Bandit networks

Charles AUGUSTE
École des Ponts
charles.auguste@ponts.org

Louis TREZZINI
École des Ponts
louis.trezzini@ponts.org

Abstract

A single-agent multi-armed bandit (MAB) problem is a problem in which a gambler, being faced with several slot machines, has to decide which machines to play, how many times and in which order to play each machine. In this project, we consider a multi-agent MAB scenario in which a group of agents connected through a social network are engaged in playing a stochastic MAB game and focus on minimizing their regret. Every time a player takes an action, the reward is observed by both the agent and its neighbors in the network. The goal of this project is to understand different collaborative policies (NAIC-type and FYL) described in [1], and to compare their performance over different network structures.

1 Introduction

This project is a review of paper [1]. As such, we will only succinctly present their model and their main results, and refer to additional results in [1] when necessary.

The interesting part of bandit networks is that we consider multiple agents playing the same multi-armed bandit (MAB) problem. Instead of playing independently, they can be connected through a graph and share their knowledge of the game (actions taken and resulting rewards) with their neighbors. Moreover, they can decide to play very diverse policies, which can be much more than simple extensions of UCB or Thomson-Sampling to the network-setting. These collaborative policies can really profit of the structure of the graph considered and even adapt to the locally available information.

2 The model

We use the definitions and notations from [1]. Let's restate them quickly.

2.1 Single-agent stochastic multi-armed bandit (MAB) problem

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of arms available to the agent. Each arm is associated with a distribution $\mathcal{P}_k$, and let μ_k be the corresponding mean, unknown to the agent. Let n be the time horizon or the total number of rounds. In each round t, the agent chooses an arm, for which he receives a reward, an i.i.d. sample drawn from the chosen arm's distribution. The agent can use the knowledge of the chosen arms and the corresponding rewards upto round $(t-1)$ to select an arm in round t. The goal of the agent is to maximize the cumulative expected reward up to round n.

2.2 Multi-agent stochastic multi-armed bandit (MAB) problem

We consider a set of users V connected by an undirected fixed network $G = (V, E)$, with $|V| = m$. Assume that each user is learning the same stochastic MAB problem i.e., faces a choice in each time from among the same set of arms $\mathcal{K}$. In the t^{th} round, each user v chooses an arm, denoted

Preprint. Work in progress.

by $a^v(t) \in \mathcal{K}$, and receives a reward, denoted by $X^v_{a^v(t)}(t)$, an i.i.d. sample drawn from $\mathcal{P}_{a^v(t)}$. In the stochastic MAB problem set-up, for a given user v, the rewards from arm i, denoted by $\{X^v_i(t) : t = 1, 2, \ldots\}$, are i.i.d. across rounds. Moreover, the rewards from distinct arms i and j, $X^v_i(t)$, $X^v_j(s)$, are independent. If multiple users choose the same action in a certain round, then each of them gets an independent reward sample drawn from the chosen arm's distribution. We use the subscripts i, v and t for arms, nodes and time respectively. The information structure available to each user is as follows. A user v can observe the actions and the respective rewards of itself and its one hop neighbors in round t, before deciding the action for round $(t+1)$.

The policy Φ^v followed by a user prescribes actions at each time t, $\Phi^v(t) : H^v(t) \rightarrow \mathcal{K}$, where $H^v(t)$ is the information available with the user till round t. A policy of the network G, denoted by Φ, comprises of the policies pertaining to all users in G. The performance of a policy is quantified by a real-valued random variable, called *regret*, defined as follows. The regret incurred by user v for using the policy Φ^v up to round n is defined as,

$$R^v_\Phi(n) = \sum_{t=1}^{n} \left(\mu^* - \mu_{a^v(t)}\right) = n\mu^* - \sum_{t=1}^{n} \mu_{a^v(t)},$$

where $a^v(t)$ is the action chosen by the policy Φ^v at time t, and $\mu^* = \max\limits_{1 \le i \le K} \mu_i$. We refer to the arm with the highest expected reward as the optimal arm. The regret of the entire network G under the policy Φ is denoted by $R^G_\Phi(n)$, and is defined as the sum of the regrets of all users in G. The expected regret of the network is given by:

$$\mathbb{E}[R^G_\Phi(n)] = \sum_{v \in V} \sum_{i=1}^{K} \Delta_i \mathbb{E}[T^v_i(n)], \tag{1}$$

where $\Delta_i = \mu^* - \mu_i$, and $T^v_i(n)$ is the number of times arm i has been chosen by Φ^v up to round n. We omit Φ from the regret notation, whenever the policy can be understood from the context. Our goal is to devise learning policies in order to minimise the expected regret of the network.

Let $\mathcal{N}(v)$ denote the set consisting of the node v and its one-hop neighbours. Let $m^v_i(t)$ be the number of times arm i has been chosen by node v and its one-hop neighbors till round t, and $\hat{\mu}_{m^v_i(t)}$ be the average of the corresponding reward samples. These are given as:

$$m^v_i(t) = \sum_{u \in \mathcal{N}(v)} T^u_i(t)$$

$$\hat{\mu}_{m^v_i(t)} = \frac{1}{m^v_i(t)} \sum_{u \in \mathcal{N}(v)} \sum_{k=1}^{t} X^u_{a^u(k)}(k) \mathbb{I}\{a^u(k) = i\},$$

where $\mathbb{I}$ denotes the indicator function. We use $m^G_i(t)$ to denote the number of times arm i has been chosen by all nodes in the network till round t.

3 Policies

3.1 UCB-Network

They propose the natural extension of the well-known single agent policy UCB to the network-setting, that they call "UCB-user". When each user in the network follows the UCB-user policy, the network policy is called UCB-Network and is outlined in Algorithm 1.

The following theorem presents an upper bound on the expected regret of a generic network, under the UCB-Network policy.

Theorem 1. *Assume that the network G follows the UCB-Network policy to learn a stochastic MAB problem with K arms. Further, assume that the rewards lie in $[0, 1]$. Then,*

(i) The expected total regret of G is upper bounded as:

$$\mathbb{E}\left[R^G(n)\right] \le b + \sum_{i:\mu_i < \mu^*} C_G \frac{8 \ln n}{\Delta_i},$$

Algorithm 1 Upper-Confidence-Bound-Network (UCB-Network)

Each user in G follows UCB-user policy
UCB-user policy for a user v:
Initialization: For $1 \leq t \leq K$
- play arm t
Loop: For $K \leq t \leq n$
- $a^v(t+1) = \underset{j}{\text{argmax}}\ \hat{\mu}_{m_j^v(t)} + \sqrt{\frac{2\ln t}{m_j^v(t)}}$

where $\Delta_i = \mu^* - \mu_i$, $\beta \in (0.25, 1)$, $b = m\left[\frac{2}{4\beta-1} + \frac{2}{(4\beta-1)^2 \ln(1/\beta)}\right]\left(\sum_{j=1}^{K} \Delta_j\right)$, *and* C_G *is a network dependent parameter, defined as follows.*

(ii) Let $\gamma_k = \min\{t \in \{1,\dots,n\} : |\{v \in V : m_i^v(t) \geq l_i = \frac{8\ln n}{\Delta_i^2}\}| \geq k\}$ *denote the smallest time index when at least* k *nodes have access to at least* l_i *samples of arm* i. *Let* η_k *be the index of the 'latest' node to acquire* l_i *samples of arm* i *at* γ_k, *such that* $\eta_k \neq \eta_{k'}$ *for* $1 \leq k, k' \leq m$. *Define* $z_k = T_i(\gamma_k) := \big(T_i^1(\gamma_k), \dots, T_i^m(\gamma_k)\big)$, *which contains the arm* i *counts of all nodes at time* γ_k. *Then,* $C_G l_i$ *is the solution of the following optimisation problem:*

$$\begin{aligned} &\max \quad \|z_m\|_1 \\ &\textit{s.t } \exists \textit{ a sequence } \{z_k\}_{k=1}^m \\ &z_j(\eta_k) = z_k(\eta_k) \ \ \forall j \geq k \\ &\langle z_k, A(\eta_k, :)\rangle \geq l_i, \quad 1 \leq k \leq m \end{aligned} \tag{2}$$

Remark 1. *We propose the result in red, which is a correction of the result from [1]. After checking the proof, we think that we can deduce this tighter upper bound on the expected total regret of G.*

Corollary 1. *For an* m*-node star network:*

$$\mathbb{E}[R^G(n)] \leq b + (m-1)\sum_{i:\mu_i<\mu^*} \frac{8\ln n}{\Delta_i} \tag{3}$$

3.2 Follow Your Leader (FYL)

They present a second network policy called Follow Your Leader (FYL) for a generic m-node network. The policy is based on exploiting high-degree hubs in the graph; for this purpose, we need to define notions of dominating sets and dominating set partitions.

Definition 3 [Dominating set of a graph] A *dominating set* D of a graph $G = (V, E)$ is a subset of V such that every node in $V \setminus D$ is adjacent to atleast one of the nodes in D.

Definition 4 [Dominating set partition of a graph] Let D be a dominating set of G. A dominating set partition based on D is obtained by partitioning V into $|D|$ components such that each component contains a node in D and a subset of its one hop neighbors.

The FYL policy for an m-node generic network is outlined in Algorithm 2. Under the FYL policy, all nodes in the dominating set are called *leaders* and all other nodes as *followers*; the follower nodes follow their leaders while choosing an action in a round. The following theorem presents an upper bound on the expected regret of an m-node star network which employs the FYL policy.

Theorem 2 (FYL regret bound, star networks)**.** *Suppose the star network G with a dominating set as the center node, follows the FYL policy to learn a stochastic MAB problem with* K *arms. Assume that the rewards lie in* $[0, 1]$. *Then,*

$$\mathbb{E}[R^G(n)] \leq d + \sum_{i:\mu_i<\mu^*}^{K} \frac{8\ln n}{\Delta_i},$$

where $d = \left[2m - 1 + \frac{2m}{4\beta-1}\left(1 + \frac{1}{(4\beta-1)\ln(1/\beta)}\right)\right]\left(\sum_{j=1}^{K} \Delta_j\right)$, $\Delta_i = \mu^* - \mu_i$ *and* $\beta \in (0.25, 1)$.

Algorithm 2 Follow Your Leader (FYL) Policy

Input: Graph G, a dominating set D and a dominating set partition
Leader - Each node in D:
Follows the UCB-user policy by using the samples of itself and its neighbors
Follower - Each node in $V \setminus D$:
In round $t = 1$:
- Chooses an action randomly from $\mathcal{K}$

In round $t > 1$:
- Chooses the action taken by the leader in its component, in the previous round $(t-1)$

A key insight obtained from Theorem 2 is that an m-node star network with the FYL policy incurs an expected regret that is lower by a factor $(m-1)$, as compared to UCB-Network.

3.3 Lower bounds on the expected regret

We decided not to focus on the lower bounds on the expected regret during this review, but it's important to notice that they prove in [1] that the regret upper bound under the FYL policy meets a lower bound universal to all policies. Hence, they conclude that the FYL policy is order optimal for star networks.

4 Our theoretical contributions

4.1 Upper bound on the expected regret for UCB-Network policy on a multiple stars graph

Suppose that you have the following m-nodes graph, that we call multiple stars graph. There are S stars on the graph. Each star j is made of a hub and its m_j children. The only connections between stars are made through their hub (central node). These connections can be can be anything ranging from full disconnection of central nodes of the stars to full connection of the central nodes of the stars.

Proposition 1 (Upper bound on the expected regret for UCB-Network policy on a multiple stars graph)**.** *Assume that a multiple stars graph G as defined above, follows a UCB-Network policy to learn a stochastic MAB problem with K arms. Further, assume that rewards are in* $[0, 1]$*. Then, regardless of the connections between the central nodes of the stars, the expected total regret of G is upper bounded as :*

$$\mathbb{E}(R^G(n)) \le (m-S) \sum_{i:\mu_i<\mu^*} \frac{8\ln(n)}{\Delta_i} + b$$

where $\Delta_i = \mu^* - \mu_i$, $\beta \in (0.25, 1)$, $b = m\left[\frac{2}{4\beta-1} + \frac{2}{(4\beta-1)^2\ln(1/\beta)}\right]\left(\sum_{j=1}^{K}\Delta_j\right)$.

Remark 2. *This result can be seen as an extension of Corollary 1 when* $S = 1$.

Proof. We use Theorem 1 of [1] to get the general result, and we only need to prove that for our multiple stars network G, we have that $C_G = \sum_{j=1}^{m} m_j = m - S$. For that, we need to solve the optimization program (2) of Theorem 1 of [1]. For our network, the formulation is that $C_G l_i$ is the solution of the following optimisation problem where A is the adjacency matrix of our graph.

$$\begin{aligned}
\underset{x}{\text{maximize}} \quad & ||z_m|| \\
\text{subject to} \quad & \exists \text{ a sequence } \{z_k\}_{k=1}^m \in \{0, l_i\}^m \\
& \exists \text{ a permutation } \{\eta_k\}_{k=1}^m \text{ of } (1, ..., m) \\
& z_l(\eta_k) = z_k(\eta_k) \qquad \forall l \ge k \\
& \langle z_k, A(\eta_k, :)\rangle \ge l_i \qquad 1 \le k \le m
\end{aligned}$$

For a multiple-stars network, the solution of this program is $l_i \sum_{j=1}^{n} m_i$. This corresponds to the scenario where none of the stars centers ever chooses the sub-optimal arm i, and every leaf of every star chooses it l_i times. It is clearly feasible because, we can have z_m be 0 for all the centers of the stars, and l_i for all the leaves. Then the order of the permutation η_k doesn't matter.

Moreover, if the centers of the stars are fully disconnected, we know that $l_i \sum_{j=1}^{S} m_j$ is indeed the maximum of the program because we can solve it star by star as they don't interact. When there is a connection between the centers, there is more information going through the graph, and therefore, the maximum can only be lower than $l_i \sum_{j=1}^{S} m_j$.

Therefore, the maximum is indeed $l_i \sum_{j=1}^{S} m_j$, and we have $C_G = \sum_{j=1}^{S} m_j = m - S$.

□

5 Experimental results

5.1 Reproducing experiments from [1]

Considering this is a review project, we decided to first reproduce the experimental results from the paper.

5.1.1 Experiment 1

Performance comparison of UCB-Network policy on various 10 node networks: 2 arms, Bernoulli rewards with means 0.5 and 0.7.

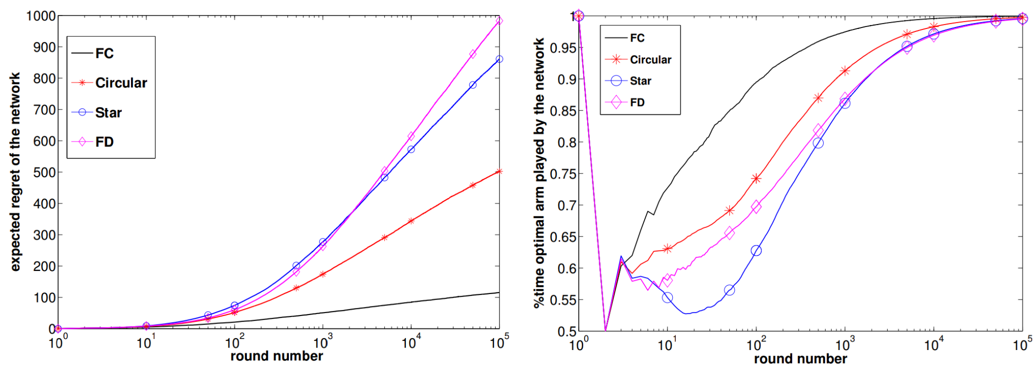

Figure 1: Results for experiment 1 from [1] (100 sample paths).

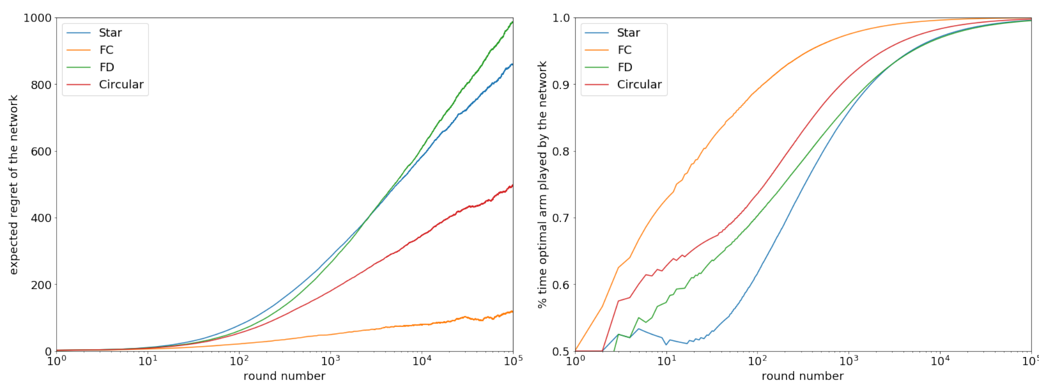

Figure 2: Our results for experiment 1 (1000 sample paths).

We observe that we obtain exactly the same results as in [1], except for the expected regret of the network at rounds > 10000. We get a noisier regret, even though we increased the number of simulations up to 1000 sample paths. We think that the authors of [1] decided to smooth their curves.

5.1.2 Experiment 2

Performance comparison of UCB-Network policy on various 20 node networks: 10 arms, Bernoulli rewards with means $0.1, 0.2, \ldots, 1$.

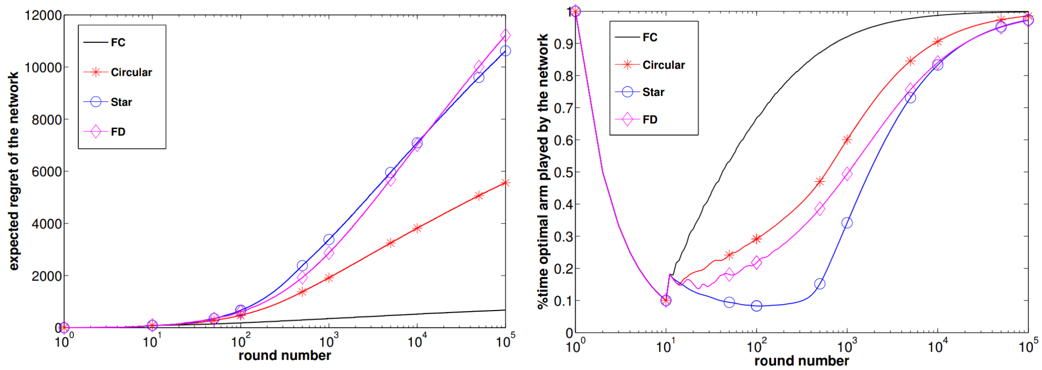

Figure 3: Results for experiment 2 from [1] (100 sample paths).

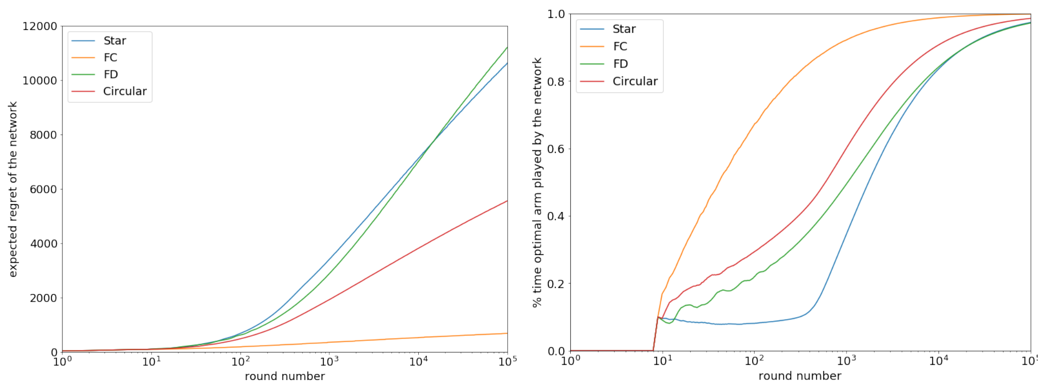

Figure 4: Our results for experiment 2 (1000 sample paths).

Here, we obtain the exact same results as in [1].

5.1.3 Experiment 3

Performance comparison of UCB-Network and FYL policies on various star networks: 2 arms, Bernoulli rewards with means 0.5 and 0.7.

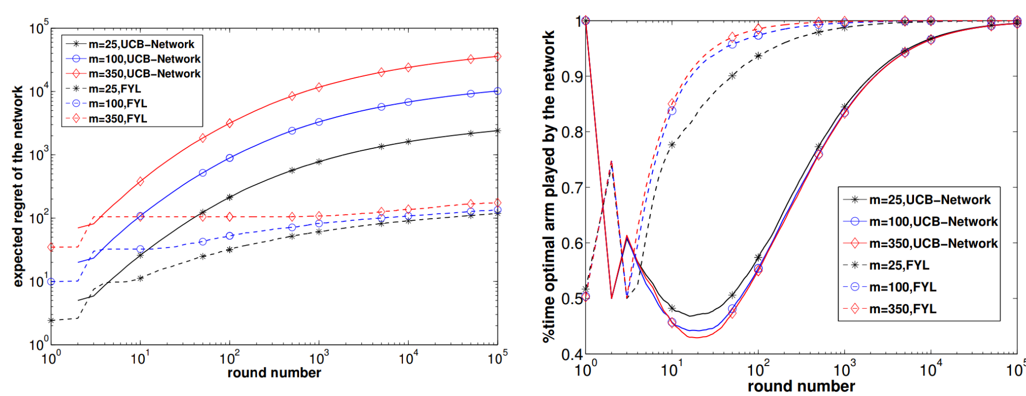

Figure 5: Results for experiment 3 from [1] (100 sample paths).

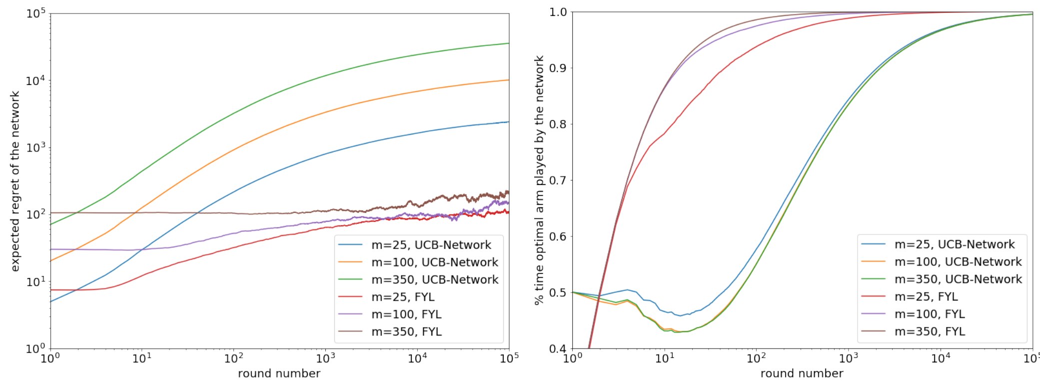

Figure 6: Our results for experiment 3 (1000 sample paths).

Same observation here as for experiment 1: we notice that we obtain exactly the same results as in [1], except for the expected regret of the network at rounds > 10000. We get a noisier regret, even though we increased the number of simulations up to 1000 sample paths. We think that the authors of [1] decided to smooth their curves.

5.2 Evaluating Follow Best Informed (FBI) policy

5.2.1 Comparison between star network and multiple stars network

Finally, we provide experimental results to demonstrate the efficiency our Follow Best Informed (FBI) policy.

We show that it provides similar results as FYL for star networks, and improves the performance for general networks (here, multiple fully connected stars network, see Figure 7).

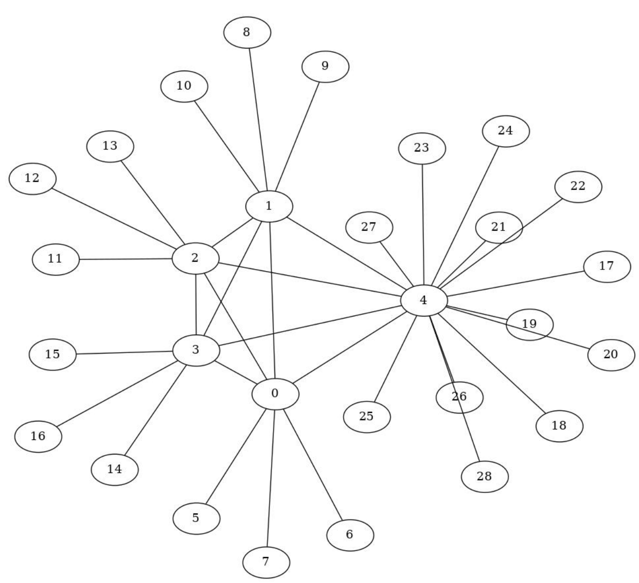

Figure 7: A 29-nodes 5-FC-stars network. It can be seen as a star network where the central hub got split into several sub-hubs.

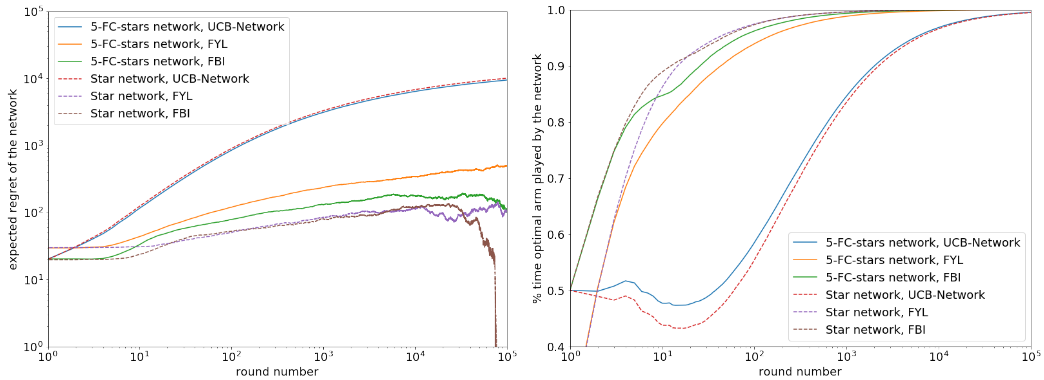

Figure 8: Performance comparison of UCB-Network, FYL, and FBI policies on a 100-nodes star network and on the 100-nodes 5-FC-stars network: 2 arms, Bernoulli rewards with means 0.5 and 0.7. (1000 sample paths).

As expected, FYL and FBI policies perform about the same on star and multiple stars networks. But the great improvement of FBI over FYL lies in its ability to naturally adapt to the graph structure. It greatly decreases expected regret for the multiple stars network, compared to FYL.

5.2.2

This gap is even larger on more pathological graph structure. Here, we compare FYL and FBI on this graph:

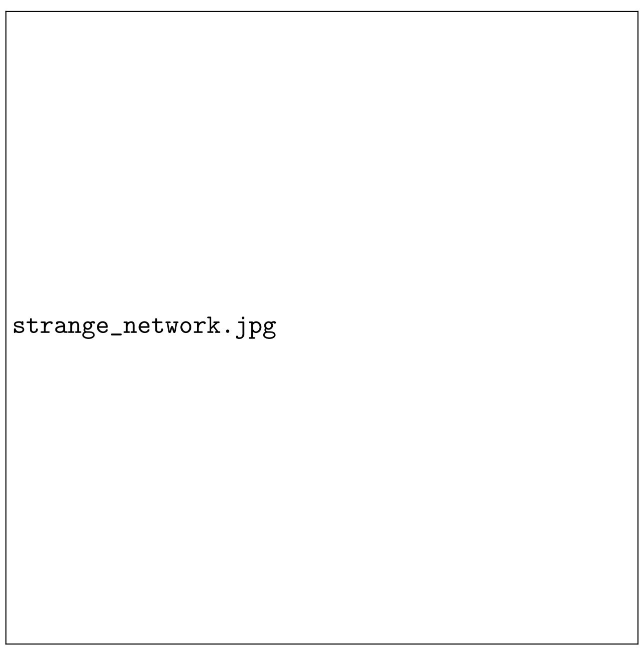

Figure 9: A pathological graph structure. The minimum dominating set is indicated in red.

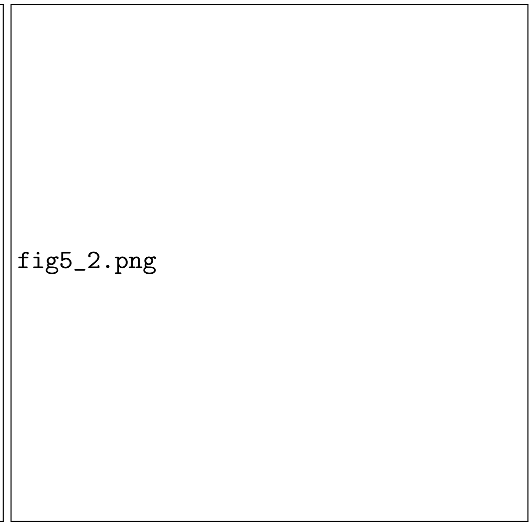

Figure 10: Performance comparison of FYL and FBI policies on the pathological graph structure: 2 arms, Bernoulli rewards with means 0.5 and 0.7. (1000 sample paths).

One issue of FYL is that it doesn't really take into account the full structure of the graph, and can only benefit from star structures. The more general the graph, the worst the policy. This comes from the fact that FYL policy relies on the dominating set (see upper bound of Theorem 6 from [1]). In our example, the dominating set doesn't allow children nodes to collect the most samples, so FBI performs much better than FYL.

6 Conclusion

References

[1] Ravi Kumar Kolla, Krishna P. Jagannathan, and Aditya Gopalan. Stochastic bandits on a social network: Collaborative learning with local information sharing. *CoRR*, abs/1602.08886, 2016.