Multi-armed Bandits in a Network

Farnood Salehi
I&C, EPFL

***Abstract*—The multi-armed bandit problem is a sequential decision problem in which we have several options (arms). We can choose one of these options at each round and receive a reward. In this kind of problem, we see only the outcome of our choice and we do not have full information about the outcomes of the other choices.**

Different patterns are used for generating the rewards. Two famous patterns are stochastic and adversarial. We will present some algorithms for predicting the optimal decision for the case of only one player [1]. One of the new research topics concerning multi-armed bandits is multi-armed bandits in networks. Different kinds of network topologies can be considered. For example in [3], the arms form a network, in which the arms are the vertices of the network, and by choosing one of the vertices, we also receive the reward of its neighbors. The second possible topology is when there is more than one player and these players form a network. Furthermore, all players have same set of arms. In [2], this type of network is studied. Their paper is about analyzing the behavior of players (with a simple myopic Bayesian learning algorithm) more than studying it as a multi-armed bandit problem.

Many algorithms are available for the classic multi-armed bandits problem, however, we believe there is a gap when several players exist and they share information between them.

***Index Terms*—Multi-armed Bandits, Learning algorithms, Learning in networks**

I. Introduction

Proposal submitted to committee: May 27th, 2015; Candidacy exam date: June 3rd, 2015; Candidacy exam committee: Ola Svensson, Friedrich Eisenbrand, Patrick Thiran, Elisa Celis.

This research plan has been approved:

Date: _____

Doctoral candidate: _____
(name and signature)

Thesis director: _____
(name and signature)

Thesis co-director: _____
(if applicable) (name and signature)

Doct. prog. director: _____
(B. Falsafi) (signature)

THE first formalization of the multi-armed bandit problem (MAB) was by Robbins in 1952 [4]. The name originates from a casino slot-machine.

MAB is a sequential decision problem, where the forecaster (or player) has K arms (actions) to choose from in order to receive a reward. After each selection, the forecaster only observes the reward of the selected action. Rewards can be different from round to round. The goal of the forecaster is to maximize his cumulative reward.

One of the possible ways to maximize this reward is to choose the arm that he guesses will give the biggest reward, but he might not have enough information about the other arms. What is the best way to play? There are two major strategies for playing: exploitation and exploration. Exploitation means choosing the arm with the highest expected payoff and exploration means choosing the arms in order to check their payoffs. If the player does only the exploitation, it is possible that after some time the best arm becomes different from what he expects, whereas if he does only exploration, he is unable to use the information he has gathered. The best approach is to combine exploitation and exploration.

In order to obtain a measure of efficiency of the predicting algorithm, the notion of regret is used, the definition of regret can be different in various cases. One of the most popular definitions of regret is the difference between the sum of the rewards of the best arm and the expected reward of the algorithm after τ rounds. This definition of regret implies that the predicting algorithm wants to perform as well as the best action. If we denote the reward of action i at time t by $g_i(t)$ and the arm selected at time t by I_t, then the regret for τ rounds is

$$R_\tau = \max_{i \in K} \sum_{t=1}^{\tau} g_i(t) - \sum_{t=1}^{\tau} g_{I_t}(t). \tag{1}$$

R_τ is a random variable. The goal is to upper bound the expected value of R_τ. As it is not easy to find an upper bound for the expected value of regret, we usually try to bound the pseudo-regret, defined on the expectation of the cumulative reward of the arms rewards and the cumulative reward of our algorithm, i.e.,

$$\bar{R}_\tau = \max_{i \in K} \mathbb{E}[\sum_{t=1}^{\tau} g_i(t)] - \mathbb{E}[\sum_{t=1}^{\tau} g_{I_t}(t)]. \tag{2}$$

When there is no possible confusion between the term "pseudo-regret" and "regret", we use the regret for convenience. For convenience, instead of using the term pseudo-regret, in the rest of the report we will use term regret.

The two types of rewards used in [1], [2] and [3] are adversarial and stochastic. With adversarial bandits, no assumption is made about the rewards; in the worst case, we can assume

an adversary, who knows our algorithm, sets the rewards of actions in the beginning of each round.

In the stochastic case, the rewards $g_i(t)$ are derived from unknown probability distributions with different means

$$\mathbb{E}[g_i(t)] = \zeta_i \text{ for } i \in K. \tag{3}$$

We denote the best arm by i^* and its expected reward by ζ^*, i.e,

$$\begin{aligned} \zeta^* &= \max_i \zeta_i, \\ i^* &= \text{argmax}_i \zeta_i. \end{aligned} \tag{4}$$

The regret is

$$\bar{R}_\tau = \tau \cdot \zeta^* - \mathbb{E}[\sum_{t=1}^{\tau} g_{I_t}(t)]. \tag{5}$$

We can also write the regret in terms of the difference of the means of rewards, i.e,

$$\bar{R}_\tau = \sum_{i=1}^{K} \triangle_i \mathbb{E}[n_i(\tau)], \tag{6}$$

where $n_i(t)$ is the number of times we choose action i and $\triangle_i = \zeta^* - \zeta_i$ (note that $\triangle_i^* = 0$).

Another version of the multi-armed bandit problem is called the contextual multi-armed bandit problem. In this type of game, we have side information (for example a d-dimensional feature vector S). There is a relation between these contexts and the rewards. After playing the game for sufficient amount of times, the algorithm should be able to extract the pattern of relations between the arms and the context vector. A new definition of regret is used in this case; it is the difference between the algorithm's cumulative reward and the reward of the best policy (i.e., a mapping from contexts to actions). In other words the algorithm should learn the best mapping $f : S \longrightarrow 1, ..., K$ from the space of contexts S to the arms. The definition of regret in contextual bandits is

$$\bar{R}_\tau^S = \max_{f:S \longrightarrow 1,...,K} \mathbb{E}\left[\sum_{t=1}^{\tau} g_{f(s_t)} - \sum_{t=1}^{\tau} g_{I_t}\right], \tag{7}$$

where $s_t \in S$ denotes the context at time t.

An example of contextual MAB is in advertising. In advertising, each person has a coefficient vector, which shows his interest for different products. For example the first element of the coefficient vector denotes the interest of person in sport products, etc. Each advertisement has also a feature vector, denoting the type of produce (e.x. the first element of feature vector denotes how much it is desirable for sport). Based on this feature vector, the advertiser wants to find the best advertisement for the person.

[[One of the main topologies that arises in various settings is MAB in a network. In this setting, b players who have a same set of actions try to choose the optimal arm. The players can share information between each other. This information can be the set of observed rewards before time t by player i, who shares this information with his neighbors. The players use this information to improve their decision: choosing the arm with the highest reward.. These players form a network G. The vertices of this network are players and the edges represent the relation between the players, i.e., the two players with an edge between them are neighbors and share information.

Another interesting setting is when we consider only one player and we present his actions by vertices of a network. By choosing an action the player receives the reward of chosen action and the rewards of neighboring actions in the network.]]

The three different settings of MABs used here are the following:
1. One player chooses from a set of actions as described above [1],
2. Multiple players form a network, and they choose from a set of actions. The players can observe the selected action (and their reward) of their neighbors [2],
3. Actions form a network, and a player chooses one and receives the reward of its neighbors as well [3].

Our research plan focuses mainly on the second topology. To our knowledge, there is no algorithm (formed of exploration and exploitation) proposed for this case.

First we will see some algorithms for the classic MAB, then we will see some network topologies that have been studied before. Finally, we will present our problem.

II. Learning algorithms

In [1], classic MAB problems and some algorithms for solving them are presented. The algorithms designed for stochastic bandits are usually deterministic, the idea behind the algorithm uses the principle of *optimism in the face of uncertainty*; the algorithm maintains an optimistic upper bound on the rewards of each arm, and selects an arm with maximal upper bound. However, randomized algorithms are used for adversarial bandits. One reason is that if we choose the actions deterministically, as the adversary knows our algorithm he will set the reward of that action (which will be chosen) to zero.

A. Upper Confidence-Bound Strategy (UCB)

The upper confidence-bound strategy (UCB) was designed for the stochastic bandits. This algorithm was introduced by Auer, Cesa-Bianchi, and Fischer [5]. Usually in the stochastic bandits, some assumptions are made on the distribution of the rewards. For UCB, the following condition should be satisfied.

Let $n_i(t-1)$ be the number of times that we select action i by time $t-1$, and let $\hat{\zeta}_{i,n_i(t-1)}$ be the empirical mean of rewards of action i seen by the player up to time $t-1$, i.e.,

$$\hat{\zeta}_{i,n_i(t-1)} = \frac{\sum_{j=1}^{n_i(t-1)} g_i(t_j^i)}{n_i(t-1)}, \tag{8}$$

recall that $g_i(t)$ is the reward of action i at time t and t_j^i is the time where action i is selected for the j^{th} time.
Let ψ be a convex function on the reals such that, for all $\lambda \geq 0$,

$$\ln \mathbb{E}[e^{\lambda \left|\hat{\zeta}_{i,n_i(t-1)} - \zeta_i\right|}] \leq \psi(\lambda), \tag{9}$$

and ψ^* be its Legendre-Fenchel transform (which is invertible), defined by

$$\psi^*(\varepsilon) = \sup_{\lambda \in \mathbb{R}^+} \{\lambda\varepsilon - \psi(\lambda)\}. \tag{10}$$

Let $\delta = t^{-\alpha}$, where α is a parameter larger than 2, which should be set by the player (depending on the rewards, it can take different values).
In the UCB algorithm, a probabilistic upper bound on the rewards, holding with probability $1-\delta$, is first obtained, then the arm with the highest upper bound is selected.
Using Markov's inequality, we have the following formula:

$$\mathbb{P}[\zeta_i - \hat{\zeta}_{i,n_i(t-1)} > \varepsilon] \leq e^{-n_i(t-1)\psi^*(\varepsilon)}. \tag{11}$$

Let $(\psi^*)^{-1}$ be the inverse of ψ^* and set $\delta = e^{-n_i(t-1)\psi^*(\varepsilon)}$, hence $\varepsilon = (\psi^*)^{-1}\left(\frac{1}{n_i(t-1)}\ln\frac{1}{\delta}\right)$. Substituting ε in Equation (11) yields that with probability at least $1-\delta$ we have

$$\hat{\zeta}_{i,n_i(t-1)} + (\psi^*)^{-1}\left(\frac{1}{n_i(t-1)}\ln\frac{1}{\delta}\right) \geq \zeta_i. \tag{12}$$

Recall that $\delta = t^{-\alpha}$, the criterion for choosing action I_t at time t is

$$I_t = \text{argmax}_i \left\{\hat{\zeta}_{i,n_i(t-1)} + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{n_i(t-1)}\right)\right\}. \tag{13}$$

The following theorem from Lai and Robbins [6], provides an upper bound on the regret of UCB.

Theorem II.1. *(Theorem 2.1 of [1])*
Assume that the reward distributions satisfy Equation (9)*. For $\alpha > 2$, the regret bound of UCB at time τ is*

$$R_\tau \leq \sum_{i,\triangle_i>0}\left(\frac{\alpha\triangle_i}{\psi^*(\triangle_i/2)}\ln\tau + \frac{\alpha}{\alpha-2}\right). \tag{14}$$

In order to see the power of algorithm in capturing the optimal action, a lower bound is derived for the regret. The following theorem states a lower bound for UCB when the distribution of rewards are Bernoulli.

Theorem II.2. *(Theorem 2 of [6])*
For any set of Bernoulli reward distribution and any algorithm that satisfies $E[n_i(\tau)] = o(\tau^b)$ (for any $b > 0$), the following inequality holds

$$\lim_{\tau\longrightarrow+\infty}\inf\frac{R_\tau}{\ln\tau} \geq \sum_{i,\triangle_i>0}\frac{\triangle_i}{kl(\zeta_i,\zeta^*)}. \tag{15}$$

Notice that this theorem is general and holds for any algorithm that satisfies $E[n_i(\tau)] = o(\tau^b)$.

B. Exponential-Weight Algorithm (EXP3)

The exponential-weight algorithm (EXP3) is a randomized algorithm introduced by Auer et al. [7] developed for adversarial bandits. In this algorithm an action is chosen according to a probability distribution and this probability distribution is updated after each round. K weights ($w_i(t)$) are defined, one for each arm and they are initialized to one, i.e.,

$$w_i(0) = 1 \quad \text{for all} \quad i \in K.$$

These weights are used to update the probability distribution.

$$p_i(t) = (1-\eta)\frac{w_i(t)}{W_t} + \frac{\eta}{K}, \tag{16}$$

where $p_i(t)$ is the probability of choosing action i and $\frac{\eta}{K}$ (η is called learning parameter) is a lower bound for the probabilities. In EXP3 an unbiased estimator is used for the rewards, then these estimations are used to update the weights, i.e.,

$$\hat{g}_j(t) = \begin{cases} \frac{g_j(t)}{p_j(t)} & \text{if arm } j \text{ is chosen} \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

the updating rule for $w_i(t)$ becomes

$$w_i(t+1) = w_i(t)e^{\frac{\eta}{K}\cdot\hat{g}_i(t)}. \tag{18}$$

Theorem II.3. *(Theorem 3.1 of [6])*
Consider an EXP3 forecaster with $\eta = \frac{K\ln K}{2\tau}$, then for any bandit the regret bound of EXP3 at time τ is

$$\bar{R}_\tau \leq 2\sqrt{2\tau K\ln K}. \tag{19}$$

C. Bandits With Side Information

Sometimes the player is provided with further information (contexts), and he can use this information to choose the optimal arm. The relation between the contexts and the rewards can have different patterns. For example, the rewards can be a linear function of context (feature vector).

First let us assume that there is no simple relation between the contexts and the rewards. At each round t, a context s_t is shown to the player (the total number of contexts is $|S|$). One way to solve this problem is to run separate EXP3 on each distinct context. This approach leads to the following theorem.

Theorem II.4. *(Theorem 4.1 of [6])*
The following upper bound holds for bandits with side information, when EXP3 is used as the forecaster.

$$\bar{R}^S_\tau \leq \sqrt{2\tau|S|K\ln K}. \tag{20}$$

D. The Expert Case

We now consider the case where the player does not access the contexts, however there is a set of N randomized policies (experts) who access them. In each time step, these experts give a probability distribution for choosing the actions, i.e.,

$$\xi^j_t \text{ for } j = 1,..,N,$$

where $\xi^j_t(i)$ is the probability distribution for choosing action i proposed by expert j at time t. EXP4 is a randomized algorithm for solving this problem.

We define weights for each expert (total number of N weights) and update these weights according to the payoff of selected action g_{I_t}. Like EXP3, these weights are initialized to one, i.e.,

$$w_i(0) = 1 \text{ for } \forall i \in N.$$

The EXP4 algorithm is as follows:
1. Receive expert advice ξ^i_t, where $\xi^i_t(j)$ is the probability of choosing arm j by expert i,

2. Calculate $W_t = \sum_{i=1}^{N} w_i(t)$,
3. Calculate the probability of choosing arm i

$$p_i(t) = (1-\eta)\sum_{j=1}^{N}\frac{w_j(t)\xi_t^j(i)}{W_t} + \frac{\eta}{K}, \tag{21}$$

4. Choose action i according to $p(t)$,
5. Receive a profit for the chosen action $g_{I_t} \in [0,1]$,
6. Set

$$\hat{g}_j(t) = \begin{cases} \frac{g_j(t)}{p_j(t)} & \text{if } j = I_t \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

7. For each expert i set

$$\hat{y}_i(t) = \sum_{j=1}^{K}\xi_t^i(j)\hat{g}_j(t), \tag{23}$$

8. Update the weight of each expert :

$$w_i(t+1) = w_i(t)\cdot e^{\frac{\eta}{K}\hat{y}_i(t)}. \tag{24}$$

Theorem II.5. *(Theorem 4.2 of [1])*
Consider EXP4 forecaster with $\eta = \sqrt{\frac{2\ln N}{\tau K}}$, then for any of experts the regret bound of EXP4 at time τ is

$$\bar{R}_\tau^{ctx} \leq \sqrt{2\tau N \ln K}. \tag{25}$$

E. Stochastic Contextual Bandits

With the stochastic contextual bandits, the policies have a known structure, i.e., each policy is a function f mapping the context s to actions K and the forecaster knows the set of policies (F). This type of bandit can be seen as supervised learning with feature space, where after each round we only have partial information about the outcome (only the action we choose). The contexts s_t and the rewards of actions $g_i(t)$ are i.i.d drawns from a fixed and unknown distribution D. The cumulative reward of the algorithm in this case is compared to the best policy with respect to distribution D, i.e.,

$$\bar{R}_\tau = \tau\cdot g_D(f^*) - \sum_{t=1}^{\tau} g_{I_t}(t), \tag{26}$$

where

$$\begin{aligned} g_D(f) &= \mathbb{E}_{(s,g)\sim D}[g_{f(s)}], \\ f^* &= \arg\sup_{f\in F} g_D(f). \end{aligned} \tag{27}$$

Definition II.1. *Consider the model y that with some parameter θ can classify points $(x_1, x_2, ..., x_n)$ without any error, then VC-dimension is the maximum number of points that can be classified correctly.*

Let the number of actions be 2 ($K = 2$).

Given a set of policies with the VC-dimension d, the VC-dimension with the exponentiation algorithm (VE) proceeds as follows:
1. For the first n rounds, choose actions uniformly at random,
2. Choose a subset of policies $F' \subseteq F$, such that for any $f \in F$ and $t = 1, .., n$ there exists $f' \in F'$ satisfying $f'(s_t) = f(s_t)$,
3. For the remaining time ($t = n+1, .., \tau$) use EXP4 by using policies in F' as experts.

Theorem II.6. *(Theorem 5 of [9])*
For any set of policies with VC-dimension d, the VE forecaster with $n = \sqrt{\tau(2d\ln\frac{e\tau}{d} + \ln\frac{2}{\delta})}$ satisfies the following upper bound for regret,

$$\tau\cdot g_D(f^*) - \sum_{t=1}^{\tau} g_{I_t}(t) \leq c\sqrt{2\tau(d\ln\frac{e\tau}{d} + \ln\frac{2}{\delta})}, \tag{28}$$

for some constant c and with probability at least $1-\delta$ with respect to randomization of rewards and contexts.

III. Learning from Neighbors

In the previous section, we defined the MAB problem for three different bandits and we presented some algorithms for predicting the optimal arm. In this section, we consider network topology, i.e., the players form a network and they have access to the information of their neighbors. In [2] the authors consider learning in networks, however, they assume agents (or players) use myopic algorithm, which means that they do not explore. There are two possible scenarios in this case: The first scenario is that they find the optimal arm and the regret is of order $O(1)$, and the second one is that they do not find the optimal arm (as they do not explore, they can get stuck in a suboptimal arm) and the regret is of order $\Omega(\tau)$ (recall that τ is the number of rounds).

Consider a society with many agents (players), these agents face a similar decision problem, they should choose an action from a set of actions and receive a reward, the distribution of rewards are constant over time, i.e., the payoff of actions are i.i.d random variables (stochastic bandits). Their decision is based on their past experience, along with their neighbors experience, which means they can see the choice of their neighbors and the rewards they get. One of the main assumptions in [2] is considering connected societies, which means for any pair of agents i and j, i is either neighbor with j or there exist agents $i_1, i_2, .., i_m$ such that i is neighbor with i_1 and i_1 is neighbor with i_2 and so on, until i_m is neighbor with j.

In [2], the authors study different network topologies and find some structure that guarantees that learning in the network finally happens, which is choosing the optimal arm. The two main question in their paper are about in which societies complete learning (choosing the optimal arm) is possible and in which societies complete learning occurs with probability 1. However, the rate of convergence of beliefs toward the optimal arm (regret) is not studied.

A. The Model

Let Θ be the set of possible states of the world (the true state is only one of $\theta \in \Theta$), X be the set of possible actions and Y be their outcomes. For each state θ, the reward of action x has a specific distribution and this distribution is known to the players. In other words, the players do not know which state is the true state of the world, but they know for each θ_i what is the distribution of the rewards of actions.

$y \in Y$ is a random variable whose conditional density given x and θ is $\phi(y|x,\theta)$. Furthermore, the reward is a function of

the selected action x and its outcome y, i.e., $r(x, y)$. Agents do not know the true state of the world, however, they have prior beliefs about the state of the world (μ), i.e,

$$\mathbb{D}(\Theta) = \{\mu = \{\mu(\theta)\}_{\theta\in\Theta} | \text{for all } \theta \in \Theta \text{ and } \sum_{\theta\in\Theta} \mu(\theta) = 1\}. \tag{29}$$

Using these beliefs, they choose action; and using the rewards, they update the beliefs. Given belief μ, one period expected utility $u(x, y)$ of taking action x is

$$u(x, \mu) = \sum_{\theta\in\Theta} \mu(\theta) \int_Y r(x, y)\phi(y|x, \theta)\, dy. \tag{30}$$

The players choose the action x, which maximizes the expected utility, i.e.,

$$G(\mu) = \{x \in X | u(x, \mu) \geq u(x', \mu) \text{ for all } x' \in X\}. \tag{31}$$

N is the set of agents and can be finite or infinite. For each agent i, $N(i)$ denotes the set of neighbors of agent i. Note that in this setting $j \in N(i)$ means that i can observe j, but it does not mean $i \in N(j)$ (j can observe i). We denote the set of agents who can observe i by $N^{-1}(i)$.

Let $\mu_{i,t}$ denote the belief of agent i at time t and $C_{i,t} = G_i(\mu_{i,t})$ denote the action of agent i at time t whose outcome is $Z_{i,t}$. Using Bayes rule, the posterior belief of agent i at time t is computed as follows,

$$\mu_{i,t+1}(\theta) = \frac{\Pi_{j\in N(i)}\phi(Z_{j,t}|C_{j,t}, \theta)\mu_{i,t}(\theta)}{\sum_{\theta'\in\Theta} \Pi_{j\in N(i)}\phi(Z_{j,t}|C_{j,t}, \theta')\mu_{i,t}(\theta')}. \tag{32}$$

B. Aggregation of Information

In this section, we mention one of the main theorems of [2]. In a connected society, every player expects the same utility in the long run.
Let Ω be the space containing sequences of realizations of actions of all agents over time (set of all possible sample paths).

Theorem III.1. *There exists $Q \in \Omega$ satisfying $\mathbb{P}_i(Q) = 1$ for all $i \in N$ such that*
if $\omega \in Q$, then for all $i \in N$, $\mu_{i,t}(\omega) \to \mu_{i,\infty}(\omega)$.

Let $X^i(\omega)$ denote the actions taken infinitely often on the sample path ω. Using Theorem III.1, we can establish the following lemma.

Lemma III.2. *Suppose $\omega \in \Omega$.*
a. If $x' \in X^i(\omega)$ then $x' \in argmax_{x\in X} u(x, \mu_{i,\infty}(\omega))$.
b. There exists a real number $U_{i,\infty}(\omega)$ such that $U_{i,t}(\omega) \to U_{i,\infty}(\omega)$, where $U_{i,t}(\omega)$ is the expected utility of chosen action at time t. In addition, if x' satisfies $U_{i,\infty}(\omega) = u(x', \mu_{i,\infty}(\omega))$, then x' is a member of $X^i(\omega)$.

Theorem III.3. *(Theorem 3.2 of [2])*
In a connected society, for all i and j in N, $U_{i,\infty}(\omega) = U_{j,\infty}(\omega)$ holds with probability 1.

This implies that even if only one player eventually learns to choose the optimal action, the rest of the agents will also choose the same optimal action in the long run.

In order to see in which communities agents eventually choose the optimal arm, we start with some definitions. First let θ_1 be the true state of the world and $G(\delta_{\theta_1})$ be the optimal actions when the true state is θ_1.

Definition III.1. *Given a sample path ω, the long run actions of agent i are said to be optimal on ω if $X^i(\omega) \subset G(\delta_{\theta_1})$. Social learning is said to occur if*

$$\mathbb{P}^{\theta_1}(\cap_{i\in N}\{X^i(\omega) \subset G(\delta_{\theta_1})\}) > 0. \tag{33}$$

Social learning is said to be complete if the probability on the left-hand side of Equation (33) is equal to 1 and incomplete if this probability is less than 1 [2].

Definition III.2. *We call an action x full informative if for all $\theta, \theta' \in \Theta$ such that $\theta \neq \theta'$ we have:*

$$\int_Y |\phi(u|x, \theta) - \phi(u|x, \theta')|\ du > 0. \tag{34}$$

We call an action x_u uninformative if $\phi(u|x, \theta)$ is independent of θ [2]. Uninformative actions are one of the reasons of incomplete learning, because once you choose them they do not provide any new information and you get stuck there, unless the information from neighbors overcome it.

Definition III.3. *The distribution of prior beliefs is heterogeneous if for every $\theta \in \Theta$, and for any open neighborhood around θ_1 (the true state of the world), there exists an agent whose prior belief lies in that neighborhood.*

Proposition III.4. *Consider a connected society with heterogeneous prior believes. If there exists a number $L > 0$ such that $\sup_{i\in N} |N(i)| \leq L$, then for any $\lambda \in (0, 1)$, we have*

$$\mathbb{P}^{\theta_1}(\cap_{i\in N}\{X^i(\omega) \subset G(\delta_{\theta_1})\}) \geq \lambda. \tag{35}$$

Now let us study the second question, in which societies complete social learning happens.

To answer this question first, we define term locally independent set. We call two agents i, j locally independent if they do not have any common neighbors, i.e., $N(i) \cap N(j) = \emptyset$.

Lemma III.5. *Fix some agent $i \in N$. For any $\lambda \in (0, 1)$ there exists a set of sample paths A_i satisfying $\mathbb{P}^{\theta_1}(A_i) \geq \lambda$ and $d(\lambda) \in (0, 1)$ such that if $\mu_{i,1} \geq d(\lambda)$ then*

$$\omega \in A_i \Rightarrow X^i(\omega) \subset G(\delta_{\theta_1}). \tag{36}$$

Fix a number $L > 0$ and a $\bar{\lambda} \in (0, 1)$. According to Lemma III.5, $\bar{d} = d(\bar{\lambda})$ exists. Now consider the collection of agents $i \in N$ that satisfy $|N(i)| \leq L$ and $\mu_{i,1}(\theta_1) \geq \bar{d}$. Let $N_{L,\bar{d}}$ be a maximal group of such locally independent agents.

Theorem III.6. *(Theorem 4.1 of [2])*
In a connected society, let $\bar{\lambda} > 0$, $\bar{d} = d(\bar{\lambda})$ and $N_{L,\bar{d}}$ be as defined above. Then

$$\mathbb{P}^{\theta_1}(\cup_{i\in N}\{X^i(\omega) \not\subset G(\delta_{\theta_1})\}) \leq (1 - \bar{\lambda})^{|N_{L,\bar{d}}|}. \tag{37}$$

A special case of the Theorem III.6 is when $|N_{K,\bar{d}}| = \infty$, which yields that complete social learning obtains (i.e., the players choose the optimal arm with the probability 1) [2].

IV. Networked Bandits with Disjoint Linear Payoffs

The paper [3] considers one player and actions in a network: The actions are connected to each other and selecting one action can invoke other actions as well. An example of this framework is advertising in social networks. In this case, we see the users as arms that are going to be chosen for advertising. Friends of a user can see the advertisement on his page. This assumption means that at each round instead of receiving one arm's reward, we receive the selected arm's reward plus its neighbors' rewards. We denote the Network by $g_t = (X_t, E_t)$, where X_t are vertices (arms). The edges E_t show the relation between the arms. We denote the neighbor of arm i at time t by $N_t(i)$ and it contains arm i itself. Note that the topology of network can change over time. The bandits used in their paper are stochastic contextual bandits. It is assumed that the expected payoff of an arm i is linear in the context s_t with coefficient w_i, i.e.,

$$y_{i,t} = s_{i,t}^T w_i + \epsilon_i, \tag{38}$$

where different arms are characterized by different weights w_i.
ϵ_i is conditionally R-sub-Gaussian where $R \leq 0$ is a fixed constant. This definition implies the two following properties for ϵ_i,

$$\begin{aligned} &\mathbb{E}[\epsilon_{i,t}|s_{i,1:t}, \epsilon_{i,1:t-1}] = 0, \\ &\mathbb{VAR}[\epsilon_{i,t}|s_{i,1:t}, \epsilon_{i,1:t-1}] \leq R^2, \end{aligned} \tag{39}$$

where $s_{i,1:t}$ denotes the sequence $s_{i,1}$, $s_{i,2}$,...,$s_{i,t}$ and similarly $\epsilon_{i,1:t-1}$ denotes the sequence $\epsilon_{i,1}$, $\epsilon_{i,2}$,..., $\epsilon_{i,t-1}$.

The idea for solving this problem is like the UCB. We construct probabilistic upper bounds for the arms (confidence sets), then we choose the arm with the highest upper bound.

For each arm i, we define $\hat{w}_i$ as the L^2-regularized least-squares estimate of w_i with regularization parameter λ, i.e.,

$$\hat{w}_i = (S_i^T S_i + \lambda)^{-1} S_i^T Y_i, \tag{40}$$

where S_i is the matrix whose rows are $s_1, .., s_{n_i(t)}$ and $n_i(t)$ is the number of times we choose arm i up to time t. For a positive self-adjoint operator V, let $||s||_V = \sqrt{\langle s, Vs\rangle}$ be the weighted norm of vector s.

Theorem IV.1. *(Theorem 1 of [3])*
According to the 'self-normalized bound for vector-valued martingales', let $V = \lambda I$, $\lambda > 0$, and $\bar{V}_t = V + \sum_{n=1}^{t-1} s_n s_n^T$ be the regularized design matrix underlying the covariates. Define $y_t = s_t^T w + \epsilon_t$ and assume that $||w||_2 \leq \alpha$. Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$, for all $t \geq 1$ we can bound w in such a confidence set:

$$C_t = \Bigg\{ w \in R^d : \\ ||\hat{w}_t - w||_{\bar{V}_t} \leq R\sqrt{2\ln\left(\frac{|\bar{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2}\alpha \Bigg\}. \tag{41}$$

We use the above theorem to derive a bound for $S^T\hat{w}$.

Theorem IV.2. *(Theorem 2 of [3])*
Let $(s_1, y_1), ..., (s_{t-1}, y_{t-1})$, $s_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ satisfy the linear model assumption. Furthermore, we have the same assumption as Theorem IV.1. Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$, for all $t \geq 1$ we have:

$$||s^T\hat{w}_t - s^T w|| \leq \\ ||s||_{\bar{V}_t^{-1}} \left(R\sqrt{2\ln\left(\frac{|\bar{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2}\alpha \right). \tag{42}$$

By using Theorem IV.2, we can derive the following probabilistic upper bound for arm i, note that this upper bound includes arm i's neighbors $N_t(i)$,

$$\sum_{j\in N_t(i)} s_{j,t}^T w_j \leq \sum_{j\in N_t(i)} s_{j,t}^T \hat{w}_j + \\ \sum_{j\in N_t(i)} ||s_{j,t}||_{\bar{V}_t^{-1}} \left(R\sqrt{2\ln\left(\frac{|\bar{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2}K \right). \tag{43}$$

The NetBandit algorithm for predicting the optimal arm is as follows:
1. Compute $\hat{w}_i$ for all i according to Equation (40),
2. Compute the confidence set according to (43),

$$B_{i,t} = \sum_{j\in N_t(i)} s_{j,t}^T \hat{w}_j + \\ \sum_{j\in N_t(i)} ||s_{j,t}||_{\bar{V}_t^{-1}} \left(R\sqrt{2\ln\left(\frac{|\bar{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2}K \right), \tag{44}$$

3. Choose arm $I_t = \text{argmax}_{i\in X_t} B_{i,t}$ (recall that X_t is the set of arms (vertices) at time t).

Theorem IV.3. *(Theorem 3 of [3])*
On the networked bandits, assume that each arm's payoff function satisfies the linear model (38)*, and assume that the contextual vector is $s_{i,t}$ for each arm $i \in X_t$, $|X_t| \leq K$ and t = 1,..., T. Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$, the cumulative regret satisfies*

$$\bar{R}_T \leq 2K\sqrt{2\beta_T(\delta)T\ln\left|I + \frac{SS^T}{\lambda}\right|}, \tag{45}$$

where

$$\beta_T(\delta) = \left(R\sqrt{2\ln\left(\frac{\left|I + \frac{SS^T}{\lambda}\right|^{1/2}}{\delta}\right)} + \lambda^{1/2}\alpha \right). \tag{46}$$

The regret bound depends on the number of invoked arms $|N_i(t)|$, and it is proportional to $\sqrt{T}$ (T is the number of rounds).

V. Conclusion and research plan

First we have defined the multi-armed-bandit problem and presented some algorithms for solving it [1]. Two different types of bandits are used here: stochastic bandits and adversarial bandits. An extension of these bandits is when we

have side information (contextual bandits). The fundamental difference with these bandits is that the more structure they have, the algorithms have a larger cumulative reward for them. For example, the regret of EXP3 for adversarial bandits is $O(\sqrt{KT\ln K})$ and the regret of UCB for stochastic bandits is $O(K\ln T)$ (where K is the number of arms). Then, we have discussed two network topologies. In the first one [2] the players form a network. A myopic algorithm is used by the players to forecast the optimal arm. Myopic means that players care only about the current round and they want to maximize the expected payoff for the current round. However, if we play this game for many rounds, we do not use a myopic algorithm (even in social networks), which means an algorithm with exploration and exploitation is better than a myopic algorithm. The consequence of using a myopic algorithm, in terms of regret, is that it is possible to be mislead into believing that the same arm is still the best option for the rest of the game ($R_T = \Omega(T)$). The main question the authors answer in [2] is whether complete learning is possible. They do not study, however, the rate of convergence toward the optimal arm (regret).
In [3], a new network topology is studied, i.e., the arms form a network and there is only one player. By choosing an arm, the player receives also the rewards of the neighbor of the chosen arm.

We plan to develop new algorithms for MABs in networks. In our problem, we assume that the players form a network containing vertices (agents) and edges (relation between vertices), which means we have a set of selfish agents (players) that form a network and each agent has a similar set of arms. In contrast, in [3] they have one player and a network of arms , and in [2] they have multiple agents using a myopic algorithm. The network of players arises in many scenarios, especially in social networks, where agents use the experience of their neighbors to improve their performance.

For example, we extend the UCB algorithm to the case when the player who uses this algorithm (we name him Player 1) has b neighbors and his neighbors use some arbitrary algorithms. In the decision rule (49), we replace the mean $\hat{\zeta}_i$ with the global mean, i.e., aggregating the information of every player and taking mean of the observed rewards. Let $n'_i(t)$ be the total number of selections of action i by players other than Player 1 and $n^1_i(t)$ be the total number of selections of action i by player 1. Then the total number of observations of the action i by Player 1 is

$$n_i(t) = n^1_i(t) + n'_i(t). \tag{47}$$

We define the following unbiased estimator for ζ_i (mean of rewards of action i)

$$\hat{\zeta}_{i,n_i(t-1)} = \frac{\sum_{j=1}^{n_i(t-1)} g_i(t_j)}{n_i(t-1)}. \tag{48}$$

We assume that the distribution of rewards satisfy Condition (9).
The criterion for choosing action I_t at time t is

$$I_t = \text{argmax}_i \left\{ \hat{\zeta}_{i,n_i(t-1)} + (\psi^*)^{-1}\left(\frac{\alpha \ln t}{n_i(t-1)}\right)\right\}. \tag{49}$$

Theorem V.1. *The regret bound of UCBN with stochastic bandits for $\alpha > 2$ is*

$$\bar{R}_\tau \leq \sum_{i,\triangle_i>0}\left(\max\left\{\max_{t=1,..,\tau}\left\{\frac{\alpha}{\psi^*(\triangle_i/2)}\ln t - n'_i(t)\right\}\triangle_i, 0\right\} + \frac{\alpha}{\alpha-2}\right). \tag{50}$$

Compared to the UCB, this regret means that if at each round we have $n'_i(t)$ free selection of the suboptimal arm i by other players, the number of selections of that specific suboptimal arm (i) would be decreased by $n'_i(t)$.

REFERENCES

[1] S. Bubeck and N. Cesa-Bianchi, "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems," Foundations and Trends in Machine Learning, 2012.
[2] V. Bala and S. Goyal, "Learning from Neighbours," The Review of Economic Studies, 1998.
[3] M. Fang and D. Tao, "Networked Bandits with Disjoint Linear Payoffs," TACM SIGKDD international conference on Knowledge discovery and data mining, 2014.
[4] H. Robbins, "Some aspects of the sequential design of experiments," In Herbert Robbins Selected Papers, 1952.
[5] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the muliti-armed bandit problem," Machine Learning Journal, 2002.
[6] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," Advances in Applied Mathematics, 1985.
[7] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire, "The non-stochastic multi-armed bandit problem," SIAM Journal on Computing, 2002.
[8] Y. Abbasi-Yadkori, C. Szepesvari, and D. Tax, "improved algorithms for linear stochastic bandits," NIPS, 2011.
[9] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. Schapire, "Contextual Bandit Algorithms with Supervised Learning Guarantees," International Conference on Artificial Intelligence and Statistics (AISTATS), 2011.