Almost Cost-Free Communication in Federated Best Arm Identification

Srinivas Reddy Kota, P. N. Karthik, and Vincent Y. F. Tan

National University of Singapore
Emails: ksvr1532@gmail.com, karthik@nus.edu.sg, vtan@nus.edu.sg

Abstract

We study the problem of best arm identification in a federated learning multi-armed bandit setup with a central server and multiple clients. Each client is associated with a multi-armed bandit in which each arm yields *i.i.d.* rewards following a Gaussian distribution with an unknown mean and known variance. The set of arms is assumed to be the same at all the clients. We define two notions of best arm—local and global. The local best arm at a client is the arm with the largest mean among the arms local to the client, whereas the global best arm is the arm with the largest average mean across all the clients. We assume that each client can only observe the rewards from its local arms and thereby estimate its local best arm. The clients communicate with a central server on uplinks that entail a cost of $C \geq 0$ units per usage per uplink. The global best arm is estimated at the server. The goal is to identify the local best arms and the global best arm with minimal total cost, defined as the sum of the total number of arm selections at all the clients and the total communication cost, subject to an upper bound on the error probability. We propose a novel algorithm FEDELIM that is based on successive elimination and communicates only in exponential time steps, and obtain a high probability instance-dependent upper bound on its total cost. The key takeaway from our paper is that for any $C \geq 0$ and error probabilities sufficiently small, the total number of arm selections (resp. the total cost) under FEDELIM is at most 2 (resp. 3) times the maximum total number of arm selections under its variant that communicates in every time step. Additionally, we show that the latter is optimal in expectation up to a constant factor, thereby demonstrating that communication is almost cost-free under FEDELIM. We numerically validate the efficacy of FEDELIM on two synthetic datasets and the MovieLens dataset.

Introduction

We study an optimal stopping variant of the federated learning multi-armed bandit (FLMAB) regret minimisation problem of Shi, Shen, and Yang (2021). The specifics of our problem setup are as follows. We consider a federated multi-armed bandit setup with a central server and $M > 1$ clients. Each client is associated with a multi-armed bandit with $K > 1$ arms in which each arm yields independent and identically distributed (*i.i.d.*) rewards following a Gaussian distribution with an unknown mean and known variance. We
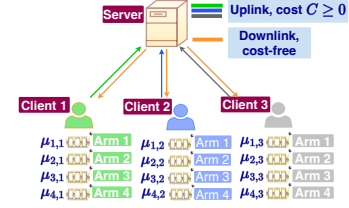
Figure 1: An illustration of our problem setup with $M = 3$ clients and $K = 4$ arms per client. The mean of arm k of client m is $\mu_{k,m}$, where $k \in [K]$ and $m \in [M]$. Communication of a scalar to the server is assumed to entail a cost of $C \geq 0$ units per usage of the uplink, whereas the downlink from the server to the client is cost-free.

assume that the set of arms is identical at all the clients. As in Shi, Shen, and Yang (2021), we consider two notions of best arm—*local* and *global*. The local best arm at a client is defined as the arm with the largest mean among the arms local to the client. The global best arm is the arm with the largest average of the means averaged across the clients (we define these terms precisely later in the paper). We assume that each client can observe the rewards generated *only* from its local arms and thereby estimate its local best arm. The clients do not communicate directly with each other, but instead communicate with the central server. Communication from each client to the server entails a fixed cost of $C \geq 0$ units per usage per uplink. The information transmitted by the clients on the uplink is used by the server to estimate the global best arm. In contrast to the work of Shi, Shen, and Yang (2021) where the goal is to minimise the regret accrued over a finite time horizon, the goal of our work is to find the local best arms of all the clients and the global best arm in a way so as to minimise the sum of the total number of arm pulls at the clients and the total communication cost, subject to an upper bound on the error probability. Figure 1 summarises our problem setup.

Motivation

The following example from the movie industry motivates our problem setup. Movies are typically categorised into various genres (for e.g., comedy, romance, action, thriller, etc.) and released in several parts (regions) of the world. The people of a region develop preferences for one or more genres

courtesy of certain region-specific demographics (for e.g., age profile, females to males ratio of the population, etc.). Suppose that there are M distinct regions and K distinct genres. The following questions are commonplace in the movie industry: (a) What genre of movies is most preferred *locally* by the people of a given region? (b) What genre of movies yields higher profits on the average *globally* across all regions? In the above example, a movie is akin to an arm and a region is akin to a client. The question in (a) above seeks to find the local best arm of each client, whereas the question in (b) seeks to find the global best arm.

Related Works

Federated learning is an emerging paradigm that focuses on a distributed machine learning scenario in which there are multiple clients and a central server training a common machine learning model while keeping each client's local data private; see McMahan et al. (2017) and Kairouz et al. (2021) and the references therein for more details. The work of Shi, Shen, and Yang (2021) extends the federated learning framework to multi-armed bandit paradigm and studies FLMAB under the theme of regret minimisation wherein the goal is to design arm selection algorithms to minimise the regret accrued over a finite time horizon. See Lattimore and Szepesvári (2020) and the references therein for more details on the regret minimisation theme and other related works on this theme. Contrary to the theme of regret minimisation, *best arm identification* falls under the theme of optimal stopping and can be embedded within the classical framework of Chernoff (1959). As noted in the work of Bubeck, Munos, and Stoltz (2011) and Zhong, Cheung, and Tan (2021), algorithms that are optimal in the context of regret minimisation are not necessarily so in the context of optimal stopping.

The problem of best arm identification is well-studied and consists in finding the best arm (i.e., the arm with the largest mean value) in a (single) multi-armed bandit. This problem is studied under two complementary settings: (a) the *fixed-confidence setting*, where the objective is to find the best arm with the smallest expected number of arm pulls subject to ensuring that the error probability is no more than a given threshold value; see Even-Dar et al. (2006); Jamieson et al. (2014), and (b) the *fixed-budget setting*, where the objective is to find the best arm as accurately as possible given a threshold on the number of arm pulls; see Audibert, Bubeck, and Munos (2010) and Bubeck, Munos, and Stoltz (2011). In this paper, we consider the fixed-confidence setting. For an excellent survey, see Jamieson and Nowak (2014).

Mitra, Hassani, and Pappas (2021) study a federated variant of the best arm identification problem with a central server and multiple clients, similar to our work. However, their problem setting differs from ours in that in their work, the arms of a single multi-armed bandit are partitioned into as many subsets as there are clients. Each client is associated with a subset of arms and can observe only the rewards generated from the arms in this subset. The central goal in their paper is to identify the global best arm, defined as arm with the largest mean among the local best arms of the clients. Notice that an arm that is not the local best arm at any client cannot be the global best arm. Therefore, it suf-

fices for each client to communicate to the server only the empirical mean of the estimated local best arm; this communication is assumed to take place periodically, only for time step $n \in \{1, H+1, 2H+1, \ldots\}$ for some fixed period H. However, in our work, the global best arm (defined as the arm with the largest average of the means averaged across the clients) may not necessarily be the local best arm at any client, because of which the clients may need to *communicate* the empirical means of their non-local best arms. Also, we propose an alternative strategy of communicating only at time steps $n = 2^t$ for $t \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, and demonstrate that this strategy, called *exponentially sparse communication*, mitigates the overall communication cost and renders communication almost cost-free.

Works on collaborative learning in bandits (e.g., Hillel et al. (2013) and Tao, Zhang, and Zhou (2019)) consider a central server and multiple clients as in our work, but with one salient difference: in the abovementioned works, the arms *and their distributions* are identical at all the clients (the goal is to establish collaboration among the clients to find the best arm faster than without collaboration). As a result, the local best arm of each client is identical to those of the other clients. In this paper, we assume that the set of arms is identical at all the clients and allow for the arms to have *different* distributions across the clients, thereby leading to possibly distinct local and global best arms.

Contributions

We now highlight the key contributions of this paper.

- We propose a novel algorithm called FEDerated learning successive ELIMination algorithm (or FEDELIM) for finding the local best arms and the global best arm (see Algorithm 1). The key feature of FEDELIM is that clients communicate to the server in only exponential time steps $n = 2^t$ for some $t \in \mathbb{N}_0$. Given any $\delta \in (0, 1)$, we show that FEDELIM declares the local best arms and the global best arm correctly with probability at least $1 - \delta$. We present problem-instance dependent upper bounds on the total number of arm selections, the communication cost, and the total cost of FEDELIM, each of which holds with probability at least $1 - \delta$ (Theorem 4). Our results show that the total cost of FEDELIM scales as $\ln(1/\delta)$ in the error probability threshold δ, and inversely as the squares of the *sub-optimality gaps* of the arms.

- For a variant of FEDELIM (called FEDELIM0) that communicates in *every time step*, we obtain a high probability problem instance-dependent upper bound on the total number of arm selections (Theorem 2). We also obtain a lower bound on the expected total number of arm selections required by *any* algorithm which outputs the correct answer with probability at least $1 - \delta$ (Theorem 3), and show that the upper and the lower bounds are tight when M is constant or when M is sufficiently large.

- The key takeaway from our paper is that for any $C \geq 0$ and sufficiently small δ, the total cost of FEDELIM is at most 3 times the total number of arm selections under FEDELIM0 with probability at least $1 - \delta$. That is, communication is *almost cost-free* in FEDELIM. Through

extensive simulations on two synthetic datasets and the large-scale, real-world MovieLens dataset, we compare the total cost of FEDELIM with that of a periodic communication protocol with period H based on successive elimination, and observe that there is a "sweet spot" for H where the total cost of the latter is minimal. Determining this sweet spot requires knowing C and other problem instance-specific constants and is infeasible in most practical settings. In comparison, FEDELIM, while being agnostic to C and other problem instance-specific constants, learns this sweet spot on-the-fly.

Although the focus of our paper is best arm identification, FEDELIM may be adapted to solve more general problems such as top-N arms identification (Kalyanakrishnan et al. 2012), thresholding in bandits (Locatelli, Gutzeit, and Carpentier 2016), ϵ-optimal arm identification (Even-Dar et al. 2006), and so on. In our paper, the Gaussian rewards assumption is merely for simplicity in the presentation. Our analyses are applicable to observations that are sub-Gaussian. For more details, see Remarks 5,6 in Reddy, Karthik, and Tan (2022).

Notations and Problem Setup

In this section, we lay down the notations used throughout the paper, and specify the problem setup. We consider a federated multi-armed bandit with a central server and M clients. Each client is associated with a multi-armed bandit with K arms (called *local* arms). We refer to the K-armed bandit associated with a client as its *local* multi-armed bandit. We write $[K] := \{1, 2, \ldots, K\}$ to denote the set of arms, and assume that $[K]$ is the same for all the clients and the server. Also, we write $[M]$ to denote the set of clients.

Local and Global Best Arms

There are M local multi-armed bandits, one associated with each client. For $n \in \mathbb{N}$, let $X_{k,m}(n)$ denote the reward (or observation) generated from local arm k of client m at time n. For each (k, m) pair, $\{X_{k,m}(n) : n \geq 1\}$ is an *i.i.d.* process following a Gaussian distribution with an unknown mean $\mu_{k,m} \in \mathbb{R}$ and known variance σ^2. For simplicity, we set $\sigma^2 = 1$. We define the *local best arm* k_m^* of client m as the arm with the largest mean among the local arms of client m, *i.e.,* $k_m^* := \arg\max_k \mu_{k,m}$; we assume that k_m^* is unique for each m. Also, let $\mu_m^* := \mu_{k_m^*,m} = \max_k \mu_{k,m}$ be the mean of the local best arm of client m. Note that two different clients may have distinct local best arms. Letting $\mu_k := \sum_{m=1}^M \mu_{k,m}/M$, we define the *global best arm* k^* as the arm with the largest value of μ_k, i.e., $k^* = \arg\max_k \mu_k$, and assume that k^* is unique. We let $\mu^* := \mu_{k^*} = \max_k \mu_k$ denote the mean of the global best arm. The global best arm may not necessarily be the local best arm at any client.

Communication Model

We assume that each client can observe only the rewards generated from its local arms, based on which the client can estimate its local best arm. Estimating the global best arm requires exchange of information among the clients. We assume that each client communicates with a central server,

and that there is no direct communication between any two clients. We also assume that the communication link from a client to the server (uplink) entails a fixed cost of $C \geq 0$ units per usage of the link, and that the communication link from the server to the client (downlink) is cost-free as in Hanna, Yang, and Fragouli (2022). Each client sends real-valued information about the rewards from one or more of its local arms on its uplink. The server aggregates the information coming from all the clients to construct a set of arms that are potential contenders for being the global best arm, and communicates this set to each of the clients on the downlink. Each client selects each arm in set received from the server to obtain a more refined estimate of the arm's empirical mean. In this way, the clients and the server communicate until there is exactly one contender arm at the server.

When $C = 0$, it is clearly advantageous for the clients to communicate with the server at every time step. When $C > 0$, it is, however, beneficial for the clients to communicate with the server only *intermittently* so that the overall communication cost will be minimized. An instance of periodic communication in federated multi-armed bandits, where the clients communicate with the server periodically, once every H time steps for a fixed integer $H > 0$, may be seen in Mitra, Hassani, and Pappas (2021). An alternative communication strategy, one that we explore in this paper, is for the clients to communicate with the server only at time steps n of the form $n = 2^t$ for $t \in \mathbb{N}_0$. As we shall see shortly, the latter strategy mitigates the communication costs significantly and renders communication almost cost-free.

Problem Instance and Algorithm

A *problem instance* is identified by the matrix $\boldsymbol{\mu} = [\mu_{k,m} : k \in [K], m \in [M]] \in \mathbb{R}^{K \times M}$ of the means of the local arms of all the clients. The actual value of $\boldsymbol{\mu}$ is unknown, and the goal is to find the local best arm at each of the clients and also the global best arm (i.e., the vector $\mathbf{S}(\boldsymbol{\mu}) := (k_1^*, k_2^*, \ldots, k_M^*, k^*) \in [K]^{m+1}$) with high confidence. Each client selects one or more of its local arms at every time $n \in \mathbb{N}$ and forms an estimate of its local best arm as the arm with the largest empirical mean at time step n.

An *algorithm* for finding the local best arms and the global best arm prescribes the following:

- A *selection rule* that specifies the arm(s) that each client must select from amongst its local arms for each n.
- A *communication rule* that specifies the condition(s) under which the clients will communicate with the server and the information that the clients will send to the server.
- A *termination rule* that specifies when to stop further selection of arms at the clients.
- A *declaration rule* that specifies the estimates $\hat{\mathbf{S}} := (\hat{k}_1^*, \hat{k}_2^*, \ldots, \hat{k}_M^*, \hat{k}^*) \in [K]^{M+1}$ of the local best arms and the global best arm to output; here, $\hat{k}_m^*$ is the estimate of the local best arm of client $m \in [M]$ and $\hat{k}^*$ is the estimate of the global best arm.

We denote an algorithm by π and define its *total cost*

$$C^{\text{total}}(\pi) = (\text{total number of arm pulls under } \pi$$
$$+ \text{ total communication cost under } \pi). \quad (1)$$

In (1), the first component on the right hand side represents the total number of arm selections made by all the clients until termination, and the second component is the total communication cost incurred across all the clients.

Objective

For $\delta \in (0,1)$, an algorithm π is said to be δ-*probably approximately correct* (or δ-PAC) if for all $\boldsymbol{\mu} \in \mathbb{R}^{K \times M}$, we have $P_{\boldsymbol{\mu}}^{\pi}(\hat{\mathbf{S}} = \mathbf{S}(\boldsymbol{\mu})) \geq 1 - \delta$; here, $P_{\boldsymbol{\mu}}^{\pi}(\cdot)$ denotes the probability measure under algorithm π and problem instance $\boldsymbol{\mu}$. Note that any δ-PAC algorithm π must declare the correct output with probability at least $1 - \delta$ for all problem instances $\boldsymbol{\mu}$, as π is oblivious to the knowledge of the underlying problem instance. Given any $\boldsymbol{\mu}$ and $\delta \in (0,1)$, our objective is to design a δ-PAC algorithm, say π^*, for finding the local best arms and the global best arm, and derive a $\boldsymbol{\mu}$-dependent upper bound, say $U(\boldsymbol{\mu}, \delta)$, on its total cost $C^{\text{total}}(\pi^*)$, such that

$$P_{\boldsymbol{\mu}}^{\pi^*}\left(C^{\text{total}}(\pi^*) \leq U(\boldsymbol{\mu}, \delta)\right) \geq 1 - \delta.$$

In the following section, we present a version of the well-known successive elimination algorithm of Even-Dar et al. (2006) for finding the local best arms and the global best arm. We interleave it with the exponentially sparse communication sub-protocol, and subsequently obtain a high probability upper bound on its total cost.

The Federated Learning Successive Elimination Algorithm (FEDELIM)

Our algorithm, termed *Federated Learning Successive Elimination Algorithm* (or FEDELIM), is presented in Algorithm 1. In the following, we provide some algorithm-specific notations and a detailed description of FEDELIM.

Algorithm-Specific Notations

The FEDELIM algorithm proceeds in several time steps; we denote a generic time step by $n \in \mathbb{N}$. An arm is said to be a *local active arm* of client m if it is still a contender for being the client's local best arm. On the other hand, an arm is said to be a *global active arm* at the central server if it is still a contender for being the global best arm. At any given time step, we let $\mathcal{S}_{l,m}$ and $\mathcal{S}_g$ denote respectively the set of local active arms at client m and the set of global active arms at the server. We write $\hat{\mu}_{k,m}(n)$ to denote the empirical mean of arm k of client m at time step n, and define $\hat{\mu}_k(n) := \sum_{m=1}^{M} \hat{\mu}_{k,m}(n)/M$. We let $\alpha_l(n) := \sqrt{\frac{2 \ln (8KMn^2/\delta)}{n}}$ and $\alpha_g(n) := \sqrt{\frac{2 \ln (8Kn^2/\delta)}{Mn}}$ denote respectively the *local confidence parameter* and the *global confidence parameter* in time step n.

Algorithm Description

At each client: In each time step n, the algorithm first computes $\mathcal{S}_m = \mathcal{S}_{l,m} \cup \mathcal{S}_g$ for each $m \in [M]$. If $|\mathcal{S}_m| > 1$, the algorithm selects each arm in $\mathcal{S}_m$ once and updates their respective empirical means (*selection rule*). Next, for each $m \in [M]$, the algorithm checks for the validity of the condition $|\mathcal{S}_{l,m}| > 1$. If this condition holds, the algorithm eliminates all those arms in $\mathcal{S}_{l,m}$ that are no more contenders

for being the local best arm of client m. This is accomplished as follows: for each $m \in [M]$, the algorithm computes $\hat{\mu}_{*,m}(n) := \max_{k \in \mathcal{S}_{l,m}} \hat{\mu}_{k,m}(n)$, and eliminates arm k from $\mathcal{S}_{l,m}$ if $\hat{\mu}_{*,m}(n) - \hat{\mu}_{k,m}(n) > 2\alpha_l(n)$. The arms remaining in $\mathcal{S}_{l,m}$ after elimination are the local active arms of client m. For each $m \in [M]$, if $|\mathcal{S}_{l,m}| = 1$ after elimination, the algorithm outputs the arm in $\mathcal{S}_{l,m}$ as the local best arm of client m (*declaration rule* for client m).

At the server: After working on $\mathcal{S}_{l,m}$ for each $m \in [M]$ as outlined above, the algorithm checks if $|\mathcal{S}_g| > 1$ and if $n = 2^t$ for some $t \in \mathbb{N}_0$. If both of these conditions hold, then each client $m \in [M]$ sends to the server its estimates $\{\hat{\mu}_{k,m}(n) : k \in \mathcal{S}_g\}$ of the empirical means of the arms in $\mathcal{S}_g$, one per usage of its uplink (*communication rule*). Because the uplink entails a cost of $C \geq 0$, the communication cost incurred at a client is $C |\mathcal{S}_g|$, and therefore the total communication cost across all the clients is $C M |\mathcal{S}_g|$. The server eliminates all those arms in $\mathcal{S}_g$ that are no more contenders for being the global best arm as follows: the server first computes $\hat{\mu}_k(n) = \sum_{m=1}^{M} \hat{\mu}_{k,m}(n)/M$ for each $k \in \mathcal{S}_g$ and also $\hat{\mu}_*(n) = \max_{k \in \mathcal{S}_g} \hat{\mu}_k(n)$, and eliminates arm k from $\mathcal{S}_g$ if $\hat{\mu}_*(n) - \hat{\mu}_k(n) > 2\alpha_g(n)$. The arms remaining in $\mathcal{S}_g$ after elimination are the global active arms. If $|\mathcal{S}_g| = 1$ after elimination, the algorithm outputs the arm in $\mathcal{S}_g$ as the global best arm (*declaration rule* for the global best arm).

Upon identifying the local best arms and the global best arm, the algorithm *terminates*. Else, if at least one of the local best arms or the global best arm is not identified, the algorithm continues to the next time step.

Remark 1. *Recall that in our problem setup, the global best arm may not necessarily be the local best arm at any client. In fact, the local best arms and the global best arms can be all distinct. As a result, even if an arm (say arm k) is eliminated from $\mathcal{S}_{l,m}$ at client m (i.e., arm k is not the local best arm of client m), it may still need to be selected further before it can be eliminated globally from $\mathcal{S}_g$, and vice-versa. It is for this reason that we set $\mathcal{S}_m = \mathcal{S}_{l,m} \cup \mathcal{S}_g$ as the set of arms to be selected at client m. In contrast, when the global best arm is always one of the local best arms, as in Mitra, Hassani, and Pappas (2021), eliminating an arm locally at a client is akin to eliminating the arm globally.*

Remark 2. *To keep the total cost of an algorithm small, it is imperative to strike a balance between the total number of arm selections and the communication cost. For instance, as is naturally expected and also demonstrated by our numerical results later in the paper, a periodic communication scheme with period H and based on successive elimination incurs a larger communication cost than our exponentially sparse communication scheme (see Figures 2b and 3b). With regard to the total number of arm selections, one might expect that the periodic communication protocol outperforms our exponential sparse communication protocol because more frequent communication in the former leads to faster identification of the global best arm. From our numerical results, we find that this is true only partially. Rather interestingly, Figures 2c and 3c indicate that the total cost of a periodic communication algorithm (based on successive elimination) with period H decreases,*

Algorithm 1: Federated Learning Successive Elimination Algorithm (FEDELIM)

Input: $K \in \mathbb{N}$, $M \in \mathbb{N}$, $\delta \in (0,1)$
Output: $(\hat{k}_1^*, \ldots, \hat{k}_M^*, \hat{k}^*) \in [K]^{M+1}$
Initialize: $n = 0$, $\hat{\mu}_{k,m}(n) = 0$ and $\mathcal{S}_{l,m} = [K]$ for all k, m, $\hat{\mu}_k(n) = 0$ and $\mathcal{S}_g = [K]$ for all k, run = true

```
1  while run = true do
2      n ← n + 1
3      for m = 1 : M do
4          S_m ← S_{l,m} ∪ S_g
5          if |S_m| > 1 then
6              for k ∈ S_m do
7                  pull arm k of client m and update its
                     empirical mean μ̂_{k,m}(n)
8          if |S_{l,m}| > 1 then
9              Set μ̂_{*,m}(n) = max_{k∈S_{l,m}} μ̂_{k,m}(n)
10             for k ∈ S_{l,m} such that
                  μ̂_{*,m}(n) − μ̂_{k,m}(n) ≥ 2α_l(n) do
11                 S_{l,m} ← S_{l,m}\{k}
12         if |S_{l,m}| = 1 then
13             Output k̂_m^* ∈ S_{l,m}
14             S_{l,m} ← ∅
15     if |S_g| > 1 and n = 2^t for some t ∈ ℕ_0 then
16         for k ∈ S_g do
17             For each m ∈ [M], client m sends
                  μ̂_{k,m}(n) to the server.
18             Set μ̂_k(n) = Σ_{m=1}^M μ̂_{k,m}(n)/M
19         Set μ̂_*(n) = max_{k∈S_g} μ̂_k(n)
20         for k ∈ S_g such that
              μ̂_*(n) − μ̂_k(n) ≥ 2α_g(n) do
21             S_g ← S_g\{k}
22     if |S_g| = 1 then
23         Output k̂^* ∈ S_g
24         S_g ← ∅
25     if |S_m| = 0 for all m ∈ [M] then
26         run = false
```

attains a minimum, and thereafter increases with increase in H, thereby suggesting that there is "sweet spot" for H, say H_{opt}, where the total cost is minimal. However, H_{opt} is, in general, a function of C and problem instance-specific constants which are not known beforehand in most practical settings, thereby making the computation of H_{opt} infeasible. Figures 2c and 3c show that FEDELIM finds this sweet spot while being agnostic to C and other problem instance-specific constants, and thereby achieves a balanced trade-off between the total number of arm selections and comm. cost.

Performance Analysis of FEDELIM

In this section, we present our theoretical results on the performance (total number of arm pulls, total communication cost, and the total cost) of FEDELIM. We only state the results below; for the detailed proofs, see Reddy, Karthik, and Tan (2022). Our first result asserts that given any $\delta \in (0,1)$, FEDELIM is δ-PAC, i.e., it identifies the local best arms and the global best arm correctly with probability at least $1 - \delta$.

Theorem 1. *Given any $\delta \in (0,1)$, FEDELIM identifies the local best arms and global best arm correctly with probability at least $1 - \delta$ and is thus δ-PAC.*

In the proof, we first show that for any $\delta \in (0,1)$, the event

$$\mathcal{E} := \bigcap_{n \in \mathbb{N}, k \in [K], m \in [M]} \left\{ \begin{array}{l} |\hat{\mu}_k(n) - \mu_k| \leq \alpha_g(n), \\ |\hat{\mu}_{k,m}(n) - \mu_{k,m}| \leq \alpha_l(n) \end{array} \right\} \quad (2)$$

has probability at least $1 - \delta$; this is established using a standard inequality on the concentration of the empirical mean around the true mean for Gaussian rewards. We then show that FEDELIM always outputs the correct answer under $\mathcal{E}$.

We now analyse a variant of FEDELIM called FEDELIM0 which communicates to the server *in every time step*. Specifically, FEDELIM0 differs from FEDELIM in line 15 of Algorithm 1, which is executed for all n in FEDELIM0 but only for $n = 2^t$ for $t \in \mathbb{N}_0$ in FEDELIM. Our interest is only in the total number of arm selections of FEDELIM0, say T_{FEDELIM0}, required to find the local best arms and the global best arm on the event $\mathcal{E}$, and how this compares with the total number of arm selections of other algorithms which also communicate in every time step. As we shall see, T_{FEDELIM0} is an important term that governs the problem instance-dependent upper bounds for the total number of arm selections and the total cost of FEDELIM. Note that T_{FEDELIM0} is also the total cost of FEDELIM0 on $\mathcal{E}$ when $C = 0$.

Performance Analysis of FEDELIM0

For $k \neq k_m^*$, let $\Delta_{k,m} := \mu_{k_m^*,m} - \mu_{k,m}$ denote the *suboptimality gap* between the means of arm k of client m and the local best arm of client m, and let $\Delta_{k_m^*,m} := \min_{k \neq k_m^*} \Delta_{k,m}$. Similarly, for $k \neq k^*$, we let $\Delta_k := \mu_{k^*} - \mu_k$ and $\Delta_{k^*} := \min_{k \neq k^*} \Delta_k$. The following result provides a problem instance-dependent upper bound on T_{FEDELIM0}.

Theorem 2. *Fix $\delta \in (0,1)$. On the event $\mathcal{E}$ defined in (2),*

$$T_{\mathrm{FEDELIM0}} \leq T := \sum_{k=1}^{K} \sum_{m=1}^{M} \max\{T_{k,m}, T_k\}, \quad (3)$$

where for each $k \in [K]$ and $m \in [M]$,

$$T_{k,m} := 102 \cdot \frac{\ln\left(\frac{64\sqrt{\frac{8KM}{\delta}}}{\Delta_{k,m}^2}\right)}{\Delta_{k,m}^2} + 1, \quad (4)$$

$$T_k := 102 \cdot \frac{\ln\left(\frac{64\sqrt{\frac{8K}{\delta}}}{M\Delta_k^2}\right)}{M\Delta_k^2} + 1. \quad (5)$$

We show in the proof that on the event $\mathcal{E}$, the total number of arm selections under FEDELIM0 required to identify arm k of client m as the client's local best arm or otherwise, say $T_{k,m}^{(1)}$, is upper bounded by $T_{k,m}$ for all $k \in [K]$ and $m \in [M]$. To establish the preceding result, we use the fact that $\alpha_1(n) \to 0$ as $n \to \infty$, and look for the smallest integer n such that $\alpha_1(n) \le \Delta_{k,m}^2/4$; call this $n_{k,m}$. We argue that $T_{k,m}^{(1)} \le n_{k,m}$ on the event $\mathcal{E}$, and subsequently show that $T_{k,m}$ is an upper bound for $n_{k,m}$. A similar procedure as above is used to upper bound the total number of arm pulls required to identify arm k as being the global best arm or otherwise at the server. Combining the two upper bounds and noting that the event $\mathcal{E}$ occurs with probability at least $1 - \delta$, we arrive at (3).

The next result shows that the upper bound in (3) is tight up to a constant factor.

Theorem 3. *Given $\delta \in (0,1)$ and a δ-PAC algorithm π, let T_δ^π denote the total number of arm selections under π required to find the local best arms and the global best arm when the clients and the server communicate in every time step. Under the problem instance $\boldsymbol{\mu}$,*

$$\inf_{\pi \text{ is } \delta\text{-PAC}} \mathbb{E}_{\boldsymbol{\mu}}^\pi[T_\delta^\pi] \ge \sum_{k=1}^K \sum_{m=1}^M \max\left\{ \frac{\ln(\frac{1}{2.4\delta})}{\Delta_{k,m}^2}, \frac{\ln(\frac{1}{2.4\delta})}{M^2 \Delta_k^2} \right\},$$

(6)

where in (6), $\mathbb{E}_{\boldsymbol{\mu}}^\pi[\cdot]$ denotes the expectation under the algorithm π and the problem instance $\boldsymbol{\mu}$.

The proof of Theorem 3 is based on the transportation lemma (Kaufmann, Cappé, and Garivier 2016, Lemma 1) which combines a certain change of measure technique and Wald's identity for *i.i.d.* processes.

Remark 3. *Theorems 2 and 3 together provide a fairly tight characterisation of the total number of arm selections under the optimal algorithm in the class of all algorithms that communicate in every time step. They show that FEDELIM0 is almost optimal in this class. Neglecting the logarithm terms and the constants, the key difference between the upper and lower bounds manifests in the second term in the maximum in (6), in which there is an additional factor of M in the denominator. When M is a constant or if M is so large so that $\Delta_{k,m} \le \sqrt{M}\,\Delta_k$ for all $(k,m) \in [K] \times [M]$ (a typical federated learning scenario in which the number of clients M is large), Theorems 2 and 3 are tight up to log factors.*

Performance of FEDELIM with Uplink Cost

We now present a high-probability upper bound on the total cost (i.e., the sum of the total number of arm pulls and the total communication cost) of FEDELIM for any $C \ge 0$. Given a problem instance $\boldsymbol{\mu}$, for each $k \in [K]$ and $m \in [M]$, let $T_{k,m}$ and T_k be as defined in (4) and (5) respectively.

Theorem 4. *Fix a problem instance $\boldsymbol{\mu}$, uplink cost $C \ge 0$, and $\delta \in (0,1)$ such that $C \ln T_k \le T_k$ for all $k \in [K]$. Let T_{FEDELIM}^C, $C_{\text{FEDELIM}}^{\text{comm}}$, and $C_{\text{FEDELIM}}^{\text{total}}$ denote respectively the total number of arm selections, the communication cost, and the total cost of FEDELIM towards identifying the local best*

arms and the global best arm. On the event $\mathcal{E}$ defined in (2), the following inequalities hold (with T as defined in (3)):

$$T_{\text{FEDELIM}}^C \le \sum_{k=1}^K \sum_{m=1}^M \max\{T_{k,m},\, 2T_k\} \le 2\,T, \quad (7)$$

$$C_{\text{FEDELIM}}^{\text{comm}} \le C \cdot M \cdot \sum_{k=1}^K \left\lceil \frac{\ln T_k}{\ln 2} \right\rceil, \quad\quad (8)$$

$$C_{\text{FEDELIM}}^{\text{total}} = T_{\text{FEDELIM}}^C + C_{\text{FEDELIM}}^{\text{comm}} \le 3\,T. \quad (9)$$

Notice that the maxima in (3) and that in (7) are identical up to the constant 2. Intuitively, the extra factor of 2 arises because if a candidate arm k is not eliminated in time step $n = 2^t$ but is eliminated in time step $n = 2^{t+1}$ for some $t \in \mathbb{N}_0$, then it must be the case that $2^{t+1} \le 2T_k$, and therefore the total number of arm selections is at most $2T_k$.

It is no coincidence that the constant 2 appears inside the maximum in (7) and also in the denominator in (8). In fact, exponential sparse communication in time steps $n = \lceil \lambda^t \rceil$ for $t \in \mathbb{N}_0$ and $\lambda > 0$, results in λ replacing 2 in both (7) and (8). Then, optimising the sum of the λ-analogues of the right hand sides of (7) and (8), we may arrive at a fairly tight upper bound on the total cost, i.e., the λ-analogue of (9). However, the optimal λ is, in general, a function of C and the problem instance-specific sub-optimality gaps which are *unknown* in most practical settings. Therefore, we do away with finding the optimal λ and instead use $\lambda = 2$. For a more detailed discussion, see Reddy, Karthik, and Tan (2022).

Remark 4. *The key takeaway result of our paper, presented in inequality (9), shows that the total number of arm selections (resp. total cost) of FEDELIM is at most 2 (resp. 3) times T. These multiplicative gaps of 2 and 3 do not depend on C. In contrast, for periodic communication (Mitra, Hassani, and Pappas 2021) with period H, it can be shown that the multiplicative gap for the total cost is $1 + C/H$, which does depend on the per usage communication cost C.*

Numerical Results

In this section, we present numerical results on the performance of FEDELIM (and FEDELIM0). We consider two synthetic datasets and one real-world dataset. The code used for obtaining the results may be accessed at https://github.com/pnkarthik/AAAI-2023-Code.

### Experiments on a Synthetic Dataset

First, we discuss our numerical results on a stylized synthetic dataset. We consider the problem instance

$$\boldsymbol{\mu} = \begin{bmatrix} 0.9 & 0.1 & 0.1 \\ 0.1 & 0.9 & 0.1 \\ 0.1 & 0.1 & 0.9 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \in \mathbb{R}^{4 \times 3}. \quad (10)$$

Notice that arm m is the local best arm of client m for each $m \in [3]$, whereas arm 4 is the global best arm. Figure 2 shows a summary of the results obtained after averaging across 100 independent trials. The error bars show ± 1 standard deviation away from the mean. Theorem 4 states that
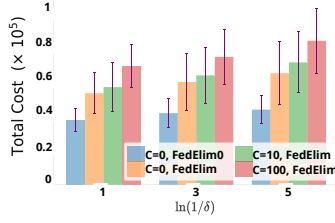
(a) A plot comparing the number of arm pulls of FEDELIM0 with the cost of FEDELIM for $C \in \{0, 10, 100\}$. FEDELIM0 has a lower cost compared to FEDELIM when $C = 0$.
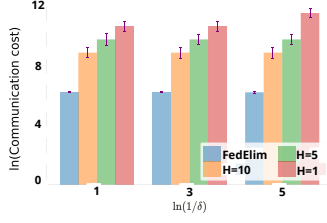
(b) A plot comparing of the communication cost incurred under the proposed FEDELIM algorithm and under periodic communication with period $H \in \{1, 5, 10\}$ and $C = 10$.
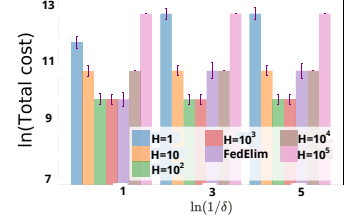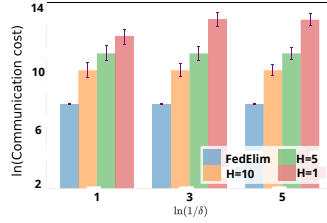
(c) A plot comparing the cost under FEDELIM and periodic communication with period $H = 10^p$ for $p \in \{0\} \cup [5]$ and $C = 10$. FEDELIM operates at a sweet spot.
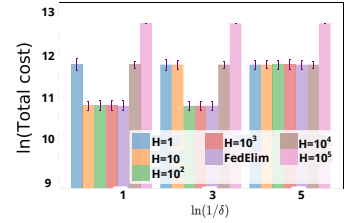
Figure 2: Numerical results on the synthetic dataset with the problem instance in (10).
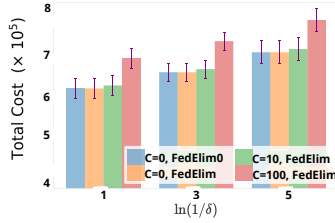
(a) Plot analogous to Figure 2a.

(b) Plot analogous to Figure 2b with $C = 10$.

(c) Plot analogous to Figure 2c with $C = 10$.

Figure 3: Numerical results on the MovieLens dataset.

for FEDELIM, the total cost when $C > 0$ is at most three times that when $C = 0$. Figure 2a strongly corroborates this. It shows the total cost for FEDELIM for various values of C as well as FEDELIM0. We observe that for each fixed value of δ, the total cost of FEDELIM is at most three times that of FEDELIM0 regardless of the value of C. In fact, the multiplicative factor three is conservative as empirically observed from Figure 2a.

In Figure 2b, we compare the communication cost of FEDELIM and periodic communication (Mitra, Hassani, and Pappas 2021). First, we see that as H increases, the communication cost decreases as expected. Second, we observe that the communication cost of FEDELIM is significantly smaller than that of the periodic communication schemes.

In Figure 2c, we compare the total cost (defined in (1)) of FEDELIM and periodic communication with period $H = 10^p$ for $p \in \{0, \ldots, 5\}$. We observe that, per Remark 2, for periodic communication, there is a "sweet spot" for H, which, in this case, occurs at around $H = 10^2$. On the other hand, FEDELIM does almost as well as the best periodic communication scheme for the optimal H. FEDELIM is, however, agnostic to the cost C, which is set to 10 here. More experimental results, specifically on a dataset of Bernoulli observations used in (Mitra, Hassani, and Pappas 2021), is available in Reddy, Karthik, and Tan (2022).

Experiments on the MovieLens Dataset

We also run our algorithm on a large-scale subsampled version of the MovieLens dataset (Cantador, Brusilovsky, and Kuflik 2011) crafted so as to simulate heterogeneity among

the clients. Specifically, we extract a subset of the MovieLens dataset containing movies that were produced in 38 different countries and across 20 different genres, resulting in a total of $8,636$ movies and about 2.04 million ratings; see Reddy, Karthik, and Tan (2022) for the details. We then set the countries and genres to be in one-to-one correspondence with the clients (so $M = 38$) and arms (so $K = 20$), respectively. Figure 3 shows the results of running FEDELIM on this dataset. The qualitative behaviours of FEDELIM, FEDELIM0 and the strategy that communicates with the server periodically match with those for the synthetic dataset. The highlight of Figure 3c is that FEDELIM attains the *absolute* minimum among all schemes that communicate periodically to the server, thus incontrovertibly demonstrating the ability of FEDELIM to effectively balance communication and the number of arm selections on a real-world, large-scale dataset.

Summary and Future Work

We designed and analyzed an algorithm called FEDELIM, and showed that it is effective in learning the local and global best arms in the context of a federated learning setting. We did not take into account the fact that communication to the server typically requires quantization of the empirical means at each of the clients. Elucidating the tradeoff between the number of bits used and the number of arm pulls (cf. Hanna, Yang, and Fragouli (2022)) is a promising area for future research. Another interesting direction is to study the performance of a track-and-stop-based algorithm for our problem.

Acknowledgements

References

Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best arm identification in multi-armed bandits. In *23rd Annual Conference on Learning Theory*, 41–53. PMLR.

Bubeck, S.; Munos, R.; and Stoltz, G. 2011. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19): 1832–1852.

Cantador, I.; Brusilovsky, P.; and Kuflik, T. 2011. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 387–388.

Chernoff, H. 1959. Sequential design of experiments. *The Annals of Mathematical Statistics*, 30(3): 755–770.

Even-Dar, E.; Mannor, S.; Mansour, Y.; and Mahadevan, S. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7(6): 1079–1105.

Hanna, O. A.; Yang, L.; and Fragouli, C. 2022. Solving multi-arm bandit using a few bits of communication. In *25th International Conference on Artificial Intelligence and Statistics*, 11215–11236. PMLR.

Hillel, E.; Karnin, Z. S.; Koren, T.; Lempel, R.; and Somekh, O. 2013. Distributed exploration in multi-armed bandits. In *26th International Conference on Neural Information Processing Systems*, 854–862.

Jamieson, K.; Malloy, M.; Nowak, R.; and Bubeck, S. 2014. lil' UCB: An optimal exploration algorithm for multi-armed bandits. In *27th Annual Conference on Learning Theory*, 423–439. PMLR.

Jamieson, K.; and Nowak, R. 2014. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *48th Annual Conference on Information Sciences and Systems*, 1–6. IEEE.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; and D'Oliveira, R. G. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC subset selection in stochastic multi-armed bandits. In *29th International Conference on Machine Learning*, 655–662. PMLR.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1): 1–42.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit Algorithms*. Cambridge University Press.

Locatelli, A.; Gutzeit, M.; and Carpentier, A. 2016. An optimal algorithm for the thresholding bandit problem. In *33rd International Conference on Machine Learning*, 1690–1698. PMLR.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *20th International Conference on Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Mitra, A.; Hassani, H.; and Pappas, G. 2021. Exploiting heterogeneity in robust federated best-arm identification. *arXiv preprint arXiv:2109.05700*.

Reddy, K. S.; Karthik, P. N.; and Tan, V. Y. F. 2022. Almost cost-free communication in federated best arm identification. *arXiv preprint arXiv:2208.09215*.

Shi, C.; Shen, C.; and Yang, J. 2021. Federated multi-armed bandits with personalization. In *24th International Conference on Artificial Intelligence and Statistics*, 2917–2925. PMLR.

Tao, C.; Zhang, Q.; and Zhou, Y. 2019. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In *60th Annual Symposium on Foundations of Computer Science*, 126–146. IEEE.

Zhong, Z.; Cheung, W. C.; and Tan, V. Y. F. 2021. On the Pareto Frontier of Regret Minimization and Best Arm Identification in Stochastic Bandits. *arXiv preprint arXiv:2110.08627*.