Bandit Algorithm (Online Machine Learning)

Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture – 58
KL-LUCB

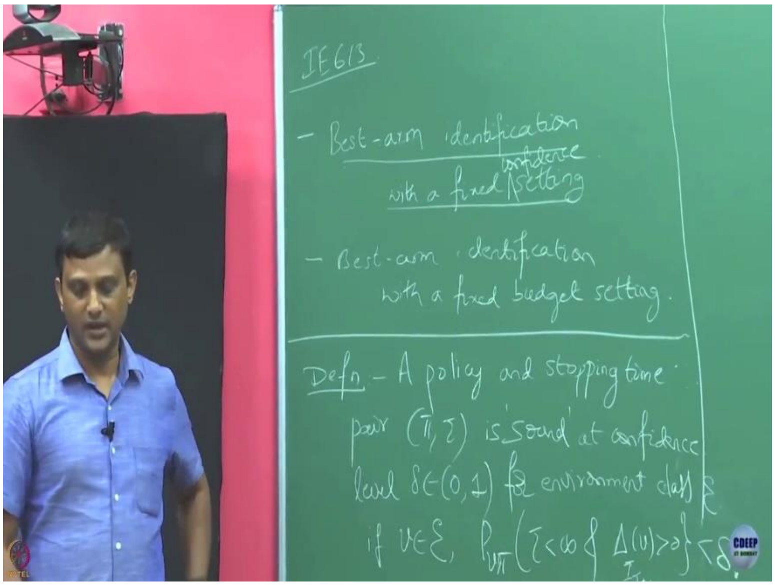
So, in the last class, we were talking about pure exploration methods, we talked about uniform exploration method and see what is the bound we are going to get from that and we also discussed what will be the lower bound right.

And then, we connected how we can obtain a policy for pure exploration from a policy which is designed for regret minimization and we then, showed that if you are going to appropriately define how you are going to choose the arm in the last round. We said that simple regret is going to be related to the cumulative regret for that policy, simply simple regret is equals to average of the cumulative regret ok.

So, today what we are going to discuss is something called fixed confidence setting, that is I will tell you with this confidence, you have to identify the optimal arm. You take whatever number of samples, you would like.

I do not care how many samples, we are going to take; when you are going to do exploration, what is the regret you are going to incur. But at the end whenever you are going to stop, if you tell me this is the arm, I want the guarantee that happens to be the optimal arm with probability at least $1-\delta$ ok.

(Refer Slide Time: 01:55)



IE613

- Best-arm identification with a fixed confidence setting

- Best-arm identification with a fixed budget setting.

Defn - A policy and stopping time pair (π, τ) is 'sound' at confidence level $\delta \in (0,1)$ for environment class $\mathcal{E}$ if $\nu \in \mathcal{E}$, $P_{\nu\pi}(\{\tau < \infty \; \& \; \Delta_{I_\tau}(\nu) > 0\} < \delta$.

So, it is called best arm identification. So, when I am doing a pure exploration right, what could be the questions I could be asking? I will give you confidence term, you do whatever number of exploration you do and at the end give me the arm which happens to be the right optimal arm with probability 1 - δ. What could be the other question kind of questions I could be asking here?

Student: (Refer Time: 02:47) the number of (Refer Time: 02:49).

So, it could something that is called fixed budget. You could be asking ok, I am going to give this much of budget that is this much of rounds, you are allowed to do whatever you want to do.

But at the end, you have to output me an arm which happens to be the correct arm with as much as high probability. So, we will formalize that; but let us say I am going to call this as fixed budget setting. The second possible question, we said at fix budget setting. So, that is. So, we will study this later, may be in the next class. But today, we are going to just focus on this part.

So, in this case, what you want to set your performance criteria as ok? What it what I said, you will be given a confidence term, you are allowed to do as many exploration you want; at the end, what you return the arm which happens to be the right one with probability at least 1 - δ, where δ is I have passed on to you.

There could be many algorithms like that right that could be doing this. But what algorithm you like in that case?

Student: The less number of.

The one which takes less number of rounds or it identifies the optimal arm with as many few rounds as possible right ok. So, let us try to formalize that. So, before I do this, I am going to give this definition.

So, when I said the best arm identification with a fixed confidence right, my only input to the algorithm is tell me what are the arms we are talking about and give it the confidence parameter. What it has to do is internally it has to do whatever exploration it has to do, but it has to come up with its own stopping criteria. Stop at that point and then, output an arm ok.

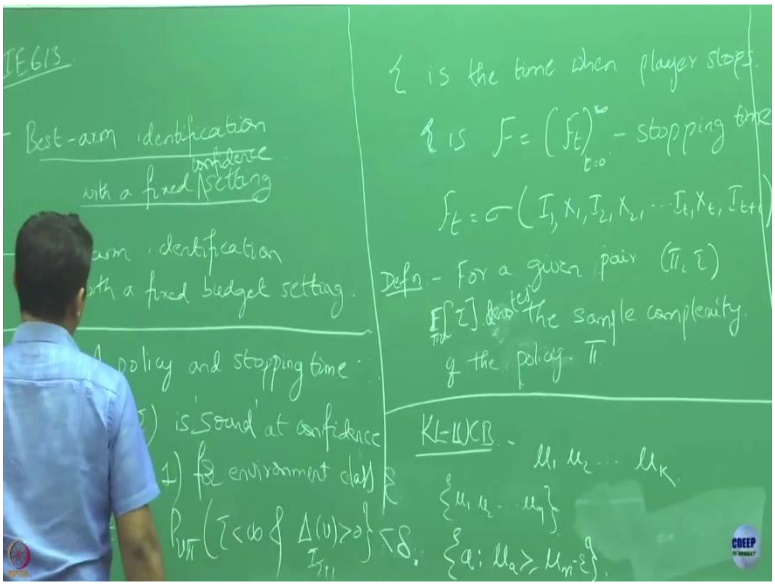
So, in all this algorithm, there will be some stop stopping criteria will be there. Let me call a policy and stopping time this ok. So, π is what a policy that you are going to apply and it has its own stopping criteria, when it is going to stop. We are going to say this policy along with a stopping criteria is going to be sound; for any δ the confidence term, we have passed on if this is going to happen.

So, what is this? It is first saying that τ is finite like it stops, it has stopped at the finite time and after that what is it τ + 1? The arm which is going to give me in the next round t + 1 that and what is $\Delta_{I_{\tau+1}}(\nu)$? This is a sub optimality gap of this arm with respect to the optimal arm, that being greater than 0. What does this mean, if this is greater than 0?

Student: (Refer Time: 07:57).

That is this $I_{\tau+1}$ happens to be sub optimal arm, that happening is upper bounded by δ. That is if its stopped at finite time and it is giving me an arm which is not optimal, that probability happens to be upper bounded by δ that means, it is giving me the correct arm with probability at least 1 - δ. So, when this happens, we are going to call this such a policy pair π τ as sound. So, I think before this, I should have mentioned this also.

(Refer Slide Time: 08:36)


IE615
Best-arm identification
with a fixed confidence setting
arm identification
with a fixed budget setting
A policy and stopping time
) is sound at confidence
1) for environment class E
$P_{\nu\pi}(\tau<\infty, \Delta(\nu)>\varepsilon) \leq \delta$
τ is the time when player stops.
τ is $\mathcal{F}=(\mathcal{F}_t)_{t=0}^{\infty}$ - stopping time
$\mathcal{F}_t=\sigma(I_1,X_1,I_2,X_2,\dots I_t,X_t,I_{t+1})$
Defn - For a given pair (π,τ)
$E_{\pi}[\tau]$ denotes the sample complexity
of the policy π
KL-LUCB
$\mu_1\ \mu_2 \dots \mu_K$
$\{\mu_1, \mu_2 \dots \mu_n\}$
$\{a: \mu_a \geq \mu_n-\varepsilon\}$

So, I am going to denote this τ here. Let me make this the τ more formal here. T is the time when player stops and we are going to say that τ is stopping time. If anybody tell me what is this τ is F_t stopping time, where we are going to say this F_t is the sigma algebra $F_t = \sigma(I_1, X_1, I_2, X_2, \dots, I_t, X_t, I_{t+1})$

So, F_t with the sigma generated by what you have observed so far till time t and the action you are going to play. You have not yet observed, what is the reward for r in the round t + 1.

So, then what does why is the what does this τ being stopping time with respect to this means? This means that so what is the definition of stopping time? This stopping time basically says that I can say something yes or no, based on till the current observation right; not needing to know what is happening in the future.

It is exactly saying this like I this is going to decide whether to stop or not based on what you have observed till that point and the action, you could potentially take in that round. But without knowing anything after reward at time x_{t+1} or anything that is going to happen in the future after that.

So, this is the stopping time associated with any policy π and the sigma algebra generated here will be indeed depend on the policy, you are going to apply right because the policy is going to govern how you are going to choose this arms and it also depends on the underlying distribution.

So, this x_1, x_2 are all generated according to the underlying distribution. (Refer Time: 11:10) for a given pair, the value at which were this algorithm is going to stop τ, can it be random?

Suppose, let us say I give you K stochastic distributions and you have an algorithm you applied your algorithm and you stopped out after certain number of rounds and gave it. Now, you are going to reapply your algorithm, again starting afresh on the same distribution, the starting time could be different.

Student: Stopping time could be different.

It could be different right. So, the τ can be in this case a random variable. So, that is why we are going to say for a given pair (π ,τ), we want to say expected value of τ. So, this is expectation is going to be induced by the policy π as well as your underlying distribution nu as the sample complexity. Now, what would be interested in is we will be interested in a policy with the sample complex, with the stopping criteria which has smallest sample complexity ok.

So, now before we start talk about how this what should how this sample complexity should be? Is it like possible that my sample complexity has to be this much irrespective of whatever for a given policy, whatever be the underlying distribution, this is the minimum sample complexity, I am going to incur?

Maybe there is some value, first we are not going to look into that that is. But what will start looking into is directly algorithm which we hope that is going to minimize the sample complexity ok.

So, I am I will. So, there are different version of this lower bounds and I have it looks like they have very complicated characterization of this lower bound. So, I will try to see which one is the simpler one, we will discuss in the next class. But today, our focus will be mostly just on the algorithm. So, today we are going to discuss one algorithm called KL-LUCB. This is actually a derivative of another algorithm called LUCB, Lower Upper Confidence Bound.

But we will just discuss the one version of it called KL-LUCB ok. Anybody has any intuition like when I have a such a pure exploration; how I should go we discussed in the previous class a bit about this.

So, one thing we will discussed is how to make use of an algorithm which is UCB kind of algorithm to make it work for the pure exploration one. So, we noticed that the kind of algorithm, we had in the regret minimization thing like UCB algorithms, not necessarily do good for the pure exploration setting right.

So, what could be other possibilities? So, one possibility is I would estimate the confidence intervals along with their means and now, I am going to compare estimated mean plus confidence term and I am going to order. And then, to check which is the highest among them and if it so happens that the highest and the next highest, if I can ensure that, the difference between them happens to be at least larger than some amount.

Then, maybe I will have enough confidence that the first guy is the best one because the second guy already ensured that it is already separated from the first guy certain amount, maybe I can use this idea.

So, what I am saying is you do like as usual like in UCB like you have the estimates plus the confidence one. So, you are going to take the optimistic value at any time for all the arms. If at any point you notice that the first the best and the next best are separated by certain amount, that amount you have to define.

If that is going to happen, then you can be confident at this point ok. So, that seems to be already separate enough separation between them, I am going to stop there. So, this KL-LUCB, we will try to formalize that. But it has to address a bit more general problem. So, when it instead of identify.

So, what we are trying to do in this case is we are always trying to identify the best arm right. Instead of I could I ask the question, find the best three arms, best four arms. So, maybe I will ask the question ok, in this class I want to identify the best five ok. So, when you when somebody gives me this five, I do not care about who is exactly first, who is exactly second, who is exactly third in that; but as long as I am if you can tell me with confidence, these are the top five. I should be fine ok.

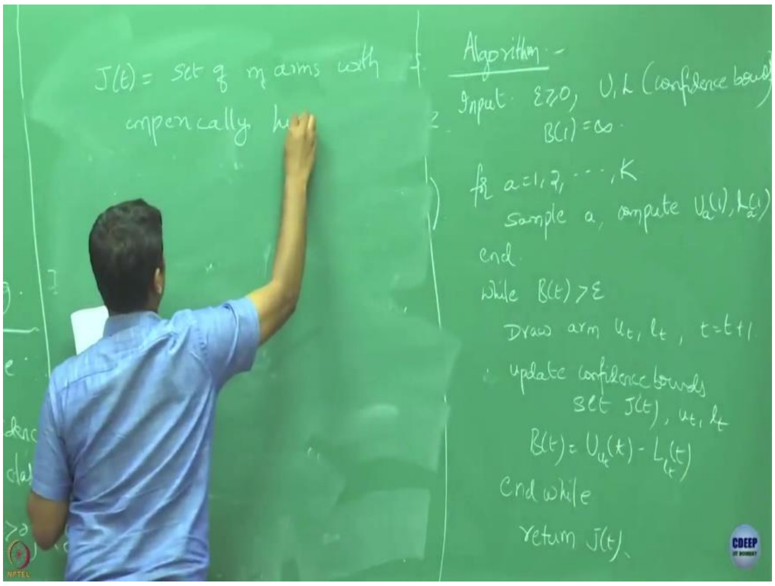
So, this KL-LUCB makes at such a generalization on this problem and it is going to solve try to identify instead of just the top one, it tries to identify the top m arms ok. So, let me just formalize that notion let us say you have this means μ_1, μ_2 … μ_k. These are the means of the arms ok. So, let us say these are all ordered already. So, μ_1 is the best, μ_2 is the second best and like that. So, the top arms are here, then simply μ_1, μ_2 …μ_ν , so if I am interested in top m arms.

Now, what I may do is instead of exactly identifying this top m arms, I will slightly weaken my condition and say that I want top m-ϵ arms. What does that mean? I will be fine with all $\{a\colon \mu_a \geq \mu_m - \epsilon\}$. So, these are exactly the top m arms right.

So, suppose, let us say I am going to set ϵ = 0; if $\epsilon = 0$, what is this set is? Set of all arms which are greater than or equals to μ_m that is exactly the top m arms. Suppose, I have slightly relax this and allowed ϵ to be positive that is given to you. Will this set include more arm than this? It is going to include more arms right. Now, I have basically but I am like as of from this bigger set, as long as you output me m arms, I am happy. I am take them as equivalent to the top m arms right.

So, whatever I have defined here is this is the special case of this with ϵ = 0 and m = 1, when I want to identify that just top m arms; but now, we are just relaxing that and I am going to say as long as you give me ϵ optimal m arms, I am fine and this algorithm is now going to look into this situation ok.

(Refer Slide Time: 20:36)

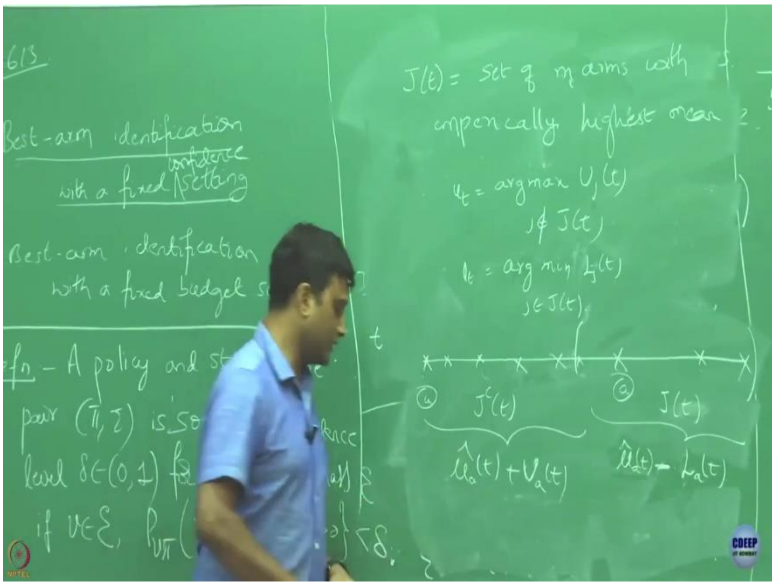


J(t) = set of m arms with empirically h

```
Algorithm:-
Input: $\epsilon \geq 0$, U, L (confidence bound)
        $B(1) = \infty$
for a = 1, 2, ..., K
    Sample a, compute $U_a(1), L_a(1)$
end
while $B(t) > \epsilon$
    Draw arm $u_t, l_t$ , $t = t+1$
    update confidence bounds
        set $J(t)$, $u_t, l_t$
    $B(t) = U_{u_t}(t) - L_{l_t}(t)$
end while
return J(t)
```

Now, the question is how to identify this top m arms. So, algorithm. So, let us let me write input and I am also going to set $B(1) = \infty$.

(Refer Slide Time: 21:18)


6/3
Best-arm identification with a fixed confidence setting
Best-arm identification with a fixed budget s
Defn – A policy and st
pair (π, τ) is so
level δ ∈ (0,1) fo
if ν ∈ ℰ, $P_{\nu\pi}$(
< δ.
t
J(t) = set of m arms with empirically highest mean
$u_t = \arg\max_{j \notin J(t)} U_j(t)$
$l_t = \arg\min_{j \in J(t)} L_j(t)$
a
$J^c(t)$
$\hat{\mu}_a(t) + V_a(t)$
a
J(t)
$\hat{\mu}_a(t) - L_a(t)$

So, let us see this. So, this algorithm takes ϵ as an input U and L, these are confidence bounds which are going to define a bit later. So, this is going to just give, U is going to give upper confidence bound and L is going to give lower confidence bound on your arms.

So, what it does is in each first K arms? First K rounds, it is going to play each arm once and for each one of them computes their upper and lower confidence bound. So, here index 1, here argument 1 means it is for the first round because they have just one-one sample and now, it is also maintaining a B function which is a function of t.

We will see what is this B t defined as? What it does is it is at every time; it is going to define two arms; U_t and L_t. U_t is coming from the upper set and L_t is coming from the lower set.

So, now let me come here what it is doing after you pull an arm, you know its empirical mean and using this upper confidence bound term, you have defined its confidence term also. So, this U and L are going to give you the confidence term ok. After you play your arms, you have the samples, you can compute their empirical mean values.

At any time you are going to maintain the set J(t), what is the J(t) is doing? It is splitting your total number of arm into two sets; the one with the highest means the first m arms with the highest mean and the other one with the rest of the arms.

So, let us say you have at any point, let us say you have ordered your. So, it. So, happened that your arm means happened to like are like this ok. Let us say in round t and let us say you are interested in m equals to 4. So, it is going to take 1, 2, 3, 4. This has one set and the remaining, it is going to put let me take one more. The remaining in the other set, this set we are going to call it as J of t and this is going to be $J^c(t)$.

So, now what is trying to do is from this set J(t), it is trying to find u_t. What is u_t in this case is going to be u_t is the. So, this is J does not belong to J(t) means, it is coming from $J^c(t)$. It will looking for the largest index of all the arms, but taking their upper confidence into account ok. So, when I have drawn here right, these arms, this is there in this set, it is there empirical values plus upper confidence bound added. It is there.

So, here all the terms here are like let us say let us say this is an arm here. This is going to be μ a hat and round t + it is coming and then, this is going to be their lower confidence bound and here, when I am going to look at let us take an arm a, it is going to be $\hat{\mu}_a(t) + U_a(t)$.

So, this is how I have taken their value as if an arm comes ok, sorry what arms say. So, this has simply the set of terms which I have divided into two partitions. These are the top m and the other ones.

Now, for each of these arms here, what I am going to take. So, these are just their empirical means. For each of the arms, I take here and what I am going to compute is their lower confidence term and what are the set elements, I am going to take from this side, I am compute the upper confidence bound.

So, suppose let us say if I am going to take this set, I am going to look at its this end. If this is the confidence interval, I am going to look at this end of this. If I am going to take an arm here, I am going to look at its upper confidence term. It is going to in each round, it is going to select these two arms and these are the one that I will going to be played in that particular round.

So, here in each round, we are going to play actually two arms; but that is same as saying each place write over two place like either I can say in the single round, I am going to play two arms or like my one round is spread over two rounds, where I have played one each.

Now, you play these two arms, for that you are going to observe a sample and then, you are going to estimate their empirical means. From that, you are going to get these two values; u_t and l_t, using the way we have defined here and then, we are going to look at the difference between the upper confidence bound for the u_t'th arm and lower confidence bound of the l_t'th arm. And now, what we are going to do is even if this guy happens to be larger than δ and ϵ, you continue. If this happens to be lower than ϵ, we are going to exit ok. So, let us understand what does this mean? Let us just for clarity, let us say this is my J(t) set and this is my $J^c(t)$ set.

So, here what I have taken from this J(t) set? I have taken my lower confidence term. So, this is my point and on this, I am going to take my upper confidence term ok. This is a partition. So, what this algorithm is trying to resolve is it is trying to resolve this set from this set; this is the top m set right.

So, what are the things that are potentially conflicting? The potentially conflicting arms are here that are at the border. So, in this set this is I am going to say this is the best and this is the other part. In the best, among the best, this is the worst guy, maybe that this guy actually belongs here, but by mistake I have put him here and may be that here among the worst, this is the best guy; maybe this best guy belongs to this part, by mistake I have put him here.

So, I need to basically be more confident about the edge points right; other points maybe like I am bit more confidence. But my resolution is I want to resolve more about these two edge points here. So, this algorithm is exactly trying to do that. So, it is trying to see. So, u_t is coming from this set.

It is going to take this upper confidence term here and this is going to take this lower confidence term here. If this difference happens to be large, then maybe I will try to I will continue ok. Just let me just confirm this is greater than ϵ or less than ϵ. So, if this difference happens to be less than, greater than ϵ whether I am confident enough?

Student: Sir, difference is positive?

Yeah.

Student: Then, $J^c(t)$ is going ahead of J(t).

Exactly. So, in this case, this is coming on the left side and this edge is on the right side right. That means, it may be possible that this guy actually belongs here and this guy is belongs here. That is why I want to resolve it and I want to continue. If that is not the case, that means, I had a enough separation between these or in the lower bound of this and upper bound of this, I had enough separation that means, I can be more confident about this and maybe I can stop. So, this algorithm says that it takes this ϵ.

So, this is like kind of a resolution parameter. It takes this as an input and based on this separation, how much you want to ensure, it is going to give this. If separation happens to be larger than ϵ, that is going to stop; otherwise its keep on doing this.