Distributed Clustering of Linear Bandits in Peer to Peer Networks

Nathan Korda NATHAN@ROBOTS.OX.AC.UK
MLRG, University of Oxford

Balázs Szörényi SZORENYI.BALAZS@GMAIL.COM
EE, Technion & MTA-SZTE Research Group on Artificial Intelligence

Shuai Li SHUAILI.SLI@GMAIL.COM
DiSTA, University of Insubria

Abstract

We provide two distributed confidence ball algorithms for solving linear bandit problems in peer to peer networks with limited communication capabilities. For the first, we assume that all the peers are solving the same linear bandit problem, and prove that our algorithm achieves the optimal asymptotic regret rate of any centralised algorithm that can instantly communicate information between the peers. For the second, we assume that there are clusters of peers solving the same bandit problem within each cluster, and we prove that our algorithm discovers these clusters, while achieving the optimal asymptotic regret rate within each one. Through experiments on several real-world datasets, we demonstrate the performance of proposed algorithms compared to the state-of-the-art.

1. Introduction

Bandits are a class of classic optimisation problems that are fundamental to several important application areas. The most prominent of these is recommendation systems, and they can also arise more generally in networks (see, e.g., (Li et al., 2013; Hao et al., 2015)).

We consider settings where a network of agents are trying to solve collaborative linear bandit problems. Sharing experience can improve the performance of both the whole network and each agent simultaneously, while also increasing robustness. However, we want to avoid putting too much strain on communication channels. Communicating every piece of information would just overload these channels. The solution we propose is a gossip-based information sharing protocol which allows information to diffuse across the network at a small cost, while also providing robustness.

Such a set-up would benefit, for example, a small start-up that provides some recommendation system service but has limited resources. Using an architecture that enables the agents (the client's devices) to exchange data between each other directly and to do all the corresponding computations themselves could significantly decrease the infrastructural costs for the company. At the same time, without a central server, communicating all information instantly between agents would demand a lot of bandwidth.

Multi-Agent Linear Bandits In the simplest setting we consider, all the agents are trying to solve the same underlying linear bandit problem. In particular, we have a set of *nodes* V, indexed by i, and representing a finite set of *agents*. At each time, t:

- a set of *actions* (equivalently, the *contexts*) arrives for each agent i, $\mathcal{D}_t^i \subset \mathcal{D}$ and we assume the set $\mathcal{D}$ is a subset of the unit ball in $\mathbb{R}^d$;
- each agent, i, chooses an action (context) $x_t^i \in \mathcal{D}_t^i$, and receives a *reward*

$$r_t^i = (x_t^i)^{\intercal}\theta + \xi_t^i,$$

  where θ is some unknown *coefficient vector*, and ξ_t^i is some zero mean, R-subGaussian noise;
- last, the agents can share information according to some protocol across a communication channel.

We define the *instantaneous regret* at each node i, and, respectively, the *cumulative regret* over the whole network to be:

$$\rho_t^i := \left(x_t^{i,*}\right)^{\intercal}\theta - \mathbb{E}r_t^i, \quad \text{and} \quad \mathcal{R}_t := \sum_{k=1}^{t}\sum_{i=1}^{|V|}\rho_t^i,$$

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

where $x_t^{i,*} := \arg\max_{x \in \mathcal{D}_t^i} x^\intercal \theta$. The aim of the agents is to minimise the rate of increase of cumulative regret. We also wish them to use a sharing protocol that does not impose much strain on the information-sharing communication channel.

Gossip protocol In a gossip protocol (see, e.g., (Kempe et al., 2003; Xiao et al., 2007; Jelasity et al., 2005; 2007)), in each round, an overlay protocol assigns to every agent another agent, with which it can share information. After sharing, the agents aggregate the information and, based on that, they make their corresponding decisions in the next round. In many areas of distributed learning and computation gossip protocols have offered a good compromise between low-communication costs and algorithm performance. Using such a protocol in the multi-agent bandit setting, one faces two major challenges.

First, information sharing is not perfect, since each agent acquires information from only one other (randomly chosen) agent per round. This introduces a bias through the unavoidable doubling of data points. The solution is to mitigate this by using a delay (typically of $O(\log t)$) on the time at which information gathered is used. After this delay, the information is sufficiently mixed among the agents, and the bias vanishes.

Second, in order to realize this delay, it is necessary to store information in a buffer and only use it to make decisions after the delay has been passed. In (Szörényi et al., 2013) this was achieved by introducing an epoch structure into their algorithm, and emptying the buffers at the end of each epoch.

The Distributed Confidence Ball Algorithm (DCB) We use a gossip-based information sharing protocol to produce a distributed variant of the generic Confidence Ball (CB) algorithm, (Abbasi-Yadkori et al., 2011; Dani et al., 2008; Li et al., 2010). Our approach is similar to (Szörényi et al., 2013) where the authors produced a distributed ϵ-greedy algorithm for the simpler multi-armed bandit problem. However their results do not generalise easily, and thus significant new analysis is needed. One reason is that the linear setting introduces serious complications in the analysis of the delay effect mentioned in the previous paragraphs. Additionally, their algorithm is epoch-based, whereas we are using a more natural and simpler algorithmic structure. The downside is that the size of the buffers of our algorithm grow with time. However, our analyses easily transfer to the epoch approach too. As the rate of growth is logarithmic, our algorithm is still efficient over a very long timescale.

The simplifying assumption so far is that all agents are solving the same underlying bandit problem, i.e. finding the same unknown θ-vector. This, however, is often unrealistic, and so we relax it in our next setup. While it may have uses in special cases, DCB and its analysis can be considered as a base for providing an algorithm in this more realistic setup, where some variation in θ is allowed across the network.

Clustered Linear Bandits Proposed in (Gentile et al., 2014; Li et al., 2016a;b), this has recently proved to be a very successful model for recommendation problems with massive numbers of users. It comprises a multi-agent linear bandit model agents' θ-vectors are allowed to vary across a clustering. This clustering presents an additional challenge to find the groups of agents sharing the same underlying bandit problem before information sharing can accelerate the learning process. Formally, let $\{U^k\}_{k=1,\dots,M}$ be a clustering of V, assume some coefficient vector θ^k for each k, and let for agent $i \in U^k$ the reward of action x_t^i be given by

$$r_t^i = (x_t^i)^\intercal \theta^k + \xi_t^i.$$

Both clusters and coefficient vectors are assumed to be initially unknown, and so need to be learnt on the fly.

The Distributed Clustering Confidence Ball Algorithm (DCCB) The paper (Gentile et al., 2014) proposes the initial centralised approach to the problem of clustering linear bandits. Their approach is to begin with a single cluster, and then incrementally prune edges when the available information suggests that two agents belong to different clusters. We show how to use a gossip-based protocol to give a distributed variant of this algorithm, which we call DCCB.

Our main contributions In Theorems 1 and 6 we show our algorithms DCB and DCCB achieve, in the multi-agent and clustered setting, respectively, near-optimal improvements in the regret rates. In particular, they are of order almost $\sqrt{|V|}$ better than applying CB without information sharing, while still keeping communication cost low. And our findings are demonstrated by experiments on real-world benchmark data.

2. Linear Bandits and the DCB Algorithm

The generic Confidence Ball (CB) algorithm is designed for a single agent linear bandit problem (i.e. $|V| = 1$). The algorithm maintains a confidence ball $C_t \subset \mathbb{R}^d$ within which it believes the true parameter θ lies with high probability. This confidence ball is computed from the observation pairs, $(x_k, r_k)_{k=1,\dots,t}$ (for the sake of simplicity, we dropped the agent index, i). Typically, the covariance matrix $A_t = \sum_{k=1}^t x_k x_k^\intercal$ and b-vector, $b_t = \sum_{k=1}^t r_k x_k$, are sufficient statistics to characterise this confidence ball. Then, given its current action set, $\mathcal{D}_t$, the agent selects the optimistic action, assuming that the true parameter sits in C_t, i.e. $(x_t, \sim) = \arg\max_{(x,\theta') \in \mathcal{D}_t \times C_t} \{x^\intercal \theta'\}$. Pseudo-code for CB is given in the Appendix A.1.

Gossip Sharing Protocol for DCB We assume that the agents are sharing across a peer to peer network, i.e. every agent can share information with every other agent, but that every agent can communicate with only one other agent per round. In our algorithms, each agent, i, needs to maintain

(1) a *buffer* (an ordered set) $\mathcal{A}_t^i$ of covariance matrices and an *active* covariance matrix $\tilde{A}_t^i$,
(2) a *buffer* $\mathcal{B}_t^i$ of b-vectors and an *active* b-vector $\tilde{b}_t^i$,

Initially, we set, for all $i \in V$, $\tilde{A}_0^i = I$, $\tilde{b}_0^i = 0$. These *active* objects are used by the algorithm as sufficient statistics from which to calculate confidence balls, and summarise only information gathered before or during time $\tau(t)$, where τ is an arbitrary monotonically increasing function satisfying $\tau(t) < t$. The buffers are initially set to $\mathcal{A}_0^i = \emptyset$, and $\mathcal{B}_0^i = \emptyset$. For each $t > 1$, each agent, i, shares and updates its buffers as follows:

(1) a random permutation, σ, of the numbers $1, \ldots, |V|$ is chosen uniformly at random in a decentralised manner among the agents,[1]
(2) the buffers of i are then updated by averaging its buffers with those of $\sigma(i)$, and then extending them using their current observations[2]

$$\mathcal{A}_{t+1}^i = \left(\left(\tfrac{1}{2}(\mathcal{A}_t^i + \mathcal{A}_t^{\sigma(i)})\right) \circ \left(x_{t+1}^i \left(x_{t+1}^i\right)^\intercal\right)\right),$$
$$\mathcal{B}_{t+1}^i = \left(\left(\tfrac{1}{2}(\mathcal{B}_t^i + \mathcal{B}_t^{\sigma(i)})\right) \circ \left(r_{t+1}^i x_{t+1}^i\right)\right),$$

$\tilde{A}_{t+1}^i = \tilde{A}_t^i + \tilde{A}_t^{\sigma(i)}$, and $\tilde{b}_{t+1}^i = \tilde{b}_t^i + \tilde{b}_t^{\sigma(i)}$.
(3) if the length $|\mathcal{A}_{t+1}^i|$ exceeds $t - \tau(t)$, the first element of $\mathcal{A}_{t+1}^i$ is added to $\tilde{A}_{t+1}^i$ and deleted from $\mathcal{A}_{t+1}^i$. $\mathcal{B}_{t+1}^i$ and $\tilde{b}_{t+1}^i$ are treated similarly.

In this way, each buffer remains of size at most $t-\tau(t)$, and contains only information gathered after time $\tau(t)$. The result is that, after t rounds of sharing, the current covariance matrices and b-vectors used by the algorithm to make decisions have the form:

$$\tilde{A}_t^i := I + \sum_{t'=1}^{\tau(t)} \sum_{i'=1}^{|V|} w_{i,t}^{i',t'} x_{t'}^{i'} x_{t'}^{i'\intercal},$$
$$\text{and } \tilde{b}_t^i := \sum_{t'=1}^{\tau(t)} \sum_{i'=1}^{|V|} w_{i,t}^{i',t'} r_{t'}^{i'} x_{t'}^{i'}.$$

where the weights $w_{i,t}^{i',t'}$ are random variables which are unknown to the algorithm. Importantly for our analysis, as a result of the overlay protocol's uniformly random choice of σ, they are identically distributed ($i.d.$) for each fixed pair (t, t'), and $\sum_{i' \in V} w_{i,t}^{i',t'} = |V|$. If information sharing was perfect at each time step, then the current covariance matrix could be computed using all the information gathered by all the agents, and would be:

$$A_t := I + \sum_{i'=1}^{|V|} \sum_{t'=1}^{t} x_{t'}^{i'} \left(x_{t'}^{i'}\right)^\intercal. \qquad (1)$$

DCB algorithm The OFUL algorithm (Abbasi-Yadkori et al., 2011) is an improvement of the confidence ball algorithm from (Dani et al., 2008), which assumes that the confidence balls C_t can be characterised by A_t and b_t. In the DCB algorithm, each agent $i \in V$ maintains a confidence ball C_t^i for the unknown parameter θ as in the OFUL algorithm, but calculated from $\tilde{A}_t^i$ and $\tilde{b}_t^i$. It then chooses its action, x_t^i, to satisfy $(x_t^i, \theta_t^i) = \arg\max_{(x,\theta) \in \mathcal{D}_t^i \times C_t^i} x^\intercal \theta$, and receives a reward r_t^i. Finally, it shares its information buffer according to the sharing protocol above. Pseudocode for DCB is given in Appendix A.1, and in Algorithm 1.

2.1. Results for DCB

Theorem 1. *Let* $\tau(\cdot) : t \to 4\log(|V|^{\frac{3}{2}} t)$. *Then, with probability* $1 - \delta$, *the regret of DCB is bounded by*

$$\mathcal{R}_t \le (N(\delta)|V| + \nu(|V|, d, t)) \, \|\theta\|_2$$
$$+ 4e^2 \left(\beta(t) + 4R\right) \sqrt{|V| t \ln\left((1 + |V|t/d)^d\right)},$$

where $\nu(|V|, d, t) := (d+1)d^2(4|V|\ln(|V|^{\frac{3}{2}} t))^3$, $N(\delta) := \sqrt{3}/((1 - 2^{-\frac{1}{4}})\sqrt{\delta})$, *and*

$$\beta(t) := R\sqrt{\ln\left(\frac{(1 + |V|t/d)^d}{\delta}\right)} + \|\theta\|_2. \qquad (2)$$

The term $\nu(t, |V|, d)$ describes the loss compared to the centralised algorithm due to the delay in using information, while $N(\delta)|V|$ describes the loss due to the incomplete mixing of the data across the network.

If the agents implement CB independently and do not share any information, which we call CB-*NoSharing*, then it follows from the results in (Abbasi-Yadkori et al., 2011), the equivalent regret bound would be

$$\mathcal{R}_t \le |V| \beta(t) \sqrt{t \ln\left((1 + t/d)^d\right)} \qquad (3)$$

Comparing Theorem 1 with (3) tells us that, after an initial "burn in" period, the gain in regret performance of DCB over CB-*NoSharing* is of order almost $\sqrt{|V|}$.

[1] This can be achieved in a variety of ways.

[2] The $\circ$ symbol denotes the concatenation operation on two ordered sets: if $x = (a, b, c)$ and $y = (d, e, f)$, then $x \circ y = (a, b, c, d, e, f)$, and $y \circ x = (d, e, f, a, b, c)$.

Corollary 2. *We can recover a bound in expectation from Theorem 1, by using the value $\delta = 1/\sqrt{|V|t}$:*

$$\mathbb{E}[\mathcal{R}_t] \leq O(t^{\frac{1}{4}}) + \sqrt{|V|t}\|\theta\|_2 + 4e^2\left(R\sqrt{\ln\left((1+|V|t/d)^d\sqrt{|V|t}\right)} + \|\theta\|_2 + 4R\right) \times \sqrt{|V|t\ln\left((1+|V|t/d)^d\right)}.$$

This shows that DCB exhibits asymptotically optimal regret performance, up to log factors, in comparison with any algorithm that can share its information perfectly between agents at each round.

Communication Complexity

If the agents communicate their information to each other at each round without a central server, then every agent would need to communicate their chosen action and reward to every other agent at each round, giving a communication cost of order $d|V|^2$ per-round. We call such an algorithm CB-*InstSharing*. Under the gossip protocol we propose each agent requires at most $O(log_2(|V|t)d^2|V|)$ bits to be communicated per round. Therefore, a significant communication cost reduction is gained when $log(|V|t)d \ll |V|$.

Using an epoch-based approach, as in (Szörényi et al., 2013), the per-round communication cost of the gossip protocol becomes $O(d^2|V|)$. This improves efficiency over any horizon, requiring only that $d \ll |V|$, and the proofs of the regret performance are simple modifications of those for DCB. However, in comparison with growing buffers this is only an issue after $O(\exp(|V|))$ number of rounds, and typically $|V|$ is large.

While the DCB has a clear communication advantage over CB-*InstSharing*, there are other potential approaches to this problem. For example, instead of randomised neighbour sharing one can use a deterministic protocol such as *Round-Robin* (RR), which can have the same low communication costs as DCB. However, the regret bound for RR suffers from a naturally larger delay in the network than DCB. Moreover, attempting to track potential doubling of data points when using a gossip protocol, instead of employing a delay, leads back to a communication cost of order $|V|^2$ per round. More detail is included in Appendix A.2.

Proof of Theorem 1

In the analysis we show that the bias introduced by imperfect information sharing is mitigated by delaying the inclusion of the data in the estimation of the parameter θ. The proof builds on the analysis in (Abbasi-Yadkori et al., 2011). The emphasis here is to show how to handle the extra difficulty stemming from imperfect information sharing, which results in the influence of the various rewards at the various peers being unbalanced and appearing with a random delay. Proofs of the Lemmas 3 and 4, and of Proposition 1 are crucial, but technical, and are deferred to Appendix A.3.

Step 1: Define modified confidence ellipsoids. First we need a version of the confidence ellipsoid theorem given in (Abbasi-Yadkori et al., 2011) that incorporates the bias introduced by the random weights:

Proposition 1. *Let $\delta > 0$, $\tilde{\theta}_t^i := (\tilde{A}_t^i)^{-1}\tilde{b}_t^i$, $W(\tau) := \max\{w_{i,t}^{i',t'} : t, t' \leq \tau,\ i, i' \in V\}$, and let*

$$C_t^i := \left\{x \in \mathbb{R}^d : \|\tilde{\theta}_t^i - x\|_{\tilde{A}_t^i} \leq \|\theta\|_2 + W(\tau(t))R\sqrt{2\log\left(\det(\tilde{A}_t^i)^{\frac{1}{2}}/\delta\right)}\right\}. \tag{4}$$

Then with probability $1-\delta$, $\theta \in C_t^i$.

In the rest of the proof we assume that $\theta \in C_t^i$.

Step 2: Instantaneous regret decomposition. Denote by $(x_t^i, \theta_t^i) = \arg\max_{x \in D_t^i, y \in C_t^i} x^\intercal y$. Then we can decompose the instantaneous regret, following a classic argument (see the proof of Theorem 3 in (Abbasi-Yadkori et al., 2011)):

$$\begin{aligned}\rho_t^i &= \left(x_t^{i,*}\right)^\intercal\theta - (x_t^i)^\intercal\theta \leq \left(x_t^i\right)^\intercal\theta_t^i - (x_t^i)^\intercal\theta \\ &= \left(x_t^i\right)^\intercal\left[\left(\theta_t^i - \tilde{\theta}_t^i\right) + \left(\tilde{\theta}_t^i - \theta\right)\right] \\ &\leq \|x_t^i\|_{\left(\tilde{A}_t^i\right)^{-1}}\left[\left\|\theta_t^i - \tilde{\theta}_t^i\right\|_{\tilde{A}_t^i} + \left\|\tilde{\theta}_t^i - \theta\right\|_{\tilde{A}_t^i}\right]\end{aligned} \tag{5}$$

Step 3: Control the bias. The norm differences inside the square brackets of the regret decomposition are bounded through (4) in terms of the matrices $\tilde{A}_t^i$. We would like, instead, to have the regret decomposition in terms of the matrix A_t (which is defined in (1)). To this end, we give some lemmas showing that using the matrices $\tilde{A}_t^i$ is almost the same as using A_t. These lemmas involve elementary matrix analysis, but are crucial for understanding the impact of imperfect information sharing on the final regret bounds.

Step 3a: Control the bias coming from the weight imbalance.

Lemma 3 (Bound on the influence of general weights)**.** *For all $i \in V$ and $t > 0$,*

$$\|x_t^i\|^2_{\left(\tilde{A}_t^i\right)^{-1}} \leq e^{\sum_{t'=1}^{\tau(t)}\sum_{i'=1}^{|V|}\left|w_{i,t}^{i',t'}-1\right|}\|x_t^i\|^2_{\left(A_{\tau(t)}\right)^{-1}},$$

$$\textit{and } \det\left(\tilde{A}_t^i\right) \leq e^{\sum_{t'=1}^{\tau(t)}\sum_{i'=1}^{|V|}\left|w_{i,t}^{i',t'}-1\right|}\det\left(A_{\tau(t)}\right).$$

Using Lemma 4 in (Szörényi et al., 2013), by exploiting the random weights are identically distributed ($i.d.$) for each fixed pair (t,t'), and $\sum_{i' \in V} w_{i,t}^{i',t'} = |V|$ under our gossip protocol, we can control the random exponential constant in Lemma 3, and the upper bound $W(T)$ using the Chernoff-Hoeffding bound:

Lemma 4 (Bound on the influence of weights under our sharing protocol)**.** *Fix some constants $0 < \delta_{t'} < 1$. Then with probability $1 - \sum_{t'=1}^{\tau(t)} \delta_{t'}$*

$$\sum_{i'=1}^{|V|} \sum_{t'=1}^{\tau(t)} \left| w_{i,t}^{i',t'} - 1 \right| \leq |V|^{\frac{3}{2}} \sum_{t'=1}^{\tau(t)} \left(2^{(t-t')} \delta_{t'} \right)^{-\frac{1}{2}},$$

$$\textit{and } W(T) \leq 1 + \max_{1 \leq t' \leq \tau(t)} \left\{ |V|^{\frac{3}{2}} \left(2^{(t-t')} \delta_{t'} \right)^{-\frac{1}{2}} \right\}.$$

In particular, for any $\delta \in (0,1)$, choosing $\delta_{t'} = \delta 2^{\frac{t'-t}{2}}$, with probability $1 - \delta/(|V|^3 t^2 (1 - 2^{-1/2}))$ we have

$$\sum_{i'=1}^{|V|} \sum_{t'=1}^{\tau(t)} \left| w_{i,t}^{i',t'} - 1 \right| \leq \frac{1}{(1 - 2^{-\frac{1}{4}}) t \sqrt{\delta}},$$

$$\text{and } W(\tau(t)) \leq 1 + \frac{|V|^{\frac{3}{2}}}{t\sqrt{\delta}}. \quad (6)$$

Thus Lemma 3 and 4 give us control over the bias introduced by the imperfect information sharing. Combining them with Equations (4) and (5) we find that with probability $1 - \delta/(|V|^3 t^2 (1 - 2^{-1/2}))$:

$$\rho_t^i \leq 2e^{C(t)} \|x_t^i\|_{\left(A_{\tau(t)}^i\right)^{-1}} (1 + C(t)) \quad (7)$$
$$\times \left[R\sqrt{2 \log \left(e^{C(t)} \det \left(A_{\tau(t)} \right)^{\frac{1}{2}} \delta^{-1} \right)} + \|\theta\| \right]$$

where $C(t) := 1/(1 - 2^{-1/4}) t\sqrt{\delta}$

Step 3b: Control the bias coming from the delay. Next, we need to control the bias introduced from leaving out the last $4\log(|V|^{3/2} t)$ time steps from the confidence ball estimation calculation:

Proposition 2. *There can be at most*

$$\nu(k) := (4|V| \log(|V|^{3/2} k))^3 (d+1) d (tr(A_0) + 1) \quad (8)$$

pairs $(i,k) \in 1, \ldots, |V| \times \{1, \ldots, t\}$ for which one of

$$\|x_k^i\|^2_{A_{\tau(k)}^{-1}} \geq e \|x_k^i\|^2_{\left(A_{k-1} + \sum_{j=1}^{i-1} x_k^j (x_k^j)^\intercal\right)^{-1}},$$

$$\textit{or } \det\left(A_{\tau(k)}\right) \geq e \det \left(A_{k-1} + \sum_{j=1}^{i-1} x_k^j (x_k^j)^\intercal \right) \textit{ holds.}$$

Step 4: Choose constants and sum the simple regret. Defining a constant

$$N(\delta) := \frac{1}{(1 - 2^{-\frac{1}{4}})\sqrt{\delta}},$$

we have, for all $k \geq N(\delta)$, $C(k) \leq 1$, and so, by (7) with probability $1 - (|V|k)^{-2}\delta/(1 - 2^{-1/2})$

$$\rho_k^i \leq 2e \|x_k^i\|_{A_{\tau(k)}^{-1}} \quad (9)$$
$$\times \left[2R \sqrt{2 \log \left(\frac{e \det \left(A_{\tau(k)} \right)^{\frac{1}{2}}}{\delta} \right)} + \|\theta\|_2 \right].$$

Now, first applying Cauchy-Schwarz, then step 3b from above together with (9), and finally Lemma 11 from (Abbasi-Yadkori et al., 2011) yields that, with probability $1 - \left(1 + \sum_{t=1}^{\infty} (|V|t)^{-2}/(1 - 2^{-1/2})\right) \delta \geq 1 - 3\delta$,

$$\begin{aligned}\mathcal{R}_t \leq & N(\delta)|V| \|\theta\|_2 + \left[|V|t \sum_{t'=N(\delta)}^{t} \sum_{i=1}^{|V|} \left(\rho_{t'}^i\right)^2 \right]^{\frac{1}{2}} \\ & \leq \left(N(\delta)|V| + \nu(|V|, d, t)\right) \|\theta\|_2 \\ & \quad + 4e^2 \left(\beta(t) + 2R\right) \left[|V|t \sum_{t'=1}^{t} \sum_{i=1}^{M} \|x_t^i\|^2_{(A_t)^{-1}} \right]^{\frac{1}{2}} \\ & \leq \left(N(\delta)|V| + \nu(|V|, d, t)\right) \|\theta\|_2 \\ & \quad + 4e^2 \left(\beta(t) + 2R\right) \sqrt{|V|t \left(2 \log\left(\det\left(A_t\right)\right)\right)},\end{aligned}$$

where $\beta(\cdot)$ is as defined in (2). Replacing δ with $\delta/3$ finishes the proof.

PROOF OF PROPOSITION 2

This proof forms the major innovation in the proof of Theorem 1. Let $(y_k)_{k \geq 1}$ be any sequence of vectors such that $\|y_k\|_2 \leq 1$ for all k, and let $B_n := B_0 + \sum_{k=1}^n y_k y_k^\intercal$, where B_0 is some positive definite matrix.

Lemma 5. *For all $t > 0$, and for any $c \in (0,1)$, we have*

$$\left| \left\{ k \in \{1, 2, \ldots\} : \|y_k\|^2_{B_{k-1}^{-1}} > c \right\} \right|$$
$$\leq (d + c) d (tr(B_0^{-1}) - c)/c^2,$$

Proof. We begin by showing that, for any $c \in (0,1)$

$$\|y_k\|^2_{B_{k-1}^{-1}} > c \quad (10)$$

can be true for only $2dc^{-3}$ different k.

Indeed, let us suppose that (10) is true for some k. Let $(e_i^{(k-1)})_{1 \leq i \leq d}$ be the orthonormal eigenbasis for B_{k-1}, and, therefore, also for B_{k-1}^{-1}, and write $y_k = \sum_{i=1}^d \alpha_i e_i$. Let, also, $(\lambda_i^{(k-1)})$ be the eigenvalues for B_{k-1}. Then,

$$c < y_k^\intercal B_{k-1}^{-1} y_k = \sum_{i=1}^d \frac{\alpha_i^2}{\lambda_i^{(k-1)}} \leq tr(B_{k-1}^{-1}),$$

$$\implies \exists j \in \{1, \ldots, d\} : \frac{\alpha_j^2}{\lambda_j^{(k-1)}}, \frac{1}{\lambda_j^{(k-1)}} > \frac{c}{d},$$

where we have used that $\alpha_i^2 < 1$ for all i, since $\|y_k\|_2 < 1$. Now,

$$\begin{aligned} & tr(B_{k-1}^{-1}) - tr(B_k^{-1}) \\ & \quad = tr(B_{k-1}^{-1}) - tr((B_{k-1} + y_k y_k^{\intercal})^{-1}) \\ & \quad > tr(B_{k-1}^{-1}) - tr((B_{k-1} + \alpha_j^2 e_j e_j^{\intercal})^{-1}) \\ & \quad = \tfrac{1}{\lambda_j^{(k-1)}} - \tfrac{1}{\lambda_j^{(k-1)} + \alpha_j^2} = \tfrac{\alpha_j^2}{\lambda_j^{(k-1)}(\lambda_j^{(k-1)} + \alpha_j^2)} \\ & \quad > \left(d^2 c^{-2} + d c^{-1}\right)^{-1} > \tfrac{c^2}{d(d+c)} \end{aligned}$$

So we have shown that (10) implies that

$$tr(B_{k-1}^{-1}) > c \text{ and } tr(B_{k-1}^{-1}) - tr(B_k^{-1}) > \frac{c^2}{d(d+c)}.$$

Since $tr(B_0^{-1}) \geq tr(B_{k-1}^{-1}) \geq tr(B_k^{-1}) \geq 0$ for all k, it follows that (10) can be true for at most $(d+c)d(tr(B_0^{-1}) - c)c^{-2}$ different k. □

Now, using an argument similar the proof of Lemma 3, for all $k < t$

$$\begin{aligned} \|y_{k+1}\|_{B_{\tau(k)}^{-1}} &\leq e^{\sum_{s=\tau(k)+1}^{k} \|y_{s+1}\|_{B_s^{-1}}} \|y_{k+1}\|_{B_k^{-1}}, \\ \text{and } \det\left(B_{\tau(t)}\right) &\leq e^{\sum_{k=\tau(t)+1}^{t} \|y_k\|^2_{B_k^{-1}}} \det\left(B_t\right). \end{aligned}$$

Therefore,

$$\begin{aligned} \|y_{k+1}\|_{B_{\tau(k)}^{-1}} \geq c\|y_{k+1}\|_{B_k^{-1}} & \text{ or } \det(B_{\tau(k)}) \geq c \det(B_k) \\ \implies \sum_{s=\tau(k)}^{k-1} & \|y_{s+1}\|_{B_s^{-1}} \geq \ln(c) \end{aligned}$$

However, according to Lemma 5, there can be at most

$$\nu(t) := \left(d + \tfrac{\ln(c)}{\Delta(t)}\right) d \left(tr\left(B_0^{-1}\right) - \tfrac{\ln(c)}{\Delta(t)}\right) \left(\tfrac{\Delta(t)}{\ln(c)}\right)^2$$

times $s \in \{1, \dots, t\}$, such that $\|y_{s+1}\|_{B_s^{-1}} \geq \ln(c)/\Delta(t)$, where $\Delta(t) := \max_{1 \leq k \leq t}\{k - \tau(k)\}$. Hence $\sum_{s=\tau(j)+1}^{k} \|y_{s+1}\|_{B_s}^{-1} \geq \ln(c)$ is true for at most $\Delta(t)\nu(|V|, d, t)$ indices $k \in \{1, \dots, t\}$.

Finally, we finish by setting $(y_k)_{k\geq 1} = \circ_{t \geq 1}(x_t^i)_{i=1}^{|V|}$.

3. Clustering and the DCCB Algorithm

We now incorporate distributed clustering into the DCB algorithm. The analysis of DCB forms the backbone of the analysis of DCCB.

DCCB Pruning Protocol In order to run DCCB, each agent i must maintain some local information buffers in addition to those used for DCB. These are:

Algorithm 1 Distributed Clustering Confidence Ball

```
**Input:** Size of network $|V|, \tau : t \to t - 4\log_2 t, \alpha, \lambda$
**Initialization:** $\forall i \in V$, set $\tilde{A}_0^i = I_d$, $\tilde{b}_0^i = \mathbf{0}$, $\mathcal{A}_0^i = \mathcal{B}_0^i = \emptyset$, and $V_0^i = V$.
**for** $t = 0, \dots \infty$ **do**
  Draw a random permutation $\sigma$ of $\{1, \dots, V\}$ respecting the current local clusters
  **for** $i = 1, \dots, |V|$ **do**
    Receive action set $\mathcal{D}_t^i$ and construct the confidence ball $C_t^i$ using $\tilde{A}_t^i$ and $\tilde{b}_t^i$
    **Choose action and receive reward:**
    Find $(x_{t+1}^i, *) = \arg\max_{(x,\tilde{\theta}) \in \mathcal{D}_t^i \times C_t^i} x^{\intercal}\tilde{\theta}$, and get reward $r_{t+1}^i$ from context $x_{t+1}^i$.
    **Share and update information buffers:**
    **if** $\|\hat{\theta}_{local}^i - \hat{\theta}_{local}^j\| > c_{\lambda}^{thresh}(t)$
    Update local cluster: $V_{t+1}^i = V_t^i \setminus \{\sigma(i)\}$, $V_{t+1}^{\sigma(i)} = V_t^{\sigma(i)} \setminus \{i\}$, and reset according to (13)
    **elseif** $V_t^i = V_t^{\sigma(i)}$
    Set $\mathcal{A}_{t+1}^i = \left(\frac{1}{2}(\mathcal{A}_t^i + \mathcal{A}_t^{\sigma(i)})\right) \circ (x_{t+1}^i \left(x_{t+1}^i\right)^{\intercal})$
    and $\mathcal{B}_{t+1}^i = \left(\frac{1}{2}(\mathcal{B}_t^i + \mathcal{B}_t^{\sigma(i)})\right) \circ (r_{t+1}^i x_{t+1}^i)$
    **else** Update: Set $\mathcal{A}_{t+1}^i = \mathcal{A}_t^i \circ (x_{t+1}^i \left(x_{t+1}^i\right)^{\intercal})$ and $\mathcal{B}_{t+1}^i = \mathcal{B}_t^i \circ (r_{t+1}^i x_{t+1}^i)$
    **endif**
    Update local estimator: $A_{local,t+1}^i = A_{local,t}^i + x_{t+1}^i \left(x_{t+1}^i\right)^{\intercal}$, $b_{local,t+1}^i = b_{local,t}^i + r_{t+1}^i x_{t+1}^i$, and $\hat{\theta}_{local,t+1} = \left(A_{local,t+1}^i\right)^{-1} b_{local,t+1}^i$
    **if** $|\mathcal{A}_{t+1}^i| > t - \tau(t)$ set $\tilde{A}_{t+1}^i = \tilde{A}_t^i + \mathcal{A}_{t+1}^i(1)$, $\mathcal{A}_{t+1}^i = \mathcal{A}_{t+1}^i \setminus \mathcal{A}_{t+1}^i(1)$. Similarly for $\mathcal{B}_{t+1}^i$.
  **end for**
**end for**
```

(1) a local covariance matrix $A_{local}^i = A_{local,t}^i$, a local b-vector $b_{local}^i = b_{local,t}^i$,

(2) and a local neighbour set V_t^i.

The local covariance matrix and b-vector are updated as if the agent was applying the generic (single agent) confidence ball algorithm: $A_{local,0}^i = A_0$, $b_{local,0}^i = 0$,

$$\begin{aligned} A_{local,t}^i &= x_t^i (x_t^i)^{\intercal} + A_{local,t-1}^i, \\ \text{and } b_{local,t}^i &= r_t^i x_t^i + b_{local,t-1}^i. \end{aligned}$$

DCCB Algorithm Each agent's local neighbour set V_t^i is initially set to V. At each time step t, agent i contacts one other agent, j, at random from V_t^i, and both decide whether they do or do not belong to the same cluster. To do this they share local estimates, $\hat{\theta}^i_t = {A^i_{local,t}}^{-1} b^i_{local,t}$ and $\hat{\theta}^j_t = {A^j_{local,t}}^{-1} b^j_{local,t}$, of the unknown parameter of the bandit problem they are solving, and see if they are further apart than a threshold function $c = c^{thresh}_{\lambda}(t)$, so that if

$$\|\hat{\theta}^i_t - \hat{\theta}^j_t\|_2 \geq c^{thresh}_{\lambda}(t), \tag{11}$$

then $V^i_{t+1} = V^i_t \setminus \{j\}$ and $V^j_{t+1} = V^j_t \setminus \{i\}$. Here λ is a parameter of an extra assumption that is needed, as in (Gentile et al., 2014), about the process generating the context sets $\mathcal{D}^i_t$:

(A) Each context set $\mathcal{D}^i_t = \{x_k\}_k$ is finite and contains $i.i.d.$ random vectors such that for all, k, $\|x_k\| \leq 1$ and $\mathbb{E}(x_k x_k^{\intercal})$ is full rank, with minimal eigenvalue $\lambda > 0$.

We define $c^{thresh}_{\lambda}(t)$, as in (Gentile et al., 2014), by

$$c^{thresh}_{\lambda}(t) := \frac{R\sqrt{2d\log(t) + 2\log(2/\delta)} + 1}{\sqrt{1 + \max\{A_{\lambda}(t, \delta/(4d)), 0\}}} \tag{12}$$

where $A_{\lambda}(t,\delta) := \frac{\lambda t}{\delta} - 8\log\frac{t+3}{\delta} - 2\sqrt{t\log\frac{t+3}{\delta}}$.

The DCCB algorithm is pretty much the same as the DCB algorithm, except that it also applies the pruning protocol described. In particular, each agent, i, when sharing its information with another, j, has three possible actions:

(1) if (11) is not satisfied and $V^i_t = V^j_t$, then the agents share simply as in the DCB algorithm;

(2) if (11) is satisfied, then both agents remove each other from their neighbour sets and reset their buffers and active matrices so that

$$\mathcal{A}^i = (0, 0, \ldots, A^i_{local}), \mathcal{B}^i = (0, 0, \ldots, b^i_{local}),$$
$$\text{and } \tilde{A}^i = A^i_{local}, \tilde{b}^i = b^i_{local}, \tag{13}$$

and similarly for agent j.

(3) if (11) is not satisfied but $V^i_t \neq V^j_t$, then no sharing or pruning occurs.

It is proved in the theorem below, that under this sharing and pruning mechanism, in high probability after some finite time each agent i finds its true cluster, i.e. $V^i_t = U^k$. Moreover, since the algorithm resets to its local information each time a pruning occurs, once the true clusters have been identified, each cluster shares only information gathered within that cluster, thus avoiding introducing a bias by sharing information gathered from outside the cluster before the clustering has been identified. Full pseudo-code for the DCCB algorithm is given in Algorithm 1, and the differences with the DCB algorithm are highlighted in blue.

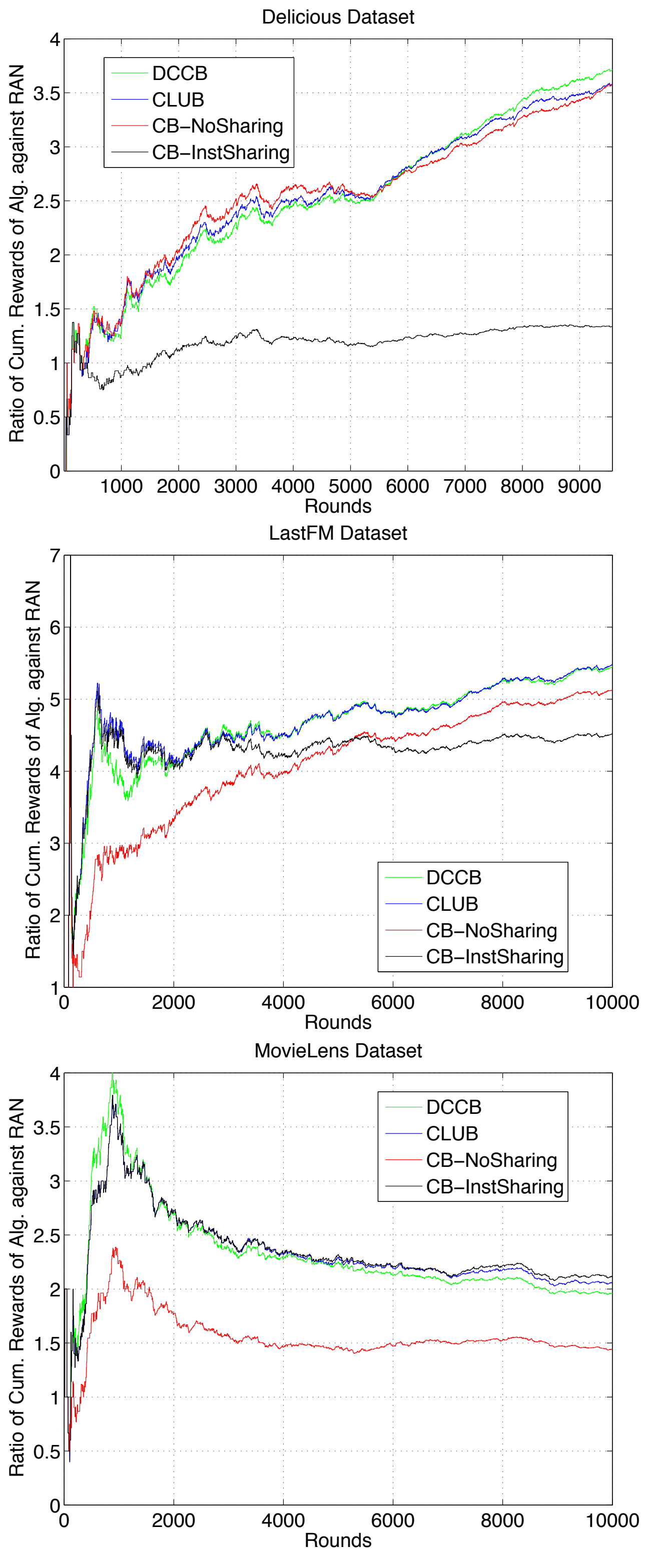

Figure 1. Here we plot the performance of DCCB in comparison to CLUB, CB-*NoSharing* and CB-*InstSharing*. The plots show the ratio of cumulative rewards achieved by the algorithms to the cumulative rewards achieved by the random algorithm.

3.1. Results for DCCB

Theorem 6. *Assume that (A) holds, and let γ denote the smallest distance between the bandit parameters θ^k. Then there exists a constant $C = C(\gamma, |V|, \lambda, \delta)$, such that with probability $1 - \delta$ the total cumulative regret of cluster k when the agents employ DCCB is bounded by*

$$\mathcal{R}_t \leq \Bigg[\max\left\{\sqrt{2}N(\delta), C + 4\log_2(|V|^{\frac{3}{2}}C)\right\} |U^k| + \nu(|U^k|, d, t) \Bigg] \|\theta\|_2 + 4e\left(\beta(t) + 3R\right)\sqrt{|U^k| t \ln\left((1 + |U^k| t/d)^d\right)},$$

where N and ν are as defined in Theorem 1, and $\beta(t) := R\sqrt{2\ln\left((1+|U^k|t/d)^d\right)} + \|\theta\|_2$.

The constant $C(\gamma, |V|, \lambda, \delta)$ is the time that you have to wait for the true clustering to have been identified,

The analysis follows the following scheme: When the true clusters have been correctly identified by all nodes, within each cluster the algorithm, and thus the analysis, reduces to the case of Section 2.1. We adapt results from (Gentile et al., 2014) to show how long it will be before the true clusters are identified, in high probability. The proof is deferred to Appendices A.4 and A.5.

4. Experiments and Discussion

Experiments We closely implemented the experimental setting and dataset construction principles used in (Li et al., 2016a;b), and for a detailed description of this we refer the reader to (Li et al., 2016a). We evaluated DCCB on three real-world datasets against its centralised counterpart CLUB, and against the benchmarks used therein, CB-*NoSharing*, and CB-*InstSharing*. The LastFM dataset comprises of 91 users, each of which appear at least 95 times. The Delicious dataset has 87 users, each of which appear at least 95 times. The MovieLens dataset contains 100 users, each of which appears at least 250 times. The performance was measured using the ratio of cumulative reward of each algorithm to that of the predictor which chooses a random action at each time step. This is plotted in in Figure 1. From the experimental results it is clear that DCCB performs comparably to CLUB in practice, and both outperform CB-*NoSharing*, and CB-*InstSharing*.

Relationship to existing literature There are several strands of research that are relevant and complimentary to this work. First, there is a large literature on single agent linear bandits, and other more, or less complicated bandit problem settings. There is already work distributed approaches to multi-agent, multi-armed bandits, not least (Szörényi et al., 2013) which examines ϵ-greedy strategies over a peer to peer network, and provided an initial inspiration for this current work. The paper (Kalathil et al., 2014) examines the extreme case when there is no communication channel across which the agents can communicate, and all communication must be performed through obesrvation of action choices alone. Another approach to the multi-armed bandit case, (Nayyar et al., 2015), directly incorporates the communication cost into the regret.

Second, there are several recent advances regarding the state-of-the-art methods for clustering of bandits. The work (Li et al., 2016a) is a faster variant of (Gentile et al., 2014) which adopt the strategy of boosted training stage. In (Li et al., 2016b) the authors not only cluster the users, but also cluster the items under collaborative filtering case with a sharp regret analysis.

Finally, the paper (Tekin & van der Schaar, 2013) treats a setting similar to ours in which agents attempt to solve contextual bandit problems in a distributed setting. They present two algorithms, one of which is a distributed version of the approach taken in (Slivkins, 2014), and show that they achieve at least as good asymptotic regret performance in the distributed approach as the centralised algorithm achieves. However, rather than sharing information across a limited communication channel, they allow each agent only to ask another agent to choose their action for them. This difference in our settings is reflected worse regret bounds, which are of order $O(T^{2/3})$ at best.

Discussion Our analysis is tailored to adapt proofs from (Abbasi-Yadkori et al., 2011) about generic confidence ball algorithms to a distributed setting. However many of the elements of these proofs, including Propositions 1 and 2 could be reused to provide similar asymptotic regret guarantees for the distributed versions of other bandit algorithms, e.g., the Thompson sampling algorithms, (Agrawal & Goyal, 2013; Kaufmann et al., 2012; Russo & Van Roy, 2014).

Both DCB and DCCB are synchronous algorithms. The work on distributed computation through gossip algorithms in (Boyd et al., 2006) could alleviate this issue. The current pruning algorithm for DCCB guarantees that techniques from (Szörényi et al., 2013) can be applied to our algorithms. However the results in (Boyd et al., 2006) are more powerful, and could be used even when the agents only identify a sub-network of the true clustering.

Furthermore, there are other existing interesting algorithms for performing clustering of bandits for recommender systems, such as COFIBA in (Li et al., 2016b). It would be interesting to understand how general the techniques applied here to CLUB are.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. We would also like to thank Gergley Neu for very useful discussions. The first author thanks the support from EPSRC Autonomous Intelligent Systems project EP/I011587. The third author thanks the support from MIUR, QCRI-HBKU, and Amazon Research Grant. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 306638.

References

Abbasi-Yadkori, Yasin, Pál, Dávid, and Szepesvári, Csaba. Improved algorithms for linear stochastic bandits. In *NIPS*, pp. 2312–2320, 2011.

Agrawal, Shipra and Goyal, Navin. Thompson sampling for contextual bandits with linear payoffs. In *ICML*, 2013.

Boyd, Stephen, Ghosh, Arpita, Prabhakar, Balaji, and Shah, Devavrat. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI): 2508–2530, 2006.

Dani, Varsha, Hayes, Thomas P, and Kakade, Sham M. Stochastic linear optimization under bandit feedback. In *COLT*, pp. 355–366, 2008.

Gentile, Claudio, Li, Shuai, and Zappella, Giovanni. Online clustering of bandits. In *ICML*, 2014.

Hao, Fei, Li, Shuai, Min, Geyong, Kim, Hee-Cheol, Yau, Stephen S, and Yang, Laurence T. An efficient approach to generating location-sensitive recommendations in ad-hoc social network environments. *IEEE Transactions on Services Computing*, 2015.

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.

Kalathil, Dileep, Nayyar, Naumaan, and Jain, Rahul. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

Kaufmann, Emilie, Korda, Nathaniel, and Munos, Rémi. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pp. 199–213. Springer, 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.

Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM, 2010.

Li, Shuai, Hao, Fei, Li, Mei, and Kim, Hee-Cheol. Medicine rating prediction and recommendation in mobile social networks. In *Proceedings of the International Conference on Grid and Pervasive Computing*, 2013.

Li, Shuai, Gentile, Claudio, and Karatzoglou, Alexandros. Graph clustering bandits for recommendation. *CoRR:1605.00596*, 2016a.

Li, Shuai, Karatzoglou, Alexandros, and Gentile, Claudio. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR Conference on Information Retrieval (SIGIR'16)*, 2016b.

Nayyar, Naumaan, Kalathil, Dileep, and Jain, Rahul. On regret-optimal learning in decentralized multi-player multi-armed bandits. *CoRR:1505.00553*, 2015.

Russo, Daniel and Van Roy, Benjamin. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Slivkins, Aleksandrs. Contextual bandits with similarity information. *JMLR*, 2014.

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

Tekin, Cem and van der Schaar, Mihaela. Distributed online learning via cooperative contextual bandits. *IEEE Trans. Signal Processing*, 2013.

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.