The Adversarial Multi-Armed Bandit Problem

The Final Project for Math829

Ke Jin, Shuying Sun, Peng Xu

1 Introduction

A multi-armed bandit problem is a sequential decision problem. At each time, an action is taken and some payoff is obtained. The goal is to maximize the total payoff obtained in a sequence of actions. The name *bandit* refers to the colloquial term for a slot machine. In a casino, the sequential decision problem faced by the player is to decide where to insert the next coin, when facing many slot machines.
Bandit problems are basic instances of sequential decision making with limited information, and address the fundamental tradeoff between exploration and exploitation in sequential experiments. The formal setting of multi-armed bandit problem is defined as follows,

---

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

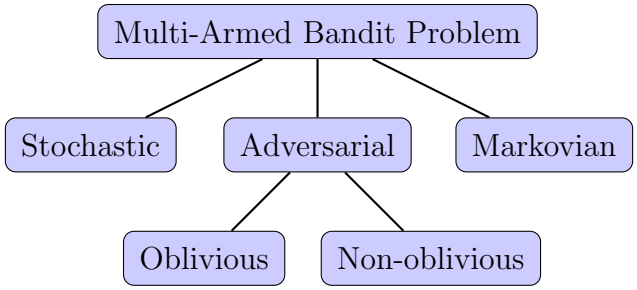
(1) the player chooses $I_t \in \{1, \ldots, K\}$, possibly with the help of external randomization;

(2) the environment (or the adversary) selects a gain vector $G_t = (g_{1,t}, g_{2,t}, \ldots, g_{K,t}) \in [0,1]^K$, possibly with the help of external randomization;

(3) the player receives the reward $g_{I_t,t}$, and (possibly) observed the gains of other arms.

According to the nature of the reward process G_t, the multi-armed bandit problem can be categorized into three different models: stochastic, adversarial and Markovian. Three distinct strategies have been developed for these different models. Specifically, we have UCB algorithm for the stochastic model, Exp3 algorithm for the adversarial model and Gittins indices for the Markovian model.

1.1 Different Models based on the nature of G_t



Stochastic model: Independent of all other things, $\{g_{i,t}\}$ are $i.i.d$ sequence for each $i \in K$.

Adversarial oblivious model: $\{I_t\}$ and $\{G_t\}$ are independent.

Adversarial non-oblivious model: Conditioned on the distribution of I_t, I_t and G_t are independent at each time t.

Markovian model: $\{g_{i,t} : I_t = i\}$ is a Markov chain for each $i \in K$.

1.2 Performance Evaluation for Multi-Armed Bandit Problem

The *regret* after n plays is defined by

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

In general, both the rewards vector G_t and the player's choices I_t might be randomized. In the following, the expectation is taken with respect to the probability space in which both $\{G_t\}$ and $\{I_t\}$ are defined. The *expected regret* is defined by

$$\mathbb{E}(R_n) = \mathbb{E}\left(\max_{i=1,\dots,K} \sum_{t=1}^{n} g_{i,t} - \sum_{t=1}^{n} g_{I_t,t} \right)$$

and the *pseudo regret* is defined by

$$\bar{R}_n = \max_{i=1,\dots,K} \mathbb{E}\left(\sum_{t=1}^{n} g_{i,t} - \sum_{t=1}^{n} g_{I_t,t} \right)$$

Note that pseudo regret is weaker than the expected regret, since it compares the average gain of the player with the average gain of each machine. The expected regret, on the other hand, compares the player's total gain with the best machine on every realization of the game. In fact, one has $\bar{R}_n \leq \mathbb{E}(R_n)$.

2 Exp3 Algorithm

In this section, we introduce the main algorithm implemented in this project the *Exp3* algorithm and its variant, along with some performance guarantee.

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

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

(2) For each arm $i = 1, \ldots, K$, update the estimated cumulative gain $\hat{G}_{i,t} = \hat{G}_{i,t-1} + \hat{g}_{i,t}$, where

$$\hat{g}_{i,t} = \begin{cases} g_{i,t} & \text{if the gains of each machine are known} \\ \frac{g_{i,t}}{p_{i,t}} \mathbb{1}_{I_t=i} & \text{if only the gain of the chosen machine is known} \end{cases}$$

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

$$p_{i,t+1} = \frac{\exp\left(\eta_t \hat{G}_{i,t}\right)}{\sum_{j=1}^{K} \exp\left(\eta_t \hat{G}_{j,t}\right)}$$

Theorem 2.1 (Pseudo regret of Exp3)**.** If Exp3 is run with $\eta_t = \sqrt{\frac{2 \ln K}{nK}}$, then

$$\bar{R}_n \leq \sqrt{2nK \ln K}$$

Moreover, if Exp3 is run with $\eta_t = \sqrt{\frac{2 \ln K}{tK}}$, then

$$\bar{R}_n \leq 2\sqrt{nK \ln K}$$

Although, the Exp3 algorithm above has the relatively small pseudo regret upper bound, our simulations indicate that the performance of Exp3 algorithm has high fluctuation, in other words, the variance of the regret is large. And it is likely that the performance of a single realization of the algorithm is very poor.

2.1 Exp3.P Algorithm

In order to get a non-trivial expected regret bound as well as high probability bound, one has the following variant of Exp3.

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

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

(2) For each arm $i = 1, \ldots, K$, update the estimated cumulative gain $\hat{G}_{i,t} = \hat{G}_{i,t-1} + \hat{g}_{i,t}$, where

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

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

$$p_{i,t+1} = (1-\gamma)\frac{\exp\left(\eta_t \hat{G}_{i,t}\right)}{\sum_{j=1}^{K} \exp\left(\eta_t \hat{G}_{j,t}\right)} + \frac{\gamma}{K}$$

Theorem 2.2 (Expected regret bound of Exp3.P)**.** If Exp3.P is run with

$$\beta = \sqrt{\frac{\ln K}{nK}}, \quad \eta = 0.95\sqrt{\frac{\ln K}{nK}}, \quad \gamma = 1.05\sqrt{\frac{K \ln K}{n}}$$

then

$$\mathbb{E}(R_n) \leq 5.15\sqrt{nK \ln K} + \sqrt{\frac{nK}{\ln K}}$$

Theorem 2.3 (High probability bound for Exp3.P)**.** If Exp3.P is run with the same β, η and γ as in Theorem 2.2, then with probability at least $1 - \delta$,

$$R_n \leq 5.15\sqrt{nK \ln K} + \sqrt{\frac{nK}{\ln K}} \ln\left(\delta^{-1}\right)$$

3 Applications

We implement the Exp3 and Exp3.P algorithm in the following three distinct situations: the simulated slot machines, the hypothetical stock trading and the repeated unknown game. We compare the effectiveness of Exp3.P strategy against the uniformly random strategy.

3.1 Simulated Slot Machines

We have K machines $M_1, M_2, \cdots, M_K$. At any time t, let $n_{i,t}$ be the number of times machine i has been chosen up to time $t-1$. Machine i returns a gain $g_{i,t} = X_{i,t} + Y_{i,t}$, where $X_{i,t}$ has distribution Bernoulli($\frac{i}{2K}$) and ,independently, $Y_{i,t}$ has distribution Beta($\alpha_{i,t}, \beta_{i,t}$) with

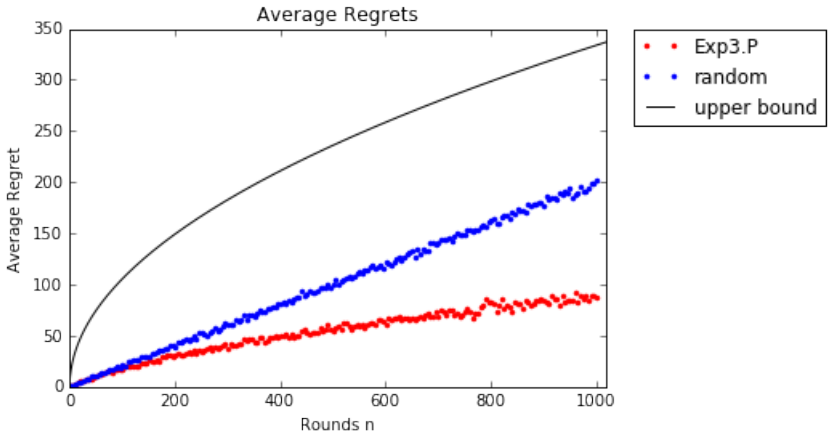
$$\alpha_{i,t} = \exp\left(-\frac{K}{t-1} n_{i,t}\right), \qquad \beta_{i,t} = \sum_{j \neq i} \alpha_{j,t}$$

Since Beta(a, b) has mean $\frac{a}{a+b}$, we have, at each time t, the mean of the sum of gains by all machines is

$$\begin{aligned} \sum_{i=1}^{K} \mathbb{E}(g_{i,t}) &= \sum_{i=1}^{K} \mathbb{E}(X_{i,t}) + \mathbb{E}(Y_{i,t}) \\ &= \sum_{i=1}^{K} \frac{i}{2K} + \frac{\alpha_{i,t}}{\alpha_{i,t} + \beta_{i,t}} = \frac{K+1}{4} + 1. \end{aligned}$$

Here, we specifically make $\sum_{i=1}^{K} \mathbb{E}(g_{i,t})$ a constant at each time t to ensure that on average, there exists some machine which generate relatively good reward. And $\alpha_{i,t}$ decreases exponentially as $n_{i,t}$ increases, and hence the mean of $Y_{i,t}$ will be small if $n_{i,t}$ is large.

Then, we simulated 5 machines defined as above, and played for 1000 rounds.

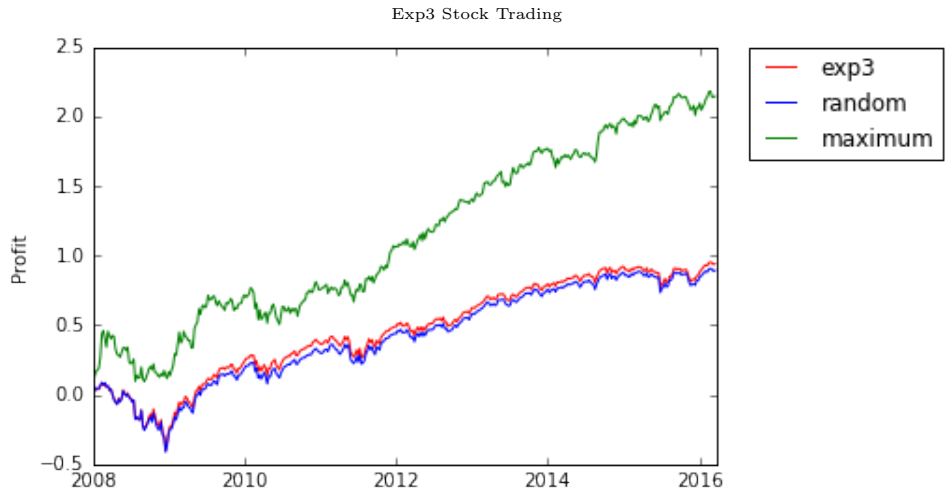


The regret using Exp3.P strategy is much smaller than choosing slot machine uniformly random each time.

3.2 Hypothetical Stock Trading

We apply the Exp3 algorithm to the 29 stocks in the Dow Jones Index(excluding Apple). We start the hypothetical trading from March 2008 to May 2016. On each

trading day we invest one dollar to one of the 29 stocks at its close price and sell it at its close price the very next trading day.



The return of this trading strategy using Exp3 is actually no better than choosing stock randomly. One reason why the Exp3 does not work in this case is: the regret bound of Exp3 is of such large magnitude which is essentially meaningless. The total gain(including the one dollar per day principal) of the best stock, which is Visa, during these 2050 days is 2052.145 dollars. The total gain of the worst stock, which is Caterpillar, during the same period is 2050.295 dollars. Hence, when choosing randomly, the expected regret of total gain cannot be worse than 1.85 dollars, the difference between the best stock and worst stock. However, the pseudo regret bound in Theorem 2.1 in this case would be $\sqrt{2 \times 2050 \times 29 \times \ln 29}$.
Hence, we conclude that the Exp3 algorithm is better suited to those problem where

$$\max_i \mathbb{E}\Big(\sum_{t=1}^{n} g_{i,t}\Big) - \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\Big(\sum_{t=1}^{n} g_{i,t}\Big) \gg \sqrt{2nK\ln K}$$

3.3 Repeated Unknown Game

In this example, the game is defined by a $K \times m$ matrix M such that each entry $M_{ij} \in [0, 1]$. At each time t, the row player chooses a row i and the opponent(column player) chooses a column j. The row player then received the payoff M_{ij}. The row player's goal is to maximize its expected total payoff over a sequence of plays.
Suppose, at time t, the row player chooses the row randomly according to a probability distribution $\boldsymbol{p} = (p_1, \ldots, p_K)^T$, and the column player similarly chooses the column according to a distribution $\boldsymbol{q} = (q_1, \ldots, q_m)^T$. Then row player's expected payoff is $\boldsymbol{p}^T M \boldsymbol{q}$.
Von Neumann's celebrated minimax theorem states that

$$\max_{\boldsymbol{p}} \min_{\boldsymbol{q}} \boldsymbol{p}^T M \boldsymbol{q} = \min_{\boldsymbol{q}} \max_{\boldsymbol{p}} \boldsymbol{p}^T M \boldsymbol{q}$$

where the maximum and minimum are taken over all distribution $\boldsymbol{p}$ and $\boldsymbol{q}$. The quantity v defined by the above equation is called the *value* of the game with matrix M. Choose

$$\bar{\boldsymbol{p}} = \operatorname*{argmax}_{\boldsymbol{p}} \min_{\boldsymbol{q}} \boldsymbol{p}^T M \boldsymbol{q}$$

To compute the *value* of the game and optimal strategy $\bar{\boldsymbol{p}}$, we could use linear programming.

Given any $\boldsymbol{p}$, it is easily seen that $\min_{\boldsymbol{q}} \boldsymbol{p}^T M \boldsymbol{q} = \min_j \sum_{i=1}^{K} M_{ij} p_i$

Therefore the problem faced by the row player can be written as

$$\max_{\boldsymbol{p}} \min_j \sum_{i=1}^{K} M_{ij} p_i$$

It is equivalent to the linear program

$$\begin{aligned} \text{maximize} \quad & z \\ \text{Subject to} \quad & z \leq \sum_{i=1}^{K} M_{ij} p_i, \quad \text{for } 1 \leq j \leq m \\ \text{and} \quad & \sum_{i=1}^{K} p_i = 1, \quad p_i \geq 0 \end{aligned}$$

If the row player knows the matrix M, then we could use the strategy $\bar{\boldsymbol{p}}$ at any time t, the expected payoff is at least v, regardless of column player's strategy. Moreover, the strategy $\bar{\boldsymbol{p}}$ is optimal in the sense that the column player can choose a strategy $\bar{\boldsymbol{q}}$ such that the row player's expected payoff is at most v, regardless of row player's strategy.

Now we put the row player in a much worse setting. Suppose, the matrix M is entirely unknown to the row player. Moreover, in each play of the game, the row player only knows the value M_{ij}, the actual gain of this play. What makes things even worse is that, not only does the column player know the entire matrix M, but it also knows, at each time t, the row player's strategy $\boldsymbol{p}_t$ *before* choosing its own strategy. So the column player can choose the column $\operatorname{argmin}_j \sum_{i=1}^{K} M_{ij} p_i^{(t)}$ with probability 1.

Theorem 3.1. Using Exp3 algorithm in the above setting for n trials, the row player's expected payoff per trial is at least $v - \sqrt{\frac{2K \ln K}{n}}$.

Proof. Let $\boldsymbol{p}_t$ and $\boldsymbol{q}_t$ denote the strategies of row player and column player respectively. Then the expected total gain of row player after n trial is $\sum_{t=1}^{n} \boldsymbol{p}_t^T M \boldsymbol{q}_t$. By

the pseudo regret bound in Theorem 2.1, we have

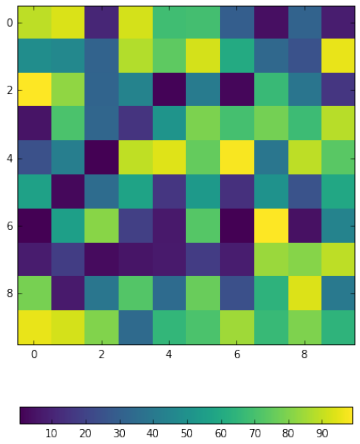
$$\sum_{t=1}^{n} \boldsymbol{p}_t^T M \boldsymbol{q}_t = \mathbb{E}\Big(\sum_{t=1}^{n} g_{I_t,t}\Big) \geq \max_i \mathbb{E}\Big(\sum_{t=1}^{n} g_{i,t}\Big) - \sqrt{2nK \ln K} \tag{1}$$

Let $\bar{\boldsymbol{p}}$ be the strategy for the row player such that $v = \min_{\boldsymbol{q}} \bar{\boldsymbol{p}}^T M \boldsymbol{q}$. Then we have

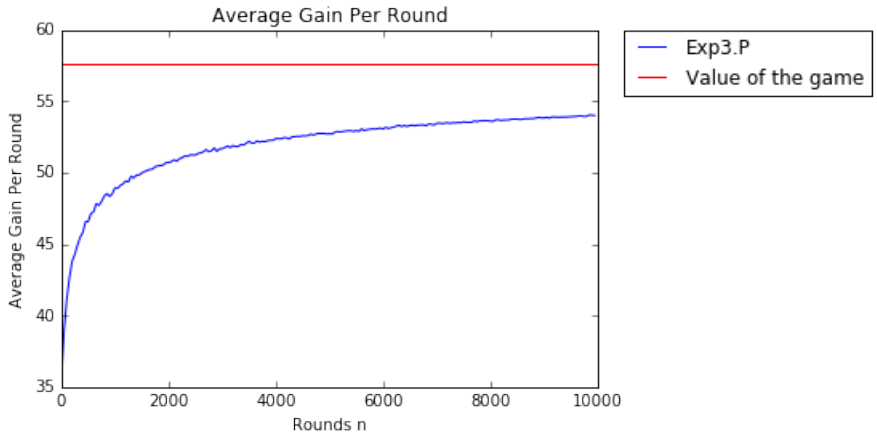
$$\begin{aligned} \max_i \mathbb{E}\Big(\sum_{t=1}^{n} g_{i,t}\Big) &\geq \sum_{i=1}^{K} \bar{p}_i \, \mathbb{E}\Big(\sum_{t=1}^{n} g_{i,t}\Big) = \mathbb{E}\Big(\sum_{t=1}^{n} \bar{\boldsymbol{p}} \cdot G_t\Big) \\ &= \sum_{t=1}^{n} \bar{\boldsymbol{p}} \cdot \mathbb{E}(G_t) = \sum_{t=1}^{n} \bar{\boldsymbol{p}} M \boldsymbol{q}_t \geq nv \end{aligned} \tag{2}$$

Combining (1) and (2) finishes the proof. □

Now we study a 10 by 10 matrix with each entry between 0 and 100.



This matrix has value $v = 57.6397722283$.
We play the unknown repeated game with the column player for 10000 round, and compute the average gain per round.



We could see that the average gain approaches the value v quickly.

References

[1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund and Robert E. Schapire, *Gambling in a rigged casino: The adversarial multi-armed bandit problem*, Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on, IEEE, (1995), 322-331.

[2] Sébastien Bubeck and Nicolo Cesa-Bianchi, *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*, arXiv preprint arXiv:1204.5721, (2012).