**Machine Learning** **Spring 2021**

# Lecture 10: Online Learning, Multi-arm Bandits Problem

*Lecturer: Liwei Wang* *Scribe: Zijian Ding, Yifan Chen, Suchen Liu, Xin Xu*



## 10.1 More on Randomized Weight Updating

In this problem, we have an adversary and a learner.
The learner is asked to learn a distribution $\vec{x}$ over [**N**].
The following interaction repeats:

**For** $t = 1, 2, \cdots$:
1. Adversary provides $\vec{f_t} \in \{0,1\}^N$
2. Learner predictes $\langle \vec{f_t}, \vec{x} \rangle$ and gives its answer to adversary
3. Adversary reveals the answer for $\langle \vec{f_t}, \vec{x} \rangle$

We would like to design a mechanism so that the leaner makes $\delta - error$ at most finite times. More precisely, $\exists T$, $\forall t > T$, $|R_t - \langle f_t, x \rangle| \leq \delta$, where $R_t$ is the prediction made by learner at time $t$.

The following algorithm bounded $T$ to $O(\frac{\log N}{\delta^2})$.

**Algorithm 10.1.1:** Randomized Weight Updating

Initialize $x_1 = (\frac{1}{N} \cdots \frac{1}{N})$, set parameter $\epsilon \in (0, \delta]$
**for** $t = 1, 2, \cdots$ **do**
  **if** $\langle f_t, x_t \rangle$-$\langle f_t, x \rangle \geq \delta$ **then**
    $x_{t+1}[i] \leftarrow (1+\epsilon) x_t[i], \quad \forall f_{t,i} = 0$
    $x_{t+1}[i] \leftarrow x_t[i], \quad \forall f_{t,i} = 1$
    Normalize $x_{t+1}$
  **end**
  **else if** $\langle f_t, x_t \rangle$-$\langle f_t, x \rangle \leq -\delta$ **then**
    $x_{t+1}[i] \leftarrow (1+\epsilon) x_t[i], \quad \forall f_{t,i} = 1$
    $x_{t+1}[i] \leftarrow x_t[i], \quad \forall f_{t,i} = 0$
    Normalize $x_{t+1}$
  **end**
  **else**
    $x_{t+1} \leftarrow x_t$
  **end**
**end**

We now prove the efficiency of the algorithm.

**Lemma 10.1** *The learner update at most* $\frac{2 \ln N}{\epsilon \delta}$ *times.*

**Proof:** Write $\{i \in [N] | f_t[i] = 0\}$ as $F_0^t$ and $[N]/F_0^t$ as $F_1^t$. When $\langle f_t, x_t\rangle - \langle f_t, x\rangle \geq \delta$, we have

$$\langle f_t, x_t\rangle - \langle f_t, x\rangle = \sum_{i\in[N]} f_t[i](x_t[i] - x[i]) = \sum_{i\in F_1^t} x_t[i] - x[i] \geq \delta.$$

For $\sum_{i=1}^N x[i] = 1 = \sum_{i=1}^N x_t[i]$, we get

$$\sum_{i\in F_0^t} x_t[i] - x[i] \leq -\delta$$

Consider potential function $D(x\|x_t)$.

$$\begin{aligned} D(x\|x_{t+1}) - D(x\|x_t) =& \sum_{i=1}^N x[i] \ln \frac{x_t[i]}{x_{t+1}[i]} \\ =& \sum_{F_0^t} x[i] \ln \frac{1+\epsilon\sum_{F_0^t} x_t[i]}{1+\epsilon} + \sum_{F_1^t} x[i] \ln(1+\epsilon\sum_{F_0^t} x_t[i]) \\ =& \ln(1+\epsilon\sum_{F_0^t} x_t[i]) - \sum_{F_0^t} x[i]\ln(1+\epsilon) \end{aligned}$$

For given $\delta$ and $\epsilon \leq \delta$, $\ln(1+\epsilon) > \epsilon(1-\delta/2)$. And $\ln(1+\epsilon\sum_{F_0^t} x_t[i]) \leq \epsilon\sum_{F_0^t} x_t[i] \leq \epsilon\sum_{F_0^t} x[i] - \epsilon\delta$. Thus,

$$D(x\|x_{t+1}) - D(x\|x_t) \leq \epsilon\sum_{F_0^t} x[i] - \epsilon\delta - \sum_{F_0^t} x[i]\epsilon(1-\delta/2) \leq -\epsilon\delta/2$$

It's similar when $\langle f_t, x_t\rangle - \langle f_t, x\rangle \leq -\delta$, we have

$$\sum_{i\in F_1^t} x_t[i] - x[i] \leq -\delta$$

and

$$\begin{aligned} D(x\|x_{t+1}) - D(x\|x_t) =& \sum_{F_1^t} x[i] \ln \frac{1+\epsilon\sum_{F_1^t} x_t[i]}{1+\epsilon} + \sum_{F_0^t} x[i] \ln(1+\epsilon\sum_{F_1^t} x_t[i]) \\ =& \ln(1+\epsilon\sum_{F_1^t} x_t[i]) - \sum_{F_1^t} x[i]\ln(1+\epsilon) \\ \leq& \epsilon\sum_{F_1^t} x[i] - \epsilon\delta - \sum_{F_1^t} x[i]\epsilon(1-\delta/2) \\ \leq& -\epsilon\delta/2 \end{aligned}$$

This means that each update of the potential function reduces at least $\epsilon\delta/2$. Due to $D(x\|x_t) \geq 0$, and $D(x\|x_1) = \sum_{i\in[N]} x[i] \ln \frac{x[i]}{1/N} = \sum_{i\in[N]} x[i]\ln x[i] + \ln N \leq \ln N$, the learner update at most $\frac{2\ln N}{\epsilon\delta}$ times. ■

## 10.2 Multi-arm Bandits Problem

### 10.2.1 Setting

In a $k$-slot machine, each arm has a probability loss $\mu_i (i \in [k])$. And the game has $T$ rounds in total, at each step $t$, the player chooses an arm $a_t$, and observes a loss $l_t(a_t)$. Be ignorant of $\mu_i$, the player wants

to minimize the total loss in expectation. To be specific, for each arm $i \in [k]$, $l_1(a_i), \cdots, l_T(a_i)$ are i.i.d. random variables under some distribution $\mathcal{D}_i$ with mean $\mu_i$. Define regret $R_T$ as

$$R_T \coloneqq \mathbb{E}_{\mathcal{A}} \left[ \sum_{t=1}^{T} \mu(a_t) - \mu^* \right]$$

where we replace $l_t(a_t)$ with mean $\mu(a_t)$ and define $\mu^* = \min_{i \in [k]} \mu_i$. Our goal is then to minimize $R_T$.

Multi-arm Bandits Problem is a simplified reinforcement learning problem. Because we don't know actual reward, the trade-off between exploration and exploitation is necessary.

### 10.2.2 UCB Algorithm

To minimize regret $R_T$, UCB algorithm gives an effective strategy, which is based on a simple principle: *Optimism in the face of uncertainty*. To be exact, after $t$ rounds, each arm has an estimated interval of loss (with high probability), and we just assume the lowest bound of the interval is the loss probability of the arm. The algorithm is called upper confidence bound because people used to consider the probability of win in the past. The implement of UCB algorithm is as follow.

**Algorithm 10.2.1:** UCB Algorithm

```
Initialization: n_0(a) = 0(∀a ∈ [k]);
n_t(a) represents #times arm a is pulled at time t;
μ̂_t(a) represents the empirical loss of arm a at time t;
for t = 1, 2, ··· , T do
    For each arm a, compute UCB_t(a) = μ̂_{t−1}(a) − sqrt(ln T / n_{t−1}(a)) ;
    Pull the arm a_t = arg min_{a∈[k]} UCB_t(a) ;
    Update n_t(a_t), μ̂_t(a_t) ;
end
```

Note that at the very beginning, if there exists $n_{t-1}(a) = 0$, then $\mathrm{UCB}_t(a) = -\infty$. Therefore the algorithm tends to explore arm $a$. Now consider a substitute. If at each step of UCB algorithm, we choose the minimum among upper bounds instead of among lower bounds, then the algorithm prefers exploitation rather than exploration. Hence it may stick in some local optima.

### 10.2.3 Upper Confidence Bound

W.L.O.G. Suppose $\mu_1 = \min_{i \in [k]} \mu_i$, which means that the first arm has the lowest average loss.

**Theorem 10.2** *Assume $\mu_1 \leq \mu_2, \cdots, \mu_k$, the regret of UCB algorithm can be bounded as:*

$$R_T = \mathbb{E}_{\mathcal{A}} \left[ \sum_{t=1}^{T} \mu(a_t) - \sum_{t=1}^{T} \mu_1 \right] \leq \sum_{a:\Delta_a > 0} \left( \frac{16 \ln T}{\Delta_a} + 2\Delta_a \right) \tag{10.1}$$

*where $\Delta_a = \mu_a - \mu_1$ denotes the gap of average loss between arm $a$ and the optimal arm.*

**Proof:** First of all, note that $R_T$ can be written as

$$R_T = \mathbb{E}_{\mathcal{A}} \left[ \sum_{t=1}^{T} \mu(a_t) \right] - \sum_{t=1}^{T} \mu_1 = \sum_{a=1}^{k} \Delta_a \cdot \mathbb{E}_{\mathcal{A}}[n_T(a)]$$

Since the regret $R_T$ equals to

$$\sum_a (\#\text{times arm } a \text{ is pulled}) \times (\text{gap of loss between arm } a \text{ and the optimal arm})$$

which simply uses the technique of double counting.

Therefore we only need to prove that, for each sub-optimal arm $a(\Delta_a > 0)$, we have

$$\mathbb{E}_{\mathcal{A}}[n_T(a)] = O\left(\frac{\ln T}{\Delta_a^2}\right) + 2$$

■

**Lemma 10.3** *For each sub-optimal arm $a$ with $\Delta_a > 0$, we have*

$$\mathbb{E}_{\mathcal{A}}[n_T(a)] = O\left(\frac{\ln T}{\Delta_a^2}\right) + 2 \tag{10.2}$$

*where $n_T(a)$ denotes #times arm $a$ is pulled in $T$ rounds.*

**Proof:** By linearity of expectation, $\mathbb{E}_{\mathcal{A}}[n_T(a)]$ could be written as

$$\mathbb{E}_{\mathcal{A}}[n_T(a)] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[a \text{ is pulled at round } t]\right] = \sum_{t=1}^{T} \Pr[a \text{ is pulled at round } t]$$

Therefore for any $n$, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{A}}[n_T(a)] &= \sum_{t=1}^{T}\sum_{k=1}^{T} \Pr[a_t = a \wedge n_t(a) = k] \\
&\le n + \sum_{t=1}^{T}\sum_{k=n+1}^{T} \Pr[a_t = a \wedge n_t(a) = k] \\
&\le n + \sum_{t=n+1}^{T} \Pr[a_t = a \wedge n_{t-1}(a) \ge n]
\end{aligned}$$

Then our goal is to better estimate $\mathbb{E}_{\mathcal{A}}[n_T(a)]$ by choosing the parameter $n$ according to $\Delta_a$. Before fine-turning $n$, we present a proposition here.

**Proposition.** If sub-optimal arm $a(\Delta_a > 0)$ is pulled at time $t$, then we can claim that at least one of the following events occur:

1. $\hat{\mu}_{t-1}(1) > \mu_1 + \sqrt{\frac{\ln T}{n_{t-1}(1)}}$
2. $\hat{\mu}_{t-1}(a) < \mu_1 + \sqrt{\frac{\ln T}{n_{t-1}(a)}} = \mu_a - \Delta_a + \sqrt{\frac{\ln T}{n_{t-1}(a)}}$

Suppose neither of them occur, then $\hat{\mu}_{t-1}(1) - \sqrt{\frac{\ln T}{n_{t-1}(1)}} \le \mu_1 \le \hat{\mu}_{t-1}(a) - \sqrt{\frac{\ln T}{n_{t-1}(1)}}$. But in UCB algorithm, arm $a$ is choosen to be $a_t = \arg\min_{a\in[k]} \left(\hat{\mu}_{t-1}(a) - \sqrt{\frac{\ln T}{n_{t-1}(a)}}\right)$, raising a contradiction.

**Union Bound.** With the proposition above, we have

$$\begin{aligned}\Pr\left[a_t = a \wedge n_{t-1}(a) \geq n\right] \leq & \Pr\left[\hat{\mu}_{t-1}(1) > \mu_1 + \sqrt{\frac{\ln T}{n_{t-1}(1)}}\right] \\ & + \Pr\left[\hat{\mu}_{t-1}(a) < \mu_a - \Delta_a + \sqrt{\frac{\ln T}{n_{t-1}(a)}} \wedge n_{t-1}(a) \geq n\right]\end{aligned}$$

And sum over $t$,

$$\begin{aligned}\sum_{t=n+1}^{T} \Pr\left[a_t = a \wedge n_{t-1}(a) \geq n\right] \leq & \sum_{t=n+1}^{T} \Pr\left[\hat{\mu}_{t-1}(1) > \mu_1 + \sqrt{\frac{\ln T}{n_{t-1}(1)}}\right] \\ & + \sum_{t=n+1}^{T} \Pr\left[\hat{\mu}_{t-1}(a) < \mu_a - \Delta_a + \sqrt{\frac{\ln T}{n_{t-1}(a)}} \wedge n_{t-1}(a) \geq n\right]\end{aligned}$$

Deal with the two summations separately. For the first summation,

$$\begin{aligned}\sum_{t=n+1}^{T} \Pr\left[\hat{\mu}_{t-1}(1) > \mu_1 + \sqrt{\frac{\ln T}{n_{t-1}(1)}}\right] &\leq \sum_{t=n+1}^{T}\sum_{k=1}^{T} \Pr\left[\hat{\mu}_{t-1}(1) > \mu_1 + \sqrt{\frac{\ln T}{k}} \wedge n_{t-1}(1) = k\right] \\ &\leq \sum_{t=n+1}^{T}\sum_{k=1}^{T} \Pr\left[\hat{\mu}_{t-1}(1) > \mu_1 + \sqrt{\frac{\ln T}{k}} \,\middle|\, n_{t-1}(1) = k\right] \\ &\leq \sum_{t=n+1}^{T}\sum_{k=1}^{T} \exp\left(-2k\left(\sqrt{\frac{\ln T}{k}}\right)^2\right) \\ &= \sum_{t=n+1}^{T}\sum_{k=1}^{T} \frac{1}{T^2} \leq 1\end{aligned}$$

where $n_t(1)$ denotes the counter of pulling arm 1, and the third inequality is a Chernoff bound.

Similarly, for the second summation, we choose $n$ such that $\Delta_a = 2\sqrt{\frac{\ln T}{n}}$, and therefore

$$\begin{aligned}&\sum_{t=n+1}^{T} \Pr\left[\hat{\mu}_{t-1}(a) < \mu_a - \Delta_a + \sqrt{\frac{\ln T}{n_{t-1}(a)}} \wedge n_{t-1}(a) \geq n\right] \\ \leq &\sum_{t=n+1}^{T} \Pr\left[\hat{\mu}_{t-1}(a) < \mu_a - \sqrt{\frac{\ln T}{n_{t-1}(a)}} \wedge n_{t-1}(a) \geq n\right] \leq 1\end{aligned}$$

(the first inequality holds because $n_{t-1}(a) \geq n$). Combining these three inequalities, we have

$$\mathbb{E}_{\mathcal{A}}[n_T(a)] \leq n + 2 = O\left(\frac{\ln T}{\Delta_a^2}\right) + 2$$

which completes the proof.

■

**Note:** The bound given above is instance-dependent regret bound since it contains $\Delta_a$. If we treat $\Delta_a$ as a constant, then $R_T = O(k \ln T)$, which provides a better bound than $O(\sqrt{T})$ in Expert Advice Problem. But

$\Delta_a$ can be small for some arms ($\Delta_a$ can not be seen as a constant in this situation), and as a consequence, the regret bound given above will be loose. In fact, if $\Delta_a$ is treated carefully, we can have a better bound for $R_T$ than this theorem does.

Next, we will give a instance-independent regret bound which is worst-case regret bound of UCB.

**Theorem 10.4** *Assume $\Delta_a$ is bounded, then the worst-case regret bound of UCB is:*

$$R_T = O\left(\sqrt{T \cdot k \ln T}\right) \tag{10.3}$$

**Proof:** Divide the arms into two groups at the gap of $\delta = \sqrt{\frac{k \cdot \ln T}{T}}$. For $\Delta_a < \delta$, we have

$$R_T^{(1)} = \sum_{t=1}^{T} \sum_{a:\Delta_a < \delta} \Delta_a \cdot \Pr[a_t = a] \leq T \cdot \delta \leq \sqrt{T \cdot k \ln T}$$

And for $\Delta_a \geq \delta$, by applying the theorem above, we have

$$R_T^{(2)} \leq \sum_{a:\Delta_a \geq \delta} \left(\frac{16 \ln T}{\Delta_a} + 2\Delta_a\right) = O\left(k \cdot \frac{\ln T}{\delta}\right) = O\left(\sqrt{T \cdot k \ln T}\right)$$

where $\Delta_a$ is bounded. Therefore $R_T = R_T^{(1)} + R_T^{(2)} = O\left(\sqrt{T \cdot k \ln T}\right)$.

■

## References

[1] https://en.wikipedia.org/wiki/Multi-armed_bandit