Collaborative Top Distribution Identifications with Limited Interaction (Extended Abstract) [§]

Nikolai Karpov*, Qin Zhang*, Yuan Zhou†
* *Computer Science Department, Indiana University*
nkarpov@iu.edu *and* qzhangcs@indiana.edu
† *Department of ISE, University of Illinois at Urbana-Champaign*
yuanz@illinois.edu

2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-7281-9621-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/FOCS46700.2020.00024

***Abstract*—We consider the following problem in this paper: given a set of n distributions, find the top-m ones with the largest means. This problem is also called *top-m arm identifications* in the literature of reinforcement learning, and has numerous applications. We study the problem in the collaborative learning model where we have multiple agents who can draw samples from the n distributions in parallel. Our goal is to characterize the tradeoffs between the running time of learning process and the number of rounds of interaction between agents, which is very expensive in various scenarios. We give optimal time-round tradeoffs, as well as demonstrate complexity separations between top-1 arm identification and top-m arm identifications for general m and between fixed-time and fixed-confidence variants. As a byproduct, we also give an algorithm for selecting the distribution with the m-th largest mean in the collaborative learning model.**

I. Introduction

In this paper we study the following problem: given a set of n distributions, try to find the m ones with the largest means via sampling. We study the problem in the multi-agent setting where we have K agents, who try to identify the top-m distributions collaboratively via communication. Suppose sampling from each distribution takes a unit time, our goal is to minimize both the running time and the number of rounds of communication of the collaborative learning process.

The problem of top-m distribution identifications originates from the literature of *multi-armed bandits* (MAB) [52], where each distribution is called an *arm*, and each sampling from a distribution is called an *arm pull*. When $m = 1$, the problem is called *best arm identification*, and has been studied extensively in the centralized setting where there is only one agent [5], [11], [24], [27], [45], [39], [33], [40], [20], [13], [29]. Some of these algorithms can be easily modified to handle top-m arm identification (e.g., [5], [12]). The problem of best arm identification has also been studied in the multi-agent collaborative learning model [31], [54]. Surprisingly, we found that in the multi-agent setting, the tasks of identifying the best arm and the top-m arms look to be very different in terms of problem complexities; the algorithm design and lower bound proof for the top-m case require significantly new ideas, and need to address some fundamental challenges in collaborative learning.

Collaborative Learning with Limited Interaction. A natural way to speed up machine learning tasks is to introduce multiple agents, and let them learn the target function collaboratively. In recent years some works have been done to address the power of parallelism (under the name of *concurrent learning*, e.g., [50], [30], [23], [22]). Most of these works assume that agents have the full ability of communication. That is, they can send/receive messages to/from each other at *any* time step. This assumption, unfortunately, is unrealistic in real-world applications, as it would be very expensive to implement unrestricted communication, which is usually the biggest drain of time, data, energy and network bandwidth. For example, once we deploy sensors/robots to unknown environment such as deep sea and outer space, it would be almost impossible to recharge them; when we train a model in a central server by interacting with hundreds of thousands of mobile devices, the communication cost will directly contribute to our data bills, not mentioning the excessive energy and bandwidth consumption.

In this paper we consider the model of *collaborative learning with limited interaction*, where the learning process is partitioned into rounds of predefined time intervals. In each round, each of the K agents takes a series of actions individually like in the centralized model, and they can only communicate at the end of

[§]Full version in https://arxiv.org/abs/2004.09454. Nikolai Karpov and Qin Zhang are supported in part by NSF IIS-1633215, CCF-1844234, and CCF-2006591. Yuan Zhou is supported in part by NSF CCF-2006526 and a JPMorgan Chase AI Research Faculty Research Award.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00024

each round. At the end of the last round before any communication, all agents should agree on the same output; otherwise we say the algorithm fails. Our goal is to minimize both the number of rounds of computation R and the running time T (assuming each action takes a unit time step).[1]

Naturally, there is a tradeoff between R and T: If $R = 1$, that is, no communication is allowed, then $T \geq T_{\mathsf{C}}$ where T_{C} is the running time of the best centralized algorithm. When R increases, T may decrease. On the other hand we always have $T \geq T_{\mathsf{C}}/K$ even when $R = T$. We are mostly interested in understanding the number of rounds needed to achieve almost *full speedup*, that is, when $T = \tilde{O}(T_{\mathsf{C}}/K)$ where $\tilde{O}(\cdot)$ hides logarithmic factors.

We do not put any constraints on the lengths of the messages that each agent can send at the end of each round, but in the MAB setting they will not be very large – the information that each agent collects can always be compressed to an array of n pairs in the form of $(x_i, \tilde{\theta}_i)$, where x_i is the number of arm pulls on the i-th arm, and $\tilde{\theta}_i$ is the empirical mean of the x_i arm pull.

Top-m Arm Identification. To be consistent with the MAB literature, we will use the term *arm* instead of *distribution* throughout this paper. The top-m arm identification problem is motivated by a variety of applications ranging from industrial engineering [41] to medical tests [55], and from evolutionary computation [49] to crowdsourcing [1]. The readers may refer to [5], [36], [21], [18], [19] and references therein for the state-of-the-art results on the top-m arm identification in the centralized model.

In this paper we mainly focus on the *fixed-time* case, where given a fixed time horizon T, the task is to identify the set of m arms with the largest means with the smallest error probability. We will also discuss the *fixed-confidence* case, where given a fixed error probability δ, the task is to identify the top-m arms with error δ using the smallest amount of time.

Without loss of generality, we assume that each of the underlying distributions has support on $(0, 1)$. In the centralized setting, Bubeck et al. [12] introduced the following complexity to characterize the hardness of an input instance V for the top-m arm identification problem. Let θ_i be the mean of the i-th arm. Let $[j]$ be the index of the arm in V with the j-th largest mean, and let $\theta_{[j]}(V)$ be the corresponding mean. Given an input instance I of n arms, let $\Delta_i^{\langle m \rangle}(I)$ be the gap between the mean of the i-th arm and that of the $[m]$-th arm or the $[m+1]$-th arm, whichever is larger. In other words,

$$\Delta_i^{\langle m \rangle}(I) \triangleq \begin{cases} \theta_i - \theta_{[m+1]}(I), \text{if } \theta_i \geq \theta_{[m]}(I), \\ \theta_{[m]}(I) - \theta_i, \text{if } \theta_i \leq \theta_{[m+1]}(I). \end{cases} \quad (1)$$

Definition 1 (Instance Complexity). *Given an input instance I of n arms and a parameter m (call it the* pivot*), we define the following quantity which characterizes the complexity of I.*

$$H^{\langle m \rangle}(I) \triangleq \sum_{i \in I} \left(\Delta_i^{\langle m \rangle}(I) \right)^{-2}.$$

*We also define a related quantity which we call the ϵ-*truncated instance complexity.

$$H_\epsilon^{\langle m \rangle}(I) \triangleq \sum_{i \in I} \max \left\{ \Delta_i^{\langle m \rangle}(I), \epsilon \right\}^{-2}.$$

To see why $H^{\langle m \rangle}(I)$ is the right measure for the instance complexity, note that if the mean of an arm is either $(\theta + \Delta)$ or $(\theta - \Delta)$ where θ is a known threshold, it takes $\Omega(\Delta^{-2})$ samples to decide whether the mean is above or below the threshold θ (as long as $\theta \pm \Delta$ are bounded away from 0 and 1). Therefore, suppose all the means are bounded away from 0 and 1, even if we are given the means of the $[m]$-th and the $[m+1]$-th arms, it still takes $\Omega(H^{\langle m \rangle}(I))$ samples to decide for each arm whether it is one of the top-m arms or not. Such intuition can be formalized to show that, in the fixed-confidence case, $\Omega(H^{\langle m \rangle}(I) \log(1/\delta))$ samples are needed to identify the top-m arms with success probability $(1-\delta)$ [51], [19]. On the other hand, there are centralized algorithms to achieve $O(H^{\langle m \rangle}(I) \log(1/\delta) + H^{\langle m \rangle}(I) \log H^{\langle m \rangle}(I))$ (see, e.g., [36]), almost matching the lower bound (up to logarithmic factors).

For the fixed-time case, in [12] it was shown that there is a centralized algorithm that identifies the top-m arms with probability at least

$$1 - \exp\left(-\tilde{\Omega}\left(\frac{T}{H^{\langle m \rangle}(I)} \right) \right) \quad (2)$$

using at most T time steps, where $\tilde{\Omega}(\cdot)$ hides logarithmic factors in n. This upper bound can also be shown to be tight up to logarithmic factors [40], [13], [51], [19]. In the collaborative learning setting, our goal is to replace the T factor in (2) with KT where K is the number of agents, so as to achieve a full speedup.

Our Contributions. We summarize our main results and their implications.

[1] We note that our model is a simplified version of the one formulated in [54]. The model defined in [54] allows each agent to perform different numbers of actions in each round, and the length of each round can be determined adaptively by the agents. However, we noticed that all the existing algorithms for collaborative learning in the literature have predefined round lengths, under which there is no point for an agent to stop early in a round.

1) We give an algorithm for the fixed-time top-m arm identification problem in the collaborative learning model with K agents and a set I of n arms. For any choice of r, the algorithm uses T time steps and $O(\log \frac{\log m}{\log K} + r)$ rounds of communication, and successfully computes the set of top-m arms with probability at least $1-\exp\left(-\tilde{\Omega}\left(\frac{K^{(R-1)/R}\cdot T}{H^{\langle m\rangle}(I)}\right)\right)$. In particular, when $r = \log K$, the algorithm uses T time steps and $O(\log \frac{\log m}{\log K} + \log K)$ rounds of communication to compute the set of top-m arms with probability at least $1-\exp\left(-\tilde{\Omega}\left(\frac{KT}{H^{\langle m\rangle}(I)}\right)\right)$, achieving a full speedup. See Section III.
2) We prove that under the same setting, any collaborative algorithm that uses $T = \frac{1}{\sqrt{K}}\cdot H^{\langle m\rangle}(I)$ time steps and aims to achieve success probability 0.99 needs at least $\Omega(\log \frac{\log m}{\log K})$ rounds of communication. By leveraging a result in [54], we can also show that any collaborative algorithm that uses $T = \frac{\alpha}{K}\cdot H^{\langle m\rangle}(I)$ time steps and aims to achieve success probability 0.99 needs at least $\Omega(\log K/(\log\log K + \log\alpha))$ rounds of communication. These indicate that our upper bound is almost the best possible. See Section IV.
3) Our lower bound gives a strong separation between the best arm identification and top-m identifications: there is a collaborative algorithm for best arm identification (i.e., when $m = 1$) that uses $T = \tilde{O}\left(\frac{1}{\sqrt{K}}\cdot H^{\langle 1\rangle}(I)\right)$ time and 2 rounds of communication (see [54], [31]), while Item 2 states that for general m, to achieve the same time bound we need $\Omega(\log K/(\log\log K + \log\alpha))$ rounds of communication.
4) We give an algorithm for the fixed-confidence top-m identification problem in the collaborative model with K agents and a set of n arms; the algorithm uses $O\left(\frac{H^{\langle m\rangle}(I)}{K}\log\left(\frac{n}{\delta}\log H^{\langle m\rangle}\right)\right)$ time steps and $O\left(\log(1/\Delta^{\langle m\rangle}_{[m]})\right)$ rounds of communication, and successfully computes the set of top-m arms with probability at least $1-\delta$. This is almost tight by a previous result in [54]. See Section V.
5) Combining Items 1, 2, and 4, we have given a separation between fixed-time and fixed-confidence top-m arm identification. We note that a similar separation result is also proved for the best arm identification problem [54], although the round complexities for top-m identification are quite different from the $m = 1$ special case (i.e., best arm identification).

Speedup. In [54] the authors introduced a concept called *speedup* for presenting the power of collaborative learning algorithms. The precise definition of speedup is rather complicated due to the definition of the instance complexity of MAB. Roughly, the speedup is defined to be the ratio between the *best* running time of centralized algorithm and that of a collaborative algorithm (given a predefined round budget R) under the condition that the two algorithms achieve the same success probability. In this paper we simply focus on a fixed success probability 0.99, and define the speedup of a collaborative algorithm which identifies the top-m arms on input instance I with accuracy 0.99 using time $T_{\mathcal{A}}(I)$ to be $T_{\mathcal{A}}(I)/H^{\langle m\rangle}(I)$, since the best centralized algorithm achieving success probability 0.99 has running time $\tilde{\Theta}(H^{\langle m\rangle}(I))$ [12]. Interpreting our results in terms of speedup, we have the following remarks:

1) Our algorithm for fixed-time top-m arm identification achieves a speedup of $\tilde{O}(K^{\frac{r-1}{r}})$ and uses $O(\log \frac{\log m}{\log K} + r)$ rounds.
2) Our lower bound shows that in order to achieve even an $\tilde{\Omega}(\sqrt{K})$ speedup, any algorithm for top-m arm identification needs at least $\Omega(\log \frac{\log m}{\log K})$ rounds.
3) Compared with the main result for the best arm identification in [54], which states that there is a R-round algorithm achieving a speedup of $\tilde{O}(K^{\frac{R-1}{R}})$, we have shown a separation between the complexities of the two problems (e.g., when $R = 2$).

Selection under Uncertainty. As a byproduct, we also get almost tight bounds for a closely related problem we call *selection under uncertainty*. This problem is similar to the classic *selection* problem where given a set of n numbers, one needs to find the m-th largest number. The difference is that now instead of having n (deterministic) numbers, we have n distributions/arms, and our goal is to find the one with the m-th largest mean via sampling. It is easy to see that this problem can be solved by first identifying the top-m arms, and then finding the worst arm in these top-m arms, which can be done in the same way as identifying the best arm.

For convenience, let us introduce a new (but very similar) definition of instance complexity for the selection under uncertainty problem:

$$\bar{H}^{\langle m\rangle}(I) \triangleq \sum_{i\neq[m]} (\theta_i - \theta_{[m]})^{-2}.$$

With $\bar{H}^{\langle m\rangle}$ we have the following immediate result: There exists an algorithm for the fixed-time m-th arm

selection problem in the collaborative learning model with K agents and a set I of n arms; the algorithm uses T time steps and $O(\log \frac{\log m}{\log K} + r)$ rounds of communication, and successfully identifies the m-th arm with probability at least

$$1 - \exp\left(-\tilde{\Omega}\left(\frac{K^{(r-1)/r} \cdot T}{\bar{H}^{\langle m \rangle}(I)}\right)\right).$$

Why Top-m Arm Identification is Difficult in the Collaborative Learning Model? Before presenting our results, let us first try to give some intuition on why top-m arm identification is difficult in the collaborative learning setting, as one may think that the top-m arm identification is a natural generalization of best arm identification (when $m = 1$), and the algorithm for the latter in [54] may be adapted to the former.

The key procedure used in previous collaborative algorithms for best arm identification [31], [54] is that in the first round, we *randomly* partition the set I of n arms into K groups, and feed each group to one agent as a subproblem. Now if each of the K agents computes the best arm in its subproblem, then we can reduce the number of best arm candidates from n to K after the first round, which is critical for us to achieve $\log K$ communication rounds. The question now is whether each subproblem can be solved time-efficiently (more precisely, in $\tilde{O}(H^{\langle 1 \rangle}(I)/K)$ time steps if we target a $\tilde{\Omega}(K)$ speedup) at each agent in the first round.

A nice property for the best arm identification is that if we randomly partition the set I of n arms to the K groups, then the group (denoted by G) containing the global best arm has a subproblem complexity $H' = \sum_{i=2}^{|G|} (\Delta'_i)^{-2}$, where Δ'_i is the difference between the mean of the best arm and that of the i-th best arm in group G. It is easy to show that

$$\mathbb{E}[H'] = \Theta\left(H^{\langle 1 \rangle}(I)/K\right). \tag{3}$$

Therefore, even though we cannot guarantee that each of the K subproblems can be solved successfully under time budget $\tilde{O}(H/K)$, we still know that the global best arm will advance to the next round with a good probability, which is enough for the algorithm to succeed.

Unfortunately, the above property does not hold in the top-m setting due to its "multi-objective" goal. First, the global m-th arm will only be assigned to one agent, and thus others do not know what pivot to use for defining its subproblem complexity. Second, even for the agent who gets the m-th arm j, it does not know what is the local rank of j, and, thus, still does not know when to stop the local pruning. Third, even if the agents know the local ranks of the m-th arm, it may not have enough time budget to solve the sub-problem; note that this is an issue only for the top-m case but *not* for the best arm case, since in the top-m case each subproblem may contain some top-m arms.

We will design an algorithm which addresses all of these challenges, and then complement it with an almost tight lower bound. Looking back, we feel that in the best arm case it was just lucky for us to have Equation (3), while in the general top-m case we have to deal with some inherent challenges in collaborative learning, which, unfortunately, also make our algorithm for top-m much more complicated than that for best arm identification. We will give a technical overview for both the algorithm and lower bound proof in Section II.

Related Work. To the best of our knowledge, the collaborative learning model studied in this paper was first proposed in [31], where the authors studied the best arm identification problem in MAB. The model was recently formalized in [54], where almost tight time-round tradeoffs for best arm identification are given.

A number of works studied *regret minimization*, which is another important problem in MAB, in various distributed models, most of which are different from the collaborative learning model considered in this paper. For example, several works [44], [48], [9] studied regret minimization in the setting of cognitive ratio network, where radio channels are models as arms, and the rewards by pulling each arm depend on the number of simultaneous pulls by the K agents (i.e., penalty is introduced for collisions). In [16] the authors considered a model where at each time step each agent can choose either to pull an arm, or broadcast a message to other agents, but cannot do both. Authors of [53], [42], [57] considered regret minimization in communication networks. Distributed regret minimization has also been studied in the non-stochastic setting [6], [37], [15].

The collaborative learning model is closely related to the *batched* model (or, *learning with limited adaptivity*), where one wants to minimize the number of policy switches in the learning process. In the batched model we want to minimize the number of policy switches when trying to achieve our learning goal. Algorithms designed in the batched model can naturally be translated to a *restricted* version of the collaborative model in which at each time step, the action taken by each agent is determined by the information (historical actions and outcomes, messages received from other agents, and the randomness of the algorithm) the agent has at the beginning of the round, and the agents cannot change their policies in the middle of the a round. A number of problems have been studied in the batched

n	number of arms in the input instance.
K	number of agents.
T	running time.
θ_i	mean of the i-th arm.
$\theta_{[i]}(V)$	the i-th largest mean among arms in V.
$Top_m(V)$	indices of the m arms with the largest means in V.
$Top_1(V)$	index of the best arm in V.
$\Delta_i^{\langle m \rangle}(V)$	mean gap of the i-th arm; defined in (1).
$H^{\langle m \rangle}(V)$	instance complexity; see Definition 1.
$H_\epsilon^{\langle m \rangle}(V)$	ϵ-truncated instance complexity; see Definition 1.

Table I: Summary of Notations

model in recent years, including best arm identification [35], [2], [34], regret minimization in MAB [47], [28], [26], Q-learning [7], convex optimization [25], online learning [14]. We note that our collaboratively learning algorithm for top-m arm identification in the fixed-confidence case also works in the batched model, and improves the algorithm in [34].

Finally, we note that there is also a large body of work on sample/communication-efficient distributed algorithms for various learning-related tasks such as classification [8], [32], [38], convex optimization [59], [58], [3], linear programming [4], [56]. Sample-efficient PAC learning in the collaborative setting is recently studied by [10], [17], [46]. However, the models considered in the papers mentioned above mainly focus on reducing the sample/communication cost, and are all different from the collaborative learning with limited interaction model we study in this paper.

Notations and Conventions. Let $Top_m(V)$ be the indices of m arms in V with the largest means, and $Top_1(V)$ be the index of the best arm in V.

We say the i-th arm is (ϵ, j)-top in V if and only if $\theta_i \geq \theta_{[j]}(V) - \epsilon$. Similarly, the i-th arm is (ϵ, j)-bottom in V if and only if $\theta_i \leq \theta_{[|V|+1-j]}(V) + \epsilon$.

In this paper we focus on the case when $\theta_{[m]}(I) > \theta_{[m+1]}(I)$, since otherwise the instance complexity of I will be infinity.

For simplicity, we will write $Top_m(V)$, $Top_1(V)$, $\theta_{[i]}(V)$, $\Delta_i^{\langle m \rangle}(V)$, $H^{\langle m \rangle}(V)$, and $H_\epsilon^{\langle m \rangle}(V)$ as Top_m, Top_1, $\theta_{[i]}$, $\Delta_i^{\langle m \rangle}$, $H^{\langle m \rangle}$, and $H_\epsilon^{\langle m \rangle}$, when $V = I$ (I is the input instance) or it is clear from the context.

We include a list of notations in Table I.

Roadmap. In the rest of this paper, we first give a technical overview of our results in Section II. We present our algorithmic result for the fixed-time case in Section III, and complement it with a matching lower bound in Section IV. Finally in Section V, we state our results for the fixed-confidence case.

II. Technical Overview

In this section we give a technical overview for our upper and lower bounds for fixed-time top-m arm identification.

A. Upper Bounds for the Fixed-Time Setting

For simplicity we consider the full speedup setting (i.e., we target a speedup of $\tilde{\Omega}(K)$); the general speedup is an easy extension. We achieve our upper bound result for fixed-time top-m arm identification in three stages. We first design an algorithm for a special time horizon $T = \tilde{\Theta}(H^{\langle m \rangle}/K)$ which uses $O(\log \frac{\log n}{\log K} + \log K)$ rounds of communication and has an error probability 0.01. We next consider general time horizon T, and target an error probability that is exponentially small in T. Finally, we try to improve the round complexity to $O(\log \frac{\log m}{\log K} + \log K)$. In each stage we face new challenges which stem from the collaborative learning model, each of which demands novel ideas.

Stage 1: A Basic Algorithm. We start with our basic algorithm. A natural idea for achieving the $T = \tilde{O}(H^{\langle m \rangle}/K)$ running time is to *randomly* partition the n arms to K agents, and then ask each agent to solve a top-η arms identification (for some value η) on its sub-instance. At the end we try to aggregate the K outputs. As briefly mentioned in the introduction, there are multiple hurdles associated with this approach. First, it is not clear how to set the value η, since we do not know how many global top-m arms will be distributed to each agent. Second, even if we know the number of global top-m arms assigned to each agent, there are cases in which the global instance complexity is rarely distributed evenly across the K agents. In other words, we cannot guarantee that each agent can solve the subproblem within our time budget $\tilde{O}(H^{\langle m \rangle}/K)$.

We resolve these issues using the following ideas: we take a *conservative* approach by setting $\eta \approx (m/K - \sqrt{n})$, and ask each agent to adopt a PAC algorithm for multiple arm identification and compute an *approximate* set of top-η arms on its sub-instance using $\tilde{O}(H^{\langle m \rangle}/K)$ time steps. The approximation error is a *random variable* depending on the random partition process. We then show that with a good probability this error is smaller than the gap between the smallest mean of the outputted arms and that of the *global* m-th arm. In this way we can guarantee that the approximate top-η arms outputted by each agent are indeed in the set of global top-m arms. Using the same idea we try to prune a set of "bottom" arms of size $\approx ((n-m)/K - \sqrt{n})$. After these operations we recurse on the rest $O(K\sqrt{n})$ arms. We continue the recursion for $O(\log \frac{\log n}{\log K})$ rounds until

the number of arms is reduced to K^{10}, and then use a simple $O(\log n)$-round collaborative algorithm which is modified from an existing centralized algorithm. Note that for $n' = K^{10}$ we have $O(\log n') = O(\log K)$, and thus overall we have used $O(\log \frac{\log n}{\log K} + \log K)$ rounds.

Stage 2: General Time Horizon. The basic algorithm only guarantees that the set of top-m arms are correctly identified with probability 0.99. Our next goal is to make the error probability exponentially small in T, which is achievable in the centralized setting. The standard technique to achieve this is to perform parallel repetition and then take the majority. That is, we guess the instance complexity to be $H = 1, 2, 4, \ldots$, and for each guess we run the basic algorithm with time horizon H for T/H times. Finally, we take the majority of the output. In the case that the budget T is larger than the actual instance complexity, at each run with probability 0.99 we are guaranteed to obtain the correct answer. Unfortunately, when T is smaller than the actual instance complexity, not much can be guaranteed. For some bad input instances, the output of the basic algorithm can be *consistently* wrong, resulting in a wrong majority.

We resolve this difficulty by introducing a notion we call *top-m certificate*, which takes form of a pair $(S, \{\tilde{\theta}_i\}_{i\in I})$, with the property that $S = \mathit{Top}_m$ and for each $i \in I$, it holds that $\left|\tilde{\theta}_i - \theta_i\right| < \Delta_i^{\langle m\rangle}/4$. We can augment our basic algorithm to output a $(S, \{\tilde{\theta}_i\}_{i\in I})$ pair (instead of simply a set of top-m arms). We then design a verification algorithm which is able to check for each $(S, \{\tilde{\theta}_i\}_{i\in I})$ pair whether it is indeed a top-m certificate. Our verification step can be fully parallelized and can finish within our guessed instance complexity H. With such a verification step at hand, the situation that we take a wrong majority will *not* happen with high probability.

Stage 3: Better Round Complexity. Our ultimate goal is to achieve an $O(\log \frac{\log m}{\log K} + \log K)$ round complexity, instead of $O(\log \frac{\log n}{\log K} + \log K)$ in the basic algorithm. We approach this by first reducing the number of arms in the input instance to $\tilde{O}(m)$, and then applying the basic algorithm. Such a reduction, however, is highly non-trivial, especially when we require the error probability introduced by the reduction to again be exponentially small in T.

Our basic idea for performing the reduction is the following: we construct a random sub-instance V by sampling each of the n arms with probability $1/m$. We can show that with constant probability, V contains exactly one global top-m arm, and $H^{\langle 1\rangle}(V) = O(H^{\langle m\rangle}/m)$. Therefore we have enough time budget to compute the best arm of V and include it into set S as a top-m candidate. We perform this subsampling procedure for $\tilde{O}(m)$ times, getting $\tilde{O}(m)$ sub-instances. By the Coupon Collector's problem we know that all global top-m arms will be included in S with a good probability.

The challenging part is to reduce the error probability of this reduction to a value that is exponentially small in T. Unfortunate, the idea of "guess-then-verify" that we have used previously does not apply here – there is simply no $(S, \{\tilde{\theta}_i\}_{i\in I})$ pair for us to verify in the reduction process.

We take the following new approach. We try to make sure that for each randomly sampled sub-instance on which we try to compute the best arm, the probability of outputting any arm in Top_m is at least *half* of that of any arm outside Top_m. This turns out to be enough for us to guarantee that the set S contains all top-m arms. We comment that the relaxation "half" is necessary here for a technical reason which we will elaborate next.

Our key observation is that if we provide sufficient time budget, say, $T \geq \lambda H^{\langle 1\rangle}(V)$ where λ is a polylogarithmic factor, for solving a randomly sampled sub-instance V, then provided that there is only one arm $a \in \mathit{Top}_m$ in V, we will output a correctly with a good probability. Now for any two arms $a \in \mathit{Top}_m$ and $b \notin \mathit{Top}_m$, by the uniformity of the sampling they will be in the sub-instance with equal probability. We are thus able to conclude that the probability of outputting a is at least as large as that of outputting b. On the other hand, if $T \leq H^{\langle 1\rangle}(V)$, then we can use our verification step to detect this event. The subtle part is the middle case when $H^{\langle 1\rangle}(V) \leq T \leq \lambda H^{\langle 1\rangle}(V)$, to handle which we perturb our time budget T such that it takes values T/λ or λT with equal probability. Using this trick we are able to "reduce" the third case to the first two cases with probability at least $1/2$, which leads to our desired property. The actual implementation of this idea is more involved, and we refer the readers to the full version for details.

B. Lower Bounds for the Fixed-Time Setting

In the lower bound part, we present two results. The first result is that $\Omega(\log K/(\log\log K + \log\alpha))$ communication rounds are needed for any algorithm with (K/α) speedup to identify the top-m arms for any m. This matches (up to logarithmic factors) the R term in the $O(\log \frac{\log m}{\log K} + R)$ rounds vs $\tilde{O}(K^{(R-1)/R})$ speedup trade-off in our upper bound result. This lower bound theorem is derived via a simple reduction together with

the similar type of lower bound proved in [54] for the $m = 1$ special case.

Our main contribution in the lower bound part is the second theorem. The theorem states that even if the goal is an $O(\sqrt{K})$ speedup, the $\log \frac{\log m}{\log K}$ term in the round-speedup trade-off is necessary. (In fact, the $\log \frac{\log m}{\log K}$ can be shown to be necessary for any K^{ζ} speedup where ζ is a positive constant.) This marks a completely different phenomenon from the $m = 1$ special case where only 2 rounds of communication are needed to achieve an $\tilde{O}(\sqrt{K})$ speedup [54], [31]. Below we sketch the proof idea for this lower bound theorem.

The need for the $\log \frac{\log m}{\log K}$ term in the round complexity stems from the hardness of collaboratively learning the splitting position (i.e., where the m-th largest arm locates), which turns out to be substantially more difficult than estimating the best arm (the $m = 1$ special case). We start from the fact that any (possibly randomized) algorithm cannot identify the number of 1's in the n-bit binary vector with success probability $\omega(n^{-1/2})$, if the algorithm is allowed to probe only $o(n)$ entries in the vector. A strengthened statement we will prove as the building block is the following lower bound for the "learning the bias" problem: given n Bernoulli arms (i.e., the stochastic reward of the arm is either 0 or 1), each of which has mean reward $(\mu+\epsilon)$ or $(\mu-\epsilon)$, then any algorithm using $o(n\epsilon^{-2}/\log(n/\epsilon))$ samples will not be able to identify the number of two types of arms with probability $\omega(n^{-1/2})$.[2]

Now we explain the connection between the learning the bias problem and the top-m arm identification problem by sketching the plan of constructing the hard instances as follows. Suppose that we set all but $n^{1/2}$ arms in the hard instance to be Bernoulli with mean reward either $(\mu+\epsilon)$ (namely "the top arms") or $(\mu-\epsilon)$ (namely "the bottom arms"). We denote the set of the rest $n^{1/2}$ arms by M, and their mean rewards are sandwiched between $(\mu + \epsilon)$ and $(\mu - \epsilon)$. We will set $m = n/2$, i.e., the goal is to identify the top half of the arms. Now, as long as the number of top arms, denoted by X, is bounded between $\frac{n}{2} - \sqrt{n}$ and $\frac{n}{2} + \sqrt{n}$, the goal is equivalent to identify the X top arms and the top-$(\frac{n}{2} - X)$ arms in M. We then vary the number of the top arms and consequently the number of the bottom arms (say, let X be uniformly randomly chosen from the range), and will argue that each agent will not be able to identify X much better than a random guess without communication, and therefore must perform one round of communication to learn X in order to identify the top-$(\frac{n}{2} - X)$ arms in M. Here, the need for communication is due to the lower bound for learning the bias and the fact that any agent in a $\sqrt{K}$-speedup algorithm is allowed to make only $O(n\epsilon^{-2}/\sqrt{K}) = o(n\epsilon^{-2})$ samples (where we make a crucial assumption that the $H^{\langle m \rangle}$ complexity of the constructed hard instance is $O(n\epsilon^{-2})$). The last piece of plan is to argue that since X is not known before the first round of communication, each agent cannot make much progress before the communication towards identifying the top-$(\frac{n}{2} - X)$ arms in M, which is a necessary sub-task. We will finally inductively prove a communication lower bound for this sub-task. Note that the number of arms in M is $n^{1/2}$, and this plan will lead to a $\log \log n$-style round complexity lower bound.

There are several challenges for the plan above. Note that in the sub-task for M, the goal is no longer to identify the top half arms, which is not well aligned with the (planned) induction hypothesis. Moreover, to make the induction work, M would naturally have the similar structure as the n-arm instance, i.e., with many top and bottom arms (possibly with different μ and ϵ parameters). However, such a construction would hardly ensure that the $H^{\langle m' \rangle}$ complexity is still $O(n\epsilon^{-2})$. Indeed, if the goal of the sub-task is to identify, for example, the top $|M|/4$ arms, since most of the top half arms are the same, the corresponding the $H^{\langle m' \rangle}$ complexity would become infinitely large. Finally, it is not clear how to make sure that any agent will not gain much information about M before the first round of communication so as to quickly identify the top $(\frac{n}{2} - X)$ arms in M whenever X is learned.

To address these challenges, we craft a more complex distribution of hierarchical instances. The main highlight is that we let M consist of multiple blocks $I_1, I_2, \ldots, I_k$, where each block has the same number of arms and is independently sampled from a recursively defined hard distribution. We restrict the possible values of $(\frac{n}{2} - X)$ to be the half multiples of the block size so that the sub-task always becomes to identify the top half arms in I_ξ for some $\xi \in \{1, 2, \ldots k\}$. We will make careful selection of the block parameters so that the $H^{\langle m \rangle}$ complexity for any instance in the support of the distribution, and the $H^{\langle m' \rangle}$ complexity of any sub-task, are all $\tilde{\Theta}(n\epsilon^{-2})$, where both upper and lower bounds are crucial to the proof.

[2]The sample complexity lower bound for a similar problem is proved in a recent work [43]. Our lower bound is different from theirs in two aspects. First, in their setting, the number of arms is not bounded and the goal is to estimate the fraction of the two types of arms up to an additive error, while in our setting, the number of arms is n, and the goal is to find out the exact numbers of arms for the two types. Second, their lower bound is for algorithms with constant success probability, while our lower bound is for algorithms with only $\omega(n^{-1/2})$ success probability.

III. A Collaborative Algorithm for the Fixed-Time Case

A. Preparation

We first introduce two centralized algorithms `CenAppTop` and `CenAppBtm` for computing (ϵ, m)-top/bottom arms. We leave their detailed description to the full version of the paper. The following lemma summarizes the guarantees of these two algorithms.

Lemma 1. *Let I be a set of n arms, $m \in \{1, \ldots, n-1\}$, and $\epsilon \in (0,1)$ be an approximation parameter. Let*

$$T_1(I, a, \epsilon, \delta) = c_1 H^{\langle a \rangle}_{\epsilon/2}(I) \cdot \log\left(H^{\langle a \rangle}_{\epsilon/2}(I)/\delta\right)$$

for a universal constant c_1. We have that

- *If $T \geq T_1(I, m, \epsilon, \delta)$, then with probability at least $(1-\delta)$, $\texttt{CenAppTop}(I, m, T, \delta)$ returns m arms each of which is (ϵ, m)-top in I using at most T time steps.*
- *If $T \geq T_1(I, n-m, \epsilon, \delta)$, then with probability at least $(1-\delta)$, $\texttt{CenAppBtm}(I, m, T, \delta)$ returns m arms each of which is (ϵ, m)-bottom in I using at most T time steps.*

The following lemma says that there is a simple collaborative algorithm `CollabTopMSimple` for top-m arm identification that uses $O(\log n)$ rounds of communication. Note that this bound is still much larger than our final target $O(\log\log m + \log K)$ rounds. `CollabTopMSimple` is a simple modification of a centralized algorithm in [12], and will be described in details in the full version of the paper.

Lemma 2. *Let I be a set of n arms, and $m \in \{1, \ldots, n-1\}$. Let*

$$T_2(I, m, \delta) = c_2 \cdot \frac{H^{\langle m \rangle}(I)}{K} \cdot \log n \cdot \log \frac{n}{\delta} \qquad (4)$$

for a universal constant c_2. There is a collaborative algorithm $\texttt{CollabTopMSimple}(I, m, T)$ such that if $T \geq T_2(I, m, \delta)$ then with probability at least $1-\delta$, one computes the set of top-m arms of I using at most T time steps and $O(\log n)$ rounds.

B. Special Time Horizon T

We are able to establish the following theorem concerning a special time horizon T.

Theorem 3. *Let I be a set of n arms, and $m \in \{1, \ldots, n-1\}$. Let*

$$T_0 = c_0 \frac{H^{\langle m \rangle}}{K}\left(\log\left(H^{\langle m \rangle} K\right) + \log^2 n\right)\log^{(2)} n \qquad (5)$$

for a large enough constant c_0. There exists a collaborative algorithm $\texttt{CollabTopM}(I, m, T)$ that computes the set of top-m arms of I with probability at least 0.99 when $T \geq T_0$, and uses at most T time steps and $O(\log \frac{\log n}{\log K} + \log K)$ rounds of communication.

Algorithm 1: $\texttt{CollabTopM}(I, m, T)$

Input: a set of n arms I, parameter m, and time horizon T.
Output: the set of top-m arms of I.

Let R be the global upper bound on the number of rounds and δ be also the global parameter equal to $1/(100R)$;
$q \leftarrow 4K\sqrt{n \log(nR)}$;
if $n > K^{10}$ **then**
  $Acc \leftarrow \emptyset$, $Rej \leftarrow \emptyset$;
  randomly assign each arm in I to one of the K agents, and let I_i be the set of arms assigned to i-th agent;
  if $m > q$ **then**
    $\ell \leftarrow (m-q)/K$;
    for *agent* $i = 1$ *to* K **do**
      $Acc_i \leftarrow \texttt{CenAppTop}\left(I_i, \ell, \frac{T}{4R}, \frac{\delta}{2K}\right)$;
    $Acc \leftarrow \bigcup_{i=1}^{K} Acc_i$;
  if $n - m > q$ **then**
    $r \leftarrow (n-m-q)/K$;
    for *agent* $i = 1$ *to* K **do**
      $Rej_i \leftarrow \texttt{CenAppBtm}\left(I_i, r, \frac{T}{4R}, \frac{\delta}{2K}\right)$;
    $Rej \leftarrow \bigcup_{i=1}^{K} Rej_i$;
  return $Acc \cup \texttt{CollabTopM}(I \setminus (Acc \cup Rej), m - |Acc|, T)$;
else
  return $\texttt{CollabTopMSimple}(I, m, T/2)$.

Our algorithm is described in Algorithm 1. Note that we have used *recursion* instead of *iteration* to omit a superscript r. But we still call each recursive step a *round*.

While the detailed proof of Theorem 3 is deferred to the full version of the paper, here we briefly state the intuition behind the algorithm and its analysis. At the beginning of each round we first randomly partition the set of arms to the K agents. Then each agent tries to identify a subset of arms Acc_i of size $\ell \approx (m/K - \sqrt{n})$ to be included to Top_m, and a subset of arms Rej_i of size $r \approx ((n-m)/K - \sqrt{n})$ to be pruned. The intuition to introduce the additive $\sqrt{n}$ term is that by a concentration bound, we have with a good probability that at least ℓ true top-m arms will be assigned to each agent, and similarly at least r non-top-m arms

will be assigned to each agent. However, even with this fact, we still cannot guarantee that each agent can identify the top and bottom arms successfully given its limited budget, which is approximately $H^{\langle m \rangle}/K$. Such a budget in some sense demands that the global instance complexity is *evenly* divided into the K agents, which is not necessary true. We thus adopt a PAC algorithm for top-m arm identification which returns a set of ℓ (ϵ, ℓ)-top arms at each agent A_i, where ϵ is a random variable which, with a high probability, is smaller than the gap between the ℓ-th top arm *locally* at A_i and that of the m-th *global* top arm. In this way we can guarantee that it is safe to include each Acc_i that A_i computes into Top_m. By essentially the same arguments, we can show that it is safe to prune the set of bottom arms Rej_i.

C. General Time Horizon T

Theorem 3 only achieves a constant error probability for a special case of the time horizon $T = \tilde{\Theta}(T_0)$ where $T_0 = H^{\langle m \rangle}/K$. Our next goal is to consider general time horizon $T \geq T_0$, and try to make the error probability decrease exponentially with respect to T/T_0. More precisely, we have the following theorem.

Theorem 4. *Let I be a set of n arms, and $m \in \{1, \ldots, n-1\}$. Let T be a time horizon. There exists a collaborative algorithm* `CollabTopMGeneral` *that computes the set of top-m arms of I with probability at least*

$$1 - n \cdot \exp\left(-\Omega\left(\frac{TK}{H^{\langle m \rangle} \cdot (\log(H^{\langle m \rangle} K) + \log^2 n) A}\right)\right),$$

where $A = \log\log n \cdot \log^2\left(TK/H^{\langle m \rangle}\right)$, using at most T time steps and $O\left(\log \frac{\log n}{\log K} + \log K\right)$ rounds.

The proof of Theorem 4 is deferred to the full version of the paper. Here, we explain its high level idea. A standard technique to achieve an error probability that is exponentially small in terms of T/T_0 is to perform parallel repetition and then take the majority. This is straightforward if we know the value T_0. Unfortunately, T_0 depends on the instance complexity which we do not know in advance. A standard trick to handle this issue is to use the doubling method. That is, we guess $T_0 = 1, 2, 4, \ldots$, and for each value we repeat T/T_0 times (ignoring logarithmic factors). We know that one of these values is very close to the actual T_0. We hope that this value is the *first* value in $\{1, 2, 4, \ldots\}$ for which the T/T_0 runs of `CollabTopM`(I, m, T) contain a majority output.

The main issue in this approach is that when $T \leq T_0$, the output of the algorithm can be *consistently* wrong, which leads to a wrong majority. Note that we do not have much control on the output of the algorithm when the time horizon is very small.

We handle this issue by introducing a concept called *top-m certificate*. We require each algorithm for top-m arm identification to output a pair $(S, \{\tilde{\theta}_i\}_{i \in I})$, where S is a subset of I of size m and $\{\tilde{\theta}_i\}_{i \in I}$ are the estimated means for *all* arms in I (not just those in S). We say a pair $(S, \{\tilde{\theta}_i\}_{i \in I})$ is a top-m certificate if it can pass an additional verification step which checks whether S is indeed the set of top-m arms of I given the estimated means $\{\tilde{\theta}_i\}_{i \in I}$. With such a verification step at hand, we do not need to worry about the case that `CollabTopM` will output a wrong answer when T is too small, since a wrong output will simply *not* pass the verification step. Finally, we make sure that this verification step is perfectly parallelizable and thus fit in our time budget.

D. An Improved Algorithm

We are able further improve the round complexity of Algorithm 1 to $O(\log \frac{\log m}{\log K} + \log K)$. Formally, we prove the following theorem in the full version of the paper.

Theorem 5. *Let I be a set of n arms, $m \in \{1, \ldots, n-1\}$, and T be the time horizon. There exists a collaborative algorithm that computes the set of top-m arms of I with probability at least*

$$1 - n \cdot \exp\left(-\Omega\left(\frac{KT}{H^{\langle m \rangle} B}\right)\right),$$

where

$$B = \log^6(KT) \log^2\left(KT/H^{\langle m \rangle}\right) \log n,$$

using at most T time steps and $O(\log \frac{\log m}{\log K} + \log K)$ rounds.

IV. Lower Bounds for the Fixed-Time Case

In this section, we state the lower bound theorems for the fixed-time setting.

Theorem 6. *For every K, m ($m \leq K$), and α ($\alpha \in [1, K^{0.1}]$), if a fixed-time collaborative algorithm $\mathcal{A}$ with K agents returns the top-m arms for every instance J with probability at least 0.99, when given time budget $\frac{\alpha}{17K} \cdot H^{\langle m \rangle}(J)$, then there exists an instance J' such that $\mathcal{A}$ uses $\Omega(\log K/(\log\log K + \log\alpha))$ rounds of communication in expectation given instance J' and time budget $\frac{\alpha}{17K} \cdot H^{\langle m \rangle}(J')$.*

In other words, to achieve (K/α) speedup for identifying the top m arms, the collaborative algorithm needs $\Omega(\log K/(\log\log K + \log\alpha))$ communication rounds.

The proof of Theorem 6 is relatively easy and resembles the round complexity lower bound $\Omega(\log K/(\log\log K+\log\alpha))$ for top arm identification in the fixed-time setting [54].

Theorem 7. *For every large enough K and m such that $K \geq \Omega(\log^4 m)$, if a fixed-time collaborative algorithm $\mathcal{A}$ with K agents returns the top-m arms for every instance J with probability at least 0.99, when given time budget $\frac{1}{\sqrt{K}}\cdot H^{\langle m\rangle}(J)$, then there exists an instance J' such that $\mathcal{A}$ uses $\Omega(\log(\log m/\log K))$ rounds of communication given instance J' and time budget $\frac{1}{\sqrt{K}}\cdot H^{\langle m\rangle}(J')$.*

In other words, even if one only aims at $\sqrt{K}$ speedup, the collaborative algorithm needs

$$\Omega(\log(\log m/\log K))$$

rounds of communication.

Theorem 7 marks the different round complexity requirement for collaborative multiple arm identification compared to the best arm identification problem. It is known that only constant number of round is needed to achieve 0.99 success probability using $\tilde{O}(K^{-\zeta}\cdot H^{\langle m\rangle}(J))$ time budget (i.e., $\tilde{O}(K^{\zeta})$ speedup) for every constant $\zeta\in(0,1)$ [31], [54]. However, Theorem 7 rules out such possibility for the multiple arm identification problem, proving it much harder than best arm identification in the collaborative setting. We note that we only prove the lower bound for $\zeta=1/2$, for the simplicity of the exposition. However, the proof can be easily extended to any constant $\zeta>0$. The only differences are that, in the theorem statement, the constraint $K\geq\Omega(\log^4 m)$ will become $K\geq\log^{f(\zeta)} m$, and the round complexity lower bound will become $\frac{1}{f(\zeta)}\cdot\log(\log m/\log K)$ where $f(\zeta)>0$ increases as ζ approaches 0.

V. The Fixed-Confidence Case

In this section we discuss the fixed-confidence case. We first present a collaborative algorithm for the fixed-confidence case. The algorithm is inspired by [31] and [12], and described in Algorithm 2.

Algorithm 2: Collaborative algorithm for fixed-confidence setting.

Input: a set of arms I, parameter m, and a confidence parameter δ.
Output: a set of top-m arms of I.

Initialize $I_0\leftarrow I$, $m_0\leftarrow m$, $Acc_0\leftarrow\emptyset$, $Rej_0\leftarrow\emptyset$, $r\leftarrow 0$, $T_{-1}\leftarrow 0$;
for $r=0,1,\ldots$, let $\epsilon_r=2^{-(r+1)}$ and $T_r=8\log(4n(r+1)^2\delta^{-1})/(K\epsilon_r^2)$;
while $I_r\neq\emptyset$ **do**
  each agent pulls each arm in I_r for T_r-T_{r-1} times;
  for each $i\in I_r$, let $\hat{\theta}_i^{(r)}$ be the estimated mean of the i-th arm in I_r after KT_r pulls (over all rounds and agents so far);
  let $\pi_r:\{1,\ldots,|I_r|\}\to I_r$ be the bijection such that $\hat{\theta}^{(r)}_{\pi_r(1)}\geq\hat{\theta}^{(r)}_{\pi_r(2)}\geq\ldots\geq\hat{\theta}^{(r)}_{\pi_r(|I_r|)}$;
  $Acc_{r+1}\leftarrow Acc_r\cup\{i\in I_r:\hat{\theta}_i^{(r)}>\hat{\theta}^{(r)}_{\pi_r(m_r+1)}+\epsilon_r\}$;
  $Rej_{r+1}\leftarrow Rej_r\cup\{i\in I_r:\hat{\theta}_i^{(r)}<\hat{\theta}^{(r)}_{\pi_r(m_r)}-\epsilon_r\}$;
  $m_{r+1}\leftarrow m-|Acc_{r+1}|$;
  $I_{r+1}\leftarrow I_r\setminus\big(Acc_{r+1}\cup Rej_{r+1}\big)$;
  $r\leftarrow r+1$;
return Acc_r.

Theorem 8. *There is an algorithm (Algorithm 2) that solves top-m arm identification with probability at least $1-\delta$, using $O\left(\log\left(1/\Delta^{\langle m\rangle}_{[m]}\right)\right)$ rounds of communication and $O\left(\frac{H^{\langle m\rangle}}{K}\log\left(\frac{n}{\delta}\log H^{\langle m\rangle}\right)\right)$ time.*

Finally we comment on the lower bound. In [54] it was shown that for the special case when $m=1$, to achieve a running time of $\tilde{O}(H^{\langle 1\rangle}/K)$ with success probability 0.99 one needs at least $\log\left(1/\Delta^{\langle 1\rangle}_{[1]}\right)$ rounds. Therefore the upper bound in Theorem 8 is tight up to logarithmic factors.

References

[1] Ittai Abraham, Omar Alonso, Vasilis Kandylas, and Aleksandrs Slivkins. Adaptive crowdsourcing algorithms for the bandit survey problem. In *COLT*, pages 882–910, 2013.

[2] Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *COLT*, pages 39–75, 2017.

[3] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *NIPS*, pages 1756–1764, 2015.

[4] Sepehr Assadi, Nikolai Karpov, and Qin Zhang. Distributed and streaming linear programming in low dimensions. In *PODS*, pages 236–253. ACM, 2019.

[5] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53, 2010.

[6] Baruch Awerbuch and Robert D. Kleinberg. Competitive collaborative learning. In *COLT*, pages 233–248, 2005.

[7] Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient q-learning with low switching cost. In *NeurIPS*, 2019.

[8] Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *COLT*, pages 26.1–26.22, 2012.

[9] Ilai Bistritz and Amir Leshem. Distributed multi-player bandits - a game of thrones approach. In *NeurIPS*, pages 7222–7232, 2018.

[10] Avrim Blum, Nika Haghtalab, Ariel D. Procaccia, and Mingda Qiao. Collaborative PAC learning. In *NIPS*, pages 2392–2401, 2017.

[11] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *ALT*, pages 23–37, 2009.

[12] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *ICML*, pages 258–265, 2013.

[13] Alexandra Carpentier and Andrea Locatelli. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *COLT*, pages 590–604, 2016.

[14] Nicolò Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *NIPS*, pages 1160–1168, 2013.

[15] Nicolò Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. Delay and cooperation in nonstochastic bandits. In *COLT*, pages 605–622, 2016.

[16] Mithun Chakraborty, Kai Yee Phoebe Chua, Sanmay Das, and Brendan Juba. Coordinated versus decentralized exploration in multi-agent multi-armed bandits. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 164–170, 2017.

[17] Jiecao Chen, Qin Zhang, and Yuan Zhou. Tight bounds for collaborative pac learning via multiplicative weights. In *NIPS*, pages 3602–3611, 2018.

[18] Lijie Chen, Anupam Gupta, and Jian Li. Pure exploration of multi-armed bandit under matroid constraints. In *COLT*, pages 647–669, 2016.

[19] Lijie Chen, Jian Li, and Mingda Qiao. Nearly instance optimal sample complexity bounds for top-k arm selection. In *AISTATS*, pages 101–110, 2017.

[20] Lijie Chen, Jian Li, and Mingda Qiao. Towards instance optimal bounds for best arm identification. In *COLT*, pages 535–592, 2017.

[21] Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *NIPS*, pages 379–387, 2014.

[22] Maria Dimakopoulou, Ian Osband, and Benjamin Van Roy. Scalable coordinated exploration in concurrent reinforcement learning. In *NeurIPS*, pages 4223–4232, 2018.

[23] Maria Dimakopoulou and Benjamin Van Roy. Coordinated exploration in concurrent reinforcement learning. In *ICML*, pages 1270–1278, 2018.

[24] Carlos Domingo, Ricard Gavaldà, and Osamu Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Min. Knowl. Discov.*, 6(2):131–152, 2002.

[25] John C. Duchi, Feng Ruan, and Chulhee Yun. Minimax bounds on stochastic batched convex optimization. In *COLT*, pages 3065–3162, 2018.

[26] Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab S. Mirrokni. Batched multi-armed bandits with optimal regret. *CoRR*, abs/1910.04959, 2019.

[27] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and markov decision processes. In *COLT*, pages 255–270, 2002.

[28] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. In *NeurIPS*, 2019.

[29] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *COLT*, pages 998–1027, 2016.

[30] Zhaohan Guo and Emma Brunskill. Concurrent PAC RL. In *AAAI*, pages 2624–2630, 2015.

[31] Eshcar Hillel, Zohar Shay Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. In *NIPS*, pages 854–862, 2013.

[32] Hal Daumé III, Jeff M. Phillips, Avishek Saha, and Suresh Venkatasubramanian. Efficient protocols for distributed classification and optimization. In *ALT*, pages 154–168, 2012.

[33] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil?ucb: An optimal exploration algorithm for multi-armed bandits. In *COLT*, pages 423–439, 2014.

[34] Tianyuan Jin, Jieming Shi, Xiaokui Xiao, and Enhong Chen. Efficient pure exploration in adaptive round model. In *NeurIPS*, pages 6605–6614, 2019.

[35] Kwang-Sung Jun, Kevin G. Jamieson, Robert D. Nowak, and Xiaojin Zhu. Top arm identification in multi-armed bandits with batch arm pulls. In *AISTATS*, pages 139–148, 2016.

[36] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, pages 227–234, 2012.

[37] Varun Kanade, Zhenming Liu, and Bozidar Radunovic. Distributed non-stochastic experts. In *NIPS*, pages 260–268, 2012.

[38] Daniel Kane, Roi Livni, Shay Moran, and Amir Yehudayoff. On communication complexity of classification problems. In *COLT*, pages 1903–1943, 2019.

[39] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *ICML*, pages 1238–1246, 2013.

[40] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *J. Mach. Learn. Res.*, 17:1:1–1:42, 2016.

[41] Lloyd W Koenig and Averill M Law. 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(3):719–734, 1985.

[42] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. On distributed cooperative decision-making in multiarmed bandits. In *ECC*, pages 243–248, 2016.

[43] Jasper CH Lee and Paul Valiant. Uncertainty about uncertainty: Near-optimal adaptive algorithms for estimating binary mixtures of unknown coins. *arXiv preprint arXiv:1904.09228*, 2019.

[44] Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Trans. Signal Processing*, 58(11):5667–5681, 2010.

[45] Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *J. Mach. Learn. Res.*, 5:623–648, 2004.

[46] Huy L. Nguyen and Lydia Zakynthinou. Improved algorithms for collaborative PAC learning. In *NeurIPS*, pages 7642–7650, 2018.

[47] Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. In *COLT*, page 1456, 2015.

[48] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits - a musical chairs approach. In *ICML*, pages 155–163, 2016.

[49] Christian Schmidt, Jürgen Branke, and Stephen E. Chick. Integrating techniques from statistical ranking into evolutionary algorithms. In *Applications of Evolutionary Computing, EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC*, pages 752–763, 2006.

[50] David Silver, Leonard Newnham, David Barker, Suzanne Weller, and Jason McFall. Concurrent reinforcement learning from customer interactions. In *ICML*, pages 924–932, 2013.

[51] Max Simchowitz, Kevin G. Jamieson, and Benjamin Recht. The simulator: Understanding adaptive sampling in the moderate-confidence regime. In *COLT*, pages 1794–1834, 2017.

[52] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

[53] Balázs Szörényi, Róbert Busa-Fekete, István Hegedűs, Róbert Ormándi, Márk Jelasity, and Balázs Kégl. Gossip-based distributed stochastic bandit algorithms. In *ICML*, pages 19–27, 2013.

[54] Chao Tao, Qin Zhang, and Yuan Zhou. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In *FOCS*, pages 126–146, 2019.

[55] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[56] Santosh S. Vempala, Ruosong Wang, and David P. Woodruff. The communication complexity of optimization. *CoRR*, abs/1906.05832, 2019.

[57] Jie Xu, Cem Tekin, Simpson Zhang, and Mihaela Van Der Schaar. Distributed multi-agent online learning based on global feedback. *IEEE Transactions on Signal Processing*, 63(9):2225–2238, 2015.

[58] Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Communication-efficient algorithms for statistical optimization. In *NIPS*, pages 1511–1519, 2012.

[59] Martin Zinkevich, Markus Weimer, Alexander J. Smola, and Lihong Li. Parallelized stochastic gradient descent. In *NIPS*, pages 2595–2603, 2010.