EE 290 Theory of Multi-armed Bandits and Reinforcement Learning Lecture 4 - 1/28/2021

Lecture 4: Analysis of Successive Elimination and UCB Algorithm

Lecturer: Jiantao Jiao *Scribe: Matt Peng, Kathy Jang*

In this lecture, we finish our analysis of the successive elimination algorithm including discussion of the instance-dependent and instance-independent cases. We then analyze the UCB1 algorithm. We conclude this lecture with discussions of an improvement to successive elimination in "Phased Successive Elimination". Note, this lecture very briefly introduced the idea of lower bound. We defer the analysis and discussion of this to the lecture 5 note.[1].

1 Finishing the Analysis of Successive Elimination

We first recap some crucial information from the last lecture in section 1.1. We then dive into the discussion of instance-dependent and instance-independent bounds in section 1.2.

1.1 Recap

The last lecture presented the crucial property:

$$\Delta(a) \triangleq \mu(a^*) - \mu(a) > 0, \quad \Delta(a) \lesssim \sqrt{\frac{\log(T)}{n_T(a)}} \quad \text{for any sub-optimal arm } a \tag{1}$$

This property simply states that an arm played too many times cannot be bad and is quite close to the optimal arm - otherwise we would have already eliminated this sub-optimal arm.

The last lecture also showcased the pseudo-regret ($R(T)$) analysis:

$$R(T) = \sum_{t=1}^{T}(\mu^* - \mu_{\mathcal{A}_t}) = \sum_{a=1}^{K} n_T(a)\Delta(a) \leq \sum_{a=1}^{T} n_T(a)\sqrt{\frac{\log(T)}{n_T(a)}} \tag{2}$$

With a further upper-bound using:

$$\sum_{a=1}^{K} n_T(a) = T \tag{3}$$

There are some algorithms that may be able to remove the $\log(T)$ factor with further discussion and references in Lecture Note 3.

1.2 Instance-dependent and Instance-independent Bounds

Before presenting these bounds, we offer formal terminology. Consider a regret bound of the form $C \cdot f(T)$ where $f(\cdot)$ does not depend on the mean rewards μ, and the constant C does not depend on T. Then the regret bound is *instance-independent* if C does not depend on μ and *instance-dependent* otherwise [1].

INSTANCE-DEPENDENT: For the problem *instance-dependent* bound, we build off equation 1 and rearrange terms to get an upper-bound for $n_T(a)$:

$$n_T(a) \lesssim \frac{\log(T)}{(\Delta(a))^2} \tag{4}$$

[1] Refer to Lecture Note 2 for notation descriptions

This can informally be interpreted as if my sub-optimality is really bad, then definitely this arm would not have been pulled many times.

But we are more interested in upper-bounding the pseudo-regret ($R(T)$) so we first upper-bound the ($\Delta(a)n_T(a)$) term:

$$\Delta(a)n_T(a) \leq \Delta(a)\frac{\log(T)}{(\Delta(a))^2} = \frac{\log(T)}{\Delta(a)} \tag{5}$$

Now we plug this bound into equation 2 to obtain the regret bound:

$$R(T) = \sum_{a=1}^{K} n_T(a)\Delta(a) \lesssim \log(T)(\sum_{a:\Delta(a)>0} \frac{1}{\Delta(a)}) \tag{6}$$

Example: A quick look into asymptotics for the ratio of $\frac{R(T)}{\log(T)}$ as T goes to infinity, we quickly show a Gaussian case limit and showcase an exact constant for the bound in the case where each arm is Gaussian with variance of 1:

$$\lim_{x\to\infty} \frac{R(T)}{\log(T)} = \sum_{a:\Delta(a)>0} \frac{2}{\Delta(a)} \tag{7}$$

Algorithms have also been discovered that exactly achieve this constant.

INSTANCE-INDEPENDENT: We now use our *instance-dependent* bound to go to the *instance-independent* bound. To do this, we separate the arms into two cases:

1. The set of arms $a \in \mathcal{S}$ where $0 < \Delta(a) \leq \epsilon$
2. The set of arms $a \in \mathcal{S}^C$ where $\Delta(a) > \epsilon$

Then we obtain our *instance-independent* bound assuming the clean event:

$$R(T) = \sum_{a=1}^{K} n_T(a)\Delta(a) = \sum_{a\in\mathcal{S}} n_T(a)\Delta(a) + \sum_{a\in\mathcal{S}^C} n_T(a)\Delta(a) \leq \epsilon \sum_{a\in\mathcal{S}} n_T(a) + \log(T) \sum_{a\in\mathcal{S}^C} \frac{1}{\Delta(a)} \tag{8}$$

Now we look to simplify by upper-bounding individual terms. Using the information from equation 3, we know that the $\epsilon \sum_{a\in\mathcal{S}} n_T(a)$ term is bounded by ϵT. And then using information of the porperty of being in the $\mathcal{S}^C$ set and the fact that there are at most K arms, we can bound the $\log(T) \sum_{a\in\mathcal{S}^C} \frac{1}{\Delta(a)}$ by $\log(T)\frac{K}{\epsilon}$

In order to get the strictest upper bound, we use the ϵ that minimizes the right side of the $R(T)$ bound. We do this by setting $\epsilon T = \log(T)\frac{K}{\epsilon}$ which gives $\epsilon = \sqrt{\frac{K\log(T)}{T}}$. We now arrive at out final bound:

$$R(T) \lesssim \sqrt{KT\log(T)} \tag{9}$$

For more on the successive elimination algorithm, this reference proves useful [2].

2 UCB1 Algorithm

We now discuss the UCB1 Algorithm: *Optimism in the face of Uncertainty.*

```
1. Try each arm once
2. In each round $t$, pick $\arg\max_a UCB_t(a)$ where $UCB_t(a) = \bar{\mu}_t(a) + r_t(a)$ and $r_t(a) = \sqrt{\frac{2\log(T)}{n_t(a)}}$
```

Algorithm 1: UCB1 Algorithm

Remark Some intuition for this algorithm. In round t, an arm a is chosen due to its large $UCB_t(a)$. The $UCB_t(a)$ is large for a couple reasons: (i) the reward is high meaning $\bar{\mu}_t(a)$ is large; (ii) $r_t(a)$ is large which may imply an under-explored arm. Motivation for either reason of $UCB_t(a)$ being large is reasonable and showcases evidence that the arm is worth choosing by providing a natural way of summing up exploration and exploitation.

For more references on the UCB1 algorithm, this reference is worth reading [3].

2.1 Analysis of UCB1 Algorithm

We want to show that $\Delta(a) \leq 2\sqrt{\frac{2\log T}{n_t(a)}}$. To show this, we start by considering a clean event. By definition, we start with the expression on the left hand side. By definition of LCB (Lower Confidence Bound), which states that $\mu_a \geq \bar{\mu}_t(a) - r_t(a)$, we can arrange the terms to achieve this inequality.

$$\mu(a) + 2r_t(a) \geq \bar{\mu}_t(a) + r_t(a) \tag{10}$$

Note that this is exactly our definition of UCB. We thus rewrite as:

$$\mu(a) + 2r_t(a) \geq \bar{\mu}_t(a) + r_t(a) = \mathrm{UCB}_t(a) \tag{11}$$

Note also that at any time t, an action a was only picked if $\mathrm{UCB}_t(a) \geq \mathrm{UCB}_t(a^*)$. This is by definition. Thus, we have:

$$\mathrm{UCB}_t(a) \geq \mathrm{UCB}_t(a^*) \geq \mu^* \tag{12}$$

Combining lines (10) and (12), we see that:

$$\mu(a) + 2r_t(a) \geq \mu^*$$

Substituting in the definition of $\Delta(a)$, we get:

$$\begin{aligned} \Delta(a) &= \mu^* - \mu(a) && (13) \\ &\leq 2r_t(a) && (14) \\ &\leq 2\sqrt{\frac{2\log T}{n_t(a)}} && (15) \end{aligned}$$

We have thus shown how to achieve the upper bound.

3 Phased Successive Elimination

Phased Successive Elimination is a variation of Successive Elimination, notably producing an upper-bound that contains a log term that is a function of K, rather than T. This is significant for some forms of the bandit problem (think infinite-horizon problems with a small number of arms).

Algorithm 1: Phased Successive Elimination

```
Initialize $A_1 = \{1, 2, ..., K\}$ ;
Let $l$ denote phase index ;
for each phase $l$ do
    Pull each active arm $a \in A_l$ $m_l$ times ;
    Let $\bar{\mu}_l$ be the average reward for arm $a$ ;
    Update the active set: $A_{l+1} \triangleq \{a : \bar{\mu}_l + 2^{-l} \geq \max_{j \in A_l} \bar{\mu}_j\}$
end
```

The upper bound of pseudo-regret of this algorithm is $\sqrt{KT \ln K}$. Another interpretation of this algorithm is that we eliminate arms with $\Delta(a) \geq 2^{-l}$.

References

[1] A. Slivkins, "Introduction to multi-armed bandits," 2019.

[2] E. Even-Dar, S. Mannor, and Y. Mansour, "Pac bounds for multi-armed bandit and markov decision processes," in *Proceedings of the 15th Annual Conference on Computational Learning Theory*, ser. COLT '02. Berlin, Heidelberg: Springer-Verlag, 2002, p. 255–270.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2–3, p. 235–256, May 2002. [Online]. Available: https://doi.org/10.1023/A:1013689704352