sample complexity of Algorithm 1 for any pre-fixed positive values ϵ and δ, as stated in the next corollary.

Corollary 4.4 *For any $\epsilon > 0$ and $0 < \delta < 1$, it suffices to run Algorithm 1 with*

$$Q = O\left(\frac{n}{\epsilon^2}\left(1 + \frac{\ln(1/\delta)}{K}\right)\right). \tag{3}$$

in order to obtain an ϵ-optimal solution with probability $1-\delta$. In other words, the sample complexity Q of the algorithm is bounded by $O\left(\frac{n}{\epsilon^2}\left(1 + \frac{\ln(1/\delta)}{K}\right)\right)$.

In fact, for $K \geq n/2$, we can obtain a better sample complexity as follows.

Theorem 4.5 *For any $0 < \delta < 1$ and $K \geq n/2$, with probability at least $1 - \delta$, there is an algorithm that can find an ϵ-optimal solution T and the number of samples used is at most*

$$O\left(\left(\frac{n-K}{K} \cdot \frac{n}{\epsilon^2}\right)\left(\frac{n-K}{K} + \frac{\ln(1/\delta)}{K}\right)\right). \tag{4}$$

When $K \geq n/2$, instead of identifying the best K arms, we can easily adapt Algorithm 1 to find the worst $(n - K)$ arms with the smallest $(n - K)$ θ_i's. In particular, the algorithm in Theorem 4.5 can be constructed as follows. Whenever we obtain a sample of value x, we use $1 - x$ as the sample value and apply Algorithm 1 to identify the top $n - K$ arms. Then we output the remaining K arms as the solution. Applying Corollary 4.4, the result of Theorem 4.5 directly follows. According to our proof in the appendix, the constants hiding in Big-O in (4) and that in (3) are the same; and since $\frac{n-K}{K} \leq 1$ when $K \geq n/2$, the bound in (4) is strictly sharper than that in (3). Also, the complexity bound in (4) captures the trivial case that when $K = n$ (i.e, selecting all arms), the sample complexity should be zero.

Remark 4.1 *To achieve the desired bounds on the regret and sample complexity, the AR stage can be substituted by a simpler process which takes a uniform number of samples from each arm and chooses the K arms with the largest empirical means. The details can be found in the appendix. We choose to present the AR subroutine in Algorithm 1 because 1) it also meets the theoretical bound in Section 4; 2) it leads to much better empirical performance.*

Remark 4.2 *The naive uniform sampling algorithm, which takes the same number of samples from each arm and chooses the K arms with the largest empirical means, does not achieve the optimal sample complexity. In general, it requires at least $\Omega(n\log(n))$ samples, which is $\log(n)$ factor worse than our optimal bound. See appendix for a detailed discussion.*

5. A Matching Lower Bound

In this section, we provide lower bounds for Bernoulli bandits where the reward of the i-th arm follows a Bernoulli distribution with mean θ_i. We prove that there is an underlying $\{\theta_i\}_{i=1}^n$ such that for any randomized algorithm $\mathcal{A}$, in order to identify an ϵ-optimal solution with probability at least $1-\delta$, the expected number of samples Q is required to be at least $\max\left\{\Omega\left(\frac{n\ln(1/\delta)}{\epsilon^2 K}\right), \Omega\left(\frac{n}{\epsilon^2}\right)\right\} = \Omega\left(\frac{n}{\epsilon^2}\left(1+\frac{\ln(1/\delta)}{K}\right)\right)$ when $K \le n/2$. When $K \ge n/2$, by a simple argument as in Theorem 4.5, the lower sample complexity bound should be $\Omega\left(\left(\frac{n-K}{K}\cdot\frac{n}{\epsilon^2}\right)\left(\frac{n-K}{K}+\frac{\ln 1/\delta}{K}\right)\right)$. According to Corollary 4.4 and Theorem 4.5, for Bernoulli bandits, our algorithm achieves the lower bound of the sample complexity for all K. To show the lower bound for $K \le n/2$, and we separate the proof into two parts: $Q \ge \Omega\left(\frac{n}{\epsilon^2}\right)$ and $Q \ge \Omega\left(\frac{n\ln(1/\delta)}{\epsilon^2 K}\right)$.

5.1. First Lower Bound for $K \le n/2$: $Q \ge \Omega\left(\frac{n}{\epsilon^2}\right)$

Theorem 5.1 *Fix the real number ϵ such that $0 < \epsilon \le 0.01$, and integers K, n such that $10 \le K \le n/2$. Let $\mathcal{A}$ be a possibly randomized algorithm, so that for any set of n Bernoulli arms with means $\theta_1, \theta_2, \ldots, \theta_n$,*

- *$\mathcal{A}$ takes at most Q samples in expectation;*
- *with probability at least 0.8, $\mathcal{A}$ outputs a set T of size K with $\mathsf{val}_K(T) \ge \mathsf{val}_K([n]) - \epsilon$.*

Then, we have that $Q \ge \Omega(\frac{n}{\epsilon^2})$.

The high level idea of the proof of Theorem 5.1 is as follows. Suppose there is an algorithm $\mathcal{A}$ which can find an ϵ-optimal solution with probability at least 0.8 and uses at most Q samples in expectation. We show that we can use $\mathcal{A}$ as a subroutine to construct an algorithm $\mathcal{B}$, which can distinguish whether a single Bernoulli arm has the mean $1/2$ or $1/2+4\epsilon$ with probability at least 0.51 (above half) with at most $\frac{200Q}{n}$ samples (Lemma 5.2). We utilize the well known result that, for any algorithm (including $\mathcal{B}$), distinguishing whether a Bernoulli arm has the mean $1/2$ or $1/2+4\epsilon$ with probability 0.51 requires at least $\Omega(\frac{1}{\epsilon^2})$ samples. Hence, we must have that $\frac{200Q}{n} \ge \Omega(\frac{1}{\epsilon^2})$, which gives the desired lower bound for Q.

Algorithm 4 Algorithm $\mathcal{B}$ (calls $\mathcal{A}$ as a subroutine)

1: Choose a random subset $S \subseteq [n]$ such that $|S| = K$ and then choose a random element $j \in S$.
2: Create n artificial arms as follows: For each $i \in [n], i \ne j$, let $\theta_i = \frac{1}{2}+4\epsilon$ if $i \in S$, let $\theta_i = \frac{1}{2}$ otherwise.
3: Simulate $\mathcal{A}$ as follows: whenever $\mathcal{A}$ samples the i-th arm:
   (1) If $i = j$, we sample the Bernoulli arm X;
   (2) Otherwise, we sample the arm with mean θ_i.
4: If the arm X is sampled by less than $\frac{200Q}{n}$ times and $\mathcal{A}$ returns a set T such that $j \notin T$, we decide that X has the mean of $\frac{1}{2}$; otherwise we decide that X has the mean of $\frac{1}{2}+4\epsilon$.

Lemma 5.2 *Let $\mathcal{A}$ be an algorithm in Theorem 5.1. There is an algorithm $\mathcal{B}$, which correctly outputs whether a Bernoulli arm X has the mean $\frac{1}{2}+4\epsilon$ or the mean $\frac{1}{2}$ with probability at least 0.51, and $\mathcal{B}$ uses at most $\frac{200Q}{n}$ samples.*

Assuming the existence of an algorithm $\mathcal{A}$ stated in Theorem 5.1, we construct the algorithm $\mathcal{B}$ in Algorithm 4. Keep in mind that the goal of $\mathcal{B}$ is to distinguish whether a given Bernoulli arm (denoted as X) has the mean $1/2$ or $1/2+4\epsilon$. From Algorithm 4, the number of samples of $\mathcal{B}$ increases by one whenever X is sampled. Since $\mathcal{B}$ stops and outputs the mean $\frac{1}{2}+4\epsilon$ if the number of samples on X reaches $\frac{200Q}{n}$, $\mathcal{B}$ takes at most $\frac{200Q}{n}$ samples from X. The intuition why the above algorithm can separate X is as follows. If X has the mean $1/2+4\epsilon$, X is no different from any other arm in S. Similarly, if X has the mean $1/2$, X is the same as any other arm in $[n]\setminus S$. If $\mathcal{A}$ satisfies the requirement in Theorem 5.1, $\mathcal{A}$ can identify a significant proportion of arms with mean $1/2+4\epsilon$. So if X has the mean $1/2+4\epsilon$, there is a good chance (noticeably larger than 0.5) that X will be chosen by $\mathcal{A}$. In the appendix, we formally prove the correctness of $\mathcal{B}$, i.e., it can correctly output the mean of X with probability at least 0.51; and thus conclude the proof of Lemma 5.2.

The second step of the proof of Theorem 5.1 is a well-known lower bound on the expected sample complexity for separating a single Bernoulli arm (Chernoff, 1972; Anthony & Bartlett, 1999).

Lemma 5.3 *Fix ϵ such that $0 < \epsilon < 0.01$ and let X be a Bernoulli random variable with mean being either $\frac{1}{2} + 4\epsilon$ or $\frac{1}{2}$. If an algorithm $\mathcal{B}$ can output the correct mean of X with probability at least 0.51, then expected number of samples performed by $\mathcal{B}$ is at least $\Omega(\frac{1}{\epsilon^2})$.*

By combining Lemma 5.2 and Lemma 5.3, we have $\frac{200Q}{n} \geq \Omega(\frac{1}{\epsilon^2})$; and therefore prove the claim that $Q \geq \Omega(\frac{n}{\epsilon^2})$ in Theorem 5.1.

5.2. Second Lower Bound for $K \leq n/2$: $Q \geq \Omega\left(\frac{n \ln(1/\delta)}{\epsilon^2 K}\right)$

Lemma 5.4 *Fix real numbers δ, ϵ such that $0 < \delta, \epsilon \leq 0.01$, and integers K, n such that $K \leq n/2$. Let $\mathcal{A}$ be a deterministic algorithm (i.e., the only randomness comes from the arms), so that for any set of n Bernoulli arms with means $\theta_1, \theta_2, \ldots, \theta_n$,*

- *$\mathcal{A}$ makes at most Q samples in expectation;*
- *with probability at least $1 - \delta$, $\mathcal{A}$ outputs a set T of size K with $\mathsf{val}_K(T) \geq \mathsf{val}_K([n]) - \epsilon$.*

Then, we have that $Q \geq \frac{n \ln(1/\delta)}{20000\epsilon^2 K}$.

The proof of Lemma 5.4 generalizes the proof for the lower bound when $K = 1$ in (Mannor & Tsitsiklis, 2004). Further, Lemma 5.4 can be easily generalized to the case where $\mathcal{A}$ is randomized, which leads to a stronger lower bound statement in the next theorem. The proofs of Lemma 5.4 and Theorem 5.5 are relegated to the appendix.

Theorem 5.5 *Fix real numbers δ, ϵ such that $0 < \delta, \epsilon \leq 0.01$, and integers K, n, such that $K \leq n/2$. Let $\mathcal{A}$ be a (possibly randomized) algorithm so that for any set of n Bernoulli arms with the mean $\theta_1, \theta_2, \ldots, \theta_n$,*

- *$\mathcal{A}$ makes at most Q samples in expectation;*
- *With probability at least $1 - \delta$, $\mathcal{A}$ outputs a set T of size K with $\mathsf{val}_K(T) \geq \mathsf{val}_K([n]) - \epsilon$.*

We have that $Q = \Omega\big(\frac{n \ln(1/\delta)}{\epsilon^2 K}\big)$.

By combining Theorem 5.1 and Theorem 5.5, we obtain the lower bound $Q = \Omega\left(\frac{n}{\epsilon^2}\left(1 + \frac{\ln(1/\delta)}{K}\right)\right)$, which further indicates that our sample complexity bound in Corollary 4.4 is sharp when $K \leq n/2$. By the equivalence between identifying the best K-arms and the worst $n - K$ arms as in the argument of Theorem 4.5, we could further establish the following lower bound, which indicates that the sample complexity bound in Theorem 4.5 is also sharp.

Theorem 5.6 *Fix real numbers δ, ϵ such that $0 < \delta, \epsilon \leq 0.01$, and integers K, n such that $K \geq n/2$. Let $\mathcal{A}$ be a (possibly randomized) algorithm such that for any set of n Bernoulli arms. Suppose that $\mathcal{A}$ can output an ϵ-optimal set T of size K, with probability at least $1 - \delta$, using at most Q samples in expectation. We have that*

$$Q = \Omega\left(\left(\frac{n-K}{K} \cdot \frac{n}{\epsilon^2}\right)\left(\frac{n-K}{K} + \frac{\ln(1/\delta)}{K}\right)\right).$$

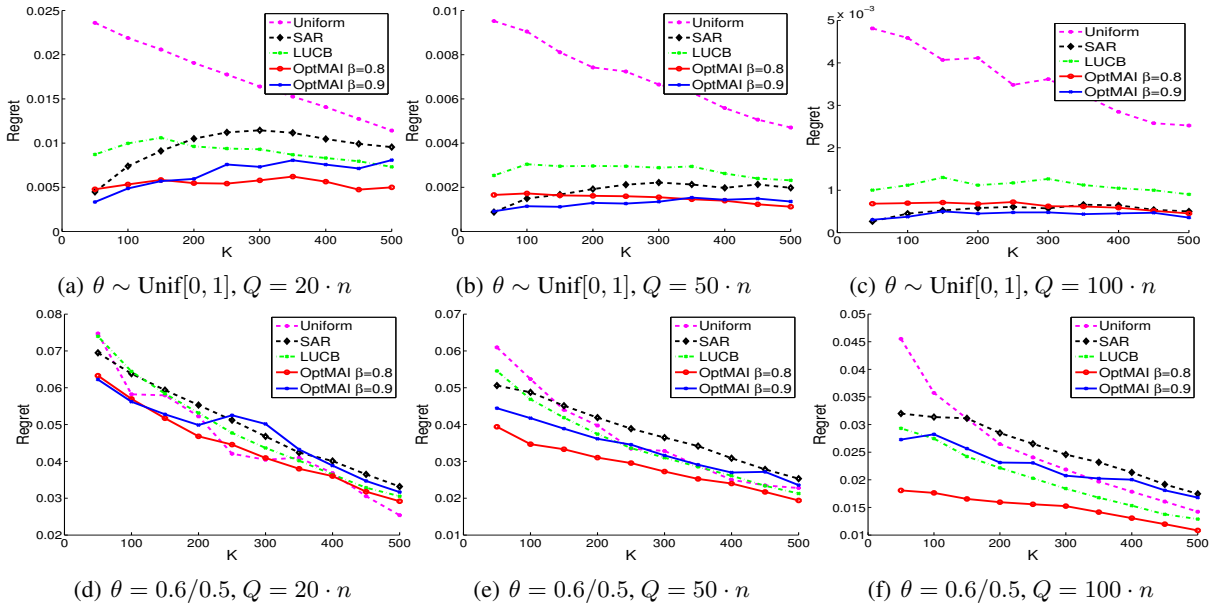
6. Experiments

In this experiment, we assume that arms follow independent Bernoulli distributions with different means. To make a fair comparison, we fix the total budget Q and compare our algorithm (OptMAI) with the uniform sampling strategy and two other state-of-the-art algorithms: SAR (Bubeck et al., 2013) and LUCB (Kalyanakrishnan et al., 2012), in terms of the aggregate regret in (1). For each experiment, we plot the average result over 100 independent runs.

The implementation of our algorithm is slightly different from its description in Section 3. First, observe that in OptMAI, Q is an upper bound of the number of samples; while $(1 - \beta^R)Q < Q$ is the actual number of samples used, where R is the total number of rounds run by the algorithm. Since all the competitor algorithms use Q samples in total, to make a fair comparison, we run OptMAI with the parameter Q', which is slightly greater than Q. In particular, since $R \leq \frac{\ln n}{\ln(4/3)}$, we could set $Q' = \frac{Q}{1-\beta^{(\ln n)/(\ln 4/3)}}$ so that the actual number of samples used by the proposed algorithm roughly equals to (but no greater than) Q. Second, in each round of QE or AR, when computing the empirical mean $\widehat{\theta}_i$, our implementation uses all the samples obtained for the i-th arm (i.e. including the samples from previous rounds). This will lead to better empirical performance especially when the budget is very limited. We note that SAR also reuses the samples from previous rounds. Third, in each round of OptMAI, the ratio of the number of samples between two consecutive rounds is set to be $\beta = e^{0.2} \cdot 0.75 \approx 0.91$. In the real implementation, one could treat this quantity as a tuning parameter to make the algorithm more flexible (as long as $\beta \in (0.75, 1)$). In this experiment, we report the results for both $\beta = 0.8$ and $\beta = 0.9$. Based on our experimental results, one could simply set $\beta = 0.8$, which will lead to reasonably good performance under different scenarios.

6.1. Simulated Experiments

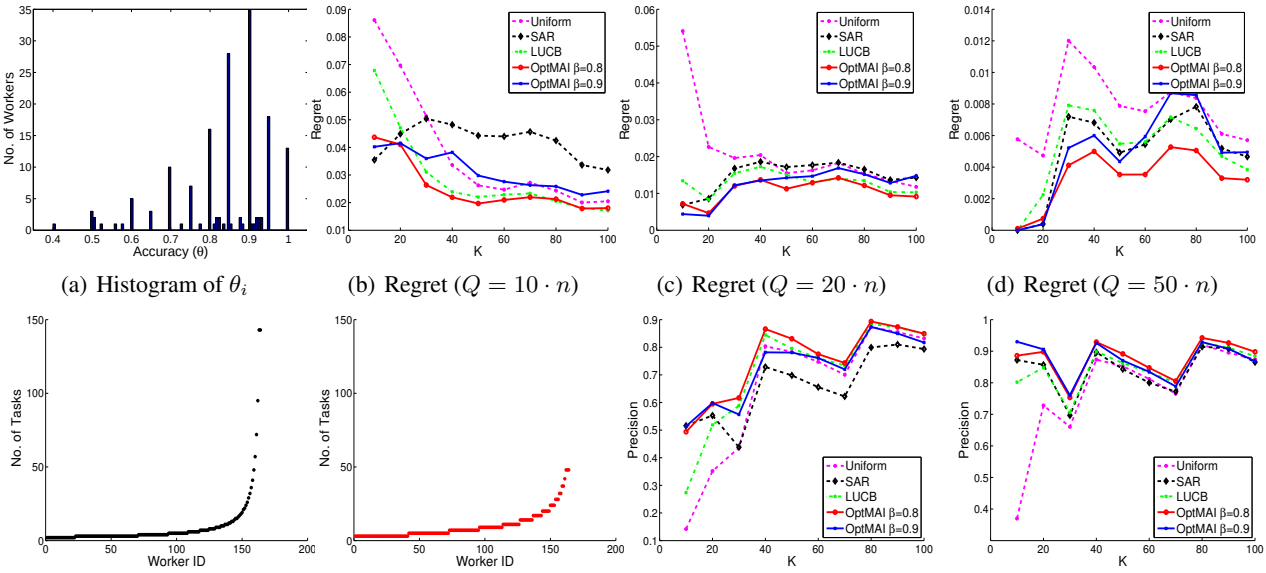
In our simulated experiment, the number of total arms is set to $n = 1000$. We vary the total budget $Q = 20n, 50n, 100n$ and $K = 10, 20, \ldots, 500$. We use different ways to generate $\{\theta_i\}_{i=1}^n$ and report the comparison results among



(a) $\theta \sim \text{Unif}[0,1]$, $Q = 20 \cdot n$ (b) $\theta \sim \text{Unif}[0,1]$, $Q = 50 \cdot n$ (c) $\theta \sim \text{Unif}[0,1]$, $Q = 100 \cdot n$

(d) $\theta = 0.6/0.5$, $Q = 20 \cdot n$ (e) $\theta = 0.6/0.5$, $Q = 50 \cdot n$ (f) $\theta = 0.6/0.5$, $Q = 100 \cdot n$

Figure 1. Performance comparison on simulated data.



(a) Histogram of θ_i (b) Regret ($Q = 10 \cdot n$) (c) Regret ($Q = 20 \cdot n$) (d) Regret ($Q = 50 \cdot n$)

(e) No. of Tasks (SAR) (f) No. of Tasks (OptMAI) (g) Precision ($Q = 10 \cdot n$) (h) Precision ($Q = 20 \cdot n$)

Figure 2. Performance comparison on the RTE data.

different algorithms:

1. $\theta_i \sim \text{Unif}[0,1]$: each θ_i is uniformly distributed on $[0,1]$ (see Figure 1(a) to Figure 1(c)).

2. $\theta_i = 0.6/0.5 : \theta_i = 0.6$ for $i = 1, \ldots, K$ and $\theta_i = 0.5$ for $i = K+1, \ldots, n$. We note that such a two point setting of θ_i is more challenging for selecting top-K arms (see Figure 1(d) to Figure 1(f)).

In Figure 1, the x-axis represents the parameter K and the y-axis represents the regret in (1). It can be seen from Figure 1 that the uniform sampling performs the worst and our method outperforms SAR and LUCB in most of the scenarios. We also observe that when K is large, the setting of $\beta = 0.8$ (red line) outperforms that of $\beta = 0.9$; while for small K, $\beta = 0.9$ (blue line) is a better choice. In the appendix, we also generate θ_i from the truncated normal distribution and the Beta distribution

and have similar observations.

6.2. Real RTE Data

We generate θ from a real recognizing textual entailment (RTE) dataset (Section 4.3 in (Snow et al., 2008)). There are 800 binary labeling tasks and 164 different workers. Since true labels of tasks are available on this data, we set each θ_i for the i-th worker to be his/her labeling accuracy. The histogram of θ_i is presented in Figure 2(a). We vary the total budget $Q = 10n, 20n, 50n$ and K from 10 to 100 and report the comparison of the regret for different approaches in Figure 2(b) to Figure 2(d). As we can see, our method with $\beta = 0.8$ (red line) outperforms other competitors for most of K's and Q's. SAR performs the best when $K = 10$, $Q = 10n$; while our method with $\beta = 0.9$ performs the best when $K = 10$ and $Q = 20n$.

In addition, we would like to highlight an interesting property of our method. As shown in Figure 2(e) and Figure 2(f) with $Q = 10n$ and $K = 20$, the empirical distribution of the number of samples (i.e., tasks) assigned to a worker using SAR is much more skewed than that using our method. This property makes our method particularly suitable for crowdsourcing applications since it will be extremely time-consuming if a single worker is assigned with too many tasks. For example, for SAR, a worker could receive up to 143 tasks (Figure 2(e)) while for our method, a worker receives at most 48 tasks (Figure 2(f)). In crowdsourcing, a single worker will take a long time and soon lose patience when performing nearly 150 testing tasks. Such an empirical observation can be theoretically justified by Theorem 4.3 (see discussions at the end of the introduction part). We also note that LUCB has the most even empirical distribution of the number of tasks assigned to a worker: a workers receives at most 17 tasks and at least 6 tasks.

In Figure 2(g) and Figure 2(h), we compare different algorithms in terms of the precision, which is defined as the number of arms in T which belong to the set of the top K arms over K, i.e., $\frac{|T \cap [K]|}{K}$. As we can see, our method with $\beta = 0.8$ achieves the highest precision followed by LUCB.

7. Acknowledgement

This work was carried out at the Simons Institute for the Theory of Computing at UC Berkeley. We are grateful to the Simons Institute for providing us such a great research environment. Yuan Zhou is supported by a grant from the Simons Foundation (Award Number 252545). Jian Li is supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61202009, 61033001, 61361136003. We would like to thank Michael I. Jordan, Qihang Lin and Denny Zhou for helpful discussions and anonymous reviewers who gave valuable suggestions to help improve the manuscript.

References

Anthony, M and Bartlett, P. L. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Audibert, J.-Y, Bubeck, S, and Lugosi, G. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 2013.

Audibert, J, Bubeck, S, and Munos, R. Best arm identification in multi-armed bandits. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, 2010.

Auer, P, Cesa-Bianchi, N, Freund, Y, and Schapire, R. The nonstochastic multiarmed bandit problem. *SIAM J. on Comput.*, 32(1):48–77, 2002a.

Auer, P, Cesa-Bianchi, N, and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002b.

Beygelzimer, A, Langford, J, Li, L, Reyzin, L, and Schapire, R. E. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2011.

Bubeck, S and Cesa-Bianchi, N. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

Bubeck, S, Wang, T, and Viswanathan, N. Multiple identifications in multi-armed bandits. In *Proceedings of the International Conference on Machine Learning*, 2013.

Chen, W, Wang, Y, and Yuan, Y. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of International Conference on Machine Learning*, 2013a.

Chen, X, Lin, Q, and Zhou, D. Optimistic knowledge gradient for optimal budget allocation in crowdsourcing. In *Proceedings of International Conference on Machine Learning*, 2013b.

Chernoff, H. *Sequential Analysis and Optimal Design*. Society for Industrial and Applied Mathematics (SIAM), 1972.

Even-Dar, E, Mannor, S, and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7:1079–1105, 2006.

Ho, C.-J, Jabbari, S, , and Vaughan, J. W. Adaptive task assignment for crowdsourced classification. In *ICML*, 2013.

Kalyanakrishnan, S and Stone, P. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of International Conference of Machine Learning*, 2010.

Kalyanakrishnan, S, Tewari, A, Auer, P, and Stone, P. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of International Conference on Machine Learning*, 2012.

Karger, D. R, Oh, S, and Shah, D. Budget-optimal task allocation for reliable crowdsourcing systems. arXiv:1110.3564v3, 11 2012.

Karnin, Z, Koren, T, and Somekh, O. Almost optimal exploration in multi-armed bandits. In *Proceedings of International Conference on Machine Learning*, 2013.

Koenig, L. W and Law, A. M. A procedure for selecting a subset of size m containing the l best of k independent normal populations, with applications to simulation. *Communications in statistics. Simulation and computation*, 14:719–734, 1985.

Mannor, S and Tsitsiklis, J. N. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5:623–648, 2004.

Raykar, V. C, Yu, S, Zhao, L. H, Valadez, G. H, Florin, C, Bogoni, L, and Moy, L. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

Schmidt, C, Branke, J, and Chick, S. E. Integrating techniques from statistical ranking into evolutionary algorithms. 2006.

Snow, R, Connor, B. O, Jurafsky, D, and Ng., A. Y. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, 2008.

Thompson, W. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

Zhou, D, Basu, S, Mao, Y, and Platt, J. Learning from the wisdom of crowds by minimax conditional entropy. In *NIPS*, 2012.