Multi Armed Bandits and Mechanism Design

Submitted as part of Honours Project (Semester 2)

Kumar Abhishek Sujit Gujar

Contents

1 Abstract

In this report, we will see the concepts of Multi-armed bandit problem (and its variants), Mechanism design and its combination are known as Multi-armed bandit mechanism. Multi-armed bandit mechanism is most useful in settings in which we want to learn hidden parameters while trying to elicit the true valuations of agents which are private information to the user. We will also see some of the applications of MAB mechanisms. In the latter part of the paper, we will see the works done using these concepts through the brief summary of research papers.

2 Multi Armed Bandit - Introduction

Multi-armed bandit problems are the most basic examples of sequential decision problems with an exploration-exploitation trade-off. This is the balance between staying with the option that gave highest payoffs in the past and exploring new options that might give higher payoffs in the future.

[2]There are three fundamental formalizations of the bandit problem depending on the assumed nature of the reward process: stochastic, adversarial, and Markovian.

Let us look at the three formalizations, Chapter 1[2]-

2.1 Intro to Stochastic bandit

In stochastic bandit problem each arm $i = 1, ..., K$ corresponds to an unknown probability distribution v_i on $[0, 1]$ (which can be scaled), and rewards $X_{i,t}$ are independent draws from the distribution v_i corresponding to the selected arm.

The stochastic bandit problem
Known parameters: Number of arms K and (possibly) number of rounds $n \geq K$.
Unknown parameters: K probability distributions $v_1, ..., v_k$ on $[0, 1]$.
For each round $t = 1, 2, ...$

1. The forecaster chooses $I_t \in \{1, ..., K\}$;
2. Given I_t the environment draws the reward $X_{I_t,t} \sim v_{I_t}$ independently from the past and reveals it to the forecaster.

The analysis of the stochastic bandit model was pioneered in the seminal paper of Lai and Robbins [1], who introduced the technique of upper confidence bound (UCB) for asymptotic analysis of regret.

2.2 Intro to Adversarial bandit

In this setting the rewards $X_{i,t}$ of the arms are set by an adversary, or opponent to some arbitrary value $g_{i,t} \in [0, 1]$. If the mechanism of setting the sequence of rewards independent of the forecaster's actions, then we call it a *oblivious* adversary. In general, however, the adversary may adapt to the forecaster's past behavior, in which case we call it *nonoblivious* adversary.

For instance, in the rigged casino, the owner may observe the way a gambler plays in order to design even more evil sequences of rewards. Clearly, the distinction between oblivious and nonoblivious adversary is only meaningful when the player is randomized (if the player is deterministic, then the adversary can pick a bad sequence of gains right at the beginning of the game by simulating the player's future actions).

The study of nonoblivious regret is mainly motivated by the connection between regret minimization and equilibria in games.

The adversarial bandit problem
Known parameters: Number of arms $K \geq 2$ and (possibly) number of rounds $n \geq K$.
For each round $t = 1, 2, ...$

1. The forecaster chooses $I_t \in \{1, ..., K\}$ possibly with the help of external randomization,
2. simultaneously the adversary selects a reward vector $g_t = (g_{1,t}, ..., g_{K,t}) \in [0, 1]^K$, possibly with the help of external randomization, and
3. the forecaster receives (and observes) the reward $g_{I_t,t}$, while the gains of the other arms are not observed.

In this adversarial setting, the goal is to obtain regret bounds in high probability or in expectation with respect to any possible randomization in the strategies used by the forecaster or the opponent, and irrespective of the opponent.

2.3 Intro to Markovian bandit

Markovian bandit assumes that the reward processes are neither $i.i.d$ (like in stochastic bandits) nor adversarial. The arms are associated with K Markov processes, each with its own state space. Each time an arm i is chosen by in-state s, a stochastic reward is drawn from a probability distribution $v_{i,s}$, and the state of arms that are not chosen remains unchanged.

Going back to our initial interpretation of bandits as sequential resource allocation processes, here we may think of K competing projects that are sequentially allocated a unit resource of work. However, unlike the previous bandit models, in this case, the state of a project that gets the resource may change. The optimal policy can be computed via dynamic programming and the problem is essential of computational nature. The seminal result of Gittins[3] provides an optimal greedy policy which can be computed efficiently.

3 Adversarial bandits

In this section, we consider the important variant of the multi-armed bandit problem where no stochastic assumption is made on the generation of rewards, Chapter 3[2].

3.1 Setting

At time step t, $g_{i,t}$ is the reward (or gain) of arm i. Assuming all rewards are bounded, say $g_{i,t} \in [0,1]$. At each time step $t = 1, 2, ...,$, simultaneously with the player's choice of arm $I_t \in \{1, ..., K\}$, an adversary assigns to each arm $i = 1, ..., K$ the reward $g_{i,t}$. Similar to the stochastic setting with the performance of the best arm through the regret

$$R_n = \max_{i=1,...,K} \sum_{t=1}^{n} g_{i,t} - \sum_{t=1}^{n} g_{I_t,t}$$

Sometimes we can consider losses rather than gains. In that case, regret is

$$R_n = \sum_{t=1}^{n} l_{I_t,t} - \min_{i=1,...,K} \sum_{t=1}^{n} l_{i,t}$$

The loss and gain versions are symmetric, in the sense that one can translate the analysis from one to the other setting via the equivalence $l_{i,t} = 1 - g_{i,t}$.

The main goal is to achieve sublinear (in the number of rounds) bounds on the regret uniformly over all possible adversarial assignments of gains to arms.

3.2 Idea

The key idea to solve the problem is to add randomization to the selection of the action I_t to play. By doing so, the forecaster can "surprise" the adversary, and this surprise effect suffices to the regret essentially as low as the regret for the stochastic model.

3.3 Exp3 Algorithm

In order to obtain nontrivial regret guarantees in the adversarial framework, it is necessary to consider randomized forecasters. Below we describe the randomized forecaster Exp3.

Exp3 (Exponential weights for Exploration and Exploitation)
Parameter: a nonincreasing sequence of real numbers $(\eta_t)_{t \in N}$.
Let p_1 be the uniform distribution over $\{1, ..., K\}$.
For each round $t = 1, 2, ..., n$

1. Draw an arm I_t from the probability distribution p_t.
2. For each arm $i = 1, ..., K$ compute the estimated loss $\tilde{l}_{i,t} = \frac{l_{i,t}}{p_{i,t}} \mathbb{1}_{I_t = i}$ and update the estimated cumulative loss $\tilde{L}_{i,t} = \tilde{L}_{i,t-1} + \tilde{l}_{i,s}$.
3. Compute the new probability distribution over arms $p_{t+1} = (p_{1,t+1}, ..., p_{K,t+1})$, where

$$p_{i,t+1} = \frac{exp(-\eta \tilde{L}_{i,t})}{\sum_{k=1}^{K} exp(-\eta \tilde{L}_{k,t})}$$

The randomized forecaster Exp3, works on two fundamental ideas,

- First, despite the fact that only the loss of the played arm is observed, with a simple trick it is still possible to build an unbiased estimator for the loss of any other arm. Namely, if the next arm I_t to be played is drawn from a probability distribution $p_t = (p_{1,t}, ..., p_{K,t})$, then

  $$\tilde{l}_{i,t} = \frac{l_{i,t}}{p_{i,t}} \mathbb{1}_{I_t = i}$$

  is an unbiased estimator (with respect to the draw of I_t) of $l_{i,t}$, as we have

  $$\mathbb{E}_{I_t \quad p_t} = \sum_{j=1}^{K} p_{j,t} \frac{l_{i,t}}{p_{i,t}} \mathbb{1}_{I_t = i} \Rightarrow l_{i,t}$$

- The second idea is to use an exponential reweighting of the cumulative estimated losses to define the probability distribution p_t from which the forecaster will select the arm I_t. Exponential weighting schemes are a standard tool in the study of sequential prediction schemes under adversarial assumptions.

3.4 Exp3.P Algorithm

We want high probability bound on the regret for which, Exp3 strategy defined in previous section is not adequate as the variance of the estimate $\tilde{l}_{i,t}$ is of order $\frac{1}{p_{i,t}}$, which can be arbitrarily large. Exp3.P algorithm does not face such problem.

Exp3.P
Parameters: $\eta \in \mathbb{R}^+$ and $\gamma, \beta \in [0,1]$.
Let p_1 be the uniform distribution over $\{1, ..., K\}$.
For each round $t = 1, 2, ..., n$

1. Draw an arm I_t from the probability distribution p_t.

2. Compute the estimated gain for each arm:

$$\tilde{g}_{i,t} = \frac{g_{i,t} \mathbb{1}_{I_t = i} + \beta}{p_{i,t}}$$

and update the estimated cumulative gain: $\tilde{G}_{i,t} = \sum_{s=1}^{t} \tilde{g}_{i,s}$.

3. Compute the new probability distribution over the arms $p_{t+1} = (p_{1,t+1}, ..., p_{K,t+1})$ where:

$$p_{i,t+1} = (1 - \gamma) \frac{exp(\eta \tilde{G}_{i,t})}{\sum_{k=1}^{K} exp(\eta \tilde{G}_{i,t})} + \frac{\gamma}{K}$$

4 Contextual bandits

4.1 Introduction

In most real-life applications, we have access to information that can be used to make a better decision when choosing amongst all actions in a MAB setting, this extra information is what gives Contextual Bandits their name. Contextual bandits is a natural extension of the multi-armed problem in which each arm is associated with some side information. Based on this side information, or context, a notion of "contextual regret" is introduced where optimality is defined with respect to the best policy (i.e., mapping from contexts to arms) rather than the best arm. A different viewpoint is obtained when the contexts are privately accessed by the policies (which are then called "experts"). In this case the contextual information is hidden from the forecaster, and arms must be chosen based only on the past estimated performance of the experts, Chapter 4[2].

4.2 Applications

- In personalized news article recommendation the task is to select, from a pool of candidates, a news article to display whenever a new user visits a website. The articles correspond to arms, and a reward is obtained whenever the user clicks on the selected article. Side information, in the form of features, can be extracted from both user and articles. For the user this may include historical activities, demographic information, and geolocation; for the articles, we may have content information and categories.

- In selecting the treatment for the patient from a possible number of treatments, we can use history of health conditions of the patient as a context to decide the best suited treatment for the patient.

4.3 Bandits with Side Information

The most basic example of contextual bandits is obtained when game rounds $t = 1, 2, ...$ are marked by contexts $s_1, s_2, ...$ from a given context set S. The forecaster must learn the best mapping $g : S \to 1, ..., K$ of contexts to arms. In this setting we are taking the case of adversarial rewards and we assume that the sequence of contexts s_t is arbitrary but fixed.

4.3.1 S-Exp3 Algorithm

One of the simplest approach to solve the above problem is to run a separate instance of Exp3 on each distinct context, which is gives the following notion of pseudo-regret,

$$\overline{R}_n^S = \max_{g: S \to \{1,...,K\}} \mathbb{E}\left[\sum_{t=1}^{n} l_{I_t,t} - \sum_{t=1}^{n} l_{g(s_t),t}\right]$$

Here $s_t \in S$ denotes the context marked the $t-$th game round.

4.4 The Expert Case

In this case we are considering contextual variant of the basic adversarial bandit model. In this variant, there is a finite set of N randomized policies. Following the setting of prediction with expert advice, no assumptions are made on the way policies compute their randomized predictions, and the forecaster experiences the contexts only through the advice provided by the policies. For this reason, in what follows we use the word expert to denote a policy. Although calling this a model of contextual bandits may sound little strange, as the structure of contexts does not seem to play a role here, however the bandit with experts have been used in practical contextual bandit problem, e.g., the news recommendation experiment[4].

4.4.1 Model

At each step $t = 1, 2, ...$ the forecaster obtains the expert advice $(\xi_t^1, ..., \xi_t^N)$ where each ξ_t^j is a probability distribution over arms representing the randomized play of expert j at time t. If $l_t = (l_{1,t}, ..., l_{K,t}) \in [0,1]^K$ is the vector of losses incurred by the K arms at time t, $\mathbb{E}_{i \sim \xi_t^j} l_{i,t}$ denotes the expected loss of expert j at time t. The expert advice depends on the realization of the forecaster's past random plays. The pseudo-regret $\overline{R}_n^{ctx}$ for the adversarial contextual bandit problem,

$$\overline{R}_n^{ctx} = \max_{i=1,\dots,N} \mathbb{E}\left[\sum_{t=1}^{n} l_{I_t,t} - \sum_{t=1}^{n} \mathbb{E}_{k\sim\xi_t^i} l_{k,t}\right]$$

4.4.2 Exp4 Algorithm

Exp4 is a simple adaptation of Exp3 to the contextual setting. Exp4 runs Exp3 over the N experts using estimates of the experts losses $\mathbb{E}_{i\sim\xi_t^j} l_{i,t}$. In order to draw arms, Exp4 mixes the expert advice with the probability distribution over experts maintained by Exp3.

Exp4 (Exponential weights algorithm for Exploration and Exploitation with Experts) without mixing:
Parameter: a nonincreasing sequence of real numbers $(\eta_t)_{t\in N}$
Let q_1 be the uniform distribution over $\{1, ..., N\}$.
For each round $t = 1, 2, ..., n$

1. Get expert advice $\xi_t^1, ..., \xi_t^N$, where each ξ_t^j is a probability distribution over arms.
2. Draw an arm I_t from the probability distribution $p_t = (p_{1,t}, ..., p_{K,t})$, where $p_{i,t} = \mathbb{E}_{j\sim q_t}\xi^j_{i,t}$.
3. Compute the estimated loss for each arm
$$\tilde{l}_{i,t} = \frac{l_{i,t}}{p_{i,t}} \mathbb{1}_{I_t=i} \qquad i = 1, ..., K$$
4. Compute the estimated loss for each expert
$$\tilde{y}_{j,t} = \mathbb{E}_{i\sim\xi_t^i} \tilde{l}_{i,t} \qquad j = 1, ..., N$$
5. Update the estimated cumulative loss for each expert $\tilde{Y}_{j,t} = \sum_{s=1}^{t} \tilde{y}_{j,s}$ for $j = 1, ..., N$.
6. Compute the new probability distribution over the experts $q_{t+1} = (q_{1,t+1}, ..., q_{N,t+1})$, where
$$q_{j,t+1} = \frac{exp(-\eta_t \tilde{Y}_{j,t})}{\sum_{k=1}^{N} exp(-\eta_t \tilde{Y}_{k,t})}$$

4.5 Competing Against the Best Context Set

4.5.1 Introduction

In 4.1 we considered the basic contextual scenario where the goal was to compete against the best mapping from contexts to arms. Now let us consider a class $\{S_\theta : \theta \in \Theta\}$ of context sets. In this new setting, each time step $t = 1, 2, ..$ is marked by the vector $(s_{\theta,t})_{\theta\in\Theta}$ of contexts, one for each set in Θ. Pseudoregret in this case is

$$\overline{R}_n^{\Theta} = \max_{\theta\in\Theta} \max_{g:S_\theta\to\{1,\dots,K\}} \mathbb{E}\left[\sum_{t=1}^{n} l_{I_t,t} - \sum_{t=1}^{n} l_{g(s_\theta,t),t}\right]$$

When $|\theta| = 1$ we get the contextual pseudoregret $\overline{R}_n^S$ defined in 4.3.1. In general when Θ contains more than one set, the forecaster must learn both the best set S_θ and the best function $g : S_\theta \to \{1, ..., K\}$ from that set to the set of arms.

4.5.2 Solution

The solution of this variant of contextual bandits involves a nontrivial combination of two of the main algorithms *Exp*4 and S-*Exp*3. In particular, we consider a scenario in which *Exp*4 uses instances of S-*Exp*3 as experts. The idea is to use *Exp*4 over the class Θ of "experts" and combine this with the S-*Exp*3 algorithm.

Intuitively, Exp4 provides competitiveness against the best context S_θ, while the instances of the $S - Exp3$ algorithm, acting as experts for *Exp*4, ensure that we are competitive against the best function $g : S_\theta \to \{1, .., K\}$ for each $\theta \in \Theta$.

4.6 Stochastic Contextual Bandits

4.6.1 Introduction

In this we will consider the case in which policies have a known structure. More specifically, each policy is a function f mapping the context space to the arm space $\{1, ..., K\}$ and the set $\mathcal{F}$ of policies is given as an input parameter to the forecaster. Under this assumption on the policies, the problem can be viewed as a bandit variant of supervised learning.

4.6.2 Setting

Here we will follow the standard notation of the supervised learning and hence will use x rather than s to denote contexts. In supervised learning, we observe data of the form (x_t, l_t). In the contextual bandit setting, the observed data are $(x_t, l_{I_t,t})$, where I_t is the arm chosen by the forecaster at time t given context $x_t \in \chi$.

In the stochastic variant of contextual bandits, contexts x_t and arm losses $l_t = (l_{1,t}, ..., l_{K,t})$ are realizations of i.i.d. draws from a fixed and unknown distribution D over $\chi \times [0, 1]^K$. In tight analogy with statistical risk $l_D(f) = \mathbb{E}_{(x,l)\sim D} l_{f(x)}$. Let

$$f^* = \arg\inf_{f\in\mathcal{F}} l_D(f)$$

the risk-minimizing policy in the class. The regret with respect to the class $\mathcal{F}$ of a forecaster choosing arms

$I_1, I_2...$ is then defined by

$$\sum_{t=1}^{n} l_{I_t,t} - nl_D(f^*)$$

This can be viewed as the stochastic counterpart of the adversarial contextual regret $\overline{R}_n^{ctx}$ introduced in 4.4.1.

4.6.3 VE Algorithm

For the case $K = 2$ arms and classes $\mathcal{F}$ of policies parameterized as $f : \chi \to \{1, 2\}$ by their VC-dimension d. For this setting, we consider the following forecaster.

```
VE (VC dimension by Exponentiation):
Parameters: number $n$ of rounds, $n^{'}$ satisfying $1 \leq n^{'} \leq n$.

  1. For the first $n^{'}$ rounds, choose arms uniformly at random.

  2. Build $\mathcal{F}^{'} \subseteq \mathcal{F}$ such that for any $f \in \mathcal{F}$ there
     is exactly $f^{'} \in \mathcal{F}^{'}$ satisfying $f(x_t) = f^{'}(x_t)$
     for all $t = 1, ..., n^{'}$.

  3. For $t = n^{'} + 1, ..., n$ play by simulating
     $Exp4.P$ [4] using policies of $\mathcal{F}^{'}$ as experts.
```

4.7 Thompson Sampling

In this section we will see the solution proposed in the paper [6] in the setting of contextual bandit with linear payoffs.

Thompson Sampling is one of the oldest heuristics for multi-armed bandit problems. It is a *randomized algorithm* based on Bayesian ideas, and has recently generated significant interest after several studies demonstrated it to have better empirical performance compared to the state-of-the-art methods. In [6], the authors have designed and analyzed the generalization of Thompson Sampling algorithm for the stochastic contextual multi-armed bandit problem with linear payoff functions, when the contexts are provided by an *adaptive adversary.*

4.7.1 Setting

In the contextual bandits setting with linear payoff functions, the learner competes with the class of all ”linear” predictors on the feature vectors. That is a predictor is defined by a d-dimensional parameter $\overline{\mu} \in \mathbb{R}^d$,, context is defined as b_i which is also d-dimensional vector associated with each arm i which is available to predictor before making a choice of which arm to play, and the predictor ranks the arms according to $b_i^T \overline{\mu}$. In this paper they have considered stochastic contextual bandit problem under linear realizability assumption, that is, we assume that there is an unknown underlying parameter $\mu \in \mathbb{R}^d$ such that the expected reward for each arm i, given context b_i, is $b_i^T \mu$. Under this realizability assumption, the linear predictor corresponding to μ is in fact the best predictor and the learner's aim is to learn this underlying parameter.

4.7.2 Algorithm

The basic idea in Thompson sampling algorithm is to assume a simple prior distribution on the underlying parameters of the reward distribution of every arm, and at every time step, play an arm according to its posterior probability of being the best arm. Following is the general structure of TS for the contextual bandits problem:

- a set Θ of parameters $\tilde{\mu}$;
- a prior distribution $P(\tilde{\mu})$ on these parameters;
- past observation $\mathbb{D}$ consisting of (context b, reward r) for the past time steps;
- a likelihood function $P(r|b, \tilde{\mu})$, which gives the probability of reward given a context b and a parameter $\tilde{\mu}$;
- a posterior distribution $P(\tilde{\mu}|\mathbb{D}) \propto P(\mathbb{D}|\tilde{\mu})P(\tilde{\mu})$, where $P(\mathbb{D}|\tilde{\mu})$ is the likelihood function.

In each round, TS plays an arm according to its posterior probability of having the best parameter.

```
Thompson Sampling for Contextual bandits
Parameters:$B = I_d, \hat{\mu} = 0_d, f = 0_d$
For each round $t = 1, 2, ...,$

  1. Sample $\tilde{\mu}(t)$ from distribution $\mathcal{N}(\hat{\mu}, v^2 B^{-1})$

  2. Play arm $a(t) := \arg\max_i b_i(t)^T \tilde{\mu}(t)$, and observe reward $r_t$.

  3. Update $B = B + b_{a(t)}(t) b_{a(t)}(t)^T, f = f + b_{a(t)}(t) r_t, \hat{\mu} = B^{-1} f$
```

where $v = R\sqrt{\frac{24}{\epsilon} d \ln(\frac{1}{\delta})}$, with $\epsilon, \delta \in (0, 1)$ and $R \geq 0$.

5 Stochastic MAB Mechanisms

5.1 MAB Mechanism Design Environment

[5] Stochastic MAB mechanisms capture the interplay between the online learning and strategic bidding. Following is the setting for the MAB mechanisms:

- There are K agents with $\mathcal{K} = \{1, 2, ..., K\}$ and they are rational and intelligent.

- Each agent i privately observes his valuation θ_i, that accounts for his preference. The value of θ_i is known to agent i and is not known to the other agents as well as the forecaster.

- The set of private values of agent i is denoted by Θ_i. The set of all type profiles is given by $\Theta = \Theta_1 \times ... \times \Theta_K$. a typical type profile is represented as $\theta = (\theta_1, ..., \theta_k)$.

- Each agent i is also parametrized by the parameter $\rho_i \in \Upsilon$. Each agent i has a certain stochastic reward associated with him that comes from the distribution v_{ρ_i} and has expectation $\mu_i \in \mathbb{R}$.

- The parameters, ρ_i and μ_i are unknown to the agents and to the forecaster, and hence need to be learnt over time.

- In order to learn the reward expectations μ_i, mechanism design problem is repeated over time. These time instances are denoted by $t \in \{1, 2, ..., T\}$.

- At any time t, the mechanism consists of a tuple ($a^t \in \mathcal{A}$, $p^t \in \mathcal{P}$) and the mechanism design problem involves finding the allocation rule ($a^t \in \mathcal{A}$ and the payment rule $p^t \in \mathcal{P}$) by eliciting the private valuations θ_i from the agents and by observing the rewards obtained so far. Let us denote set of all possible mechanisms by χ.

- At any time t, allocation rule $a^t = \{a_1^t, a_2^t, ..., a_k^t\}$ denotes whether an agent i is allocated at time t or not i.e. $a_i^t \in 0, 1 \; \forall t\{1, 2, ..., T\}$. If agent i is allocated at time t then $a_i^t = 1$ and is 0 otherwise. Note that an agent corresponds to an arm of multiarmed bandit and allocation rule a provides an arm pulling strategy where $a_i^t = 1$ implies pulling an arm i at time t.

- Given an allocation rule a^t, $X_i(t) \sim v_{\rho_i} \in \mathbb{R}$ denotes the stochastic reward obtained from agent i at time t. Note that rewards are observed only for those agents who are allocated i.e. whose $a_i^t = 1$. We assume that $X_i(t) = 0$ if agent is not allocated at time t i.e. $a_i^t = 0$.

- Utility function instance t is given by $u_i^t : \chi \times \Theta_i \times \mathbb{R} \rightarrow \mathbb{R}$. given $x^t \in \chi$ and $\theta_i^t \in \Theta_i$, $X_i(t) \sim v_{\rho_i} \in \mathbb{R}$, the value $u_i^t(x^t, \theta_i^t, X_i(t))$ denotes the payoff that agent i, having type $\theta_i \in \Theta_i$, reward $X_i(t)$, receives from an outcome $x^t \in \chi$.

- The agents might be strategic and may not report their true valuations θ_i^t to the mechanism designer so as to increase their utilities.

- Total expected utility of player i is denoted by $u_i = \mathbb{E}_{X_i(t) \sim v_{\rho_i}}[\sum_{t=1}^{T}(u_i^t(x^t, \theta_i^t, X_i(t)))]$

- In some settings private valuations θ_i^t can change over time and in some cases agents are asked to report their private values only once.

5.2 MAB Mechanisms: Key Notions

When learning is not involved, allocation and payment depend only on the bids provided by the agents. However, when there is learning involved, allocation and payment functions at any round t also depend on how the rewards or success are observed in the previous allocations. Lets take an example of Single Slot Sponsored Search Auction[5] (SSA) which is also known as pay-per-click auction. Let $s \in \{0, 1\}^{K \times T}$ be the reward realization matrix in the case of SSA and bids by each advertiser i is b_i. (Note: All the other symbols means same as defined in previous subsection)

5.2.1 Ex-Post Monotone Allocation Rule

We say that an allocation rule "a" is ex-post monotone if allocation rule "a" is monotone for every reward realization i.e., $\forall i \in \mathcal{K}$, $\forall b_i \geq b_i^-$,

$$a_i(b_i, b_{-i}; s) \geq a_i(b_i^-, b_{-i}; s),$$

$$\forall b_{-i} \in \Theta_{-i}, s \in \{0, 1\}^{K \times T}.$$

Here, $a_i(b_i, b_{-i}; s) = \sum_{t=1}^{T} a_i^t(b_i, b_{-i}; s)$
Note that the allocation at any time t can only depend on the reward realization observed till time $t - 1$.

5.2.2 Stochastic Monotone Allocation Rule

We say that an allocation rule "a" is stochastic monotone if it is monotone in expectation with respect to reward realizations,

$$\mathbb{E}_s[a_i(b_i, b_{-i}; s)] \geq \mathbb{E}_s[a_i(b_i^-, b_{-i}; s)],$$

$$\forall b_i \geq b_i^-, \forall b_{-i} \in \Theta_{-i}.$$

5.2.3 Ex-Post Incentive Compatible Mechanism

We say that a mechanism is ex-post incentive compatible if all the bidders are truthful for every reward realization irrespective of the bids of other workers,

$$u_i(a_i(\theta_i, b_{-i}; s), p_i(\theta_i, b_{-i}; s), \theta_i; s) \geq u_i(a_i(b_i, b_{-i}; s), p_i(b_i, b_{-i}; s), \theta_i; s)$$

$$\forall \theta_i \in \Theta_i, b_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0, 1\}^{K \times T}$$

5.2.4 Stochastic Incentive Compatible Mechanism

We say that a mechanism is stochastic incentive compatible if all the bidders are truthful in expectation with respect to reward realization irrespective of the bids of other workers, bids of other workers,

$$\mathbb{E}_s[u_i(a_i(\theta_i, b_{-i}; s), p_i(\theta_i, b_{-i}; s), \theta_i; s)] \geq \mathbb{E}_s[u_i(a_i(b_i, b_{-i}; s), p_i(b_i, b_{-i}; s), \theta_i; s)]$$

$$\forall \theta_i \in \Theta_i, b_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0,1\}^{K \times T}$$

5.2.5 Ex-Post Individual Rational Mechanism

A mechanism $M = (a, p)$ is said to be ex-post individually rational if participating in the mechanism always gives any advertiser non-negative utility. That is, $\forall i \in \mathcal{K}$,

$$u_i(a_i(\theta_i, b_{-i}; s), p_i(\theta_i, b_{-i}; s), \theta_i; s) \geq 0,$$

$$\forall \theta_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0,1\}^{K \times T}$$

6 Research Paper Summaries

6.1 Research Paper 1

A Contextual-Bandit Approach to Personalized News Article Recommendation.
Authors : *Lihong Li, John Langford, Robert E. Schapire.*[7]

- *What is the problem addressed?*
  In the paper the authors have tried to model personalized recommendation of news articles as a contextual bandit problem, a principled approach in which a learning algorithm sequentially selects articles to serve users based on contextual information about the users and articles, while simultaneously adapting its article-selection strategy based on user-click feedback to maximize total user clicks(also known as click through rate). That is it addresses the challenge of identifying the most appropriate web-based content at the best time for individual users.

- *Challenges:*
  Following are the main challenges in solving the problem:

  - First, the content of web-service repository changes dynamically, undergoing frequent insertions and deletions. So it is crucial to quickly identify interesting content for users.
  - Challenges involved in the process of gathering and storing user attributes, managing content assets, and, based on an analysis of current and past users' behavior, delivering the individually best content to the present user being served.
  - Main challenge is to how to use features of users and content to deliver the best content to the present user in efficient way in terms of getting maximum clicks from the users in small duration of time.

- *What is the solution proposed?*
  In the paper they have shown taht a confidence interval can be computed efficiently in closed form when the payoff model is linear and called this algorithm LinUCB, as given some parametric form of payoff function, a number of methods exist to estimate from data the confidence interval of the parameters with which we can compute a UCB of the estimated arm payoff.
  Following is the overview of how a contextual bandit algorithm proceeds in discrete trials $t = 1, 2, 3...$ In trial t:

  - The algorithm observes the current user u_t and a set A_t of arms or actions together with their feature vectors $x_{t,a}$ for each $a \in A_t$. The vector $x_{t,a}$ summarizes information of both the user u_t and arm a_t and will be referred as context.
  - Based on observed payoffs in previous trials, A chooses an arm $a_t \in A_t$, and receives payoff r_{t,a_t} whose expectation depends on both the user u_t and the arm a_t.
  - The algorithm then improves its arm-selection strategy with the new observation, $(x_{t,a_t}, a_t, r_{t,a_t})$. It is important to emphasize here that no feedback (namely, the payoff $t_{t,a}$) is observed for unchosen arms $a \neq a_t$.

  The authors have proposed two variants of algorithms LinUCB with Disjoint Linear Models and LinUCB with Hybrid Linear Models.
  Second, they have addressed the problem of offline evaluation showing this is possible for any explore/exploit strategy when interactions are independent and identically distributed as might be a reasonable assumption for different users.
  Third using offline evaluation method they successfully applied our new algorithm to a Yahoo! Front Page Today Module dataset containing over 33 million events. Results showed a 12.5% click lift compared to a standard context-free bandit algorithm.

- *Claims of the author?*

  - First, its computational complexity is linear in the number of arms and at most cubic in the number of features.
  - Second, the algorithm works well for a dynamic arm set, and remains efficient as long as the size of A_t is not too large. This case is true in many applications. In news article recommendation, for instance, editors add/remove articles to/from a pool and the pool size remains essentially constant.
  - Third, if the arm set At is fixed and contains K arms, then the confidence interval decreases fast enough with more and more data, and then prove the strong regret bound of $\tilde{O}(\sqrt{KdT})$.

```
LinUCB Algorithm
Inputs: $\alpha \in \mathbb{R}_+$ for $t = 1, 2, 3, ..., T$ do
  Observe features of all arms $a \in \mathbb{A}_t$ :
  $x_{t,a} \in \mathbb{R}^d$
  for all $a \in \mathbb{A}_t$ do
    if $a$ is new then
      $A_a \leftarrow I_d$
      $b_a \leftarrow 0_{d \times 1}$
    end if
    $\hat{\theta}_a \leftarrow A_a^{-1} b_a$
    $p_{t,a} \leftarrow \hat{\theta}_a^T x_{t,a} + \alpha \sqrt{x_{t,a}^T A_a^{-1} x_{t,a}}$
  end for  Choose arm $a_t = \arg\max_{a \in \mathbb{A}_t} p_{t,a}$ with ties
  broken arbitrarily, and observe a real-valued
  payoff $r_t$
  $A_{a_t} \leftarrow A_{a_t} + x_{t,a_t} x_{t,a_t}^T$
  $b_{a_t} \leftarrow b_{a_t} + r_t x_{t,a_t}$
end for
```

6.2 Research Paper 2

Knapsack Based Optimal Policies for Budget–Limited Multi-Armed Bandits
Authors : *Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R. Jennings*[9]

- *What is the problem addressed?*
  In budget–limited multi-armed bandit (MAB) problems, the learner's actions are costly and constrained by a fixed budget. Consequently, an optimal exploitation policy may not be to pull the optimal arm repeatedly, as is the case in other variants of MAB, but rather to pull the sequence of different arms that maximizes the agent's total reward within the budget.

- *What is the solution proposed?*
  They have proposed two new pulling policies, namely:

  - KUBE (Knapsack-based upper confidence bound exploration and exploitation)
  - Fractional KUBE

  In both learning algorithms, they do not explicitly separate exploration from exploitation. Instead, they explore and exploit at the same time by adaptively choosing which arm to pull next, based on the current estimates of the arms' rewards.
  Following are the steps KUBE algorithm as follows:

  - At each time step, KUBE calculates the best set of arms that provide the highest total upper confidence bound of the estimated expected reward, and still fits the residual budget, using **unbounded knapsack model** to determine the best set.
  - **Unbounded knapsack model** is known to be an NP-hard problem, hence uses efficient approximation method called *density-ordered greedy* approach.
  - KUBE then uses the **frequency** that each arm occurs within the approximated best set as a *probability* with which to randomly choose an arm to pull in the next time step.
  - The reward that is received is then used to update the estimate of the upper confidence bound of the pulled arm's expected reward, and the unbounded knapsack problem is solved again.

  In fractional KUBE algorithm the difference from above algorithm is that instead using the *density-ordered greedy* to solve the underlying bounded knapsack problem, fractional KUBE relies on a computationally less expensive approach, namely the *fractional relaxation-based algorithm.*

  Fractional KUBE algorithm It also approximates the underlying unbounded knapsack problem at each time step t in order to determine the frequency of arms within the estimated best set of arms. In fractional relaxation-based algorithm solely chooses the arm with highest estimated confidence bound-cost ratio(i.e. the item type) with the highest estimated confidence bound-cost ratio(i.e. item density), fractional KUBE does not need to randomly choose an arm.

- *Implication in the result:*
  One of the implications of the numerical results is that although fractional KUBE has a better bound on its performance regret than KUBE, the latter typically outperforms the former in practice.

6.3 Research Paper 3

Efficient Crowdsourcing of Unknown Experts using Multi-Armed Bandits
Authors : *Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings.*[8]

- *What is the problem addressed?*
  The problem addressed is an expert crowdsourcing problem, where the crowdsource work requires an effort by experts of the particular field, for example in development and testing of large software applications. The problem is an expert crowdsourcing problem as a *budget-limited* MAB with additional constraints that workers can only complete(bounded) number of tasks. MAB setting is used to learn the rewards as workers produce results of varying quality, the utility of each assigned task is unknown and can vary with the workers and tasks. So, the problem sums up as to assign tasks within a limited bud-

get to a set of workers such that its total utility is maximized.

- *Challenges:*
  - First, the quality of a completed task can vary greatly both between workers and even subsequent tasks completed by the same worker.
  - Second, there is typically little or no prior knowledge about the expected quality of a worker, as the online labor market is inherently open and dynamic in nature with little historical information about worker performance.
  - Third, experts often demand widely varying prices for their services. This can be due to differences in skill level but is similarly influenced by individual expectations, local wages and the cost of living in the worker's country of residence.
- *What is the solution proposed?*
  They have proposed a novel algorithm called *bounded $\epsilon-$ first*, that efficiently tackles the bounded MAB as follows:
  - To deal with the unknown performance characteristics of workers, our algorithm divides its budget into two amounts (as dictated by a ϵ parameter) to be used in two sequential phases — an initial exploration phase and exploitation phase.
  - In exploration phase, it uniformly samples the performance of a wide range of workers using the first part of its budget.
  - In exploitation phase, it selects only the best workers using its remaining budget. In the latter, the algorithm chooses the best set of workers by solving a *bounded knapsack* problem.
- *Claims of the author?*
  The author claims that the *performance regret* (i.e. the difference between the performance of a particular algorithm and that of the theoretical optimal solution) of the bounded $\epsilon-$first approach is at most O($B^{\frac{2}{3}}$) with high probability, where B is the total budget. The sub-linear theoretical bound implies that the algorithm has zero-regret property.

  The author claims that their approach outperforms current crowdsourcing techniques by up to 155%, and achieves 75% of the optimal.
- *Drawbacks of the algorithm:*
  Mentioned in [9]. Following are the drawbacks budget-limited $\epsilon-$first method suffers:
  - The performance of $\epsilon-$first approaches depend on the value of ϵ chosen. Finding a suitable ϵ for a particular problem instance is a challenge since settings with different budget limits or arm costs (which are not known beforehand) will typically require different values of ϵ.
  - The regret bound that $\epsilon-$first provides is O($B^{\frac{2}{3}}$), where B is the budget limit, whereas theoretical best possible regret is typically a logarithmic function of the number of pulls.
- *Terminology*
  - Bounded knapsack problem: Given N types of items, each type i has a corresponding value v_i, and weight w_i. In addition, there is also knapsack with the weight capacity of C. The bounded knapsack problem selects integer units of those types that maximize the total value of items in the knapsack, such that the total weight of the items does not exceed the knapsack weight capacity.
  - Zero-regret property: In above case, we have sub-linear regret then we will say the algorithm has the zero-regret property, that as B is increased, the average regret (i.e. the performance regret divided by the total budget) tends to 0. This property is a key measure of efficiency within the bandit literature. Indeed, the zero–regret property guarantees that our algorithm asymptotically converges to the optimal solution with probability 1 as B tends to infinity.

6.4 Research Paper 4

Online Decision Making in Crowdsourcing Markets: Theoretical Challenges
Authors : *Aleksandrs Slivkins, Jennifer Wortman Vaughan*[10]

- *Introduction*
  The paper offers a detailed reflection on the modeling issues that inhibit theoretical research on the repeated decision making in crowdsourcing. This paper does not gives us a single unified model but gives us a multitude of modeling choices that exist.

  Despite the vast scope of the existing works, crowdsourcing brings an array of domain-specific challenges that require novel solutions. To address these challenges in a principled way, one would like to formulate a unified collection of well-defined algorithmic questions with well-specified objectives, allowing researchers to propose novel solutions and techniques that can be easily compared, leading to a deeper understanding of the underlying issues. However, it appears very difficult to capture all of the relevant aspects of crowdsourcing in a coherent model.

As a result, many of the existing theoretical papers on crowdsourcing propose their own new models. This makes it difficult to compare techniques across papers, and leads to uncertainty about which parameters or features matter most when designing new platforms or algorithms.
Following are the reasons for why developing a unified model of crowdsourcing is difficult:

  - There is a tension between the desire to study existing platforms like Mechanical Turk or to look ahead and look ahead and model alternate platforms with novel features that could potentially lead to improved crowdsourcing markets. A plethora of different platform designs are possible, and different designs lead to both different models and different algorithmic questions.
  - Second, even after a model of the platform has been determined, one must take into account the diversity of tasks and workers that use the platform.
  - Third, crowdsourcing workers are human beings, they may act strategically, maximizing their own welfare in response to the incentives provided by the requesters and the platform.

- *Informal Problem*
  We can see the problem through 3 point of views, i.e, there are 3 parties: *workers*, *requesters*, and the *crowdsourcing platform*. Following are some features of the problem:

  - Overtime requesters submit tasks to the platform and workers get matches with requesters, perform tasks, and receive payments.
  - Different workers have varying level of expertise in different tasks.
  - All parties make repeated decisions over time and can learn over time which can improve the decision making.
  - All decision makers receive the partial feedback they observe the consequences of their decisions.
  - Workers and requesters can behave strategically, so their incentives need to be taken into account.

  The problem is to design algorithms for decision making, on behalf of the platform, the requesters, or perhaps even the workers.

6.5 Research Paper 5

A Truthful Budget Feasible Multi-Armed Bandit Mechanism for Crowdsourcing Time-Critical Tasks

Authors : *Arpita Biswas, Shweta Jain, Debmalya Mandal, Y. Narahari.*[12]

- *What is the problem addressed?*
  The paper address the problem faced by service requesters on modern crowdsourcing platforms. Following are the features of the problem that this paper address:

  - The requester wishes to crowdsource a number of tasks but has a fixed budget which leads to a trade-off between cost and quality while allocating tasks to workers.
  - Each task has a fixed deadline and a worker who is allocated a task is not available until the deadline.
  - The qualities (probability of completing a task successfully within deadline) of crowd workers are not known.
  - The crowd workers are strategic about their costs.

  The aim of the paper is to maximize the expected number of successfully completed tasks, assuring budget feasibility, incentive compatibility, and individual rationality.

- *What is the solution proposed?*
  They proposed a new algorithm "*Budgeted MAB mechanism with task deadline*". The input parameters to the algorithm are the bid vector $\hat{c}$, task deadline τ, budget B and maximum allowed bid vector $\overline{c}$. The mechanism is exploration separated in which the mechanism divides the budget B into the exploration budget and exploitation budget. B_1 is the exploration budget.

  The algorithm first explores the workers by allocating the tasks to the workers in a round robin fashion till budget B_1 is exhausted. Such rounds are known as exploration rounds. The algorithm maintains the running average of quality $\hat{q}_k$ obtained from each worker k in the exploration round. The per allocation payment for each worker during exploration phase is $\hat{c}$. After the exploration rounds .the τ workers according to decreasing order of $\frac{q_k^+}{c_k}\forall k \in N$, are chosen to be played to be sequentially one by one in each round, where $\hat{q}_k^+ = \hat{q}_k + \sqrt{\frac{K\overline{c}\ln(KB)}{2B_1}}$ is the upper bound quality estimate for worker k.

- *What does the paper claim?*

  - A MAB mechanism that takes into account limited budget, task deadlines, unknown qualities, and strategic workers (strategic about their costs). Note that the quality of a worker refers to the probability of the worker completing a task successfully within the given deadline. They claim the mechanism maximizes the expected number of tasks completed successfully subject to budget feasibility, incentive compatibility, and individual rationality.

  - Established an upper bound $O(B^{\frac{2}{3}}(K\ln(KB))^{\frac{1}{3}})$ on the expected regret of the proposed mechanism with respect to an appropriate be benchmark algorithm, where B is the total budget and K is the number of workers where uncertainty in the availability of a worker, the budget constraint, and the strategic nature of the workers render the regret analysis are challenging.
  - They have proved that the exploration separated property is necessary for any truthful and IR mechanism for the given setting.

- *What are the new ideas the author claims?*
  The authors have claimed that there is no existing work that deals with budgeted multi-armed problems which additionally captures the task deadlines and strategic nature of workers over their tasks.

- *How does the work compare with previous works?*
  The budgeted multi-armed bandit that is close to the setting considered by Tran-Thanh[8, 9] without strategic agents and task deadlines. Budgeted MAB problems have also been widely studied for pricing tasks (or items) in crowdsourcing (or dynamic procurement) problems. With workers arriving online with a fixed and known distribution, Singer[11] considered a budgeted setting with a goal to maximize the total number of allocated tasks to the workers. The workers were assumed to complete the task successfully if allocated and thus, the goal was to design a pricing mechanism to complete the tasks within a budget. In the setting, workers complete the allocated task with a fixed probability which is unknown and we wish to design an auction mechanism by incentivizing the workers to bid their true cost of effort.

6.6 Research Paper 6

Contextual Bandit Algorithms with Supervised Learning Guarantees
Authors : *Alina Beygelzimer, John Langford, Lihong Li, Lev ReyzinRobert E. Schapire.*[13]

- *What is the problem addressed?*
  The paper address the problem of competing with large set of N policies in the non-stochastic bandit setting, where the learner must repeatedly select among K actions but observes only the reward of chosen action.

- *What does the paper claim?*
  They have proposed a new algorithm $Exp4.P$. The algorithm is based on a careful composition of the $Exp4$ and $Exp3.P$ algorithms. The paper advances on a basic argument, namely, that *exploration problems* are solvable in almost the same sense as supervised learning problems, with suitable modifications to existing learning algorithms. In particular they have shown that learning to compete with any set of strategies in the contextual bandit setting requires only a factor of K more experience than for supervised learning (to achieve the same level of accuracy with the same confidence).

- *What are the new ideas the author claims?*
  The authors have claimed that the algorithm addressed competing with a finite (but possibly exponential in T) set of policies. They have shown that using $Exp4.P$ in a black ox fashion to guarantee a high probability regret bound of $O(\sqrt{Td\ln T})$, in the case of infinite set of policies with a finite VC-dimension d. $Exp4.P$ provides a practical framework for incorporating more expressive expert classes and it is efficient when N is polynomial in K and T and it may be possible to run $Exp4.P$ efficiently in certain cases when working with a family of experts that is exponentially large, but well structured.

- *How does the work compare with previous works?*
  The author claims that the algorithm $Exp4.P$ with high probability achieves $O(\sqrt{TK\ln N})$ regret in the adversarial contextual bandit setting which improves on the $O(T^{\frac{2}{3}}(k\ln N)^{\frac{1}{3}})$ with high probability bound in the stochastic setting. Previously, this result was known to hold in expectation for the algorithm Exp4, but a high probability statement did not hold for the same algorithm, as per-round regrets on the order of $O(T^{\frac{-1}{4}})$ were possible. Succeed-ing with high probability is important because reliably useful methods are preferred in practice.

- *Limitation*
  $Exp4.P$ does retain one limitation of its predecessors, it requires keeping explicit weights over the experts, so in the case when N is too large, the algorithm becomes inefficient.

6.7 Research Paper 7

Mean Field Analysis of Multi-Armed Bandit Games
Authors : *Ramki Gummadi, Ramesh Johari, Sven Schmit, Jia Yuan Yu* .[14]

- *Introduction:*
  Much of the classical work on algorithms for multi-armed bandits focuses on rewards that are stationary over time. By contrast, we study multi-armed bandit (MAB) games, where the rewards obtained by an agent also depend on how many other agents

choose the same arm (as might be the case in many competitive or cooperative scenarios).

In many cases, an agent appears to be solving an MAB problem, but in fact the rewards earned on the arms may be highly dependent on the actions of other agents who are also solving their own MAB problems. For example, in wireless resource sharing, when a device chooses to transmit in a given channel, it competes directly with other devices exploring the same channel. Similarly, in online ad auctions, advertisers considering bidding on different keywords must consider the likelihood of winning against other bidders competing on the same keywords.

When agents interact in this way, the overall system can no longer be analyzed through the eyes of a single agent; rather, we have to view the agents' interactions as a dynamic game, that they have called as a *multi-armed bandit (MAB) game*. Somewhat surprisingly, while MAB problems and variants have been extensively studied, there is little structural insight into dynamic games where agents solve interlinked MAB problems. In this paper, we provide significant insight into this class of *strategic models*.

- *Challenges:*
  To illustrate the difficulty, first suppose that two agents each choose among a finite set of arms each period, and that the (random) rewards they obtain are dependent on both an agent specific parameter, as well as whether or not the other agent pulled the same arm. It should be clear that this problem is significantly more complex than the classical MAB, because by pulling one arm, an agent learns both about that arm's reward distribution and her opponent's strategy. For example, in a resource sharing scenario, if an agent does not receive a reward on an arm, this would likely increase her belief that another player may have been taking the same action at the same time. Thus proper analysis of dynamic equilibrium in such game requires modeling each player's beliefs, their beliefs over other players beliefs, etc.

- *Perfect Bayesian Equilibrium*
  In such a dynamic game with finitely many players, the standard equilibrium concept is perfect Bayesian equilibrium (PBE); PBE requires that:

  - Agents maintain beliefs over all that they learn about their competitors.
  - Agents play optimally after any history, given their beliefs.

  The resulting equilibrium concept is both intractable (as it requires exceedingly complex state information) and implausible (since in practice, agents may not track fine-scale behavior of their competitors).

- *What is the problem addressed?*
  Due to the problems in establishing PBE and solving the modified MAB problem In many practical bandit scenarios, adoption of classical bandit algorithms designed for a stationary environment seems commonplace, despite the fact that the environment could be nonstationary as a result of being dependent on the population wide actions which leads to few questions like,

  - Does it matter that agents make the stationarity assumption?
  - What are the conditions under which the environment does become stationary?

  In this paper the authors have addressed the above mentioned problems.

- *What is the solution proposed?*
  They proposed MAB game in a mean field regime, inspired by an approximation where the number of agents becomes large. In particular, suppose agents' conjecture that competitors pull arms at a frequency given by their long run average; they have referred to this long run average of arm frequencies across agents as the population profile. Under this conjecture, their environment appears stationary, so the agent's optimization problem becomes a classical MAB problem. Of course, a consistency check is required: the conjectured population profile must arise from agents' chosen policy. We refer to the resulting fixed point as a mean field steady state (MFSS).

- *What are the main results in the paper?*

  1. Existence of MFSS for MAB games. In the model they have considered, they have assumed that agents play fixed policy; for example, this may be a regret-optimal policy for the classical (stationary) MAB setting. They have established existence of MFSS for this model. While fixing the policy agents use may be ill justified for an equilibrium concept like the PBE, this approach is sensible for MAB games for reasons outlined above; for example, if agents use a regret-minimizing policy (such as UCB), we can show that it is approximately optimal for an MAB game.
  2. Uniqueness and convergence: They have identified a contraction condition on the arm rewards that ensures the MFSS is unique, and that starting from any initial state, the dynamics will converge to this MFSS (in the sense that

eventually the population profile becomes constant). The contraction condition requires that the agent population is sufficiently mixing and that the sensitivity of the reward function to the population profile is low enough.

3. Approximation: Under the same contraction condition used to establish uniqueness, they have shown an approximation result that justifies the use of MFSS. In particular, they have established that if the number of agents grows large, then the dynamics of the finite agent system converge to the dynamics of the mean field model.

References

[1] T. L. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” Advances in Applied Mathematics, vol. 6, pp. 4–22, 1985.

[2] Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems By Sebastien Bubeck and Nicolo Cesa-Bianchi

[3] J. C. Gittins, “Bandit processes and dynamic allocation indices,” Journal of the Royal Statistical Society. Series B (Methodological), pp. 148–177, 1979.

[4] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. E. Schapire, “Contextual bandit algorithms with supervised learning guarantees,” in Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR Workshop and Conference Proceedings Volume 15, 2011.

[5] Mechanisms with Learning for Stochastic Multi-armed Bandit Problems Shweta Jain, Satyanath Bhat, Ganesh Ghalme, Divya Padmanabhan, and Y. Narahari.

[6] Thompson Sampling for Contextual Bandits with Linear Payoffs, Shipra Agrawal, Navin Goyal, Proceedings of the 30^{th} International Conference on Ma- chine Learning, Atlanta, Georgia, USA, 2013. JMLR: W &CP volume 28.

[7] A Contextual-Bandit Approach to Personalized News Article Recommendation, Lihong Li, John Langford, Robert E. Schapire.

[8] Efficient Crowdsourcing of Unknown Experts using Multi-Armed Bandits, Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings.

[9] Knapsack Based Optimal Policies for Budget–Limited Multi-Armed Bandits, Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R. Jennings.

[10] Online Decision Making in Crowdsourcing Markets: Theoretical Challenges, Aleksandrs Slivkins, Jennifer Wortman Vaughan

[11] Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In Proceedings of the 22nd International Conference on World Wide Web, pages 1157-1166, 2013.

[12] A Truthful Budget Feasible Multi-Armed Bandit Mechanism for Crowdsourcing Time-Critical Tasks, Arpita Biswas, Shweta Jain, Debmalya Mandal, Y. Narahari.

[13] Contextual Bandit Algorithms with Supervised Learning Guarantees, Alina Beygelzimer, John Langford, Lihong Li, Lev ReyzinRobert E. Schapire.

[14] Mean Field Analysis of Multi-Armed Bandit Games, Ramki Gummadi, Ramesh Johari, Sven Schmit, Jia Yuan Yu.