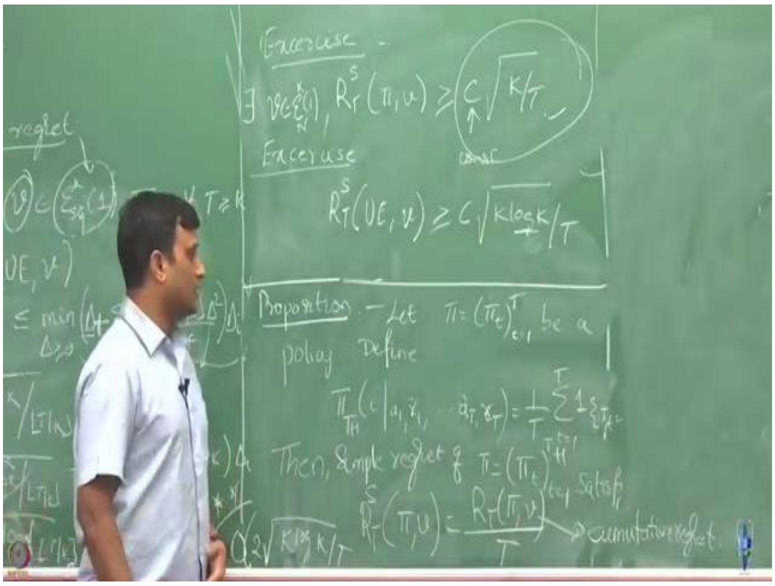
# Bandit Algorithm (Online Machine Learning)
# Prof. Manjesh Hanawal
# Industrial Engineering and Operations Research
# Indian Institute of Technology, Bombay

## Lecture – 57
## Uniform Exploration

(Refer Slide Time: 00:21)

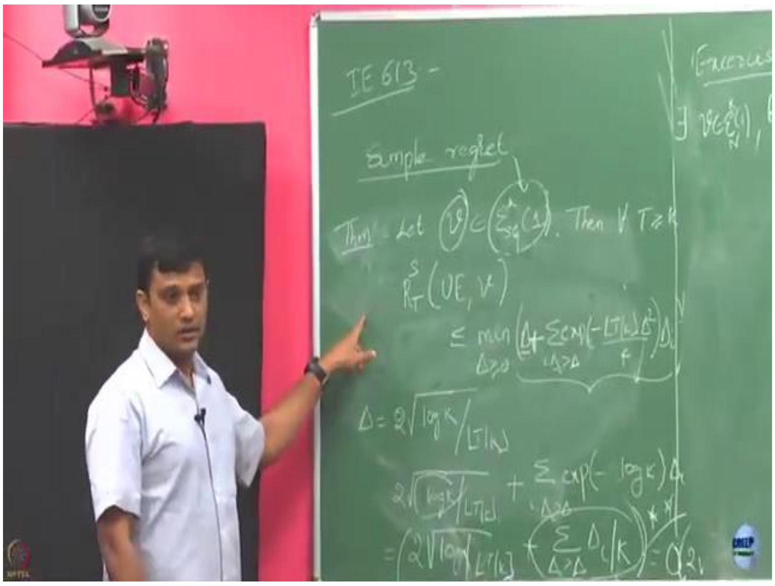


This is a first exercise problem you are going to do for assignment. So, this is going to be the assign exercise one in the 33rd chapter in this book, book by it is Csaba Szepesvari and Lattimore. So, what this exercise wants you to show is, it ask you to show that fix the number of rounds; says that there exists a distribution ν such that whatever policy you are going to apply, the regret is going to like grow like $\exists\ \nu \in \epsilon_N^K(1), s.t., \quad R_T^s(\pi,\nu) \geq C\sqrt{\frac{K}{T}}$

So, it is saying that there exists $\nu$ and it says that this is the class of sub Gaussian random variables with parameter 1. You would say that there exists a ν belonging to this class such that irrespective of whatever $\pi$ you are going to choose, we are going to incur regret which is of the order $\sqrt{\frac{K}{T}}$ for some constant C. So, how to work out this?

This is an exercise and this has a hint also in that exercise like this follows along the similar lines which we did it in the adversarial case, right. You there also came up with an algorithm where you showed that there exist an adversary who can select a sequence such a way that, your regret is going to be like at least a  grow like $\sqrt{T}$ along that you are going to show this.

(Refer Slide Time: 02:31)


IE 613 -
Simple regret
Thm: Let $\nu \in \mathcal{E}_{SG}^{k}(1)$. Then $\forall\, T \geq k$
$R_T^S(UE, \nu)$
$\leq \min_{\Delta \geq 0} \left(\Delta + \sum_{i:\Delta_i > \Delta} \exp\left(-\frac{\lfloor T/k \rfloor \Delta_i^2}{4}\right) \Delta_i\right)$
$\Delta = 2\sqrt{\log k / \lfloor T/k \rfloor}$
$2\sqrt{\log k / \lfloor T/k \rfloor} + \sum_{i:\Delta_i > \Delta} \exp(-\log k)\Delta_i$
$= \left(2\sqrt{\log k / \lfloor T/k \rfloor} + \sum_{\Delta_i > \Delta} \Delta_i / k\right)$
Exercise
$\exists\, \nu \in \mathcal{E}_N^k(1)$,

An another thing, so if you are now going to look into this is your uniform exploration is already optimal. For uniform exploration, what is the boundary we got? We got it like $2\sqrt{\frac{k \log k}{T}}$.

So, 2 is a constant, we can ignore that. But if you are going to compare for the parameters K and T, the only extra parameter I have here is log(k); otherwise it is it is already $\sqrt{\frac{k}{T}}$. So, that means, it is already optimal up to this factor log(k). Another exercise in the same chapter it asks you to show that, yes this uniform exploration is optimal up to this factor log(k); but you cannot avoid this factor log(k) here, for uniform exploration it will be always there.

So, you have to show that, if you are going to use uniform exploration, this is going to be always $C\sqrt{\frac{k \log k}{T}}$ (Refer Time: 03:54). So, for uniform exploration you cannot avoid this.

Maybe there is an other policy, better policy which will go into achieve this lower bound; but uniform policy is not that optimal, it is this extra factor log(k).

Is there a policy which will achieve this lower bound for my simple regret? The answer turns out to be positive; if that is the case, then the question we have to answer is, what is that policy?

Suppose let us say I am going to, this is also a stochastic case right; instead of cumulative regret we are looking at simple regret. Suppose let us say instead of worrying about a special algorithm for this, I will just take one of the algorithm which I have already for the stochastic multi armed bandit, let us say UCB or KLUCB. I will take and for the whatever it is going to do, let it do it in the exact way it is doing.

But whatever arm it is going to select in the T +1'th round, for that round I am going to compute its instantaneous regret. So, do you think if I do that, it should give me a better performance like that should be good, ok.

So, let us say I want to, I will give a time T; what you do is, you just apply your UCB algorithm, let us say whatever you did till T round you did it. And whatever you are going to, it asks you to select in the T + 1'th round that you gave me as output in the T +1'th round and then I am going to compute simple regret on that.

Do you think the performance of that is going to be good or bad?

Student: Bad.

But there also it has to be doing some optimization, right. If it is not bothering about optimization at each round, its regret would not be sublinear; it has to also do some optimization right for every round.

Student: We are exploring (Refer Time: 06:13).

We are exploring less over there.

Student: Compared to this.

But so in that case, most of the time it will be picking the optimal arm right or it will be exploiting. If it has, if it is doing a good job, if it is already exploiting it; it is better be exploiting the best arm. If it is exploiting some arm right; that means it believes that that is the best arm, right.

So, it could be I also exploring in that; the way the UCBs are selecting, it has both exploitation term plus exploration term in, right. It may be happening at some point, it may in the ; you sure unlucky it may be happening in the T + 1'th round, it went into exploitation term become dominant and it is exploiting exploring some arm.

So, in that case it would not be giving a good performance to you, right.

Student: Sir last week, (Refer Time: 07:07) sum into mathematics?

But that is not UCB, I am just saying if you have to just plainly take it and apply it, give it whatever it is doing. But, other way like thinking like you; suppose let us say I took UCB. So, I have given time t, I just took the UCB.

Student: Yes sir.

And I just see how many arms it played till this point, each one of the arms and then I am going to pick the arm which is played highest number of times. If I do this, so you expect it should have a better simple regret.

Student: linear linearly linear amount of time (Refer Time: 07:47).

Yeah.

Student: So.

So that means, if you are going to just pick the arm which has been so far played highest number of time?

Student: Should.

You should, do you expect that to be already the best one or that should give the smallest sample regret. So, then, then you are you are saying it depends on t; if t is large enough.

Student: Yeah.

Then maybe like I would have explored all of them sufficiently good amount and I have good confidence in each one of them;

Student: So.

Then maybe I will just pick the one which has the which have been selecting most number of the times. If t is small, I do not have that much confidence, right.

Student: Yes.

Because who knows like I may not have.

Student: (Refer Time: 08:30) two arm.

Yeah.

Student: Then ucb will help us differential between those two perhaps in a better way as compared to.

See like rest I distinguished fine, but the two 1's which are very close to each other.

Student: Yes.

If I have not yet resolved them, finally when I have to give one arm.

Student: No, but I (Refer Time: 08:54).

Which was.

Student: But UCB will pick both of those linear type right order of.

Right.

Student: But this one will pick t by k time.

So, fine, so in that case like if. So, if you are in a bad situation where two arms are very close.

Student: This thing is .

It has not yet figured out which one of them is better and so it might be playing them equal number of times. So, you take the whatever the empirical means; what I want is at that time in the T + 1 whatever it is going to play.

Student: Yes.

Whether it is going to, how far it is going to from the optimal arm is my question?

Student: Yes.

Whether it is going to be good or not?

Student: Yes.

It may be right like still; suppose if you are saying two arms are very close to each other and it has been playing them for quite some time to distinguish which one of them is better.

Now, in the T + 1'th round, you just ask pick one; it should be picking one of these to right, because it is time to resolve between these two. And because these two are close, it may happen that even their sample regret will also would not be that much; because they are almost like very, they are their difference is not that much.So you are saying oing UCB is not bad in that case we should.

Student: yes.

Better take this sample. So, fine, that is what I am asking like whether you will go for a UCB and just take whatever it says in the T + 1 round.

Student: (Refer Time: 10:25).

For the simple regret, so it depends on the problem instance, right.

Student: Yes.

So, you cannot like blindly apply it on any problem instance given to you. So, for a simple regret, you have to be about more careful. But still if I want to still get a bound on, if I want to adopt an algorithm that is there for standard which is designed for me regret minimization to a case where I want to do.

Student: Simple regret.

Simple regret minimization, what could be the case? One thing simple thing I could be, one possible thing I could do is; based on number of times I have observed each of these arms till time t, I am going to now construct a probability distribution on the arms which is proportional to the number of times I have played each one of them. And then I am going to pick a one according to this probability distribution.

So, we will see that, if there is a good algorithm for the regret minimization problem and if you are going to do like this in the T + 1'th round; that is you are going to construct a probability distribution based on how many times your regret minimization algorithm has played till time t, then even your simple thing would be good.

So, let us understand this. So, first I want to express a simple regret in terms of the cumulative regret, so this is a proposition I am going to write. So, let cumulative regret. So, let π defined like this be a policy. So, this is defined for the first T rounds.

Now, we are going to define this policy for the T + 1'th round in this fashion. What you are going to say? I am going to say that, we are going to play I'th arm given that you have been observing ok, this is like $I_1$ in our notation, $I_T$ that is you played $I_1$ in the first round observed reward $r_1$ and in the $I_T$ th round you played $I_T$ arm and you observed reward $I_T$.

We are going to define this quantity to be what? To be the average of this indicator. What is this going to give you? It is going to give you. So, what is this $\pi_{T+1}$ is going to give you?

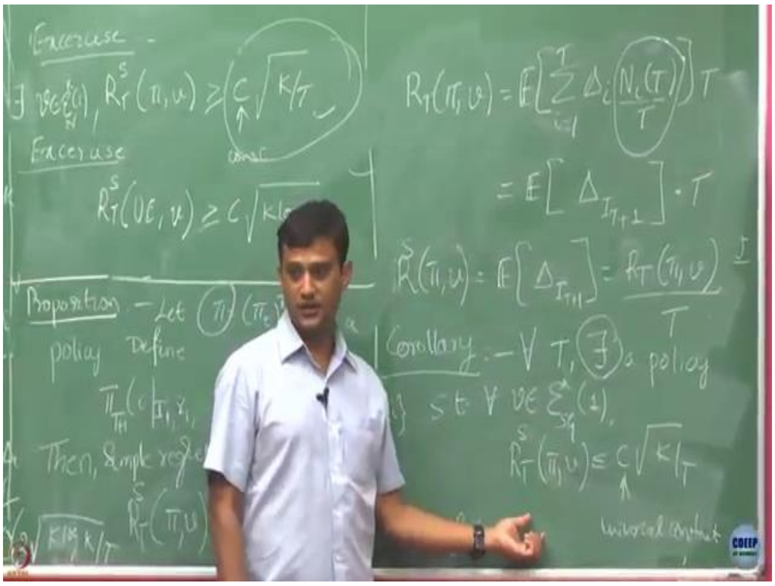
Student: Arm (Refer Time: 14:53).

It is going to give you a distribution on the arms, right. And what is, how is this distribution is going to be? That is going to be proportional to the number of times you have played that particular arm.

Now, if you are going to do this and now if you are going to compute a simple regret of the policy, where you take the first T rounds like this you and for the T + 1'th round, you have exactly done like this. Then the simple regret of that algorithm is going to be cumulative regret of the policy π that was playing like this in the first whatever it is; so it was whatever the policies, it has done something till first round first T rounds like $\pi_1$, $\pi_2$ all the way up to $\pi_t$.

Then this simple regret is going to be nothing, but the cumulative regret divided by this T; that is the average of this cumulative regret is exactly equals to your simple regret. So, why is that true?

(Refer Slide Time: 16:03)



So, let us write my cumulative regret, I know my cumulative regret is given by what? $R_T(\pi, \nu) = E[\sum_{\{i=1\}}^{T} \Delta_i N_i(T)]$, we have we know that this is by the regret decomposition theorem. Now, let me divide it by $N_i(T)$. So, what is $N_i(T)$ here is?

Student: (Refer Time: 16:44).

$N_i(T)$ is gives you the number of times you have played arm i till round T. So, is this $N_i(T)$ is going to be same as this quantity here? No, what is this? This is nothing, but this distribution $\pi_{T+1}$, right. So, then can I write this as this is this expectation as nothing, but I just $E[\Delta_{I_{T+1}}].T.$ So, then this is nothing, but and this is nothing but, this is the simple regret. So, in my simple regret, so if you give me any policy which I can apply to get a good cumulative, which I can apply for the cumulative regret minimization problem.

And if I am going to and then if I am going to sample by T + 1'th arm according to the distribution defined like this; then the cumulative regret, sorry the cumulative regret, the average of a cumulative regret is nothing, but my simple regret for that policy $\pi$.

So, what I basically done is? I took a policy $\pi$ here which I am applying it on the standard bandit problem to minimize cumulative regret. But the way I am pulling the T + 1'th round is based on this new distribution. If I do this, I am just saying that this is how the simple regret and the cumulative regret are related. This is the simple regret that I would have got in T + 1'th round and this is the cumulative regret in the first T rounds, ok.

Now, can you tell me what is the best bound, so given this relation holds? Now can you tell me what is the best simple regret bound I can get? $O(\sqrt{\frac{K}{T}})$ Order root K by T. So, why is that? Moss algorithm if I use my moss algorithm that R and T guarantee me some $O(\sqrt{KT})$. But because of the denominator T here, I am going to get it as $O(\sqrt{\frac{K}{T}})$, which is what this lower bound also tells.

So, if I have a good algorithm for my cumulative regret, I have an good algorithm for simple regret. But it does not like look like as simple as it is not necessary that any algorithm which performs well on the that gives a good cumulative regret, necessarily has to yield a good simple regret.

So, that we will talk a bit later, but let me just for complete this statement here as a corollary; for all there exist, for all there exists a policy such that for all which is. So, as you already see that, what is that there exists a policy; this policy which is giving me this bound is already moss policy, right. So fine.

So, let me first discuss. So, this corollary says that, for all T there exists the policy such that, for any instance this relation holds $R_T(\pi, \nu) \leq C \sqrt{\frac{K}{T}}$. We are giving an upper bound lower bound we are already given here; this is irrespective of what is your policy, there exist an instance such that this lower bound holds. Now we have to give an, if see that we can achieve this. What we are saying is, there exist a policy such that for any instance this holds. Now you already know what is that policy, this policy is moss policy; because that gives me $O(\sqrt{KT})$ and from that I am going to achieve $O(\sqrt{\frac{K}{T}})$..