CSE599i: Online and Adaptive Machine Learning **Winter 2018**

Lecture 5: Non-Stochastic Bandits

Lecturer: Kevin Jamieson *Scribes: Anran Wang, Beibin Li, Brian Chan, Shiqing Yu, Zhijin Zhou*

Notation

The following notations are widely used in this manuscript for the discussion of multiarmed bandit (MAB) game:

1. In a bandits machine, t represents time point t. T represents the total play time.
2. I_t represent the arm pulled at time t. When n represents the number of arms in a bandit machine, then $I_t \in [n] = \{1, 2, ..., n\}$.
3. $\ell_{i,t} \in [0, 1]$ represents the lose of pulling arm i at time t.
4. $\Delta(i, t) = \ell_{i,t}$ represents the gap of benefits if the player pull arm i at time t.
5. $g_{i,t} = 1 - \ell_{i,t}$ represents the gain of pulling arm i at time t.
6. $\mathbb{1}\{I_t = i\}$ is an indicator function. When $I_t = i$ (i.e. i-th arm is pulled at time t), $\mathbb{1}\{I_t = i\}$ equals to one; otherwise, it equals to zero.
7. Total regret the player received in the whole game is $R(T) = \max_i \sum_{t=1}^{T} \ell_{I_t,t} - \sum_{t=1}^{T} \ell_{i,t}$.

The following definitions are helpful in the manuscript.

Definition 1. *Expected regret:* $\mathbb{E}[R(T)] = \mathbb{E}\left[\max_i \sum_{t=1}^{T} \ell_{i,t} - \sum_{t=1}^{T} \ell_{I_t,t}\right]$

Definition 2. *Pseudo-regret:* $\tilde{R}(T) = \max_i \mathbb{E}\left[\sum_{t=1}^{T} \ell_{i,t} - \sum_{t=1}^{T} \ell_{I_t,t}\right]$

1 Problem Definition

All the previous bandits problems we analyzed were stochastic, which means the rewards are determined before the game. However, in the real world, the distribution of the reward X_t usually depends on previous actions/rewards. For instance, if an online advertisement system (e.g. Google Ads) shows the same type of ads for a customer (i.e. the system is choosing the same "arm" constantly), then the customer would click the advertisement less and less often. The definition of non-stochastic bandits is shown below.

Bandit 1: Non-Stochastic Multi-Arm Bandits

Input : number of trials T, and the non-stochastic n-arm bandit machine
Output: Player's choices $I_1, I_2, \cdots, I_T$, and the regret $R(T)$.
(I_i represents the arm the player has chosen at time i)
for $t = 1, 2, \cdots, T$ **do**
  Player chooses an arm $I_t \in [n]$
  Adversary simultaneously chooses losses for each arm $(\ell_{1,t}, \ell_{2,t}, \ldots, \ell_{n,t}) \in [0,1]^n$.
  Player receives and observes loss $\ell_{I_t,t}$.
end
Total regret $R(T) = \max_i \sum_{t=1}^{T} \ell_{I_t,t} - \sum_{t=1}^{T} \ell_{i,t}$.

Non-stochastic bandits are sometimes referred to as adversarial bandits, where an adversary is changing the future rewards for arms. It's easy to see no matter what deterministic algorithm the player chooses, the adversary can always design a game to defeat the player such that the player's algorithm suffers $R(T) \geq \frac{T}{2}$. For instance, in a two-arm bandit case, the adversary can learn the player's deterministic algorithm and predict the player's choice, and then the adversary can set the rewards of the chosen arm to zero and the reward of the unchosen arm to one; hence, the total rewards received by the player will be less than or equal to $T/2$, but the maximum possible reward is T. A generalized proof for n-arm bandit machine is shown below.

Theorem 1. *A game exists s.t. any deterministic algorithm suffers $R(T) \geq \frac{T}{2}$ in an adversarial bandit.*

Proof. Because the player's algorithm f is deterministic, $I_t = f\left(\{\ell_{I_s,s}, I_s\}_{s=1}^{t-1}\right)$, where s is a time point before time t. This fact means player's choice at time t depends on all his/her previous actions and observed losses.

The adversary can choose the strategy for the n arms for each time t:

1. if player chooses $I_t = 1$, then the adversary could set $\ell_{1,t} = 0$, and $\ell_{i,t} = 1\ \forall i \neq 1$.
2. if player chooses $I_t \neq 1$, then the adversary could set $\ell_{1,t} = 1$, and $\ell_{i,t} = 0\ \forall i \neq 1$.

Because the best reward at each time is 1, the maximum possible losses for this game is $\sum_{t=1}^{T} \ell_{I_t,t} = \sum_{t=1}^{T}(1-0) = T$. On the other hand, the player receive reward 0 at each time, and hence total reward the player get is: $\min_i \sum_{t=1}^{T} \ell_{i,t} = 0 \leq T/2$.

So, $R(T) \geq \sum_{t=1}^{T} \ell_{I_t,t} - \min_i \sum_{t=1}^{T} \ell_{i,t} = T - T/2 = T/2$.

□

Because the adversary can design a game to make the player suffer at least $T/2$ regret after game, we have to use a randomized strategy to surprise the adversary.

In the normal adversarial bandits, the pseudo-regret has no direct interpretation for an adapting adversary who can change the future rewards for each arm during the game. However, the pseudo regret equals to true expected regret in an **oblivious adversary**, where $\{\ell_{i,t}\}$, does not depend on $\{I_s\}_{s=1}^{t-1}$. This fact means the adversary can look at your algorithm but all the losses ($l_{I_t,t}$ for $\forall t, \forall I_t$) in the game must be picked before the start of the game.

Before the game, the adversary filled out matrix L, where entry $L_{i,t}$ represents the distribution of lose for arm i at time t. In the stochastic setting, the distribution of lose for each arm won't change across time (i.e. $L_{i,t} = L_{i,t'}$ for $\forall_{i,t,t'}$); on the other hand, in oblivious adversary, the distribution of lose for each arm would change over time (i.e. it's possible $L_{i,t} \neq L_{i,t'}$). So, under the circumstance of oblivious adversary, the player should modify its algorithm from stochastic bandit to adapt to the fact that the rewards/losses from each arm can change.

2 EXP3 Algorithm

Exponential-weight algorithm for Exploration and Exploitation (EXP3) is then developed to solve the non-stochastic problem with an oblivious adversary. The term "exponential" comes from the formula $w_{i,t+1} = \exp\left(\eta \sum_{s=1}^{t} \tilde{\ell}_{i,s}\right) = w_{i,t} \exp\left(-\eta \tilde{\ell}_{i,t}\right)$, where the weight for an arm is updated by the exponential of its losses. The concept of exponential-weight is widely used in computer science and well explained in [1].

In the following EXP3 algorithm, $w_{i,t}$ represents the weight of arm i at time t, which is used to calculate and update exponential weights. For simplicity, assume the initial weight for every arm is the same and $w_{i,1} = 1$ for $\forall i \in \{1, 2, ..., n\}$. W_t is the cumulative weights (i.e. total weight of all arms) at time t, and $p_{i,t} = \frac{w_{i,t}}{W_t}$ represents the percentage weight of arm i at time t for $\forall i$.

Player's choice I_t (choice at time t) is randomly chosen from probability distribution p_t of all arms, and this step is the random component of the algorithm to surprise the adversarial. After each action, the unbiased estimator for loss is calculated as $\tilde{\ell}_{i,t} = \frac{\mathbb{1}\{I_t = i\}}{p_{i,t}} \ell_{i,t}$. It is an unbiased estimator because

$$\mathbb{E}[\tilde{\ell}_{i,t}] = \sum_{j=1}^{n} p_{j,t} \frac{\mathbb{1}\{j = i\}}{p_{i,t}} \ell_{i,t} = \ell_{i,t}, \tag{1}$$

where $p_{j,t}$ is the probability (i.e. percentage of weight) that arm i will be chosen at time t. In the algorithm, p_1 represents the probability distribution of arm choice before the game.

Bandit 2: EXP3 Algorithm

Input : a non-increasing sequence of real numbers $(\eta_t)_{t \in \mathbb{N}}$, an n-arm non-stochastic bandit
Initialize $w_{i,1} = 1$ for every arm i.
for $t = 1, 2, \cdots, T$ **do**
  Update $W_t = \sum_{i=1}^{n} w_{i,t}$, $p_{i,t} = \frac{w_{i,t}}{W_t}$.
  Player chooses an arm $I_t \in [n]$ from the probability distribution p_t
  for *each arm* $i = 1, \cdots, n$ **do**
    Compute the estimated loss $\tilde{\ell}_{i,t} = \frac{\mathbb{1}\{I_t = i\}}{p_{i,t}} \ell_{i,t}$
    Update $w_{i,t+1} = \exp\left(\eta_t \sum_{s=1}^{t} \tilde{\ell}_{i,s}\right) = w_{i,t} \exp\left(-\eta_t \tilde{\ell}_{i,t}\right)$.
  end
end

The pseudo-regret of EXP3 algorithm, $\tilde{R}(T)$, is bounded by $\sqrt{2nT \log(n)}$, and consider using $\log(\frac{W_{T+1}}{W_1})$ for mathematical convenience. The proof is shown below, which contains three steps: 1. get lower bound for $\log(\frac{W_{T+1}}{W_1})$; 2. get upper bound for $\log(\frac{W_{T+1}}{W_1})$; 3. combine the lower bound and upper bound to bound the regret of EXP3 algorithm. The formal proof is shown below.

Theorem 2. *The pseudo-regret of EXP3 algorithm is bounded, and* $\tilde{R}(T) \leq \sqrt{2nT \log(n)}$.

Proof. For the EXP3 algorithm, assume $\eta_t = \eta$ as a constant number for $\forall t \in [T]$.

Note $\mathbb{E}[\tilde{\ell}_{i,t}^2] = \frac{\ell_{i,t}^2}{p_{i,t}}$ and $\mathbb{E}[\ell_{I_t,t}] = \sum_{j=1}^{n} p_{j,t} \ell_{j,t} = \sum_{j=1}^{n} p_{j,t} \mathbb{E}[\tilde{\ell}_{i,t}]$.

Then, for the loss of i-th arm, we have

$$\mathbb{E}\left[\left(\tilde{\ell}_{i,t} - \mathbb{E}[\tilde{\ell}_{i,t}]\right)^2\right] \leq \mathbb{E}[\tilde{\ell}_{i,t}^2] - \mathbb{E}[\tilde{\ell}_{i,t}]^2 = \frac{\ell_{i,t}^2}{p_{i,t}} - \ell_{i,t}^2 = \ell_{i,t}^2 \left(\frac{1}{p_{i,t}} - 1\right)$$

By dividing W_{T+1} with W_1 and then take a *log*, we can get

$$\log\left(\frac{W_{T+1}}{W_1}\right) = \sum_{t=1}^{T} \log\left(\frac{W_{t+1}}{W_t}\right) \tag{*}$$

First, the lower bound on left hand side of equation (*) is

$$\begin{aligned}
\log\left(\frac{W_{T+1}}{W_1}\right) &= \log\left(\frac{\sum_{i=1}^n w_{i,t+1}}{n}\right) && \text{(because } W_1 = n\text{)} \\
&= -\log(n) + \log\left(\sum_{i=1}^n w_{i,t+1}\right) && \text{(take } n \text{ out of log)} \\
&= -\log(n) + \log\left(\sum_{i=1}^n \exp\left(-\eta \sum_{t=1}^T \tilde{\ell}_{i,t}\right)\right) && \text{(expand } w_{i,t+1} \text{ by definition)} \\
&\geq -\log(n) + \log\left(\exp\left(-\eta \sum_{t=1}^T \tilde{\ell}_{j,t}\right)\right) && \text{(use one term to lowerbound sum)} \\
&= -\log(n) - \eta \sum_{t=1}^T \tilde{\ell}_{j,t} && \text{(combine log and exp)} \qquad (2)
\end{aligned}$$

where the last inequality holds for *any* fixed $j \in [n]$. Second, the upper bound on right hand side of equation (*) is

$$\begin{aligned}
\log\left(\frac{W_{t+1}}{W_t}\right) &= \log\left(\sum_{i=1}^n \frac{w_{i,t}}{W_t} \exp(-\eta\tilde{\ell}_{i,t})\right) && \text{(by definition of cumulative weight } W\text{)} \qquad (3) \\
&= \log\left(\sum_{i=1}^n p_{i,t} \exp(-\eta\tilde{\ell}_{i,t})\right) && \text{(by definition of percentage weight } p_{i,t}\text{)} \qquad (4) \\
&\leq \log\left(\sum_{i=1}^n p_{i,t} - \eta\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t} + \frac{\eta^2}{2}\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t}^2\right) && \text{(from Taylor series: } \forall x \leq 0\text{, there is } e^x \leq 1 + x + \frac{x^2}{2}\text{)} \qquad (5) \\
&\leq -\eta\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t} + \frac{\eta^2}{2}\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t}^2. && \text{(because } \sum_i p_{i,t} = 1\text{, } e^x \geq 1 + x\text{)} \qquad (6)
\end{aligned}$$

At last, we combine the lower bound (2) and upper bound (6) for (*) together to obtain, for any $j \in [n]$:

$$-\log(n) - \eta\sum_{t=1}^T \tilde{\ell}_{j,t} \leq -\eta\sum_{t=1}^T\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t} + \frac{\eta^2}{2}\sum_{t=1}^T\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t}^2$$

Rearranging the terms, we get

$$\sum_{t=1}^T\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t} - \sum_{t=1}^T \tilde{\ell}_{j,t} \leq \frac{\log(n)}{\eta} + \frac{\eta}{2}\sum_{t=1}^T\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t}^2.$$

By incorporating the identity $\mathbb{E}\left[\sum_{t=1}^T\sum_{i=1}^n p_{i,t}\tilde{\ell}_{i,t} - \sum_{t=1}^T \tilde{\ell}_{j,t}\right] = \mathbb{E}\left[\sum_{t=1}^T \ell_{I_t,t} - \sum_{t=1}^T \ell_{j,t}\right]$ (for a given arm $j \in [n]$),

the pseudo regret can be bounded by

$$\begin{aligned}
\tilde{R}(T) &= \max_{j\in[n]} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{I_t,t} - \sum_{t=1}^{T} \ell_{j,t}\right] \\
&\leq \max_{j\in[n]} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n} p_{i,t}\tilde{\ell}_{i,t} - \sum_{t=1}^{T} \tilde{\ell}_{j,t}\right] \\
&\leq \frac{\log(n)}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{n} p_{i,t} \underbrace{\mathbb{E}[\tilde{\ell}_{i,t}^2]}_{\ell_{i,t}^2/p_{i,t}} \\
&\leq \frac{\log(n)}{\eta} + \frac{\eta n T}{2}.
\end{aligned}$$

We minimize the right-hand side of the last display by choosing $\eta = \sqrt{\frac{2\log(n)}{nT}}$, which gives the desired bound on the pseudo-regret $\tilde{R}(T) \leq \sqrt{2nT\log(n)}$.

□

The non-stochastic bandits problem is at least as hard as stochastic bandits, because the stochastic bandits is a special case of non-stochastic bandits, where the distribution of rewards for a given arm is defined and wouldn't change overtime. We also showed the pseudo-regret for EXP3 algorithm, bounded by $\sqrt{2nT\log(T)}$, is comparable to the result for the stochastic multi-arm bandit problem, where the expected regret for UCB algorithm for stochastic bandits is bounded by $O(\sqrt{nT\log(T)})$ [2]. It's nice that algorithm can solve the non-stochastic bandits problem as well as the stochastic bandits problem.

EXP3 algorithm can be used for both stochastic and non-stochastic bandits. However, UCB is better for stochastic bandits, because its expected regret transitions from $\sqrt{nT\log(T)}$ into $O(\Delta^{-1}\log(T))$ for large T (Δ, a constant, is the maximum possible regret for a time), while EXP3 is still in $O(\sqrt{nT\log(T)})$. Loss for UCB and EXP3 is shown in Figure 1, where x-axis is the time played in game, and y-axis is the total regret to time t.

The variance of EXP3 is high, which means there is large probability that player gets a much large regret than the expected one. In order to resolve the deficiencies in EXP3, a new algorithm, EXP3.P, is introduced.

3 EXP3.P Algorithm

The problem of the above EXP3 algorithm is, although the pseudo regret $\tilde{R}(T)$ can be bounded by $\sqrt{2nT\log(n)}$ given Theorem 2, the actual regret has a large variance since the loss $\ell_{i,t}$ is inverse-proportional to $p_{i,t}$. Hence, if at some time t, the probability of choosing some I_t is too small, the resulting reward would have a large fluctuation.

One idea is to never let $p_{i,t}$ be too small. To achieve this, we can mix the $p_{i,t}$ with uniform distribution to smooth the probability distribution.

Specifically, suppose p_t is the probability distribution of choosing some I_t before mixing, and u_t is the uniform distribution. We mix them linearly: $p'_t = p_t(1-\gamma) + u_t\gamma$ where $\gamma \in [0,1]$ is a constant. The intuition behind this is obvious: we increase the randomness of the selection and hence we do more exploration than exploitation.

The essential step when doing the mixing is how to choose the mixing parameter γ. If we set it too low, we cannot avoid the large variance problem. If we set it too high, too much smoothing would increase the total regret. We address this dilemma by introducing a *biased gain* instead of the *loss* metric in the EXP3 algorithm:

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

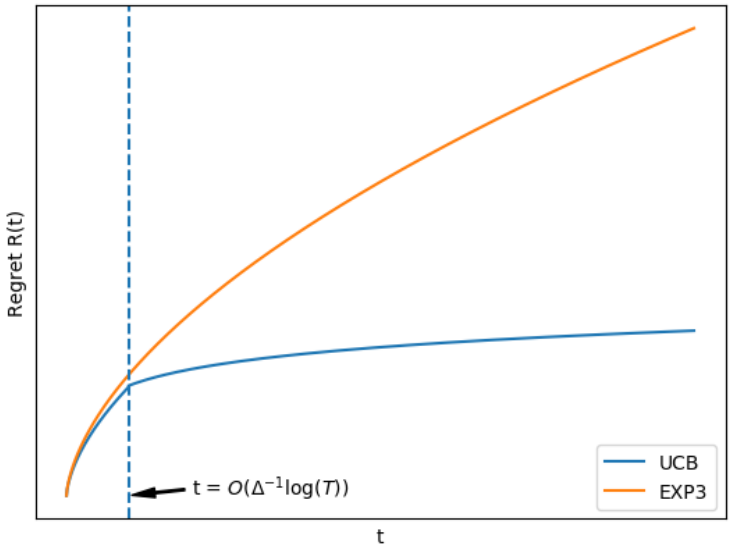

Figure 1: Regret for UCB and EXP3

where $g_{i,t}$ is the *gain* of pushing arm i at time t and can be represented as $1 - \ell_{i,t}$. In particular, as we will see in the following proof, the biased gain introduces guaranteed probabilistic bound on the regret R.

EXP3.P algorithm is proposed to achieve an upper-bound of the total regrets $R(T)$ at time T, by taking the above intuitions.

Bandit 3: EXP3.P Algorithm

Input : $\eta > 0$, $\gamma, \beta \in [0, 1]$

Initially at time $t = 1$, let p_1 be the uniform distribution over $\{1, \ldots, n\}$

for $t = 1, 2, \cdots, T$ **do**

Player chooses an arm $I_t \in [n]$ from the probability distribution p_t

For each arm $i = 1, \cdots, n$ compute the estimated biased gain

$$\tilde{g}_{i,t} = \frac{g_{i,t}\mathbb{1}\{I_t = i\} + \beta}{p_{i,t}} \tag{7}$$

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

Compute the new probability distribution over arms p_{t+1}:

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

end

By specifically choosing the parameters η, β, γ, we can achieve a high probability bound for the total regret at time T. The following are two different choices of the parameters.

Theorem 3. *For the EXP3.P algorithm, for any given confidence $\delta \in (0, 1)$, we set*

$$\beta = \sqrt{\frac{\log(n\delta^{-1})}{nT}}, \eta = 0.95\sqrt{\frac{\log(n)}{nT}}, \gamma = 1.05\sqrt{\frac{n\log(n)}{T}}.$$

Then

$$R(T) \leq 5.15\sqrt{nT\log(n\delta^{-1})}$$

with probability at least $1-\delta$.

Proof. Let $\mathbb{E}_t[X]$ be the expectation of random variable X conditioned on $I_1, \cdots, I_{t-1}$. We can easily calculate that $\mathbb{E}_t\left[\beta\frac{g_{i,t}\mathbb{1}\{I_t=i\}}{p_{i,t}}\right] = \beta g_{i,t}$. Using Taylor expansion we know $\exp(x) \leq 1+x+x^2$ and $\exp(x) \geq 1+x$ for $x \leq 1$. Hence,

$$\begin{aligned}
&\mathbb{E}_t\left[\exp((\beta g_{i,t} - \beta\tilde{g}_{i,t}))\right] \\
=\ &\mathbb{E}_t\left[\exp\left(\beta g_{i,t} - \beta\frac{g_{i,t}\mathbb{1}\{I_t=i\}+\beta}{p_{i,t}}\right)\right] \\
\leq\ &\left(1+\mathbb{E}_t\left[\beta g_{i,t} - \beta\frac{g_{i,t}\mathbb{1}\{I_t=i\}}{p_{i,t}}\right] + \mathbb{E}_t\left[\beta g_{i,t} - \beta\frac{g_{i,t}\mathbb{1}\{I_t=i\}}{p_{i,t}}\right]^2\right)\exp\left(-\frac{\beta^2}{p_{i,t}}\right) && (\exp(x)\leq 1+x+x^2) \\
\leq\ &\left(1+0+\mathbb{E}_t\left[\beta\frac{g_{i,t}\mathbb{1}\{I_t=i\}}{p_{i,t}}\right]^2\right)\exp\left(-\frac{\beta^2}{p_{i,t}}\right) && \left(\mathbb{E}_t\left[\beta\frac{g_{i,t}\mathbb{1}\{I_t=i\}}{p_{i,t}}\right] = \beta g_{i,t}\right) \\
\leq\ &\left(1+\beta^2\frac{g_{i,t}^2}{p_{i,t}}\right)\exp\left(-\frac{\beta^2}{p_{i,t}}\right) && (\text{definition}) \\
\leq\ &\exp\left(\frac{\beta^2 g_{i,t}^2}{p_{i,t}}\right)\exp\left(-\frac{\beta^2}{p_{i,t}}\right) && (\exp(x)\geq 1+x) \\
\leq\ &1
\end{aligned}$$

Adding all time $t \leq T$ together (multiplying the $\mathbb{E}_t$ for $t = 1, \cdots, T$)

$$\mathbb{E}\left[\exp\left(\beta\sum_{t=1}^{T} g_{i,t} - \beta\sum_{t=1}^{T}\frac{g_{i,t}\mathbb{1}\{I_t=i\}+\beta}{p_{i,t}}\right)\right] \leq 1$$

Then we apply Markov's inequality $\mathbb{P}(X > \log(\delta^{-1})) \leq \delta\mathbb{E}[e^X]$

$$\beta\sum_{t=1}^{T} g_{i,t} - \beta\sum_{t=1}^{T}\frac{g_{i,t}\mathbb{1}\{I_t=i\}+\beta}{p_{i,t}} \leq \log(\delta^{-1}) \tag{8}$$

with probability at least $1-\delta$. Recall that the total regret at time T is defined as

$$R(T) = \max_i \sum_{t=1}^{T} g_{i,t} - \sum_{t=1}^{T} g_{I_t,t}$$

To simplify, we assign $k = \arg\max_i \sum_{t=1}^{T} g_{i,t}$. Given Equation 7, we know

$$\mathbb{E}_{i\sim p_t}[\tilde{g}_{i,t}] = g_{I_t,t} + \beta n$$

Hence,

$$R(T) = \beta nT + \sum_{t=1}^{T} g_{k,t} - \sum_{t=1}^{T}\mathbb{E}_{i\sim p_t}[\tilde{g}_{i,t}]$$

Now we focus on the last term $-\mathbb{E}_{i\sim p_t}[\tilde{g}_{i,t}]$. We let u be an independent random variable uniformly distributed over the n arms and $\omega_t = \frac{p_t - u\gamma}{1-\gamma}$, the distribution induced by EXP3.P at time t without the mixing[1]. Hence $\frac{\omega_t}{p_t - u\gamma} = \frac{1}{1-\gamma}$.

$$\begin{aligned}
-\mathbb{E}_{i\sim p_t}[\tilde{g}_{i,t}] &= -(1-\gamma)\mathbb{E}_{i\sim\omega_t}[\tilde{g}_{i,t}] - \gamma\mathbb{E}_{i\sim u}[\tilde{g}_{i,t}] \\
&= (1-\gamma)\left(\frac{1}{\eta}\log\mathbb{E}_{i\sim\omega_t}[\exp(\eta(\tilde{g}_{i,t} - \mathbb{E}_{k\sim\omega_t}[\tilde{g}_{k,t}]))] - \frac{1}{\eta}\log\mathbb{E}_{i\sim\omega_t}[\exp(\eta\tilde{g}_{i,t})]\right) - \gamma\mathbb{E}_{i\sim u}[\tilde{g}_{i,t}]
\end{aligned}$$

Given $\log(x) \le x - 1$, $\exp(x) \le 1 + x + x^2$ for all $x \le 1$, and $(1+\beta)\eta n \le \gamma$, we have

$$\begin{aligned}
\log\mathbb{E}_{i\sim\omega_t}[\exp(\eta(\tilde{g}_{i,t} - \mathbb{E}_{k\sim p_t}[\tilde{g}_{k,t}]))] &= \log\mathbb{E}_{i\sim\omega_t}[\exp(\eta\tilde{g}_{i,t})] - \eta\mathbb{E}_{k\sim p_t}[\tilde{g}_{i,t}] \\
&\le \mathbb{E}_{i\sim\omega_t}[\exp(\eta\tilde{g}_{i,t}) - 1 - \eta\tilde{g}_{i,t}] \\
&\le \mathbb{E}_{i\sim\omega_t}[\eta^2\tilde{g}_{i,t}^2] \\
&\le \frac{1}{1-\gamma}\eta^2\sum_{i=1}^n p_{i,t}\tilde{g}_{i,t}^2 && \left(\frac{\omega_{i,t}}{p_{i,t}} \le \frac{\omega_{i,t}}{p_{i,t} - u\gamma} = \frac{1}{1-\gamma}\right) \\
&\le \frac{\max_i g_{i,t}\mathbb{1}\{I_t = i\} + \beta}{1-\gamma}\eta^2\sum_{i-1}^n \tilde{g}_{i,t} && \left(\tilde{g}_{i,t} = \frac{g_{i,t}\mathbb{1}\{I_t = i\} + \beta}{p_{i,t}}\right) \\
&\le \frac{1+\beta}{1-\gamma}\eta^2\sum_{i=1}^n \tilde{g}_{i,t} && (g_{i,t} \in [0,1])
\end{aligned}$$

Finally, we combine the above two inequalities and sum up $\mathbb{E}_{i\sim\omega_t}[\tilde{g}_{i,t}]$ for each $t < T$

$$-\sum_{t=1}^T \mathbb{E}_{i\sim p_t}[\tilde{g}_{i,t}] \le (1+\beta)\eta\sum_{t=1}^T\sum_{i=1}^n \tilde{g}_{i,t} - \frac{1-\gamma}{\eta}\sum_{t=1}^T \log\left(\sum_{i=1}^n \omega_{i,t}\exp(\eta\tilde{g}_{i,t})\right)$$

Recall that[2]

$$\omega_{i,t} = \frac{\exp(\eta\tilde{G}_{i,t-1})}{\sum_{k=1}^n \exp(\eta\tilde{G}_{k,t-1})},$$

$\tilde{G}_{i,0} = 0$ from EXP3,

$$\begin{aligned}
-\sum_{t=1}^T \mathbb{E}_{i\sim p_t}[\tilde{g}_{i,t}] &\le (1+\beta)\eta\sum_{t=1}^T\sum_{i=1}^n \tilde{g}_{i,t} - \frac{1-\gamma}{\eta}\log\left(\prod_{t=1}^T \frac{\sum_{i=1}^n \exp(\eta\tilde{G}_{i,t})}{\sum_{i=1}^n \exp(\eta\tilde{G}_{i,t-1})}\right) \\
&= (1+\beta)\eta\sum_{t=1}^T\sum_{i=1}^n \tilde{g}_{i,t} - \frac{1-\gamma}{\eta}\log\left(\frac{\sum_{i=1}^n \exp(\eta\tilde{G}_{i,T})}{\sum_{i=1}^n \exp(\eta\tilde{G}_{i,0})}\right) \\
&\le (1+\beta)\eta n\max_j \tilde{G}_{j,T} + \frac{\log(n)}{\eta} - \frac{1-\gamma}{\eta}\log\left(\sum_{i=1}^n \exp(\eta\tilde{G}_{i,T})\right) \\
&\le (1+\beta)\eta n\max_j \tilde{G}_{j,T} + \frac{\log(n)}{\eta} - (1-\gamma)\max_j \tilde{G}_{i,T} \qquad (9) \\
&\le -(1-\gamma-(1+\beta)\eta n)\max_j\sum_{t=1}^T g_{j,t} + \frac{\log(n\delta^{-1})}{\beta} + \frac{\log(n)}{\eta},
\end{aligned}$$

[1] The equation for ω_t in [2] is wrong.

[2] The corresponding equation in [2] is wrong.

where in (9) we used the fact that for any real numbers $a_1, \ldots, a_n$,

$$\log\left(\sum_{i=1}^{n}\exp(a_i)\right) = \log\left(\sum_{i=1}^{n}\exp(a_i - \max_j a_j)\right) + \max_j a_j \geq \log\left(\exp(\max_j a_j - \max_j a_j)\right) + \max_j a_j \geq \max_j a_j.$$

[3] We combine Equation 8 with this, resulting in

$$R(T) \leq \beta n T + \gamma T + (1+\beta)\eta n T + \frac{\log(n\delta^{-1})}{\beta} + \frac{\log(n)}{\eta} \tag{10}$$

with probability at least $1-\delta$.

We put

$$\beta = \sqrt{\frac{\log(n\delta^{-1})}{nT}}, \eta = 0.95\sqrt{\frac{\log(n)}{nT}}, \gamma = 1.05\sqrt{\frac{n\log(n)}{T}}$$

into Equation 10 and yield the claimed bound.

□

One problem of this is that we have to know the confidence δ before choosing the parameters. This makes it difficult to compute the expectation of regret. Here, we set a different set of parameters to address this problem.

Theorem 4. *For*

$$\beta = \sqrt{\frac{\log(n)}{nT}}, \eta = 0.95\sqrt{\frac{\log(n)}{nT}}, \gamma = 1.05\sqrt{\frac{n\log(n)}{T}},$$

then

$$R(T) \leq \sqrt{\frac{nT}{\log(n)}}\log(\delta^{-1}) + 5.15\sqrt{nT\log(n)} \tag{11}$$

with probability at least $1-\delta$.

Proof. Similarly using Equation 10 we yield this claim. □

Intuitively, if we use EXP3.P algorithm rather than the EXP3 algorithm, we might have a higher expected regret since we are trading exploitation with exploration. We extend Theorem 4 and have the following theorem

Theorem 5. *For*

$$\beta = \sqrt{\frac{\log(n)}{nT}}, \eta = 0.95\sqrt{\frac{\log(n)}{nT}}, \gamma = 1.05\sqrt{\frac{n\log(n)}{T}},$$

the expected regret of EXP3.P algorithm is

$$\mathbb{E}[R(T)] \leq 5.15\sqrt{nT\log(n)} + \sqrt{\frac{n\log(n)}{T}}$$

Proof. We can integrate the Equation 11 over δ to calculate the expectation. Specifically, let

$$W = \sqrt{\frac{\log(n)}{nT}}(R(T) - 5.15\sqrt{nT\log(n)})$$

We have

[3] In [2], the last term of the second line, $\log(\sum_{i=1}^{n}\exp(\eta\tilde{G}_{i,T}))$, is mistakenly written as $\log(\sum_{i=1}^{T}\exp(\eta\tilde{G}_{i,n}))$.

$$\mathbb{E}[W] \leq \int_0^1 \frac{1}{\delta}\mathbb{P}(W > \log(\frac{1}{\delta}))d\delta \leq 1$$

Then we get

$$\mathbb{E}[R(T)] \leq 5.15\sqrt{nT\log(n)} + \sqrt{\frac{n\log(n)}{T}}$$

□

Hence, we got a slightly higher expected regret using EXP3.P algorithm than using EXP3 algorithm.

4 Adaptive Bounds

The bound we have derived so far has its weakness: it holds over all possible adversarial assignments of gains to arms. Thus, it is natural for us to ask if it is possible to have strategies with minimax optimal regret, but also with much smaller regret when the loss sequence is not the worst case.

The first adaptive bound in this direction was proven by [3]. He showed that for the *gain version* (compare to losses) of the problem and against an *oblivious adversary*, EXP3 algorithm has a pseudo-regret of $O\left(\sqrt{nG_T^*}\right)$, where n is the number of arms, and $G_T^* \leq T$ is the maximal cumulative reward of the optimal arm after T rounds. This result was further improved by Audibert and Bubeck [4]. Denote by $g_{i,t}$ the reward (or gain) of arm i at time step t. He shows that by using the gain estimate:

$$\tilde{g}_{i,t} = -\frac{\mathbb{1}\{I_t = i\}}{\beta}\log\left(1 - \frac{\beta g_{i,t}}{p_{i,t}}\right)$$

One can bound the regret with high probability again *any* adversary.

Hazan and Kale[5] proved from another direction that one can attain regret of $O(\sqrt{\sum_{i=1}^n V_{i,T}})$ in the full information setting, where

$$V_{i,T} = \sum_{t=1}^{T}\left(\ell_{i,t} - \frac{1}{T}\sum_{s=1}^{T}\ell_{i,s}\right)^2$$

is the total variation of the loss for arm i. The main ingredient of this analysis is that they used the "reservoir sampling" procedure. Reservoir sampling is a family of randomized algorithms for randomly choosing a sample of k items from a list S containing n items, where either $n \gg k$ or n is an unknown number. Typically n is large enough that the list doesn't fit into main memory. For example, if we are given a big array of numbers, and we need to write an efficient function to randomly select k numbers where $1 \ll k \ll n$. A simple solution is to create a reservoir (i.e., an array) of maximum size k and randomly select an item from stream $\{0, \ldots, n-1\}$. If the selected item is not previously selected, then put it in the reservoir. The time complexity of this algorithm is $O(k^2)$. Readers may refer to [5] for more details.

5 EXP3++ Algorithm

Seldin and Slivkins [6] present an algorithm, EXP3++ that is applicable to both stochastic and non-stochastic multi-armed bandit problems without distinguishing between them. This algorithm is proved to achieve almost optimal regret in both setting, without the knowledge of the environment setting in advance: if the enviroment happens to be adversarial, the proposed algorithm is just a factor of 2 worse than the performance of the EXP3 algorithm which we will show later; if the enviroment happens to be stochastic, the performance of the algorithm is comparable to UCB1 proposed by [3].

The EXP3++ algorithm is based on augmentation of the EXP3 algorithm with a new control lever in the form of exploration parameters that are tailored individually for each arm. This algorithm combines

two *independent* mechanisms. The first mechanism controls the performance of the algorithm in adversarial environments through a standard EXP3-like playing strategy in the form of a Gibbs distribution over actions. The second mechanism exploits the residual degree of exploration freedom for detection and exploitation of suboptimality gaps.

To be more specific, this proposed algorithm has two control levers: the learning rate η_t and the exploration parameter $\xi_t(i)$. In each round t of the game, the algorithm chooses one action I_t among n possible arms according to probability $\tilde{\rho}_t(i)$, which is updated everyone based on the observed cumulative losses $\tilde{L}_{i,t-1}$, the learning rate η_t, as well as the exploration parameter $\xi_t(i)$. The algorithm proceeds as the following.

Bandit 4: EXP3++ Algorithm

$\forall i : L_0(i) = 0$
for $t = 1, 2, \dots T$ **do**
  $\beta_t = \frac{1}{2}\sqrt{\frac{\log n}{tn}}$
  $\forall i : \varepsilon_t(i) = \min\left\{\frac{1}{2n}, \beta_t, \xi_t(i)\right\}$
  $\forall i : \rho_t(i) = e^{-\eta_t \tilde{L}_{i,t-1}} \Big/ \sum_{i'} e^{-\eta_t \tilde{L}_{i',t-1}}$
  $\forall i : \tilde{\rho}_t(i) = \sum_{i'} (1 - \varepsilon_t(i'))\rho_t(i) + \varepsilon_t(i)$
  Draw action I_t according to $\tilde{\rho}_t$ and play it.
  Observe and suffer the loss $\ell_{I_t,t}$.
  $\forall i : \tilde{\ell}_{i,t} = \frac{\ell_{I_t,t}}{\tilde{\rho}_t(i)}\mathbb{1}\{I_t = i\}$.
  $\forall i : \tilde{L}_{i,t} = \tilde{L}_{i,t-1} + \tilde{\ell}_{i,t}$
end

Note that the EXP3 algorithm is a special case of EXP3++ with $\eta_t = 2\beta_t$ and $\xi_t(i) = 0$. The main innovation of this algorithm is the introduction of the exploration parameter η_t. This parameter is tuned individually for each arm based on its past performance. This algorithm can be proved to have the following properties under adversarial regime and stochastic regime:

Adversarial Regime By tunning only η_t, it is sufficient to bound regret in adversarial setting. Specifically, we have:

Theorem 6. *For any $\eta_t = \beta_t$ and any $\xi_t(i) \geq 0$ the regret of EXP3++ for any t satisfies:*

$$R(t) \leq 4\sqrt{nT \log n}$$

This regret is just a factor of 2 worse than the regret of EXP3 (i.e., $2\sqrt{nT \log n}$)

Stochastic Regime To control the regret of the algorithm in the stochastic regime, it suffices to tune the exploration parameters $\eta_t \geq \beta_t$.

Adversarial regime with a gap An adversarial regime is named by *an adversarial regime with a gap* if there exists a round τ and an arm i^*_τ that persists to be the best arm in hindsight for all rounds $\tau' > \tau$. If we define a deterministic gap $\Delta(t, i)$ for arm i at round τ as:

$$\Delta(\tau, i) = \min_{t \geq \tau}\left\{\frac{1}{t}(\lambda_t(i) - \lambda_t(i^*_\tau))\right\}$$

where $\lambda_t(i)$ is the cumulative loss of arm i, we have the following theorem.

Theorem 7. *For each arm i, we define the gap $\Delta(i) = \mu(i) - \mu(i^*)$. Assume that the gaps $\Delta(i)$ are known. For any choice of $\eta_t \geq \beta_t$ and any $c \geq 18$,the regret of EXP3++ with $\xi_t(i) = \frac{c \log (t\Delta(i))^2}{t\Delta(i)^2}$ in the stochastic*

regime satisfies:

$$R(t) \leq \sum_i O\left(\frac{\log(t)^2}{\Delta(i)}\right) + \sum_i \tilde{O}\left(\frac{n}{\Delta(i)^3}\right)$$

For the proof of the above theorem, readers may refer to [6] for more details.

6 Switching Adversaries

Generally, we expect to obtain a sublinear regret, implying that the per-round expected regret, $\mathbb{E}[R(T)]/T$ tends to zero. But arbitrary adaptive adversaries can easily force the regret to grow linearly. In order to obtain a sublinear regret, we need to focus on reasonably weaker adversaries, which have constraints on the loss functions they can generate.

The weakest version is oblivious adversary, which determines the loss on round t based only on the current action I_t. A stronger version is the *oblivious adversary with switching costs*. This adversary is similar to the oblivious adversary defined above, but charges the player an additional switching cost of 1 whenever $I_t \neq I_{t-1}$. This setting of considering switching cost among arms is very natural in many settings. Consider a single-stock investor, who faces a commission cost at every trade. If the investor keeps his position in a stock for multiple trading days, he is exempt from any additional fees, but when he sells one stock and buys another, he incurs a fixed commission. More generally, this setting allows us to capture any situation where choosing a different action involves a costly change of state.

This setting can be formulized as the following. This adversary defines his sequence of loss functions in two steps: first he chooses an oblivious sequence of loss functions, $\ell_1, \ell_2, \ldots$. Then, he sets switching cost $f_1(x) = \ell_1(x)$, and

$$\forall t \geq 2, f_t(I_{1:t}) = \ell_t(I_t) + \mathbb{1}\left\{I_t \neq I_{t-1}\right\}.$$

The switching cost f_t above is defined as a function of I_t and I_{t-1}. A more general case, called *an adaptive adversary with a memory of 1* whose loss functions can depend on the *previous action* in an arbitrary way.

This adversary can also be further strengthened to *bounded memory adaptive adversary*[7], which has a bounded memory of an arbitrary size. Compared to adaptive adversary with a memory of 1, this adversary is allowed to set his loss function based on the players m most recent past actions, where m is a predefined parameter.

The above adversaries have been studied, some of the current state-of-the-art results are as the following. FLL algorithm proposed by [8] designed for the switching costs setting guarantees that under full-information feedback the oblivious component of the player's expected regret (without the switching costs), as well as the expected number of switches, is upper bounded by $O(T)$. The work in [9] focuses on the bounded memory adversary with bandit feedback and guarantees an expected regret of $O(T^{2/3})$. This bound naturally extends to the full-information setting.

The work by [10], studied the problem of prediction with expert advice against different types of adversaries, ranging from the oblivious adversary to the general adaptive adversary, as mentioned above. They proved a upper bound of:

- $O(\sqrt{T})$ for an oblivious adversary with switching costs, with full-information feedback;
- $O(\sqrt{T})$ for an oblivious adversary with bandit feedback;
- $O(T^{2/3})$for a bounded memory adversary with bandit feedback.

They also proved a lower bound of:

- $\Omega(T^{2/3})$ with switching costs and bandit feedback;
- $\Omega(T^{2/3})$ with bounded memory and full-information feedback.

The above bounds suggest that predicting with bandit feedback is strictly more difficult than predicting with full-information feedback even in terms of the dependence on T, and even on small finite action sets. Also, in the full-information setting, predicting against a switching-cost adversary is strictly easier than predicting against an arbitrary adversary with a bounded memory.

References

[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[2] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[3] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[4] Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(Oct):2785–2836, 2010.

[5] Elad Hazan and Satyen Kale. Better algorithms for benign bandits. *Journal of Machine Learning Research*, 12(Apr):1287–1311, 2011.

[6] Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *International Conference on Machine Learning*, pages 1287–1295, 2014.

[7] Chris Mesterharm. On-line learning with delayed label feedback. In *International Conference on Algorithmic Learning Theory*, pages 399–413. Springer, 2005.

[8] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[9] Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. *arXiv preprint arXiv:1206.6400*, 2012.

[10] Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems*, pages 1160–1168, 2013.