Lecture 6: Idea behind lower bound of the Bandit Problem

25/01/2023

Lecturer: Prof. Subrahmanya Swamy Peruru *Scribe: Pranjal Praneel & Piyush Kumar*

Previously we discussed a general version of Thompson sampling based on the Gaussian TS algorithm. We discussed the fact that Bayesian Regret ($E_{env \approx prior}[R(T; env; algo)]$) is a useful metric when we are dealing with cases where we have prior knowledge about the underlying distribution by measuring the performance of our algorithm on environments that we are most likely to see according to our prior. In this lecture, we look at the ideas deriving the lower bound for the regret of the Bandit Problem.

1 Defining the Lower Bound

A natural idea behind defining the regret for the n-armed bandit problem would be that the best we can do for an algorithm is saying that for any algorithm, the regret is at least $\Omega(\sqrt{kT})$ over all environments. In more technical terms -

$$E[R(T; env; algo)] \geq \Omega(\sqrt{kT}) \, \forall \, algo, \, \forall \, env$$

However, this method for defining the lower bound is not correct. A simple counter-example is the case where all the arms have identical distribution. In such a case, we have no regret and hence the above inequality doesn't hold here.

1.1 Correct Statement

$$\forall \, algo, \exists \, env \; s.t \; E[R(T; env; algo)] \geq \Omega(\sqrt{kT})$$

This statement is a minmax lower bound which can also be written as

$$\min_{algo} \max_{env} E[R(T; env; algo)] \geq \Omega(\sqrt{kT})$$

The statements above state that for any given environment, we can always find an algorithm that can give us our lowest regret bound of $\Omega(\sqrt{kT})$. In this statement, using max over env takes care of the worst case environment and using min over algo takes care of the fact that even the best algorithm cannot do better than this regret.

1.2 Hypothesis Testing

For proving the lower bound on the regret, we first need to understand the concept of hypothesis testing. Let us assume that we are given T samples such that all i.i.d samples are either from $N(0, 1)$ or $N(\Delta, 1)$, i.e., pure samples from one of the given distributions D_1 and D_2, respectively.

We want to predict whether the samples are from D_1 or D_2. A very simple method is to look at the mean of the samples and decide. Compute $\bar{x}_T = \sum_{i=1}^{T} x_i$ and predict that samples belong to D_1 if $\bar{x}_T \leq \frac{\Delta}{2}$ or D_2 if $\bar{x}_T > \frac{\Delta}{2}$.

It can be shown that for any T, we can always have a Δ small enough such that there is a constant probability of making errors in our predictions.

$$if\ \Delta < \frac{1}{\sqrt{T}}\, is\, chosen,\ \mathbb{P}(Prediction\, error) \geq e^{\frac{-1}{8}}$$

2 Ideas Involded in Regret analysis

A Rough analysis of Regret (Not Rigorous): Let us say our environment has two arms with $D_{a_1} = \mathcal{N}(0,1)$ and $D_{a_2} = \mathcal{N}(\Delta,1)$ being the reward distributions. Also, assume that we are given T "free" trials during which we can explore without paying any regret. Even after those free trials, we could still get confused between the two arms with a constant probability (as stated in the previous section). Hence in the next 'T' actual rounds of playing we can incur an expected regret of

$$T \cdot \Delta \cdot \mathbb{P}(error) \simeq T \cdot \frac{1}{\sqrt{T}} \cdot e^{-\frac{1}{8}} = \mathcal{O}(\sqrt{T})$$

A Slightly better Argument: Since we are interested in bounds for the "worst-case" environment and showing that no algorithm can do better than $\mathcal{O}(\sqrt{kT})$ in such an environment, we are going to do the following: We are going to construct two environments and say that in one of the two environments, every algorithm will incur $\Omega(\sqrt{kT})$ regret.

The two environments are:

$$Env1 := \{\mathrm{a}_1 \sim \mathcal{N}(\Delta,1)\,\mathrm{and}\,\mathrm{a}_2 \sim \mathcal{N}(0,1)\}$$

$$Env2 := \{\mathrm{a}_1 \sim \mathcal{N}(\Delta,1)\,\mathrm{and}\,\mathrm{a}_2 \sim \mathcal{N}(2\Delta,1)\}$$

Observe that these two environments satisy two properties:

1. $Env1$ and $Env2$ are very similar (assuming Δ is small)
2. The best possible arm in $Env1$ is the worst arm in $Env2$.

Important Observations: The decision of what actions to take at time 't' depends only on the actions & rewards played/observed till time '(t-1)'. i.e. a(t) depends only on $(a_1, r_1, a_2, r_2, ..., a_{t-1}, r_{t-1})$

More importantly, Irrespective of what environment we are dealing with ($Env1$ or $Env2$), if the samples seen so far are the same then the algorithm will take the same actions in both environments.

Since we assume that both $Env1$ & $Env2$ are very similar, there is a good chance (i.e., a constant probability) of seeing the same samples, and hence the same expected number of times we sample actions (lets say arm a_1) in both environments.

$$i.e.,\ E_{env1}[n_t(a_1)] \simeq E_{env2}[n_t(a_1)] = x$$

Regret in Environment 1

$$E[R(T; env1)] = (T - E_{env1}[n_T(a_1)]) \cdot \Delta$$

Regret in Environment 2

$$E[R(T; env2)] = E_{env2}[n_T(a_1)] \cdot \Delta$$

Worst-case regret

$$\inf_x max\{\Delta T - \Delta x, \Delta x\}$$

It is easy to see that $x = \frac{T}{2}$ is the solution to the equation above. We get the worst case regret as $\frac{\Delta T}{2}$, since all our analysis is based on the fact that $\Delta \simeq \frac{1}{\sqrt{T}}$, it simplifies to $\frac{\sqrt{T}}{2}$. Also in the above statement, inf over x is like choosing the best possible algorithm.

Note: Some important things to keep in mind-

1. These arguments are not very rigorous since we assumed $E_{env1}[n_T(a_1)] \simeq E_{env2}[n_T(a_2)]$
2. In the next lecture, we will use the notion of KL divergence to make these arguments rigorous.

3 Some useful Theorems for Hypothesis Testing

We define $P \sim \mathcal{N}(0, 1)$ and $Q \sim \mathcal{N}(\Delta, 1)$.

Define an event A, upon occuring which we predict that samples belong to distribution P.

$$A = \{sample\, mean \leq a\}, \text{for some a.}$$

$$A^c = \{sample\, mean \geq a\}, \text{e.g. a} = \frac{\Delta}{2}$$

We further define the probability of 2 events.

$$P(A^c) = \mathbb{P}(\text{wrong predictions | samples from P})$$

$$Q(A) = \mathbb{P}(\text{wrong predictions | samples from Q})$$

If $P = Q$, then it is impossible to predict whether the sample came from P or Q as they are identical. Randomly deciding P or Q is the best we can do.

$$\text{i.e., } P(A^c) = \frac{1}{2} \text{ and } Q(A) = \frac{1}{2}$$

for $P = Q$ case.

However when P & Q are different, we have the following theorem which lower bounds these probabilities using $KL(P||Q)$.

Theorem: Bretagnolle-Huber inequality: Let P & Q be two distributions on the same sample space, then for any event A, we have

$$P(A^c) + Q(A) \geq \frac{1}{2} exp(-KL(P, Q)), \forall \text{A}$$

References

1. Chapter 13, 14 from "Bandit Algorithms" book by T. Lattimore, C. Szepesvari
2. Lectures notes by "Prof. Sanjay Shakkottai"