MULTI-ARMED BANDITS IN MULTI-AGENT NETWORKS

Shahin Shahrampour**, Alexander Rakhlin*†*, and Ali Jadbabaie*‡

*Department of Electrical Engineering, Harvard University
†Department of Statistics at the Wharton School, University of Pennsylvania
‡Institute for Data, Systems, and Society, Massachusetts Institute of Technology

ABSTRACT

This paper addresses the multi-armed bandit problem in a multi-player framework. Players explore a finite set of arms with stochastic rewards, and the reward distribution of each arm is player-dependent. The goal is to find the best global arm, i.e., the one with the largest expected reward when averaged out among players. To achieve this goal, we develop a distributed variant of the well-known `UCB1` algorithm. Confined to a network structure, players exchange information locally to estimate the global rewards, while using a confidence bound relying on the network characteristics. Then, at each round, each player votes for an arm, and the majority vote is played as the network action. The whole network gains the reward of the network action, hoping to maximize the global welfare. The performance of the algorithm is measured via the notion of network regret. We prove that the regret scales logarithmically with respect to time horizon and inversely in the spectral gap of the network. Our algorithm is optimal in the sense that in a complete network it scales down the regret of its single-player counterpart by the network size. We demonstrate numerical experiments to verify our theoretical results.

Index Terms— Sequential decision-making, multi-armed bandits, multi-agent networks, distributed learning.

1. INTRODUCTION

The multi-armed bandit (MAB) problem has been extensively studied in the literature [1–6]. In its classical setting, the problem is defined by a set of *arms* or *actions*, and it captures the exploration-exploitation dilemma for a *learner*. At each time step, the learner chooses an arm and receives its corresponding *payoff* or *reward*. The term *bandit* indicates that only the reward of the chosen arm is revealed to the learner, while the rewards of other arms remain undisclosed at that particular time. The learner then hopes to maximize the total payoff obtained from sequentially selecting the arms. Equivalently, the learner aims to minimize the *regret* by competing with the best single arm in hindsight. The reward model for arms could be stochastic or non-stochastic, and optimal algorithms for both cases are proposed in the literature [6]. While early studies on MAB dates back to nine decades ago, the problem has received considerable attention due to its modern applications. MAB could be an instance of sequential decision-making for ad placement, website optimization, packet routing, cognitive compressive sensing, and etc. [6–9].

In this paper, we depart from the classical setting and address the stochastic MAB in a multi-player network. We consider a scenario where a group of *players* or *agents* collaborate to achieve a team task. In this framework, each arm may possess different reward distributions for distinct players. The goal is to reach consensus on the arm which best fits the network. We consider this arm to be the arm with the largest expected reward when averaged out among players. As a motivating example, consider a number of brands (arms) selling a product. We have several users (players) in a social network who need to decide on one specific brand for the whole network. The users rate these brands differently and must agree upon using one brand. The decision is made by the majority vote to maximize the global welfare.

Therefore, in our setup, each arm has a true *global* payoff that can be written in terms of an average of *individual* rewards. Agents are not able to identify the best global arm since they do not receive any informative signal about the global reward. Therefore, they need to benefit from side observations gained from local communication. The model has a flavor of distributed detection algorithms where the parameter of interest is not fully observable to an individual agent [10, 11]. However, there is an additional restriction in this work, as we consider a bandit setup where the player only receives the signal of the *chosen* action.

To find the best global arm, players wish to maximize a global objective. As standard in online algorithms, we translate this objective to minimizing a network *regret*. We then propose an algorithm dubbed Distributed Upper Estimated Reward (`d-UER`) to minimize the network regret. The algorithm estimates the global reward of the arms in a purely *distributed* fashion. To concentrate around the true value of the global rewards with high probability, the algorithm exploits a *confidence bound* that relies on the network topology. Then, players iteratively vote for their favorite arm, and each time the majority vote is played as the network action. The whole network scores the reward of the network action, hoping to maximize the global welfare. This algorithm can be viewed as a distributed version of the well-known `UCB1` algorithm.

We prove that the regret of our algorithm scales logarithmically with respect to time horizon. It also scales inversely in the gap between the arms. The inverse dependence to gap is quite natural since similar arms to the best arm are hard to distinguish. Furthermore, the regret relies on the *spectral gap* of the network as in the exploration phase information needs to be propagated throughout the network. Our algorithm is optimal in the sense that in a complete network the regret is scaled down by the network size comparing to its single-player analog. This is due to variance reduction when we distribute samples between agents throughout the network. We finally provide numerical experiments to verify the impact of network size and spectral gap in practice.

Related Work: Several variants of decentralized MAB have been studied in the literature. In [12–14], decentralized MAB has been

This work was supported by ONR BRC under Grant N00014-12-1-0997 and NSF under Grant CDS&E-MSS 1521529.

formulated for applications in cognitive radio networks and multichannel communication systems. Unlike our setup, in these works simultaneous selection of one arm by a few players is not recommended and reduces the reward in some sense. Some other works focused on decentralized MAB with application to advertising systems. In [15], the interaction between users in a social network provides information for an external, centralized decision-maker. In [16] only a single *major* agent in the network has access to its reward sequence, while other agents are aware of the sampling pattern of the major agent. The works of [17, 18] are particularly relevant to our work; however, the arms in those settings are player independent, whereas in our setup they are player dependent. Our work is also related to distributed detection and learning under *full information* setting [11, 19–26]. In these models, the world is governed by a fixed true *state* (arm), aimed to be recovered collaboratively. However, agents receive information about all states per round, whereas our *bandit* setup entails only one-state feedback per round.

Notation

$[n]$	The set $\{1, 2, ..., n\}$ for any integer n
$\mathbf{1}\{\cdot\}$	The indicator function
$\mathbb{1}$	The vector of all ones
$\mathbb{E}[\cdot]$	The expectation operator
$x^\top$	Transpose of the vector x
$x(k)$	The k-th element of vector x
$\sigma_i(W)$	The i-th largest singular value of matrix W

2. PROBLEM FORMULATION

We have a multi-agent network with N *players* or *agents* sequentially selecting *arms* or *actions*. The number of arms is a finite number K, a common knowledge in the network. Pulling arm $k \in [K]$ at time $t \in [T]$ rewards the player $i \in [N]$ with a random variable $X_{i,t}(k) \in [0, 1]$. The rewards are independent and identically distributed over time for an individual player. We further assume that they are independent across players and arms. Hence, the expected value $\mu_i = \mathbb{E}[X_{i,t}] \in \mathbb{R}^K$ is fixed over time though the characteristics of each arm is different among players. That is, for arm k and players i, j, the values $\mu_i(k)$ and $\mu_j(k)$ are not equal in general. The *true* reward of arm $k \in [K]$ is the network average as follows

$$\mu(k) := \frac{1}{N}\sum_{i=1}^{N}\mu_i(k) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[X_{i,t}(k)].$$

Agents aim to maximize the *global* welfare in the sense of finding the arm k^* such that $\mu(k) \leq \mu(k^*)$ for all $k \in [K]$. However, no individual player can make a correct inference by simply collecting individual signals $\{X_{i,t}\}_{t=1}^{T}$ as these only approximate μ_i over time. Therefore, players must collaborate with each other to estimate the true rewards in a distributed fashion. Based on a distributed protocol discussed in the next section, each agent votes for an arm at each round. We represent by the random variable $I_{i,t}$, the arm that is chosen by player i at time t. The network action I_t at time t is then the majority vote, i.e., the random variable I_t is defined as

$$I_t := \text{the most repeated element of the set } \{I_{i,t}\}_{i=1}^{N}.$$

Consistently with the bandit setting, after this selection is made, player i only observes the corresponding payoff $X_{i,t}(I_t)$ at period t.

Common to the MAB framework, let us reformulate the problem in terms of *regret*. Without loss of generality, we assume the following order for the true rewards,

$$\mu(1) \geq \mu(2) \geq \cdots \geq \mu(K),$$

for the rest of the paper. For any pair $k \leq m$, we define the *gap* as

$$\Delta_{k,m} := \mu(k) - \mu(m) = \frac{1}{N}\sum_{i=1}^{N}\mu_i(k) - \mu_i(m),$$

to capture the suboptimality of arm m comparing to arm k. We use $n_t(k)$ to denote the number of times that arm k has been chosen by the network as the majority vote until time t. It is easy to observe that maximizing the global welfare is equivalent to minimizing the network *regret* in the following sense,

$$\mathbf{R}_T := T\mu(1) - \sum_{t=1}^{T}\mathbb{E}\left[\mu(I_t)\right] = \sum_{k=2}^{K}\Delta_{1,k}\mathbb{E}[n_T(k)], \qquad (1)$$

where the expectation is taken over the randomness in the choice of arms.

In Section 3, we propose an online, distributed algorithm to solve (1). We prove that the algorithm incurs a strongly sub-linear regret with respect to time. Furthermore, we show that the regret depends on the network characteristics since players communicate with each other to understand the true value of the rewards. In other words, the information dissemination over the network requires a time which appears as a network penalty in the regret. This is in contrast with the classical (one-player) MAB, where the player only investigates the arms in the exploration phase. In our setup (multi-player MAB), an extra information-exchange time is also required in the exploration period to estimate the true rewards.

Network Structure: One individual player is not able to track the best arm in isolation as the signals $\{X_{i,t}\}_{t=1}^{T}$ are not informative enough for approximation of μ. Therefore, agents communicate with each other iteratively to approximate the true rewards. We use the symmetric and doubly stochastic matrix W to capture the interaction between agents. This matrix has positive diagonals, and a positive $[W]_{ij} > 0$ means that player i assigns a weight $[W]_{ij} = [W]_{ji}$ to observations of player $j \neq i$. When $[W]_{ij} = 0$, agents i and j never communicate with each other directly. Therefore, we have

$$\sum_{j\in\mathcal{N}_i}[W]_{ij} = \sum_{j=1}^{N}[W]_{ij} = \sum_{i=1}^{N}[W]_{ij} = \sum_{i\in\mathcal{N}_j}[W]_{ij} = 1,$$

where $\mathcal{N}_i := \{j \in [N] : [W]_{ij} > 0\}$ is the *local* neighborhood of agent i. We assume that the underlying network is *connected*, i.e., there exists a *path* from any player $i \in [N]$ to any player $j \in [N]$. This assumption guarantees the information flow over the network.

3. DISTRIBUTED UPPER ESTIMATED REWARD

We now describe the d-UER algorithm to minimize regret in (1). The algorithm can be cast as a cooperative version of the well-known UCB1 [3]. As we discussed in Section 2, the feedback setup does not allow an individual player to solely identify the best arm. Therefore, players need to cooperate with each other to collectively explore the arms.

```
Algorithm 1 Distributed Upper Estimated Reward
Input : Number of agents $N$ and arms $K$. Parameter $d > 0$.
Initialization : Each action is played once, and the reward of action $k \in [K]$ for player $i \in [N]$ is stored in $\psi_{i,1}(k)$. For each $i \in [N]$ and $k \in [K]$, let $\phi_{i,0}(k) = 0$ and $n_{i,0}(k) = 1$, respectively.
for $t = 1$ to $T$ do
  for $i = 1$ to $N$ do
    $\phi_{i,t} = \sum_{j=1}^{N} [W]_{ij} \phi_{j,t-1} + \psi_{i,t}$.        % estimation of true rewards
    $C_t(k) = \sqrt{2 \log t \left( \frac{1}{N n_{t-1}(k)} + \frac{2d}{n_{t-1}^2(k)} \right)}$.        % upper confidence bound for each action
    $I_{i,t} = \text{argmax}_{k \in [K]} \left\{ \frac{\phi_{i,t}(k)}{n_{t-1}(k)} + C_t(k) \right\}$.        % agent's action
  end for
  Let $I_t$ be the most frequent element of the set $\{I_{i,t}\}_{i=1}^{N}$.        % network's action or majority vote
  For any $k \in [K]$, update the counter as $n_t(k) = n_{t-1}(k) + \mathbf{1}\{k = I_t\}$.
  The network scores $\mu(I_t)$, and player $i \in [N]$ observes $X_{i,t}(I_t)$.        % private observation
  Let $\psi_{i,t+1}(k) = X_{i,t}(k) \mathbf{1}\{k = I_t\}$ for any $k \in [K]$.
end for
```

The d-UER algorithm is summarized in the table above. It provides a completely decentralized method to estimate the true rewards. Players accumulate local observations, and they take into account an upper confidence bound to make individual decisions. The term $\phi_{i,t}$ in the algorithm (normalized by the number of times each arm has been played) is an estimator of the true rewards. The confidence bound C_t allows agents to concentrate around the true value of the rewards with high probability. Then, agent i makes the individual decision $I_{i,t}$ at time t, the majority vote I_t among agents is chosen as the network action, and agents observe the corresponding payoff of the action chosen by the majority vote.

Note that since the setup is multi-player, d-UER exploits a confidence bound relying on the network structure through parameter d. We will see that this parameter must be tuned as an upper bound on a quantity that depends on network size and spectral gap.

The following lemma provides a closed-from solution for $\{\phi_{i,t}\}_{t=1}^{T}$ and shows the importance of the mixture behavior of the Markov chain W in estimation.

Lemma 1. *Any update of the form* $\phi_{i,t} = \sum_{j=1}^{N} [W]_{ij} \phi_{j,t-1} + \psi_{i,t}$ *can be expressed as,*

$$\phi_{i,t} = \sum_{\tau=1}^{t} \sum_{j=1}^{n} \left[W^{t-\tau} \right]_{ij} \psi_{j,\tau},$$

whenever the update is initialized at $\phi_{i,0}(k) = 0$, *for any* $i \in [N]$ *and* $k \in [K]$. *Also, given the connectivity of the network, the doubly stochastic matrix* W *with positive diagonal satisfies*

$$\sum_{\tau=1}^{t} \sum_{j=1}^{N} \left| \left[W^{t-\tau} \right]_{ij} - \frac{1}{N} \right| \leq \boldsymbol{dE}_1,$$

for any $i \in [N]$, *where*

$$\boldsymbol{dE}_1 := \frac{2}{1 - \sigma_2(W)} + \frac{\log N}{\log \left[\sigma_2(W)^{-1} \right]},$$

and $\sigma_2(W) < 1$ *is the second largest singular value of* W.

Proof. See the Appendix in [27]. □

The lemma suggests that the update $\phi_{i,t}$ accumulates new information from the environment and averages out the past. It is important to have network connectivity, since it allows $W^t \to \frac{1}{N} \mathbb{1}\mathbb{1}^\top$ as $t \to \infty$. Indeed, when the underlying network topology is disconnected, information cannot spread in the whole network. Therefore, players cannot observe some of informative signals dispersed throughout the network. In this case, they are not able to make a correct inference about the true reward of the arms, resulting in the network regret increasing linearly in time.

Finally, the lemma implies that the performance of the algorithm relies on the mixture properties of the Markov chain W. This is proved by the dependence of the quantity $\boldsymbol{dE}_1$ to $\sigma_2(W)$.

Theorem 2. *Let for each* $k \in [K]$ *and* $i \in [N]$, *the sequence* $\{X_{i,t}(k)\}_{t=1}^{T}$ *be i.i.d. samples from a stationary distribution with* $\mu_i(k) = \mathbb{E}[X_{i,t}(k)]$. *Let also the independence over arms and agents hold such that*

$$\mathbb{E}[X_{i,t}(k) X_{j,t}(k')] = \mathbb{E}[X_{i,t}(k)] \mathbb{E}[X_{j,t}(k')],$$

for $i \neq j$ *or* $k \neq k'$. *Given the connectivity of the network, the regret of* d-UER *algorithm, defined in* (1), *satisfies the following bound*

$$\mathbf{R}_T \leq \sum_{k=2}^{K} 4 \max \left\{ \frac{12 \log T}{N \Delta_{1,k}}, Nd \right\} + K \sum_{k=2}^{K} \left(2.5 \left(1 + \log \left[\frac{4}{\Delta_{1,k}} \right] \right) \boldsymbol{dE}_1 \boldsymbol{dE}_2 + \frac{2\pi^2}{3} \Delta_{1,k} \right),$$

whenever $d \geq \boldsymbol{dE}_1$, *where*

$$\boldsymbol{dE}_1 := \frac{2}{1 - \sigma_2(W)} + \frac{\log N}{\log \left[\sigma_2(W)^{-1} \right]},$$

and

$$\boldsymbol{dE}_2 := \frac{\log N}{\log \left[\sigma_2(W)^{-1} \right]}.$$

Proof. See the Appendix in [27]. □

Theorem 2 indicates that the regret depends on the network size as well as the second largest singular value of W. Defining the *spectral gap* of the network as

$$\gamma(W) := 1 - \sigma_2(W),$$

the theorem further shows that the regret bound scales inversely in the spectral gap of the network. Note that regret serves as a *non-asymptotic* performance metric, reiterating the importance of the spectral gap in the finite-time analysis.

The local feedback does not provide each player with adequate information, yielding a delay in proper decision-making. For instance, in cycle and path networks where the diameter is $\mathcal{O}(N)$ the incurred penalty $\boldsymbol{dE}_1 = \mathcal{O}(N^2 \log N)$ is large, whereas in a complete network $W = \frac{1}{N}\mathbb{1}\mathbb{1}^\top$ the Markov chain is mixed from the outset, and there is no network penalty. The latter can be seen as N copies of a single-player MAB where $\sigma_2(W) = 0$. In this case, the network errors become $\boldsymbol{dE}_1 = 2$ and $\boldsymbol{dE}_2 = 0$, and the well-known result of [3] for `UCB1` algorithm is recovered (scaled down by a factor of N). The $\frac{1}{N}$ factor is an advantage gained through reducing the variance of samples by distributing N samples among N individuals.

4. NUMERICAL EXPERIMENTS

Our theoretical results indicate that the size and spectral gap of the network are decisive in the performance of `d-UER` . In this section, we study the impact of these two factors via numerical experiments. In the first scenario, we consider networks of the same size differing in the spectral gap, while in the second scenario we consider complete networks of different sizes. For both cases, we plot the regret versus time horizon to investigate the performance.

For all of our experiments, we deal with $K = 4$ arms. We divide the agents into two groups, for whom the expected value of the rewards are as follows,

	$\mu_i(1)$	$\mu_i(2)$	$\mu_i(3)$	$\mu_i(4)$
$i \in$ Group 1	0.0149	0.7161	0.7944	0.6749
$i \in$ Group 2	0.5144	0.5108	0.9955	0.4778

generated randomly in the unit interval. This gives rise to the following global rewards

$\mu(1)$	$\mu(2)$	$\mu(3)$	$\mu(4)$
0.2646	0.6135	0.8950	0.5764

and the best global arm would be arm 3.

4.1. The Impact of Spectral Gap

For the first scenario, we fix the network size to $N = 50$. We would like to evaluate the performance of `d-UER` algorithm in three networks: complete, cycle and 4-regular (all with self-loops). For the reward values given in the tables, we run 100 experiments and average out regret over these runs. We then plot Fig. 1 which shows regret for these networks with respect to time ($T = 3000$).

As verified in theoretical results, the regret bound scales inversely with the spectral gap $\gamma(W) = 1 - \sigma_2(W)$. We can observe the impact in Fig. 1 where the networks are sorted correctly with respect to this metric. The complete network (largest spectral gap) has the best performance, while the 4-regular outperforms the cycle (due to its larger spectral gap). The result is also consistent with the intuition that the networks should be sorted according to their connectivity. The more the connectivity, the less the regret.

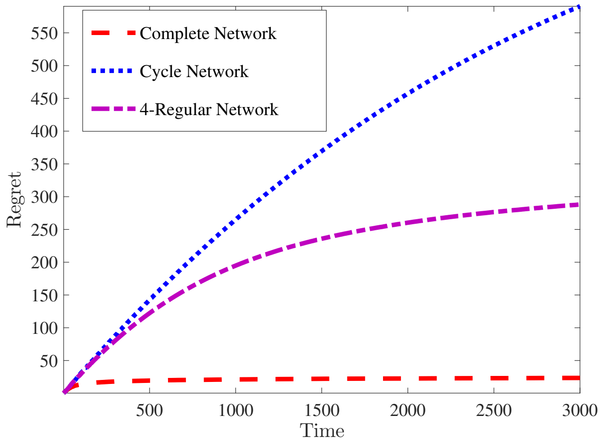

Fig. 1. Performance of `d-UER` in complete, cycle, and 4-regular networks. The regret scales inversely in the spectral gap. Hence, as the spectral gap grows larger, the regret becomes smaller.

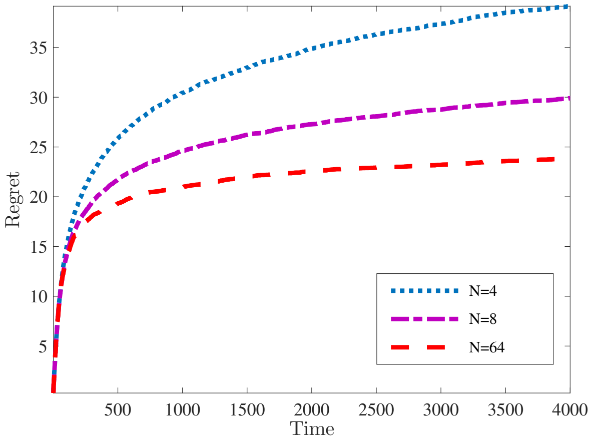

Fig. 2. Performance of `d-UER` in complete networks of size $N \in \{4, 8, 64\}$. As network grows larger, the estimation variance for agents decreases, and the regret becomes smaller.

4.2. The Impact of Network Size

Theorem 2 indicates that the regret scales logarithmically with time in the long run, and the dominant term has a $1/N$ factor. Therefore, the network size proves to be crucial in the performance. We choose the complete networks such that $W = \frac{1}{N}\mathbb{1}\mathbb{1}^\top$, for different values of $N \in \{4, 8, 64\}$ to investigate the performance of `d-UER` algorithm. We study complete networks to remove the effect of spectral gap and focus on size. In this case, the spectral gap always remains to be one and does not change with network size.

For the reward values given in the tables, we run 100 experiments and average out regret over these runs. Fig. 2 shows the regret for the three networks with respect to time ($T = 4000$). The simulation certifies that larger network size results in a lower regret.

5. REFERENCES

[1] Tze Leung Lai and Herbert Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[4] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári, "Exploration–exploitation tradeoff using variance estimates in multi-armed bandits," *Theoretical Computer Science*, vol. 410, no. 19, pp. 1876–1902, 2009.

[5] Kehao Wang and Lin Chen, "On optimality of myopic policy for restless multi-armed bandit problem: An axiomatic approach," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 300–309, 2012.

[6] Sébastien Bubeck and Nicolo Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[7] Aditya Mahajan and Demosthenis Teneketzis, "Multi-armed bandit problems," in *Foundations and Applications of Sensor Management*, pp. 121–151. Springer, 2008.

[8] Sattar Vakili, Keqin Liu, and Qing Zhao, "Deterministic sequencing of exploration and exploitation for multi-armed bandit problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 759–767, 2013.

[9] Saeed Bagheri and Anna Scaglione, "The restless multi-armed bandit formulation of the cognitive compressive sensing problem," *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1183–1198, 2015.

[10] Dušan Jakovetić, José MF Moura, and João Xavier, "Distributed detection over noisy networks: Large deviations analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4306–4320, 2012.

[11] Shahin Shahrampour, Alexander Rakhlin, and Ali Jadbabaie, "Distributed detection : Finite-time analysis and impact of network topology," *IEEE Transactions on Automatic Control*, vol. 61, 2016.

[12] Keqin Liu and Qing Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, 2010.

[13] Cem Tekin and Mingyan Liu, "Performance and convergence of multi-user online learning," in *Game Theory for Networks*, pp. 321–336. Springer, 2012.

[14] Dileep Kalathil, Naumaan Nayyar, and Rahul Jain, "Decentralized learning for multiplayer multiarmed bandits," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2331–2345, 2014.

[15] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff, "Multi-armed bandits in the presence of side observations in social networks," in *IEEE Conference on Decision and Control (CDC)*, 2013, pp. 7309–7314.

[16] Soummya Kar, H Vincent Poor, and Shuguang Cui, "Bandit problems in networks: Asymptotically efficient distributed allocation rules," in *IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 1771–1778.

[17] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard, "On distributed cooperative decision-making in multiarmed bandits," *arXiv preprint arXiv:1512.06888*, 2015.

[18] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard, "Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms," *arXiv preprint arXiv:1606.00911*, 2016.

[19] Shahin Shahrampour and Ali Jadbabaie, "Exponentially fast parameter estimation in networks using distributed dual averaging," in *IEEE Conference on Decision and Control (CDC)*, 2013, pp. 6196–6201.

[20] Anusha Lalitha, Anand Sarwate, and Tara Javidi, "Social learning and distributed hypothesis testing," in *International Symposium on Information Theory (ISIT)*, 2014, pp. 551–555.

[21] Anusha Lalitha and Tara Javidi, "On the rate of learning in distributed hypothesis testing," in *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 1–8.

[22] Angelia Nedić, Alex Olshevsky, and César A Uribe, "A tutorial on distributed (non-bayesian) learning: Problem, algorithms and results," in *26th IEEE Conference on Decision and Control (CDC)*, 2016, pp. 6795–6801.

[23] A. Nedic, A. Olshevsky, and C.A. Uribe, "Nonasymptotic convergence rates for cooperative learning over time-varying directed graphs," in *American Control Conference (ACC)*, July 2015, pp. 5884–5889.

[24] S. Shahrampour, M.A. Rahimian, and A. Jadbabaie, "Switching to learn," in *American Control Conference (ACC)*, July 2015, pp. 2918–2923.

[25] Lili Su and Nitin H Vaidya, "Non-bayesian learning in the presence of byzantine agents," in *International Symposium on Distributed Computing*. Springer, 2016, pp. 414–427.

[26] Lili Su and Nitin H Vaidya, "Asynchronous non-bayesian learning in the presence of crash failures," in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*. Springer, 2016, pp. 352–367.

[27] Shahin Shahrampour, Alexander Rakhlin, and Ali Jadbabaie, "Multi-armed bandits in multi-agent networks," *https://sites.google.com/site/shahinshahrampour/publications*, 2016.