Social Learning in Multi Agent Multi Armed Bandits

Abishek Sankararaman

abishek@utexas.edu

The University of Texas at Austin
Austin, Texas, USA

Ayalvadi Ganesh

A.Ganesh@bristol.ac.uk

University of Bristol
Bristol, UK

Sanjay Shakkottai

sanjay.shakkottai@utexas.edu

The University of Texas at Austin
Austin, Texas, USA

ABSTRACT

We introduce a novel decentralized, multi agent version of the classical Multi-Arm Bandit (MAB) problem, consisting of n agents, that collaboratively and simultaneously solve the same instance of K armed MAB to minimize individual regret. The agents can communicate and collaborate among each other *only* through a pairwise asynchronous gossip based protocol that exchange a limited number of bits. In our model, agents at each point decide on (i) which arm to play, (ii) whether to, and if so (iii) what and whom to communicate with.

We develop a novel algorithm in which agents, whenever they choose, communicate only arm-ids and not samples, with another agent chosen uniformly and independently at random. The per-agent regret achieved by our algorithm is $O\left(\frac{\lceil \frac{K}{n} \rceil + \log(n)}{\Delta} \log(T)\right)$, where Δ is the difference between the mean of the best and second best arm. Furthermore, any agent in our algorithm communicates (arm-ids to an uniformly and independently chosen agent) only a total of $\Theta(\log(T))$ times over a time interval of T .

We compare our results to two benchmarks - one where there is no communication among agents and one corresponding to complete interaction, where an agent has access to the entire system history of arms played and rewards obtained of all agents. We show both theoretically and empirically, that our algorithm experiences a significant reduction both in per-agent regret when compared to the case when agents do not collaborate and each agent is playing the standard MAB problem (where regret would scale linearly in K), and in communication complexity when compared to the full interaction setting which requires T communication attempts by an agent over T arm pulls. Our result thus demonstrates that even a minimal level of collaboration among the different agents enables a significant reduction in per-agent regret.

KEYWORDS

Decentralized Systems; Gossip Algorithms; Online Learning

ACM Reference Format:

Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. 2020. Social Learning in Multi Agent Multi Armed Bandits. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '20 Abstracts)*, June 8–12, 2020, Boston, MA, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3393691.3394217>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS '20 Abstracts, June 8–12, 2020, Boston, MA, USA

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7985-4/20/06.

<https://doi.org/10.1145/3393691.3394217>

The Multi Armed Bandit (MAB) problem is a fundamental model to study online learning and the exploration-exploitation trade offs associated with them. Motivated by the increasing need to design learning algorithms for several large scale networked systems, we introduce a collaborative multi-agent version of the classical MAB problem. Some common examples of such networked systems include (i) social and peer-to-peer recommendation services catering to large number of users who are in turn connected by a network ([4]), (ii) a collection of distributed sensors or Internet of Things (IoT) devices learning about the underlying environment (such as road traffic conditions) and connected with each other through some communication infrastructure such as the wireless spectrum ([6]), and (iii) online marketplaces with many services catering to the same customer base, where the different services can potentially share data about the users in some privacy compatible form ([2]). Detailed examples can be found in the full version [5]. A common theme in many of these applications is the presence of a single MAB instance, which many agents are simultaneously playing to minimize their own cumulative regret. Our objective is to understand the benefit of collaboration in speeding up learning.

Model overview

Our setting consists of n agents, that collaboratively solve the same instance of a stochastic K -armed MAB problem. The objective of each agent is to take actions to minimize their own individual cumulative regret. If there were just one agent, or if the agents were oblivious to each other and did not collaborate, then each agent is playing independently, the classical K armed MAB problem. In our model, the agents can potentially collaborate with each other in solving the MAB problem by sending messages to each other over a communication network connecting them. Agents take action at epochs of a Poisson process (asynchronous issue) consists of (i) an arm to play and observe a reward, (ii) whether to communicate, and if so, (iii) what and with whom to communicate. Our model imposes three constraints on the communications among agents. Firstly, each agent, whenever it chooses to communicate, can do so with only one other agent and is thus, 'gossip style communications'. Secondly, agents can only communicate a limited $O(\log(nK))$ bits, each time they choose to communicate. The number of bits communicated in each communication attempt cannot scale with time or communicate real numbers upto arbitrary precision (such as empirical arm-means). Thirdly, each agent can access the communication medium only $o(T)$ times over any horizon of T pulls of arms. This restriction disallows agents from communicating each time they pull an arm and observe a reward. Furthermore, the agents are *decentralized* - namely the actions of each agent can only depend on the agents past history of arms played, rewards obtained and messages received.

Figure 1: A plot consisting of 80 agents and 40 arms (means chosen randomly) comparing the single agent UCB without communications with our scheme. The curve is averaged over 10 runs with 95% confidence interval.

Main Result

We develop a novel algorithm that leverages collaboration across agents, such that the per-agent regret after an agent has played for T times is $\Theta\left(\frac{\log(n) + \lceil \frac{K}{n} \rceil}{\Delta} \log(T) + \frac{\log^3(n) \log \log(n)}{\Delta^2}\right)$, where Δ is the arm-gap between the best and the second best arm. Moreover, in a time interval of T , an agent communicates for about $\log(T)$ times, where each communication is an arm-id, i.e., uses at-most $O(\log_2(K))$ bits per communication.

Key Idea in Algorithm

The main idea in our algorithm is to use the communication medium only to *recommend arms*, rather than to exchange observed scores or rewards. Our policy restricts agents to only play from the set of arms they are *aware* of at any instant of time. Each agent is only aware of a small set of arms in the beginning, and this set increases with time as agents receive recommendations. Agents in our algorithm communicate with another agent chosen uniformly and independently at random, and thus the communication is ‘gossip style’ [7].

Qualitatively, our regret scaling occurs due to two reasons: (i) The (local-explore + gossip) mechanism underlying our algorithm ensures that the *best arm spreads quickly through the network* to all agents. As agents only play from among arms they are aware of, it is not apriori certain that all agents become aware of the best-arm. (ii) Nevertheless, our algorithm ensures that each agent in the network only ever *explores a vanishingly small fraction* $\Theta(\frac{1}{n} + \frac{\log(n)}{K})$ of the arms. In other words, the sub-optimal arms do not spread and thus not all agents need to learn and discard the sub-optimal arms.

Comparison with Benchmark Systems

We quantify the effect of collaboration with limited communication on the learning performance by comparing our result with two baselines - no communication among agents and a one with complete interactions among agents.

1. No Communication regime - If the players do not interact at all, then each player will see a standard MAB problem consisting of K arms. Thus, from well known results (for ex. [1]), each agent

after playing the MAB problem for T time steps, must incur a regret that scales as $O(\frac{K}{\Delta} \log(T))$ [3].

2. Full Interaction Regime - On the other end is the perfect collaboration model in which every agent at each instance it plays an arm and observes a reward, *broadcasts* both the arm played and reward obtained to all other agents. This does not satisfy the constraints of our communication. In this case, every agent before playing an arm, has access to the entire system history and thus can jointly simulate a single agent optimal scheme. Thus, after a total of T clock ticks of a particular agent, the total number of arm pull by all agents is roughly nT and thus the overall system regret is $O(\frac{K}{\Delta} \log(nT))$. As there are n agents in total, the per agent regret in this case scales as $\frac{1}{n} O(\frac{K}{\Delta} \log(nT))$ which is of order

$O(\frac{K}{n} \frac{1}{\Delta} \log(T) + \frac{K \log(n)}{n \Delta})$. This is the best possible per-agent regret scaling one can hope for in this networked setting and no other collaborative policy can beat this regret scaling. However, to achieve this, each agent must communicate, both its arms and the observed reward to all other agents, after each pull (i.e., communicate T times over T plays)

Thus, despite communication constraints, we show that our algorithm achieves, both a significant reduction in the per-agent regret compared to the setting of no interactions among agents by bringing the order from K to $\lceil \frac{K}{n} \rceil + \log(n)$ as the leading term in front of $\frac{\log(T)}{\Delta}$. On the other hand, Our algorithm is also only a factor $\log(n)$ off from the setting of complete interaction among agents, which has a factor of $\frac{K}{n}$ in front of the $\frac{\log(T)}{\Delta}$ term. Moreover, our algorithm achieves the reduced regret scaling with a much smaller communication resources where an agent only uses the communication channel of order $\log(T)$ times over T times of play of an agent. We plot in Figure 1, a representative situation showing the regret growth of our algorithm against that of the no communication and full interaction case.

Acknowledgements - This work is partially supported by NSF Grant CNS-1704778, ARO grant W911NF-17-1-0359 and the US DoT supported D-STOP Tier 1 University Transportation Center and the Simons Foundation Grant (#197892 to UT Austin).

REFERENCES

- [1] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.
- [2] Edward Boon, Leyland Pitt, and Esmail Salehi-Sangari. How to manage information sharing in online marketplaces – an exploratory study. In *Ideas in Marketing: Finding the New and Polishing the Old*, pages 538–541. Springer International Publishing, 2015.
- [3] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.
- [4] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548. ACM, 2016.
- [5] Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35, 2019. URL: <https://dl.acm.org/doi/10.1145/3366701>, doi: 10.1145/3366701.
- [6] Prabodini Semasinghe, Setareh Maghsudi, and Ekram Hossain. Game theoretic mechanisms for resource management in massive wireless iot systems. *IEEE Communications Magazine*, 55(2):121–127, 2017.
- [7] Devavrat Shah. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009.