IE613: Online Machine Learning **Jan-Apr 2018**

Lecture 25: Pure Exploration

Lecturer: M. K. Hanawal *Scribes: Chhavi Sharma*

So far, we have studied adversarial bandit setting and stochastic multi-armed bandit setting where at each round t, learner finds an arm which minimizes/maximizes the expected cumulative regret/reward while exploring and exploitation. In this lecture, we consider classical multi-armed bandit setting and find an ϵ-optimal arm with high probability at the end of round T (pure exploration) instead of finding an arm which maximizes the expected cumulative reward. In this setting, the sample complexity is the number of rounds required to achieve an ϵ- optimal arm with high probability (say 1 - δ). Here, the learner has two objectives. First objective is to minimize the number of rounds to get an arm which is ϵ close to the optimal arm with high probability and the second objective is to minimize the probability of not identifying the best arm for given number of T rounds. These objectives can be formulated as the following optimization problems.

Definition 25.1 ***Error constrained*** **(δ)**

Error constrained (δ) problem minimizes the number of rounds to get the best arm with a given probability atleast $1-\delta$.

$$\begin{aligned} &\text{minimize} \quad T \\ &\text{subject to} \quad \mathbb{P}_r\left(|\mu_{I_T} - \mu^\star| > \epsilon\right) \leq \delta \end{aligned}$$

where μ_{I_T} is the reward corresponding to the arm I_T returned by the algorithm after T rounds and $\mu^\star$ is the optimal reward.

Definition 25.2 ***Budget constrained (T)***

Budget constrained (T) minimizes the probability of not identifying the best arm with given number of rounds T.

$$\underset{I_T}{\text{minimize}} \quad \mathbb{P}_r\left(|\mu_{I_T} - \mu^\star| > \epsilon\right)$$

Definition 25.3 **(ϵ, δ)-*PAC Algorithm***

An algorithm is said to be (ϵ, δ)-PAC algorithm if it outputs an ϵ-optimal arm with probability atleast $1-\delta$, when it terminates i.e., $\mathbb{P}_r\left(|\mu_{I_T} - \mu^\star| \leq \epsilon\right) \geq 1-\delta$ where μ_{I_T} is the reward corresponding to the arm I_T returned by the algorithm when it terminates and $\mu^\star$ is the optimal reward.

25.1 Pure Exploration

A more general set up for pure exploration is given by algorithm 1.

Algorithm 1 Pseudo Code of Pure exploration

1: **Input:** K arms, δ
2: **for** t = 1,2,.... **do**
3: play i_t and observe reward from arm i_t.
4: If stopping criteria is met, stop and output the best arm, otherwise continue.

From algorithm 1, which arm i_t should be played by the learner and what is the stopping criteria is a legitimate question. A naive approach is given by the following Naive(ϵ, δ) algorithm.

Algorithm 2 Naive (ϵ, δ)

1: For every arm $i \in [K]$, sample it for $l = \frac{4}{2\epsilon^2} \log\left(\frac{2K}{\delta}\right)$ times.
2: $\forall\ i \in [K]$, estimate $\hat{\mu}_i$
3: Output $j = \arg\max_{i \in [K]} \hat{\mu}_i$

Theorem 25.4 *The algorithm Naive (ϵ, δ) is an (ϵ, δ)-PAC algorithm with sample complexity $O\left(\frac{K}{\epsilon^2} \log \frac{K}{\delta}\right)$.*

Proof: To prove that Naive (ϵ, δ) is an (ϵ, δ)-PAC algorithm, we show that $\mu_j < \mu^\star - \epsilon$ with probability atmost δ. Suppose $\mu_j < \mu^\star - \epsilon$. Therefore,

$$\begin{aligned}
\mathbb{P}_r\left(\hat{\mu}_j > \hat{\mu}_{i^\star}\right) &\leq \mathbb{P}_r\left(\hat{\mu}_j > \mu_j + \tfrac{\epsilon}{2} \text{ or } \hat{\mu}_{i^\star} < \mu_{i^\star} - \tfrac{\epsilon}{2}\right) \\
&\leq \mathbb{P}_r\left(\hat{\mu}_j > \mu_j + \tfrac{\epsilon}{2}\right) + \left(\hat{\mu}_{i^\star} < \mu_{i^\star} - \tfrac{\epsilon}{2}\right). \\
&\leq 2\exp^{-2l(\epsilon/2)^2} \text{ (From Hoeffdings inequality).} \\
&= 2\exp^{-\log 2K/\delta} = \tfrac{\delta}{K}.
\end{aligned}$$

$\mathbb{P}_r\left(\cup_{j=1}^{K} \hat{\mu}_j > \hat{\mu}_{i^\star}\right) \leq \sum_{j=1}^{K} \mathbb{P}_r\left(\hat{\mu}_j > \hat{\mu}_{i^\star}\right) \leq K \times \frac{\delta}{K} = \delta$. Therefore, $\mathbb{P}_r\left(|\mu_j - \mu^\star| > \epsilon\right) \leq \delta$. Hence, Naive (ϵ, δ) is an (ϵ, δ)-PAC algorithm.

Algorithm 2 takes $K \times \frac{4}{2\epsilon^2} \log\left(\frac{2K}{\delta}\right) = \frac{2K}{\epsilon^2} \log \frac{2K}{\delta} = \frac{2K}{\epsilon^2}\left(\log 2K - \log \delta\right) = \frac{2K}{\epsilon^2}\left(\log 2 + \log \frac{K}{\delta}\right) \approx \frac{2K}{\epsilon^2} \log \frac{K}{\delta}$ samples. The term $\log 2$ disappears because δ is very small and hence $\log \frac{K}{\delta}$ term dominates $\log 2$. Hence, sample complexity of Naive (ϵ, δ) is $O\left(\frac{K}{\epsilon^2} \log \frac{K}{\delta}\right)$.

■

Now, we will study some other algorithms which return $\epsilon-$ optimal arm with probability atleast $1 - \delta$.

25.1.1 Successive Elimination

Assumptions:

1. Expected rewards are known.

2. $\mu_1 > \mu_2 > .... > \mu_K$ and hence, $\Delta_i = \mu_1 - \mu_i \geq 0$.

In each round, SE eliminates the arm with minimum empirical mean and repeat it until we get a single arm with highest empirical mean. The pseudo code for Successive Elimination (SE) is given by algorithm 3.

Algorithm 3 Successive Elimination

1: **Input:** $\delta > 0$, bias of arms $\mu_1, \mu_2, ..., \mu_K$.
2: Set S = [K], $t_i = \left\lceil \frac{4}{\Delta_i^2} \log\left(\frac{K}{\delta}\right) \right\rceil$; $t_{K+1} = 0$.
3: $\forall\ a \in [K]$, $\hat{\mu_a} = 0$.
4: **for** i = 0,1,...,K-2 **do**
5: sample all $a \in [K]$ for $(t_{K-i} - t_{K-i+1})$ times.
6: estimate $\hat{\mu}_a\ \forall\ a\ \in S$.
7: $a_{min} = \arg\min_{a \in S} \hat{\mu}_a,\ S = S \setminus a_{min}$.
8: Output S.

Theorem 25.5 *The successive elimination with known biases is an $(0, \delta)$-PAC algorithm and its sample complexity is* $O\left(\left(\log \frac{K}{\delta}\right) \sum_{i=2}^{K} \frac{1}{\Delta_i^2}\right)$.

Proof:

At first round, K arms are sampled t_K times. In second round, K-1 arms are sampled $(t_{K-1} - t_K)$ times. In general, in round $1 \leq i < K$, we sample $K - i + 1$ arms $(t_{K-i} - t_{K-i+1})$ times. Therefore, the sample complexity is

$$Kt_K + (K-1)(t_{K-1} - t_K) + ... + 2(t_2 - t_3) = \sum_{i=2}^{K} t_i = \sum_{i=2}^{K} \left\lceil \frac{4}{\Delta_i^2} \log\left(\frac{K}{\delta}\right) \right\rceil = O\left(\left(\log \frac{K}{\delta}\right) \sum_{i=2}^{K} \frac{1}{\Delta_i^2}\right).$$

Now we will show that optimal arm 1 is not eliminated in any of the rounds i.e., $a_{\min}$ is not optimal arm in each round with high probability.

$$\sum_{i=0}^{K-2} \mathbb{P}\{\hat{\mu}_1 < \hat{\mu}_j\ \forall\ j\ \neq\ 1\}\ \leq\ \sum_{i=0}^{K-2} \mathbb{P}\{\hat{\mu}_1 < \hat{\mu}_j\ \forall\ j = 2, ..., K-i\}\ \leq\ \sum_{i=0}^{K-2} \mathbb{P}\{\hat{\mu}_1 < \hat{\mu}_{K-i}\} \leq \frac{\delta}{K}(K-1)\ \leq\ \delta.$$

Hence, in each round, best arm is removed with probability atmost δ. ■

Now we relax the assumption that reward of each arm is known in advance and modify the above SE algorithm such that it works in any set of biases. The modified algorithm is known as Successive Elimination with unknown biases.

Algorithm 4 Successive Elimination with unknown biases / SE(δ)

1: **Input:** $\delta > 0$.
2: Set S = [K].
3: **for** t = 1,2,.... **do**
4: sample all arms in S.
5: let $\hat{\mu}_{\max}(t) = \max_{a \in S} \hat{\mu}_a(t)$ and $\alpha_t = \sqrt{\frac{\log(Kt/\delta)}{2t}}$.
6: for every arm $a \in S$ such that $\hat{\mu}_{max}(t) - \hat{\mu}_a(t) \geq 2\alpha_t$, set $S = S \backslash a$.
7: t = t + 1.
8: If $|S| > 1$, Go to step 4, else output S.

Theorem 25.6 *SE(δ) is a $(0, \delta)$-PAC algorithm and with probability atleast $(1 - \delta)$, its arm sample complexity is bounded by* $O\left(\sum_{i=2}^{K} \frac{1}{\Delta_i^2} \log\left(\frac{K}{\delta \Delta_i}\right)\right)$.