Gossip-based distributed stochastic bandit algorithms

Balázs Szörényi[1,2] SZORENYI@INF.U-SZEGED.HU
Róbert Busa-Fekete[2,3] BUSAROBI@INF.U-SZEGED.HU
István Hegedűs[2] IHEGEDUS@INF.U-SZEGED.HU
Róbert Ormándi[2] ORMANDI@INF.U-SZEGED.HU
Márk Jelasity[2] JELASITY@INF.U-SZEGED.HU
Balázs Kégl[4] BALAZS.KEGL@GMAIL.COM
[1]INRIA Lille - Nord Europe, SequeL project, 40 avenue Halley, 59650 Villeneuve d'Ascq, France
[2]Research Group on AI, Hungarian Acad. Sci. and Univ. of Szeged, Aradi vértanúk tere 1., H-6720 Szeged, Hungary
[3]Mathematics and Computer Science, University of Marburg, Hans-Meerwein-Str., 35032 Marburg, Germany
[4]Linear Accelerator Laboratory (LAL) & Computer Science Laboratory (LRI), CNRS/University of Paris Sud, 91405 Orsay, France

Abstract

The multi-armed bandit problem has attracted remarkable attention in the machine learning community and many efficient algorithms have been proposed to handle the so-called *exploitation-exploration dilemma* in various bandit setups. At the same time, significantly less effort has been devoted to adapting bandit algorithms to particular architectures, such as sensor networks, multi-core machines, or peer-to-peer (P2P) environments, which could potentially speed up their convergence. Our goal is to adapt stochastic bandit algorithms to P2P networks. In our setup, the same set of arms is available in each peer. In every iteration each peer can pull one arm independently of the other peers, and then some limited communication is possible with a few random other peers. As our main result, we show that our adaptation achieves a linear speedup in terms of the number of peers participating in the network. More precisely, we show that the probability of playing a suboptimal arm at a peer in iteration $t = \Omega(\log N)$ is proportional to $1/(Nt)$ where N denotes the number of peers. The theoretical results are supported by simulation experiments showing that our algorithm scales gracefully with the size of network.

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

1. Introduction

The recent appearance of large scale, unreliable, and fully decentralized computational architectures provides a strong motivation for adapting machine learning algorithms to these new computational architectures. One traditional approach in this area is to use gossip-based algorithms, which are typically simple, scalable, and efficient. Besides the simplest applications, such as computing the average of a set of numbers (Kempe et al., 2003; Jelasity et al., 2005; Xiao et al., 2007), this approach can be used to compute global models of fully distributed data. To name a few, Expectation-Maximization for Gaussian Mixture learning (Kowalczyk & Vlassis, 2005), linear Support Vector Machines (Ormándi et al., 2012), and boosting (Hegedűs et al., 2012) were adapted to this architecture. The goal of this paper is to propose a gossip-based stochastic multi-armed bandit algorithm.

1.1. Multi-armed bandits

Multi-armed bandits tackle an iterative decision making problem where an agent chooses one of the K previously fixed arms in each round t, and then it receives a random reward that depends on the chosen arm. The goal of the agent is to optimize some evaluation metric such as the *error rate* (the expected percentage of playing a suboptimal arm) or the *cumulative regret* (the expected difference of the sum of the obtained rewards and the sum of the rewards that could have been obtained by selecting the best arm in each round). In the *stochastic* multi-armed bandit setup, the distributions can vary with the arms but do not change with time. To achieve the desired goal, the agent has to trade off using arms found to be good based on earlier plays (exploitation) and trying arms that have not been tested enough times (exploration) (Auer et al., 2002; Cesa-Bianchi & Lugosi, 2006; Lai & Robbins, 1985). According to a result by Lai & Robbins (1985), no algorithm can have an error rate $o(1/t)$. One can thus consider policies with error rate $O(1/t)$ to be asymptotically optimal. An example of such a method is the ϵ-GREEDY algorithm of Auer et al. (2002).

Multi-armed bandit algorithms have generated significant theoretical interest, and they have been applied to many real applications. Some of these decision problems are clearly relevant in a distributed context. Consider, for example, a fully decentralized recommendation system, where we wish to recommend content based on user feedback without running through a central server (e.g., for privacy reasons). Another example is real-time traffic planning using a decentralized sensor network in which agents try to optimize a route in a common environment. Our algorithm is probably not applicable per se to these settings (in the first example, contextual bandits (Langford & Zhang, 2007) are arguably more adequate, and in the second example the environment is non-stationary and the agents might be adversarial (Cesa-Bianchi & Lugosi, 2006)), but it is a first step in developing theoretically sound and practically feasible solutions to problems of this kind.

1.2. P2P networks

A P2P network consists of a large collection of nodes (peers) that communicate with each other directly without any central control. We assume that each node has a unique address. The communication is based on message passing. Each node can send messages to any other node assuming that the address of the target node is available locally. This "knows about" relation defines an *overlay network* that is used for communication.

In this paper two types of overlay networks are considered. In the theoretical analysis (Sections 2 and 3) we will use the PERFECTOVERLAY protocol in which each node is connected to exactly two distinct neighbors, which means that the communication graph is the union of disjunct circles. Within this class, the neighbor assignment is uniform random, and it changes in each communication round. This protocol has no known practical decentralized implementation, so in our experiments (Section 5) we use the practically feasible NEWSCAST protocol of Jelasity et al. (2007). In this protocol each node sends messages to two distinct nodes selected randomly in each round. The main difference between the two protocols is that in PERFECTOVERLAY each node receives *exactly two* messages in each round whereas in NEWSCAST the number of received messages by any node follows a Poisson distribution with parameter 2 (when N is large).

1.3. P2P stochastic bandits and our results

In our P2P bandit setup, we assume that each of the N peers has access to the same set of K arms (with the same unknown distributions that does not change with time—hence the setting is stochastic), and in every round each peer pulls one arm independently. We also assume that on each peer, an individual instance of the same bandit algorithm is run. The peers can communicate with each other by sending messages in each round exclusively along the links of the applied overlay network. In this paper we adapt the stochastic ϵ-GREEDY bandit algorithm[1] of Auer et al. (2002) to such an architecture.

Our main theoretical goal is to assess the achievable speedup as a function of N. First, note that after T rounds of arm-pulling and communicating, the number of total plays is NT so (recalling the bound by Lai & Robbins 1985) the order of magnitude of the best possible error rate is $1/(NT)$. In Section 3, we show that our algorithm achieves error rate $O(1/(d^2Nt))$ for a number of rounds $T = \Omega(\log N)$, where d is a lower bound on the gap between the expected reward on i^* and any suboptimal arm. Consequently, the regret is also of the order $O\big(\log(NT)/d^2 + N\min(t, \log N)\big)$, where $N\min(t, \log N)$ is essentially the cost of spreading the information in the network. [2] The simulation experiments (Section 5) also show that our algorithm scales gracefully with the size of the network, giving further support to our theoretical results.

1.4. Related research

Gelly et al. (2008) addresses the exploration-exploitation dilemma within a distributed setting. They introduce a heuristic for a multi-core parallelization of the UCT algorithm (Kocsis & Szepesvári, 2006). Note, however, that multi-core parallelization is simpler than tackling fully distributed environments. The reason is that in the multi-core architectures, the individual computational units have access to a shared memory making

[1]See Section E in the Supplementary for a more detailed discussion about our choice, and why applying algorithms like UCB (Auer et al., 2002) directly would be suboptimal in this setup.

[2]If, in each round, each peer communicates with a constant number of peers, it takes $\Omega(\log N)$ time for a peer to spread information to at least a linear portion of the rest of the peers. See a more detailed discussion in Section E in the supplementary material.

information exchange cheap, quick, and easy. The large data flow generated by a potentially complete information exchange in a fully distributed environment is clearly not feasible in real-life applications.

Awerbuch & Kleinberg (2008) consider a problem where a network of individuals face a sequential decision problem: in each round they have to choose an action (for example, a restaurant for dinner), then they receive a reward based on their choice (how much they liked the place). The individuals can also communicate with each other, making it possible to reduce the regret by sharing their opinions. This distributed recommendation system can be interpreted as a multi-armed bandit problem in a distributed network, just like ours, but with three significant differences. The first is that they consider the adversarial setting (that is, in contrast to our stochastic setting, the distributions of the arms can change with time). The second is that their bound on the regret is $O((1 + K/N)(\log N)T^{2/3}\log T)$ per individual, and thus the total regret over the whole network of individuals is $O((N+K)(\log N)T^{2/3}\log T)$. This is linear in the number of peers, contrary to our logarithmic dependence. Finally, they allow for a communication phase of $\log N$ rounds between the consecutive arm pulling, which makes the problem much easier than in our setup.

Both the fact that peers act in parallel, and that we introduce a delay between pulling the arms relates our approach to setups with delayed feedback (Joulani, 2012). (Similar, but not bandit problem is considered by (Langford et al., 2009)) In this model, in round t, for each arm i a random value $\tau_{i,t}$ is drawn, and the reward for pulling arm i is received in round $t + \tau_{i,t}$. However, the regret bounds in Joulani (2012) grow linearly in the length of the expected delay, which is unusable in our setup where the delay grows exponentially with T.

Our algorithm shows some superficial resemblance with the Epoch-greedy algorithm introduced by Langford & Zhang (2007). Epoch-greedy is also based on the ϵ-greedy algorithm and, just like ours, it updates the arm selection rule based on new information only at the end of the epochs. However, besides these similarities the two algorithms are very different, and provide solutions to completely different problems. In epoch-greedy the original epsilon-greedy algorithm is modified in several crucial points, of which the most important is that they decouple the exploration and exploitation steps: exploration is only done in the last round of the epochs. This is favorable in that specific contextual bandit setting they work with, but would be harmful in our setup, since it would generate too large regret.

Finally, it should be stressed that our main contribution is the general approach to adapt ϵ-greedy to decentralized architectures with limited communication, such as P2P networks. It is not clear though how to do this with other algorithms.

2. P2P-ϵ-greedy: a peer-to-peer ϵ-greedy stochastic bandit algorithm

In this section, we present our algorithm. Let $N, K \in \mathbb{N}^+$ denote the number of peers and the number of arms, respectively. For the easier analysis we assume that N is a power of 2, that is $N = 2^m$ for some $m \in \mathbb{N}$. Throughout the description of our algorithm and its analysis, we use the PerfectOverlay protocol which means that each peer sends messages to two other peers and receives messages from the same two peers in each round.

Arms, peers, and rounds will be indexed by $i = 1, \ldots, K$, $j, j' = 1, \ldots, N$, and $t, t' = 1, \ldots, T$, respectively. μ^i denotes the mean of the reward distribution for arm i. The indicator $\mathbb{I}^i_{j,t}$ is 1 if peer j pulls arm i in round t, and 0 otherwise. The immediate reward observed by peer j in round t is $\xi_{j,t}$. In the standard setup, if all rewards were communicated immediately to all peers, μ_i would be estimated in round t by $\hat{\mu}^i_t = s^i_t/n^i_t$ where $s^i_t = \sum_{t'=1}^{t}\sum_{j'=1}^{N}\mathbb{I}^i_{j',t'}\xi_{j',t'}$ is the sum of rewards and $n^i_t = \sum_{t'=1}^{t}\sum_{j'=1}^{N}\mathbb{I}^i_{j',t'}$ is the number of times arm i was pulled. Using the PerfectOverlay protocol, each peer j sends its s and n estimates to its two neighbors, peer j_1 and j_2,[3] then peer j updates its estimates by averaging the estimates of its neighbors. Formally, in each round t, the estimates at each peer j can be expressed as *weighted* sums $s^i_{j,t} = \sum_{t'=1}^{t}\sum_{j'=1}^{N} w^{j',t'}_{j,t}\mathbb{I}^i_{j',t'}\xi_{j',t'}$ and $n^i_{j,t} = \sum_{t'=1}^{t}\sum_{j'=1}^{N} w^{j',t'}_{j,t}\mathbb{I}^i_{j',t'}$, where the weights are defined recursively as

$$w^{j',t'}_{j,t} = \begin{cases} 0 & \text{if } t < t' \vee (t = t' \wedge j \neq j') \\ N & \text{if } t = t' \wedge j = j' \\ \frac{1}{2}\left(w^{j',t'}_{j_1,t-1} + w^{j',t'}_{j_2,t-1}\right) & \text{if } t > t'. \end{cases} \quad (1)$$

It is then obvious that for $t > 1$, $s^i_{j,1} = \mathbb{I}^i_{j,1}\xi_{j,1}$ and

$$s^i_{j,t} = \tfrac{1}{2}(s^i_{j_1,t-1} + s^i_{j_2,t-1}) \ . \quad (2)$$

Once we have an estimate $\hat{\mu}^i_{j,t} = s^i_{j,t}/n^i_{j,t}$, the stan-

[3] j_1 and j_2 can change in every round, so we should write $j_{1,j,t}$ and $j_{2,j,t}$. We use j_1 and j_2 to ease notation.

dard ϵ-GREEDY policy of Auer et al. (2002) is to choose the optimal arm (arm i for which $\hat{\mu}^i_{j,t}$ is maximal) with probability $1-\epsilon_t$, and a random arm with probability ϵ_t, with ϵ_t converging to 0 at a speed of $1/t$. The problem with this strategy in the P2P environment is that rewards received in recent rounds do not have time to spread, making the standard $s^i_{j,t}/n^i_{j,t}$ biased. To control this bias, we do not use rewards $\xi_{j,t}$ immediately after time t, rather we collect them in auxiliary variables and work them into the estimates only after a delay that grows exponentially with time. For the formal description, let $s^i_{j,t}(t_1,t_2) = \sum_{t'=t_1}^{t_2}\sum_{j'=1}^{N} w^{j',t'}_{j,t}\mathbb{I}^i_{j',t'}\xi_{j',t'}$ and $n^i_{j,t}(t_1,t_2) = \sum_{t'=t_1}^{t_2}\sum_{j'=1}^{N} w^{j',t'}_{j,t}\mathbb{I}^i_{j',t'}$, and let $\mathcal{T}(t) = 2^{\lfloor\log(t-1)\rfloor}$ the "$\log_2$ floor" (the largest integer power of 2 which is less then of t). With this notation, the reward estimate of P2P-ϵ-GREEDY is

$$\hat{\mu}^i_{j,t} = c^i_{j,t}/d^i_{j,t}, \tag{3}$$

where $c^i_{j,t} = s^i_{j,t}\big(1, \mathcal{T}(t/2)-1\big)$ and $d^i_{j,t} = n^i_{j,t}\big(1,\mathcal{T}(t/2)-1\big)$. The simple naive implementation of the algorithm would be to communicate the weight matrix $\big[w^{j',t'}_{j,t}\big]^{t'=1,\dots,t}_{j'=1,\dots,j}$ between neighbors in each round t, and to compute $\hat{\mu}^i_{j,t}$ according to (3) and (1). This would, however, imply a linear communication cost in terms of the number of rounds t. It turns out that it is sufficient to send six vectors of size K to each neighbor to compute (3). Indeed, the quantities $a^i_{j,t} = s^i_{j,t}\big(\mathcal{T}(t),t\big)$, $b^i_{j,t} = n^i_{j,t}\big(\mathcal{T}(t),t\big)$, $r^i_{j,t} = s^i_{j,t}\big(\mathcal{T}(t/2),\mathcal{T}(t)-1\big)$, $q^i_{j,t} = n^i_{j,t}\big(\mathcal{T}(t/2),\mathcal{T}(t)-1\big)$, $c^i_{j,t}$, and $d^i_{j,t}$ can be updated by

$$\begin{aligned} c^i_{j,t} &= c^i_{j,t} + r^i_{j,t}, & d^i_{j,t} &= d^i_{j,t} + q^i_{j,t} \\ r^i_{j,t} &= a^i_{j,t}, & q^i_{j,t} &= b^i_{j,t} \\ a^i_{j,t} &= 0, & b^i_{j,t} &= 0 \end{aligned} \tag{4}$$

each time when t is an integer power of 2, and by

$$a^i_{j,t+1} = a^i_{j,t} + N\mathbb{I}^i_{j,t}\xi_{j,t} \text{ and } b^i_{j,t+1} = b^i_{j,t} + N\mathbb{I}^i_{j,t} \tag{5}$$

in every round t. In addition, in each iteration t, preceeding (5) and (4), all the six vectors are updated by aggregating the neighbors, similarly to (2).

The intuitive rationale of the procedure is the following. A run is divided into epochs: the ℓ-th epoch starts in round $t = 2^\ell$ and ends in round $t = 2^{\ell+1}-1$. During the ℓth epoch, the rewards $\xi_{j,t}$ are collected in the vector $\mathbf{a}_{j,t} = [a^i_{j,t}]_{i=1,\dots,K}$ and counted in $\mathbf{b}$. At the end of the epoch, they are copied into $\mathbf{r}$ and $\mathbf{q}$ respectively. The rewards and the counts are finally copied into $\mathbf{c}$ and $\mathbf{d}$, respectively, at the end of epoch $(\ell+1)$. In other words, a reward obtained in iteration t will not be used to estimate the expected reward until the iteration $2\cdot 2^{\log\lfloor t-1\rfloor}$. This procedure allows the rewards to "spread" in the network for a certain time before being used to estimate te expected reward, which makes is possible to formally control the bias of the estimates.

The pseudocode of P2P-ϵ-GREEDY is summarized in Algorithm 1. Formally, a *model* $\mathcal{M}$ is a 6-tuple $(\mathbf{c},\mathbf{d},\mathbf{r},\mathbf{q},\mathbf{a},\mathbf{b})$ where each component is a vector in $\mathbb{R}^K$. Peer j requests models $\mathcal{M}_{j_1,t}$ and $\mathcal{M}_{j_2,t}$ from its two neighbors j_1 and j_2 (Line 1), aggregates them into a new model $\mathcal{M}_{j,t}$ (Line 3), chooses an arm $i_{j,t}$ based on $\mathcal{M}_{j,t}$ (Lines 7–8), and then updates $\mathcal{M}_{j,t}$ based on the obtained reward (Line 10). When j is asked for send a model, it sends its updated $\mathcal{M}_{j,t+1}$.

3. Analysis

Before stating the main theorem, we introduce some additional notations. The index of the unique optimal arm is denoted $i^* = \arg\max_{1\le i\le K}\mu_i$. Let $\Delta_i = \mu_{i^*} - \mu_i$. We assume (as Auer et al. 2002) that there exist a lower bound d on the difference between μ^*_i and the expected reward of the second best arm, that is, $\exists d: 0 < d \le \min_{i\ne i^*}\Delta_i$. Our main result is the following.

Theorem 1. *Consider a P2P network of N peers with a* PERFECTOVERLAY *protocol. Assume that the same K arms are available at each peer and that the rewards come from $[0,1]$. Then, for any $c>0$, the probability of selecting a suboptimal arm $i\ne i^*$ at any peer by* P2P-ϵ-GREEDY *after $t \ge cK/(d^2N)$ iterations is at most*

$$\frac{c}{d^2tN} + 2\left(\frac{c}{d}\ln\frac{Ntd^2e^{1/2}}{cK}\right)\left(\frac{cK}{Ntd^2e^{1/2}}\right)^{\frac{c}{3d}} + \\ + \frac{4e}{d^2}\left(\frac{cK}{Ntd^2e^{1/2}}\right)^{\frac{c}{2}} + \frac{4608}{\Delta_i^2}N^3 2^{-t/2} \; . \tag{6}$$

The first three terms of (6) correspond to the bound given by Auer et al. (2002) for their version of the ϵ-greedy algorithm. The last term corresponds to the P2P overhead: it results from the imperfect information of a peer about the rewards received throughout the network. This last term decays exponentially and it becomes insignificant after $O(\log N)$ rounds.

The following corollary is a reformulation of Theorem 1 in terms of the regret. Stochastic bandit algorithms are usually evaluated in terms of the expected regret $R^t = \sum_{i\ne i^*}\Delta_i\sum_{t'=1}^{t}\mathbb{P}[i_{t'}=i]$, where $\sum_{t'=1}^{t}\mathbb{P}[i_{t'}=i]$ is the expected number of times arm i is pulled up to round t. In our P2P setup, an arm is pulled in each round t and at each peer j, so we

Algorithm 1 P2P-ϵ-GREEDY at peer j in iteration t

```
 1: Receive $\mathcal{M}_{j_1,t}$ and $\mathcal{M}_{j_2,t}$ from the two current neighbors
 2: Let $\epsilon_t = \min\left(1, \frac{cK}{d^2tN}\right)$          ▷ $c > 0$ is a real-valued parameter controlling the exploration
 3: $\mathcal{M}_{j,t} = \text{AGGREGATE}(\mathcal{M}_{j_1,t}, \mathcal{M}_{j_2,t})$
 4: if $t = 1$ then
 5:     Let $i_{j,t} = j \mod K$                                   ▷ Initial (arbitrary) arm-selection
 6: else
 7:     With probability $1-\epsilon_t$ let $i_{j,t} = \arg\max\{c^i_{j,t}/d^i_{j,t} : 1 \le i \le K, d^i_{j,t} > 0\}$     ▷ exploitation step
 8:     and with probability $\epsilon_t$ let $i_{j,t}$ be the index of a random arm     ▷ exploration step
 9: Pull arm $i_{j,t}$ and receive reward $\xi_{j,t}$
10: The model to be sent is $\mathcal{M}_{j,t+1} = \text{UPDATE}(\mathcal{M}_{j,t}, \xi_{j,t}, i_{j,t}, t)$
11: function AGGREGATE($\mathcal{M}' = (\mathbf{c}', \mathbf{d}', \mathbf{r}', \mathbf{q}', \mathbf{a}', \mathbf{b}'), \mathcal{M}'' = (\mathbf{c}'', \mathbf{d}'', \mathbf{r}'', \mathbf{q}'', \mathbf{a}'', \mathbf{b}'')$)
12:     $\mathbf{c} = (1/2)(\mathbf{c}' + \mathbf{c}'')$, $\mathbf{d} = (1/2)(\mathbf{d}' + \mathbf{d}'')$     ▷ Elementwise vector operators
13:     $\mathbf{r} = (1/2)(\mathbf{r}' + \mathbf{r}'')$, $\mathbf{q} = (1/2)(\mathbf{q}' + \mathbf{q}'')$
14:     $\mathbf{a} = (1/2)(\mathbf{a}' + \mathbf{a}'')$, $\mathbf{b} = (1/2)(\mathbf{b}' + \mathbf{b}'')$
15:     return $\mathcal{M} = (\mathbf{c}, \mathbf{d}, \mathbf{r}, \mathbf{q}, \mathbf{a}, \mathbf{b})$
16: function UPDATE($\mathcal{M} = (\mathbf{c}, \mathbf{d}, \mathbf{r}, \mathbf{q}, \mathbf{a}, \mathbf{b}), \xi, i, t$)
17:     if $t$ is an integer power of 2 then
18:         $\mathbf{c} = \mathbf{c} + \mathbf{r}$,   $\mathbf{d} = \mathbf{d} + \mathbf{q}$, $\mathbf{r} = \mathbf{a}$,   $\mathbf{q} = \mathbf{b}$, $\mathbf{a} = \mathbf{b} = \mathbf{0}$
19:     $a^i = a^i + N\xi$,   $b^i = b^i + N$
20:     return $\mathcal{M}$
```

are interested in upper bounding the sum of the expected regrets incurred at each peer

$$\mathcal{R}^t = \sum_{i \neq i^*} \Delta_i \sum_{t'=1}^{t} \sum_{j'=1}^{N} \mathbb{P}\left[i_{j',t'} = i\right], \tag{7}$$

where $\mathbb{P}\left[i_{j,t} = i\right] = \mathbb{P}\left[\mathbb{I}^i_{j,t} = 1\right]$ is the probability that peer j pulls arm i in round t. Since the last term of (6) becomes close to 0 only after $O(\log N)$ rounds, we will not bound the total regret starting at round zero, rather starting at round $\tilde{t}(N) = O(\log N)$. This implies that the total regret will be increased by a $O(N \log N)$ term, as explained in Section 1.3.

Corollary 2. *Let $\mathcal{R}^t$ (7) denote the expected regret for the whole network after t iterations in the* P2P-ϵ-GREEDY *algorithm. Then $\mathcal{R}^t - \mathcal{R}^{\tilde{t}(N)} = O\big(\log(Nt)/d^2\big)$ for some $\tilde{t}(N) = O(\log N)$.*

We start the analysis by investigating $\mathbf{c}_{j,t}$ in a particular peer j. For any arm $1 \le i \le K$ and any peer $1 \le j \le N$, each component of $\mathbf{c}_{j,t}$ can be rewritten as the weighted sum of individual rewards received up to iteration $\mathcal{T}(t/2) - 1$, and then decomposed as

$$\begin{aligned} c^i_{j,t} &= s^i_{j,t}\big(1, \mathcal{T}(t/2) - 1\big) = \sum_{t'=1}^{\mathcal{T}(t/2)-1} \sum_{j'=1}^{N} w^{j',t'}_{j,t} \mathbb{I}^i_{j',t'} \xi_{j',t'} \\ &= \underbrace{\sum_{t'=1}^{\mathcal{T}(t/2)-1} \sum_{j'=1}^{N} (w^{j',t'}_{j,t} - 1) \mathbb{I}^i_{j',t'} \xi_{j',t'}}_{z^i_{j,t}\text{, corresponds to (A) in the proof}} \\ &\quad + \underbrace{\sum_{t'=1}^{\mathcal{T}(t/2)-1} \sum_{j'=1}^{N} \mathbb{I}^i_{j',t'} \xi_{j',t'}}_{\text{recovered sum ((B) in the proof)}} \end{aligned} \tag{8}$$

The following lemma states some important properties of the weights.

Lemma 3. *For any rounds t' and $t \ge t'$, and any peer j', the weights of the reward $\xi_{j',t'}$ in round t sum up to N: $\sum_{j=1}^{N} w^{j',t'}_{j,t} = N$. Furthermore, for any $t > t'$, the weight $w^{j',t'}_{j,t}$ is a random variable, it is independent of j and $\xi_{j',t'}$, and the distribution of $w^{j',t'}_{j,t}$ is identical at each peer j.*

Proof. The first statement follows trivially from the definition of the weights (1). The independence of the weights of the peer indices and of the rewards is true since the random assignments of neighbors of the PERFECTOVERLAY protocol is independent of the bandit game. □

The following lemma can be thought of as bounding the "horizontal variance": focusing on just one specific reward $\xi_{j',t'}$, it bounds the variance of its weights $w_{j,t}^{j',t'}$ throughout the network $j = 1, \dots, N$ in a given iteration t.

Lemma 4. *For any $t' \geq 1$, $t > t'$, and $1 \leq j' \leq N$, we have $\mathbb{E}\left[\sum_{j=1}^{N}(w_{j,t}^{j',t'} - 1)^2\right] \leq N^2/2^{t-t'}$. Furthermore, $\mathbb{E}_j\left[(w_{j,t}^{j',t'} - 1)^2\right] \leq N/2^{t-t'}$.*

Proof. The proof of the first statement follows Kempe et al. (2003) and Jelasity et al. (2005) and it is included in the supplementary material. The last claim is true because the distributions of $w_{j,t}^{j',t'}$, $j = 1, \dots, N$, are identical (Lemma 3). □

Using Lemma 4 we can now bound the variance of the first term on the right hand side in (8), and the variance of the first term of a similar decomposition of $d_{j,t}^i$. We start with the latter.

Lemma 5. *For any $t \geq 1$, any $1 \leq j \leq N$, and any $1 \leq i \leq K$, the random variable $y_{j,t}^i = \sum_{t'=1}^{\mathcal{T}(t/2)-1}\sum_{j'=1}^{N}(1 - w_{j,t}^{j',t'})\mathbb{I}_{j',t'}^i$ has zero mean and variance of at most $12N^3 2^{-t/2}$.*

Proof. The zero mean is a consequence of Lemma 3. For the variance, we have

$$\begin{aligned}\operatorname{Var}\left[y_{j,t}^i\right] &= \sum_{t',t''=1}^{\mathcal{T}(t/2)-1}\sum_{j',j''=1}^{N}\mathbb{E}\Big[\mathbb{I}_{j',t'}^i\mathbb{I}_{j'',t''}^i\left(1 - w_{j,t}^{j',t'}\right) \\ &\qquad\qquad \times\left(1 - w_{j,t}^{j'',t''}\right)\Big] \\ &\leq N\sum_{t',t''=1}^{\mathcal{T}(t/2)-1}\sqrt{\tfrac{1}{2^{t-t'}}}\sqrt{\tfrac{1}{2^{t-t''}}} \qquad (9) \\ &\leq N^3 2^{-t/2}\left(\tfrac{1}{1-1/\sqrt{2}}\right)^2 \leq 12N^3 2^{-t/2} ,\end{aligned}$$

where (9) follows from Lemma 4 and the Cauchy-Schwarz inequality. □

Lemma 6. *For any $t \geq 1$, any $1 \leq j \leq N$, and any $1 \leq i \leq K$, the random variable $z_{j,t}^i = \sum_{t'=1}^{\mathcal{T}(t/2)-1}\sum_{j'=1}^{N}(1 - w_{j,t}^{j',t'})\mathbb{I}_{j',t'}^i\xi_{j',t'}$ has zero mean and variance of at most $\operatorname{Var}\left[z_{j,t}^i\right] \leq 12N^3 2^{-t/2}$.*

Proof. The first step is to exploit the fact that $\xi_{j,t} \in [0, 1]$. Then the proof is analogous to the proof of Lemma 5. □

Proof. of Theorem 1 **(sketch)** We first control the first term (A) in (8) by analyzing a version of ϵ-GREEDY where N independent plays are allowed per iteration. We follow closely the analysis of ϵ-GREEDY of Auer et al. (2002) with some trivial modifications. Then in (B) we relate this to P2P-ϵ-GREEDY and show that the difference is negligible.

Assume that $t \geq cK/(d^2N)$, let $\epsilon_j = cK/(d^2jN)$, and let $x_0 = \frac{N}{2K}\sum_{j=1}^{t}\epsilon_j$. The probability of choosing some arm i in round t at peer j is

$$\mathbb{P}\left[i_{j,t} = i\right] \leq \tfrac{\epsilon_t}{K} + (1 - \epsilon_t)\,\mathbb{P}\left[c_{j,t}^i/d_{j,t}^i \geq c_{j,t}^{i^*}/d_{j,t}^{i^*}\right],$$

where $i^* = \arg\max_{1\leq i\leq K}\mu_i$. The second term can be decomposed as

$$\begin{aligned}&\mathbb{P}\left[c_{j,t}^i/d_{j,t}^i \geq c_{j,t}^{i^*}/d_{j,t}^{i^*}\right] \\ &\leq \mathbb{P}\left[\tfrac{c_{j,t}^i}{d_{j,t}^i} \geq \mu_i + \tfrac{\Delta_i}{2}\right] + \mathbb{P}\left[\tfrac{c_{j,t}^{i^*}}{d_{j,t}^{i^*}} \leq \mu_{i^*} - \tfrac{\Delta_i}{2}\right]. \quad (10)\end{aligned}$$

Now let $C_t^i = \sum_{t'=1}^{\mathcal{T}(t/2)-1}\sum_{j'}^{N}\xi_{j',t'}\mathbb{I}_{j',t'}^i$ ((B) in (8)) and $D_t^i = \sum_{t'=1}^{\mathcal{T}(t/2)-1}\sum_{j'}^{N}\mathbb{I}_{j',t'}^i$. Using the union bound, we bound the first term of (10) by

$$\begin{aligned}\mathbb{P}\left[\tfrac{c_{j,t}^i}{d_{j,t}^i} \geq \mu_i + \tfrac{\Delta_i}{2}\right] &\leq \mathbb{P}\left[C_t^i - \mu_i D_t^i \geq \tfrac{\Delta_i}{8}D_t^i\right] \\ &\quad + \mathbb{P}\left[c_{j,t}^i - C_t^i \geq \tfrac{\Delta_i}{8}D_t^i\right] \\ &\quad + \mathbb{P}\left[\mu_i\left(D_t^i - d_{j,t}^i\right) \geq \tfrac{\Delta_i}{8}D_t^i\right] \\ &\quad + \mathbb{P}\left[\tfrac{\Delta_i}{2}\left(D_t^i - d_{j,t}^i\right) \geq \tfrac{\Delta_i}{8}D_t^i\right] \\ &= T_1 + T_2 + T_3 + T_4\end{aligned}$$

We can upper bound T_1 following Auer et al. (2002). To upper bound T_2 recall that, by Lemma 6,

$$c_{j,t}^i - C_t^i = \sum_{t'=1}^{\mathcal{T}(t/2)-1}\sum_{j'=1}^{N}(1 - w_{j,t}^{j',t'})\mathbb{I}_{j',t'}^i\xi_{j',t'} = z_{j,t}^i$$

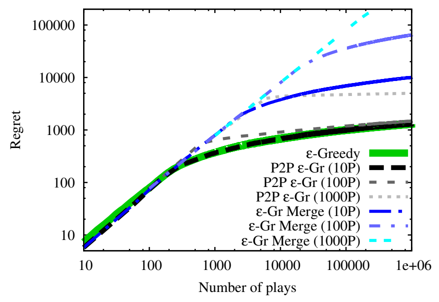
has expected value $\mathbb{E}\left[z_{j,t}^i\right] = 0$ and variance $\operatorname{Var}\left[z_{j,t}^i\right] \leq 12 \cdot N^3 2^{-t/2}$. Now apply Chebyshev's inequality for $z_{j,t}^i$ to get

$$T_2 \leq \mathbb{P}\left[\left|z_{j,t}^i\right| \geq \tfrac{\Delta_i}{8}\right] \leq \tfrac{768}{\Delta_i^2}N^3 2^{-t/2}.$$

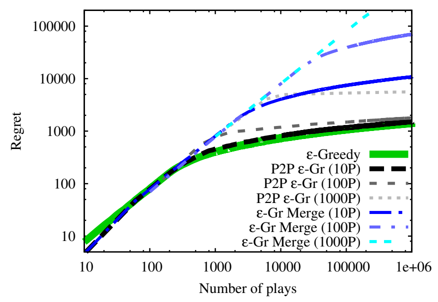
T_3 and T_4 can be upper bounded the same way using Lemma 5, so $T_2 + T_3 + T_4 \leq \frac{2304}{\Delta_i^2}N^3 2^{-t/2}$, and the second term of (10) can be upper bounded following the same steps. The proof can then be completed by a slight modification of the original proof of Auer et al. (2002) (see the supplementary material). □

4. P2P-ϵ-greedy.slim: a practical algorithm

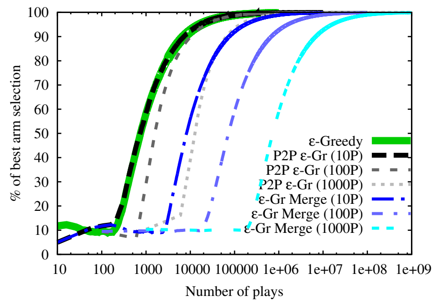
In P2P-ϵ-GREEDY, each peer sends its model to two other peers, inducing a network-wise communication



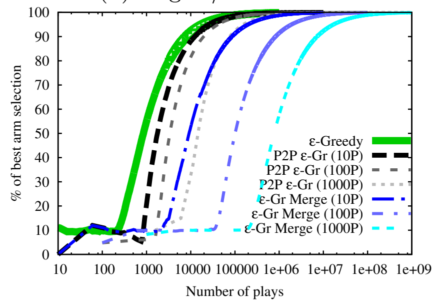
(a) Regret/PerfectOverlay



(b) Regret/Newscast



(c) Accuracy/PerfectOverlay



(d) Accuracy/Newscast

Figure 1. Comparison of ϵ-greedy and P2P-ϵ-greedy in terms of regret (upper panels) and accuracy (lower panels). We used the PerfectOverlay protocol in 1(a) and 1(c) and the Newscast protocol in 1(b) and 1(d).

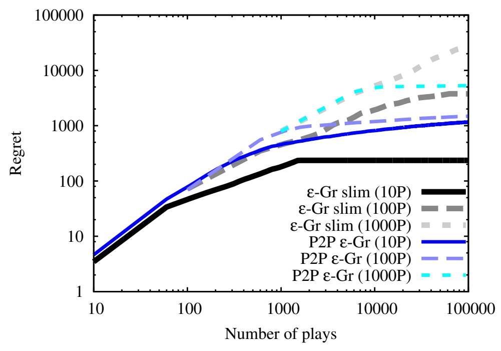
cost of $O(NK)$. This is impractical when K is large (e.g., $K \approx N$). In this section we present a practical algorithm with $O(N)$ communication cost. The main idea is that each peer sends and receives models about *only one* arm in each round. We have no formal proof about the convergence of the algorithm, but in experiments (Section 5) we found that it worked almost as well as P2P-ϵ-greedy.

In P2P-ϵ-greedy.slim, the model becomes $\mathcal{M} = (i, c, d, r, q, a, b)$, where $i \in \{0, \ldots, K\}$ is the index of the arm $\mathcal{M}$ stores information about, and c, d, r, q, a and b are scalar values corresponding to the vector variables in P2P-ϵ-greedy. In each iteration, peer j has its current model $\mathcal{M}$ corresponding to arm i, and it receives two models $\mathcal{M}_1$ and $\mathcal{M}_2$ corresponding to arms i_1 and i_2. Then it proceeds as follows. (For complete pseudocode see Section D.)

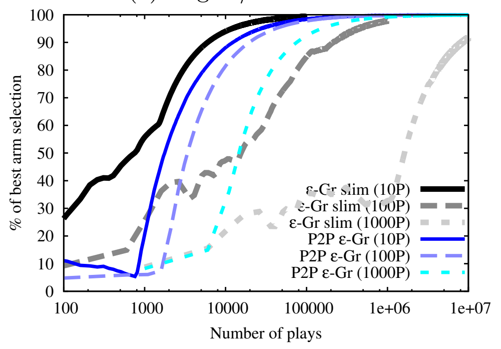
1. If $i_1 \neq i_2$, then let $\mathcal{M}' = \mathcal{M}_1$ if $c_1/d_1 > c_2/d_2$, and $\mathcal{M}' = \mathcal{M}_2$ otherwise. Go to Step 3.
2. If $i_1 = i_2$, then let $\mathcal{M}'$ the result of the aggregation of $\mathcal{M}_1$ and $\mathcal{M}_2$. Go to Step 3.
3. If $i' = i$, then let $\mathcal{M} = \mathcal{M}'$ (replace the current model with the incoming model). Go to Step 5.
4. If $i' \neq i$, then let $\mathcal{M}$ be the better (with the larger c/d) of $\mathcal{M}$ and $\mathcal{M}'$ with probability $1-\epsilon$, and the worse of the two models with probability ϵ. Go to Step 5.
5. Pull the arm i corresponding to the new model $\mathcal{M}$. Observe reward ξ. Add $N\xi$ to a and N to b. If t is an interger power of 2, update the model variables analogously to P2P-ϵ-greedy. Finally, send the updated model to its neighbors according to the particular protocol.

5. Experiments

In the first experiments, we verified our theoretical results in experiments on synthetic data. Our first goal was to verify the main claim of the paper, namely that the ϵ-greedy algorithm can achieve logarithmic regret after $\Omega(\log N)$ iterations in a P2P. Our second goal was to give empirical support to our epoch-based technique. We compared the performance of ϵ-greedy, P2P-ϵ-greedy, and a simplified version of the P2P-ϵ-greedy which only aggregates the models in each iteration and works the rewards into the mean estimates $(\mathbf{c}_{j,t}/\mathbf{d}_{j,t})$ immediately. We will refer to this simplified P2P algorithm as P2P-ϵ-Gr-merge. Although our regret analysis was carried out by assuming PerfectOverlay protocol, we also tested the P2P algorithms using the Newscast protocol. We used P2P networks with various sizes: $N = 10, 100, 1000$. We compared the performances of the algorithms in terms of their regret and their accuracy (rate of plays on which the best arm is selected). The test problem consisted of



(a) Regret/NEWSCAST



(b) Accuracy/NEWSCAST

Figure 2. Comparison of the P2P-ϵ-GREEDY and the P2P-ϵ-GREEDY.SLIM algorithms in terms of (a) regret and (b) accuracy using P2P networks of various sizes. The NEWSCAST protocol was used in every case.

$K = 10$ arms with Bernoulli distributions whose parameters were set to $\mu_i = 0.1 + 0.8(i-1)/(K-1)$. Accordingly, we set the parameter d to $0.07 < \min_i \Delta_i$. The only hyperparameter of the ϵ-greedy methods is set to $c = 0.1$. The performance measures (regret and accuracy) of the algorithms are plotted against number of plays in Figure 1. The results show averages over 10 repetitions of the simulation. We remark that the P2P adaptations of ϵ-greedy algorithm pulls N arms in each iteration, thus the curves concerning to P2P algorithms start at the Nth play.

The plots show that, first, the performance of P2P-ϵ-GREEDY scales gracefully with respect to the number of peers and its regret grows at the same speed as that of ϵ-GREEDY in accordance with our main result (Corollary 2). Furthermore, their regrets are also on a par with respect to the number of plays. Second, P2P-ϵ-GR-MERGE converges slower than P2P-ϵ-GREEDY which confirms empirically the need to delay using the rewards in the estimates.[4] Third, the performance of P2P-ϵ-GREEDY does not deteriorate significantly with the NEWSCAST protocol, which is an important experimental results from a practical point of view. Finally, note that the significant leap in the regret when $N = 1000$ is due to the $N \log N$ cost of spreading the information.[5]

[4]See more on this in Section E in the Supplementary.

In the second experiment, we compared the performance of P2P-ϵ-GREEDY.SLIM and P2P-ϵ-GREEDY using the same stochastic bandit setup as in the first experiment. We used NEWSCAST in the test runs. Both algorithms were run with the same parameters ($c = 0.1, d = 0.07$) using P2P networks of sizes $N = 10, 100, 1000$. Figure 2 shows the regret and accuracy against number of plays. The results are averaged over 10 repetitions of the simulation. P2P-ϵ-GREEDY.SLIM is slightly worse than P2P-ϵ-GREEDY but asymptotically it performs comparably for a K times smaller communication cost.

6. Conclusions and Further work

In this paper, we adapted the ϵ-GREEDY stochastic bandit algorithm to P2P architecture. We showed that P2P.ϵ-GREEDY preserves the asymptotic behavior of its standalone version, that is, the regret bound is $O(tN)$ for the P2P version if $t = \Omega(\log N)$, and thus achieves significant speed-up. Moreover, we presented a heuristic version of P2P.ϵ-GREEDY which has a lower network communication cost. Experiments support our theoretical results. As a further work, we plan to investigate how to adapt some appropriately randomized version of the UCB bandit algorithm(Auer et al., 2002) to P2P environment.

Acknowledgments

This work was supported by the ANR-2010-COSI-002 grant of the French National Research Agency, the European Union and the European Social Fund through project FuturICT.hu (grant no .: TAMOP-4.2.2.C-11/1/KONV-2012-0013), and by the Future and Emerging Technologies programme FP7-COSI-ICT of the European Commission through project QLectives (grant no.: 231200). M. Jelasity was supported by the Bolyai Scholarship of the Hungarian Academy of Sciences.

References

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

Awerbuch, Baruch and Kleinberg, Robert. Competitive collaborative learning. *J. Comput. Syst. Sci.*,

[5]See the discussion before Corollary 2 and in Section E.1 in the Supplementary.

74(8):1271–1288, December 2008. ISSN 0022-0000. doi: 10.1016/j.jcss.2007.08.004. URL http://dx.doi.org/10.1016/j.jcss.2007.08.004.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, NY, USA, 2006.

Gelly, S., Hoock, J.B., Rimmel, A., Teytaud, O., and Kalemkarian, Y. The parallelization of Monte-Carlo planning. In *Proceedings of of the Fifth International Conference on Informatics in Control, Automation and Robotics*, pp. 244–249, 2008.

Hegedűs, I., Busa-Fekete, R., Ormándi, R., Jelasity, M., and Kégl, B. Peer-to-peer multi-class boosting. In *International European Conference on Parallel and Distributed Computing (EURO-PAR)*, pp. 389–400, 2012.

Jelasity, M., Montresor, A., and Babaoglu, O. Gossip-based aggregation in large dynamic networks. *ACM Trans. on Computer Systems*, 23(3): 219–252, August 2005.

Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.-M., and van Steen, M. Gossip-based peer sampling. *ACM Transactions on Computer Systems*, 25(3):8, 2007.

Joulani, Pooria. Multi-armed bandit problems under delayed feedback. Msc thesis, Department of Computing Science, University of Alberta, 2012.

Kempe, D., Dobra, A., and Gehrke, J. Gossip-based computation of aggregate information. In *Proc. 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03)*, pp. 482–491. IEEE Computer Society, 2003.

Kocsis, L. and Szepesvári, Cs. Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, pp. 282–293, 2006.

Kowalczyk, W. and Vlassis, N. Newscast EM. In *17th Advances in Neural Information Processing Systems*, pp. 713–720, Cambridge, MA, 2005. MIT Press.

Lai, T.L. and Robbins, H. Asymptotically efficient allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Langford, John and Zhang, Tong. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, 2007.

Langford, John, Smola, Alex, and Zinkevich, Martin. Slow Learners are Fast. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22*, pp. 2331–2339. 2009.

Ormándi, R., Hegedüs, I., and Jelasity, M. Gossip learning with linear models on fully distributed data. *Concurrency and Computation: Practice and Experience*, 2012. doi: 10.1002/cpe.2858.

Xiao, L., Boyd, S., and Kim, S.-J. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67 (1):33–46, January 2007.