Network of Bandits

Raphaël Féraud
Orange Labs
raphael.feraud@orange.com

Abstract

The distribution of machine learning tasks on the user’s devices offers several advantages for application purposes: scalability, reduction of deployment costs and privacy. We propose a basic brick, DISTRIBUTED MEDIAN ELIMINATION, which can be used to distribute the best arm identification task in various schemes. In comparison to MEDIAN ELIMINATION run on a single player, we showed a near optimal speed-up factor in $O(\alpha.N/K \ln^2 K)$, where $K$ is the number of actions, $N$ is the number of players, and $\alpha \in (0, 1/2]$. This speed-up factor is reached with a near optimal communication cost. Experiments illustrate and complete the analysis: in comparison to MEDIAN ELIMINATION with unconstrained communication cost, the distributed version shows practical improvements.

1 Introduction

Since the beginning of this century a vast quantity of connected devices has been deployed. These devices dialog together through wired and wireless networks thus generating infinite streams of events. By interacting with streams of events, machine learning algorithms are used for instance to optimize the choice of ads on a website, to choose the best human machine interface, to recommend products on a web shop, to insure self-care of set top boxes, to assign the best wireless network to mobile phones. With the now rising internet of things, the number of decisions (or actions) to be taken by more and more autonomous devices further increases. In this context, the distribution of machine learning algorithms on the user’s devices offers decisive advantages:

- scalability, thanks to parallel processing,
- reduction of deployment costs, thanks to the removal of the centralized processing architecture,
- privacy, by processing the user’s data on his own device.

However, the massive distribution of machine learning algorithms on datastreams raises two major concerns. Firstly, the cost of this massive distribution can be huge in terms of communications. Indeed, the naive distribution of machine learning algorithms transmits each incoming event on each device to all devices leading to a maximum communication cost in $O(NT)$, where $N$ is the number of players (i.e. nodes) and $T$ the time horizon. The communication cost has to be precisely quantified and controlled. Secondly, the proliferation of autonomous machine learning algorithms, which choose actions for users, implies to control the potential cost of bad choices. We need algorithms with strong theoretical guarantees.

Most of applications necessitate to take and optimize decisions with a partial feedback. This well known problem is called multi-armed bandits (MAB). In its most basic formulation, it can be stated
as follows: there are \( K \) decisions, each having an unknown distribution of bounded rewards. At each step, one has to choose a decision and receives a reward. The performance of a MAB algorithm is assessed in terms of regret (or opportunity loss) with regards to the unknown optimal decision. Optimal solutions have been proposed to solve this problem using a stochastic formulation in [Auer et al (2002)a], using a Bayesian formulation in [Kaufman et al (2012)], or using an adversarial formulation in [Auer et al (2002)b]. While these approaches focus on the minimization of the expected regret, the \((\epsilon, \delta)\)-best-arm identification, based on the PAC setting (see Vailant (1984)), focuses on the sample complexity (i.e. the number of time steps) needed to find an \( \epsilon \)-approximation of the best arm with a failure probability of \( \delta \). This formulation has been studied for MAB problem in [Even-Dar et al (2002), Bubeck et al (2009)], for dueling bandit problem in [Urvoy et al (2013)], for linear bandit problem in [Soare et al (2014)], and for the contextual bandit problem in [Péaude et al (2016)].

Recent years have seen an increasing interest for the study of the collaborative distribution scheme: \( N \) players collaborate to solve a multi-armed bandit problem. The distribution of non-stochastic experts has been studied in [Kanade et al (2012)]. The distribution of stochastic multi-armed bandits has been studied for peer to peer network in [Szönyi et al (2013)]. In [Hillel et al (2013)], the analysis of the distributed exploration is based on the sample complexity need to find the best arm with an approximation factor \( \epsilon \). When only one communication round is allowed, an algorithm with an optimal speed-up factor of \( \sqrt{N} \) has been proposed. The algorithmic approach has been extended to the case where multiple communication rounds are allowed. In this case a speed-up factor of \( N \) is obtained while the number of communication rounds is in \( O(\ln 1/\epsilon) \). The authors focused on the trade-off between the number of communication rounds and the number of pulls per player. This analysis is natural when one would like to distribute the best arm identification task on a centralized processing architecture. In this case, the bandwidth between processors is broad and the number of communication rounds is the true cost. The massive distribution of best arm identification task, that we would like to address is more challenging.

First of all, when bandit algorithms are deployed on the user’s devices, the bandwidth between the devices and the processors can no longer be considered as broad. In this case, rather than the number of communications, the cost of communications is modeled by the number of transmitted bits. The first part of this study focuses on the analysis of the trade-off between the number of transmitted bits and the number of pulls per player. In the next section, we propose a distributed best arm identification algorithm, which is near optimal both in terms of number of pulls per player and in terms of number of transmitted bits, and which benefits from a speed-up factor in \( O(\alpha \cdot N/K \ln^2 K) \) in comparison to MEDIAN ELIMINATION [Even-Dar et al (2002)] run on a single player.

More importantly, when bandit algorithms are deployed on the user’s devices, the event player is active is modeled by an indicator random variable. Indeed, a player can choose an action only when an uncontrolled event occurs such as: the device of the user is switched on, the user has launched a mobile phone application, the user connects to a web page... Unlike in [Hillel et al (2013)], where the draw of players is deterministic, here we consider that the players are drawn from a distribution. In the last part of the analysis, we provide an upper bound of the number of draws of players needed to identify the best arm.

2 Distributed Median Elimination

2.1 Principle

The distribution of the best arm identification task is a collaborative game. For instance, when a mobile phone application plays an action such as personalize its interface, a feedback is received from the user. This feedback is used to update the estimated mean reward of the chosen action given the mobile phone. If the mobile phone application would like to eliminate the action, it transmits it to the synchronization system. If enough mobile phones would like to eliminate the action, the synchronization system sends the suppressed action to each mobile phone. From a theoretical point of view any device can be used as the synchronization system. However, in practice the synchro-
organization system must be reachable at any time, and thus a server has to be used to share information between devices (see Figure 1).

### 2.2 Algorithm

Let $N$ be the number of players (i.e. the number of user’s devices). To model the distribution of the best arm identification task on the user’s devices, we assume that at each time step a player $n$ is drawn according to a probability $P(z)$. Let $N_{\gamma} = |\{ n \leq N, P(z = n) > \gamma \}|$ be the number of active players. Let $\mathcal{A}$ be a set of $K$ actions.

![Figure 1: Principle: the events are processed on the user’s devices, and the synchronization server shares information between devices. This distribution scheme is similar to the one called Site Predictive Model in [Kanade et al (2012)].](image)

Let $y \in [0, 1]^K$ be a vector of bounded random variables, $y_k$ be the random variable denoting the reward of the action $k$, and $y^n_k$ be the random variable denoting the reward of the action $k$ chosen by the player $n$. Let $\mu^n_k = \mathbb{E}[y^n_k] = \mathbb{E}[y_k \cdot 1_{z=n}]$ be the expected reward of the action $k$ for the player $n$. We study the distribution the best arm identification task, when the following assumptions hold:

**Assumption 1:** the mean reward of an action does not depend on the player: $\forall n \in \{1, \ldots, N\}$ and $\forall k \in \{1, \ldots, K\}$, $\mu^n_k = \mu_k$.

**Assumption 2:** each transmitted message through the communication network is coded using a binary code\(^1\). For instance, when the synchronization server notifies to all players that the action $k = 7$ is eliminated, it sends to all players the code ‘111’.

Now, we can derive and analyze a simple and efficient algorithm to distribute the best arm identification task (see Algorithm 1). Each player collaborates with each other by sharing the same set of remaining actions $\mathcal{A}$. When a player $n$ is drawn, it plays a remaining action $k$ from $\mathcal{A}$. It receives a feedback $y^n_k$ and uses it to update its estimated mean reward $\hat{\mu}^n_k$. Rather than to transmit the estimated mean reward to other players, which would lead to use a broad bandwidth, DISTRIBUTED MEDIAN ELIMINATION transmits to the synchronization server the actions $k$, that it would like to eliminate (Algorithm 1 function SendToServer($k, n$) line 15). For the current player, the worst half actions are candidate for elimination when each remaining action has been played enough (Algorithm 1 condition line 10). When a message is received by the synchronization server from a player (Algorithm 2 function MessageFromPlayer($k, n$) line 5) the flag $\lambda^n_k$ corresponding to the tuple (action,player) is set to one. When the number of flags set to one for an action is enough high, the

\[^1\text{a prefix code such as a truncated binary code or a Huffman code (see [Cover and Thomas (2006)]) would be more efficient. To simplify the exposition of ideas, we have restricted the analysis to binary code.}\]
Algorithm 1 DISTRIBUTED MEDIAN ELIMINATION

Inputs: $\epsilon \in [0, 1)$, $\delta \in (0, 1]$, $N_\gamma \leq N$, $\alpha \in (0, 1/2]$  
Output: an $\epsilon$-approximation of the best arm

1: \[ \eta = \left(\frac{\delta}{2}\right)^{-\frac{1}{\alpha N}}, l = 1, \forall n \eta_i^n = \eta/2 \text{ and } \epsilon_i^n = \epsilon/4, \forall (k, n) \] \[ t_i^n = 0, \hat{\mu}_k^n = 0, \lambda_k^n = 0 \]

2: repeat
3: \[ \text{if MessageFromServer}(k) \text{ then} \]
4: \[ A = A \setminus \{k\} \]
5: \[ \text{end if} \]
6: Draw a player $n \sim P(z)$
7: Play sequentially each action $k \in A$
8: Receive $y_k^n$
9: \[ t_i^n = t_i^n + 1, \hat{\mu}_k^n = \frac{y_k^n}{t_i^n} + \frac{t_i^n - 1}{t_i^n} \hat{\mu}_k^n \]
10: if $\forall k, t_i^n \geq 4/(\epsilon_i^n)^2 \ln(3/\eta_i^n)$ then
11: \[ \text{Let } m_i^n \text{ be the median of } \hat{\mu}_k^n \text{ such that } \lambda_k^n = 0 \]
12: for all $k \in A$ do
13: \[ \text{if } (\hat{\mu}_k^n < m_i^n \text{ and } \lambda_k^n = 0) \text{ then} \]
14: \[ \text{SendToServer}(k, n), \lambda_k^n = 1 \]
15: \[ \text{end if} \]
16: \[ \text{end for} \]
17: \[ \epsilon_i^{n+1} = 3.\epsilon_i^n / 4, \eta_i^{n+1} = \eta_i^n / 2, l = l + 1 \]
18: end if
19: until $(\forall n, \sum_k \lambda_k^n = |A| - 1)$

Algorithm 2 SYNCHRONIZATION SERVER

1: Inputs: $N_\gamma \leq N$  
2: Output: an $\epsilon$-approximation of the best arm  
3: Initialization: $\forall (k, n) \lambda_k^n = 0$
4: repeat
5: \[ \text{if MessageFromPlayer}(k, n) \text{ then} \]
6: \[ \lambda_k^n = 1 \]
7: \[ \text{if } (\sum_n \lambda_k^n \geq \alpha N_\gamma) \text{ then} \]
8: \[ \text{SendToPlayers}(k), \forall n \lambda_k^n = 0 \]
9: \[ \text{end if} \]
10: \[ \text{end if} \]
11: until $(\forall n, \sum_k \lambda_k^n = |A| - 1)$

synchronization server sends this information to all players (Algorithm 2 function SendToPlayer(k) line 8), and the action is eliminated for all players (Algorithm 1 lines 3-5).

3 Analysis

Theorem 1 states that in comparison to MEDIAN ELIMINATION [Even-Dar et al (2002)] run on each player (i.e. zero communication cost), the potential speed-up factor of DISTRIBUTED MEDIAN ELIMINATION is $O\left(\alpha N/K \ln^2 K\right)$, while the communication cost is $2N(K - \alpha + 1)|\log_2 K|$ bits.

Definition 1: An $\epsilon$-approximation of the best arm $k^* = \arg\max_{k \in A} \mu_k$ is an arm $k \in A$ such that $\mu_k^* \leq \mu_k + \epsilon$.

Theorem 1: when $N_\gamma = N$ with a probability at least $1 - \delta$, DISTRIBUTED MEDIAN ELIMINATION finds an $\epsilon$-approximation of the optimal arm, transmitting less than $2N(K - \alpha + 1)|\log_2 K|$
bits through the communication network and using
\[ t^* = O \left( \frac{K^2 \ln^2 K}{\alpha N \epsilon^2} \ln \frac{1}{\delta} \right) \text{ pulls per player.} \]

The proof of Theorem 1 is based on two Lemmas.

**Lemma 1:** when \( N_\gamma = N \) with a probability at least \( 1 - \delta \), DISTRIBUTED MEDIAN ELIMINATION finds an \( \epsilon \)-approximation of the optimal arm, using
\[ t^* = O \left( \frac{K^2 \ln^2 K}{\alpha N \epsilon^2} \ln \frac{1}{\delta} \right) \text{ pulls per player.} \]

**Proof.** Using Lemma 1 in [Even-Dar et al (2002)], we state that each single player finds an \( \epsilon \)-approximation of the best arm with a failure probability \( \eta \). The analysis of the sample complexity of a single player departs from the one of MEDIAN ELIMINATION since the arms are eliminated only when the synchronization server sends an elimination message. As a consequence, even if a player has decided to eliminate an arm, it still plays this arm until enough players would like to eliminate it. From line 10 of Algorithm 1, the sample complexity of any single player \( n \) is:
\[ t^*_n = \sum_{l=1}^{\log_2 K} \frac{4 K_n^l}{(c^l)^2 \ln \frac{3}{\eta_n^l}}, \text{ where } K_n^l \text{ is the number of actions at epoch } l \text{ of the player } n. \]

We have \( \eta_n^l = \eta/2^l, c^l = (3/4)^{l-1}, \epsilon/4, \text{ and } K_n^l \leq K \). We obtain:
\[ t^*_n \leq \frac{16 K}{\epsilon^2} \sum_{l=1}^{\log_2 K} \left( \frac{16}{9} \right)^{l-1} \left( l \ln 2 + \ln \frac{3}{\eta} \right). \]

This implies:
\[ t^*_n \leq \frac{16 K \log_2 K}{\epsilon^2} \left( \frac{16}{9} \right)^{\log_2 K} \left( \log_2 K \ln 2 + \ln \frac{3}{\eta} \right) \leq \frac{16 K^2 \log_2 K}{\epsilon^2} \left( \log_2 K \ln 2 + \ln \frac{3}{\eta} \right) \]
\[ \Rightarrow t^*_n \leq \frac{16 K^2 \log_2 K}{\epsilon^2} \ln \frac{3}{\eta} (\log_2 K + 1) = O \left( \frac{K^2 \ln^2 K}{\epsilon^2} \ln \frac{1}{\eta} \right) \]

They are two possible reasons of failure at time \( t^* \): (1) The set of remaining arms does not contain an \( \epsilon \)-approximation of the best arm: as an arm is eliminated only when \( \alpha N \) players eliminate it, an \( \epsilon \)-approximation of the best arm is eliminated with a probability \( \eta^{\alpha N} \).

(2) The set of remaining arm contains a suboptimal arm: at least \( (1 - \alpha).N \) players have not eliminate a suboptimal arm with a probability \( \eta^{(1-\alpha)N} \). Then, \( \alpha \leq 1/2 \) implies \( \eta^{(1-\alpha)N} \leq \eta^{\alpha N} \).

So DISTRIBUTED ACTION ELIMINATION fails with a probability lower than \( \delta = 2 \eta^{\alpha N} \). Replacing \( \eta \) by \( \left( \frac{3}{4} \right)^{\alpha N} \) in inequality we provide the upper bound of the number of pulls per player.

**Lemma 2:** when \( N_\gamma = N \), DISTRIBUTED MEDIAN ELIMINATION stops transmitting less than
\[ 2N(K - \frac{\alpha + 1}{2})\lfloor \log_2 K \rfloor \text{ bits.} \]
Proof. Each action is sent to the server no more than once per player. If the algorithm does not fail, at least \( \alpha.N \) players do not send the code of the best action. Thus the number of sent codes to the server is upper bounded by \( N.K - \alpha.N \).

Then, the statement \( \forall n \lambda_n^k = 0 \) line 8 of algorithm \[ and the fact that no more than one action per player is sent to the server ensure that the synchronization server sends each suboptimal action only once. Thus the number of sent codes to players is \( (K-1).N \).

The optimal length of a binary code needed to code an alphabet of size \( K \) is \( \lceil \log_2 K \rceil \). Thus, the total number of transmitted bits is upper bounded by \( 2N(K - \frac{\alpha + 1}{2})\lceil \log_2 K \rceil \).

Theorem 2 shows that Distributed Action Selection is near optimal both in terms of transmitted number of bits and number of pulls per player.

**Theorem 2:** It exists a distribution \( D_y \) such that any distributed algorithm on \( N \) players needs to transmit at least \( 2N(K - 1)\lceil \log_2 K \rceil \) bits to find an \( \epsilon \)-approximation of the best arm in at least

\[
\Omega \left( \frac{K}{N.\epsilon^2 \ln \frac{1}{\delta}} \right) \text{ pulls per player.}
\]

Proof. First, Theorem 1 in [Mannor and Tsitsiklis (2004)] provides the lower bound of the number of pulls for a player. When \( N \) players collaborate to find the best arm, the maximum speed-up factor is \( N \), otherwise one of them violates the lower bound. Then, let us assume that it exists an algorithm with a speed-up factor \( N \) that transmits less than \( 2N(K - 1)\lceil \log_2 K \rceil \) bits. There are only three possibilities to achieve this goal:

1. a player does not transmit information about an action to the server,
2. or the server does not transmit an eliminated action to a player,
3. or this algorithm transmits less than \( \lceil \log_2 K \rceil \) bits for each action.

If a player does not transmit an information about an action to the server (condition 1), then this action may not be eliminated. If this remaining action is not the optimal one, the algorithm fails or the speed-up factor \( N \) is not reached.

If an eliminated action is not transmitted to a player (condition 2), this player will not eliminate this action at time \( t^* \). As a consequence, the speed-up factor \( N \) is not reached.

Thus, the number of sent messages cannot be less than \( N.(K - 1) \) messages from players to the server plus \( N.(K - 1) \) messages from server to the players. The minimum information that can be transmitted about an action is its index. Using a binary code (see Assumption 2), the number of bits needed to transmit the index of an action cannot be less than \( \lceil \log_2 K \rceil \) (condition 3).

The first part of the analysis seems to show that the proposed algorithm allows to efficiently distribute the best arm identification task on millions of devices, and thus benefits from a huge theoretical speed-up factor in \( O(N) \). However, when the best arm identification task is distributed on a large number of devices, we need to take into account the probability of players. In particular, some of devices could simply be switched off, and thus have a zero probability to be drawn. A simple way to handle the case where some players have a low or even a zero probability is to use only the \( N_\gamma \) most active players to eliminate the actions. It requires a global knowledge of the distribution of players. The speed-up factor becomes \( \alpha.N_\gamma \) instead \( \alpha.N \), but one can insure that the proposed algorithm stops. Theorem 3 provides an upper bound of the number of draws of players (i.e. the time steps) needed to find an \( \epsilon \)-approximation of the best arm with high probability.

**Theorem 3:** with a probability at least \( 1 - 2.\delta \), Distributed Action Selection identifies the optimal arm, using
draws of players, where \( N_\gamma = |\{ n \leq N, P(z = n) > \gamma \}|. \)

**Proof.** At each time step, a player \( n \) is drawn according to a probability \( P(z = n) \). Let \( N_\gamma = \{ n \leq N, P(z = n) > \gamma \} \) be the set of players, where \( P(z = n) > \gamma \). Let \( T \) be the time step where each player \( n \in N_\gamma \) has been drawn at least \( t \) times. Let \( f \) be the number of times where a player has not been drawn at time step \( T \). \( f \) follows a negative binomial distribution with parameters \( t, 1 - \gamma \). By definition of the negative binomial distribution, We have:

\[
E[f] = \frac{(1 - \gamma)t}{\gamma}, \quad \text{and} \quad V[f] = \frac{(1 - \gamma)t}{\gamma^2}
\]

Using the Chebyshev’s inequality, we have:

\[
P\left( P(f) \geq \frac{(1 - \gamma)t}{\gamma} + \epsilon \right) \leq \frac{(1 - \gamma)t}{\epsilon^2 \gamma^2} = \delta
\]

Then, we have:

\[
P\left( P(f) \geq \frac{(1 - \gamma)t}{\gamma} + \frac{1}{\gamma} \sqrt{\frac{(1 - \gamma)t}{\delta}} \right) \leq \delta
\]

Let \( T^* \) be the time step needed to obtain \( t^* \) draws of each player. The number of draws \( T^* \) is the sum of the number of draws of the less pulled player and the draws which do not contain this player. Hence, the following inequality is true with a probability \( 1 - 2\delta \):

\[
T^* \leq t^* + \frac{(1 - \gamma)t^*}{\gamma} + \frac{1}{\gamma} \sqrt{\frac{(1 - \gamma)t^*}{\delta}} = \frac{1}{\gamma} \left( t^* + \sqrt{\frac{(1 - \gamma)t^*}{\delta}} \right)
\]

Then using Theorem 1 with \( N_\gamma \) players, and replacing \( t^* \) by its value, we provide the proof.

In comparison to MEDIAN ELIMINATION [Even-Dar et al (2002)] run without constraints on number of transmitted bits, which benefits from an optimal sample complexity bound in \( O\left( \frac{K^2 \ln K}{\gamma^2} \right) \), the worst case for DISTRIBUTED MEDIAN ELIMINATION is when the \( N \) players are uniformly distributed. In this worst case, the left term of Theorem 3 shows that the speed of factor is less than \( O\left( \frac{K^2 \ln K}{\gamma^2} \right) \leq 1 \), while the right term models the additional cost in terms of time steps of the low communication cost. In comparison to MEDIAN ELIMINATION run on a single player, the speed of factor is still in \( O\left( \frac{\alpha N \ln K}{\gamma^2} \right) \). In the experiments, we consider this worst case to compare the performance of the proposed algorithm with these two baselines.

### 4 Experimentation

To illustrate and complete the analysis of the proposed algorithm, we consider a simple best arm identification task: 10 arms; the optimal arm has a mean reward \( \mu_0 = 0.7 \), the second one \( \mu_1 = 0.5 \), the third one \( \mu_2 = 0.3 \), and the other have a mean reward of 0.1. MEDIAN ELIMINATION was designed to be optimal for the worst case, where the arms have means close to each other. This toy problem, where a few number of arms have high mean rewards and the other low mean rewards, is difficult for MEDIAN ELIMINATION. Indeed, this algorithm suppresses half arms at each step without taking into account the gap with the estimated best arm. That is why despite its optimality, for this problem it can spend too much times to eliminate the suboptimal arms.
The number of players versus the regret at time horizon is plotted for Distributed Median Elimination, and Median Elimination with minimum and maximum communication costs $(0, 2NT)$, which are used as baselines (see Figure 2). Firstly, we observe that in this worst case, where the players are uniformly distributed, Distributed Median Elimination is outperformed by Median Elimination with a maximum communication cost when the number of players is large. For 256 players, Distributed Median Elimination and Median Elimination on a single player does not end the first elimination epoch. Secondly, it is interesting to notice that Distributed Median Elimination fixes the main drawback of Median Elimination. For this difficult problem, where mean rewards are not close to each other, the distribution of elimination epochs introduces variation in the sets of remaining actions. The more the mean reward of an action is high, the more its probability to be less the median is low. That is why, for this difficult problem Distributed Median Elimination outperforms Median Elimination when the number of players is reasonable in comparison to the time horizon.

Figure 2: The number of players versus the regret at time horizon $10^6$ averaged over 100 trials is plotted for each algorithm with $\epsilon = 0.5$ and $\delta = 0.05$, and for Distributed Median Elimination with different values of $\alpha$.

Figure 3: The time horizon versus the regret averaged over 100 trials is plotted for each algorithm with $\epsilon = 0.5$, $\delta = 0.05$, and $N = 16$. 

8
When the parameter $\alpha$ is maximum ($\alpha = 1/2$), Distributed Median Elimination performs as well as Median Elimination (see Figure 3), while its communication cost is negligible in comparison to the one of Median Elimination (531 bits versus $32.10^6$). When the parameter $\alpha$ is lower than $1/2$, Distributed Median Elimination clearly outperforms the two baselines (see Figure 3).

5 Conclusion and future works

In order to distribute the best identification task as close as possible to the user’s devices, we have proposed a new problem setting, where the players are drawn from an unknown distribution. We provided and analyzed the algorithm Distributed Median Elimination for this problem. We have showed that its communication cost is near optimal in terms of transmitted number of bits, while the number of pulls per player is near optimal. In an illustrative experiment, we have compared the proposed algorithm with two natural baselines: Median Elimination with minimum and maximum communication cost. The problem is chosen to be difficult for Median Elimination approaches. At each epoch, introducing variations in the set of remaining actions, the proposed algorithm fixes the main drawback of Median Elimination and clearly outperforms the two baselines. Finally, when the number of events per player is not high enough, the negligible communication cost of the proposed algorithm allows to use a two stage approach: the first stage uniformly clusters the devices (for instance by cell id for mobile phones). Distributed Median Elimination is played on the clusters at the second stage. This two stage approach provides a speed-up factor linear in terms of number of clusters, facilitates privacy by processing data close to the end user, and in practice benefits of better performances.

These results are obtained when Assumption 1 holds: the mean reward of actions does not depend on the player. Future works will extend this distributed approach to case where Assumption 1 does not hold, and in particular for the contextual bandit problem. Indeed, Distributed Median Elimination is a basic brick, which can be extended to the selection of variables to build a distributed decision stump and then a distributed version of Bandit Forest ([Féraud et al (2016)]).

References

[Auer et al (2002)a] Auer, P., Cesa Bianchi, N., Fischer, P.: Finite-time Analysis of the Multiarmed Bandit Problem, Machine Learning, 47, 235-256, 2002.

[Auer et al (2002)b] Auer, P., Cesa-Bianchi, N., Freund, Y ., Schapire, R. E.: The nonstochastic multiarmed bandit problem, SIAM J. COMPUT., 32 48-77, 2002.

[Bubeck et al (2009)] Bubeck, S., Wang, T., Stoltz: Pure exploration in multi-armed bandits problems, COLT, 2009.

[Cover and Thomas (2006)] Cover, T. M., Joy A. Thomas, T. A.: Elements of Information Theory, Wiley-Interscience, 2006.

[Even-Dar et al (2002)] Even-Dar, E., Mannor, S., Mansour Y .: PAC Bounds for Multi-armed Bandit and Markov Decision Processes, COLT, 255-270, 2002.

[Féraud et al (2016)] Féraud, R., Allesiardo, R., Urvoy, T., Clérot, F.: Random Forest for the Contextual Bandit Problem, AISTATS, 2016.

[Szörényi et al (2013)] Szörényi, B., Busa-Fekete, R., Hegedűs, I., Ormándi, R., Jelasity, M., Kégl, B.: Gossip-based distributed stochastic bandit algorithms, ICML, 2013.

[Hillel et al (2013)] Hillel, E., Karnin, Z., Koren, T. Lempel, R., Somekh, O.: Distributed Exploration in Multi-Armed Bandits, NIPS, 2013.

[Kanade et al (2012)] Kanade, V., Liu Z., Radunović, B.: Distributed Non-Stochastic Experts, NIPS, 2012.

[Kaufman et al (2012)] Kaufman, E., Korda, N., Munos, R.: Thompson sampling: An asymptotically optimal finite time analysis, COLT, 2012.
[Mannor and Tsitsiklis (2004)] Mannor, S., and Tsitsiklis, J. N.: The Sample Complexity of Exploration in the Multi-Armed Bandit Problem, JMLR, 2004

[Soare et al (2014)] Soare, M., Lazaric, L., Munos, R.: Best-Arm Identification in Linear Bandits, NIPS, 2014.

[Urvoy et al (2013)] Urvoy, T., Clérot, F., Féraud, R., Naamane, S.: Generic Exploration and K-armed Voting Bandits, ICML, 2013.

[Vailant (1984)] Valiant, L.: A theory of the learnable, Communications of the ACM, 27, 1984.