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

## Lecture - 36
## KL–UCB Algorithm

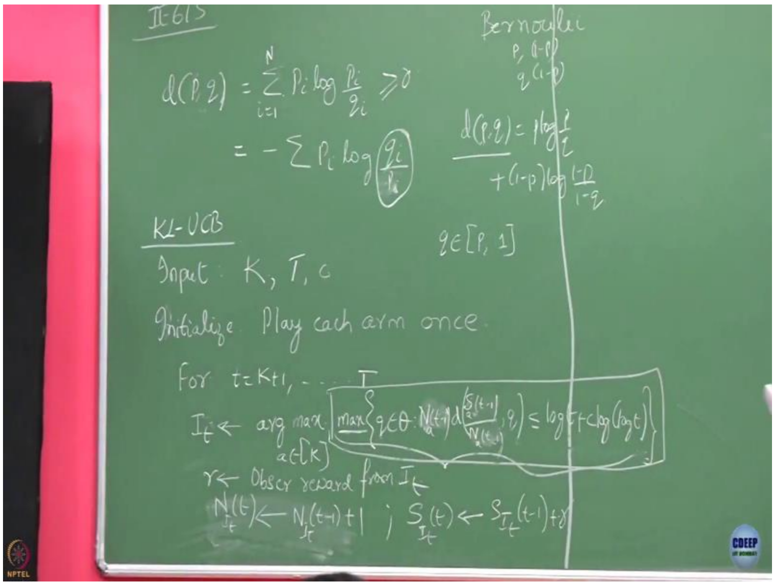
Let us start like in in our last class, we saw the proof or regret bound of us UCB Algorithm right. So, we finally showed that, the expected number of pulls of any suboptimal arm is like order log t and then, based on that we showed that expected regret of UCB's order log t or it depends like k times a log t order log t.

So, we will see another algorithm called KL-UCB. So, this KL-UCB is it is again the algorithm was same, the only thing is the way indexes are computed is based on different confidence terms here and these confidence terms are derived based on slightly different method than what has been defined used in UCB, in the sense that these are slightly stronger confidence bounds.

Then we will show that, using that we will get a slightly actually a better regret bounds than what we have got it in terms of a for the UCB. It is not better in terms of log t, but in terms of the constants and how in terms of the dependency and the problem parameters itself ok.

So, let us discuss those things. So, first I am going to give you the algorithm directly. So before giving the algorithm, I am going to define this term called divergence, which is going to be used here.

(Refer Slide Time: 01:49)



IE 613

$$d(p,q) = \sum_{i=1}^{N} p_i \log \frac{p_i}{q_i} \geq 0$$

$$= -\sum p_i \log \left(\frac{q_i}{p_i}\right)$$

Bernoulli

$p, (1-p)$

$q\,(1-q)$

$$d(p,q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$

$q \in [p, 1]$

KL-UCB

Input: $K, T, c$

Initialize: Play each arm once.

```
For $t = K+1, \ldots, T$
    $I_t \leftarrow \arg\max_{a \in [K]} \left[ \max \left\{ q \in \Theta : N_a(t-1)\, d\left(\frac{S_a(t-1)}{N_a(t-1)}, q\right) \leq \log t + c \log(\log t) \right\} \right]$
    $r \leftarrow$ Obser reward from $I_t$
    $N_{I_t}(t) \leftarrow N_{I_t}(t-1) + 1$ ; $S_{I_t}(t) \leftarrow S_{I_t}(t-1) + r$
```

So, in general let us p and q be any distributions ok, we are going to define divergence between these two quantities as summation. Let us say i is 1 to some N. So, both let us say q and p are distribution on N objects, they are going to define distribution on some set of N values and then, the divergence between them is defined like this.

So, you can always verify that this quantity is a non-negative quantity. And in a sense, it is measures the kind of distance between these two distributions. It is not actually true distance metric; but in a sense, it has some of the properties ok.

First thing you can notice that when $p_i$ equals to $q_i$ for all i, this is already 0; that means, if both the distributions are identical, then this is value is going to be 0 and as if that pretty far, in the sense that they are not putting same amount of mass; but they are putting on different different objects differently, this quantity is going to be large ok.

So, fine. So, that is the now we are going to see that the concentration bounds, they we will use is going to one can express them in terms of this divergence and they are going to be that will give us a tighter bounds on the divergence probabilities; then, we what we already have for using the Hoeffding’s bounds ok.

Student: Sir?

Let us see. You want to prove it. Now, this is correct right?

Student: q i by p i.

$q_i$ by $p_i$. So, do you know Jensen’s inequality? Can you apply Jensen’s inequality on this? So, now define this as the value of your random variable and this is the associated probability. So, then this is nothing but, expected value of that random variable. So, can you apply Jensen’s inequality and see whether this is indeed correct? So, what log is what kind of function?

Student: Concave.

It is the concave function. Now, you are looking basically at an expected value of a concave function. So, see what happens then? It directly follows from that. Now, right away I am going to give the KL-UCB bound. So, this requires input which is the number of arms and let us say we also need to get the horizon for which you are running it over, then play; so, initialize. Now, for.

So, I am taking input K and T and then, I am going to run it for T number of rounds and in every rounds, so before I have played each arm once, so I have one sample from each of these arms. Now, I am going to play an arm by choosing an arm which has the highest index and that index is defined like this.

So, you see that this is the step where, we are differing from UCB. So, what you are doing is basically you taken a. What you did in UCB? There you basically took for that arm, you took its empirical mean and then added the confidence term which look like 2 times log t divided by number of pulls of that arm.

But now, this is a different you say different creature here. Now, for every i, you are counting this. Now, this is this entire thing is acting as an index for that arm ok. So, now let us see how this is defined? So, this one is now actually with my new notation, this is like S of a t minus 1, N of a t minus 1, N of a t minus 1.

So, N of a t minus 1 denotes number of pulls of arm a till the previous round. Now, I am looking the product of number of pulls and this divergence, is it larger than this quantity and then, I am looking it for every q and I am going to choose the largest q which satisfies this?

So, what is this quantity here? This is the divergence between q and this quantity. What is this quantity here? S a t minus 1 here ok. So, let me complete this, then again come back. So, whatever this gives right, let us say for some arm this gives me the maximum value, I am going to choose that as my current arm to play.

I am going to play that arm and I am going to observe reward for that arm, then I am going to increment the count for that arm, number of plays I have done and I am also this S is keeping track of the cumulative reward from that arm and I have updated the cumulative arm for that arm.

So, now can you see what is this ratio is giving you this is the cumulative reward for that arm a divided by total number of times you have played that arm till t minus 1. So, this will give what? This will give empirical mean of that arm till that point. So, this is the total cumulative sum and this is the number of times I have played that arm.

So, now what am computing here? I am computing the divergence between the empirical mean and that given q and now, I am looking for that largest q which falls within my parameter range.

So, this theta is going to define what is the parameter range, I will be looking into. That is basically saying where my true means are coming from. We have already always assumed that my distributions have a finite support right; that means, I already know in the means are also going to be in some, so that is captured by this theta. Now, I am looking for that q, which takes the, which maximizes that largest q which satisfies this condition ok. Let us see this now.

So, now you can think. So, q is one parameter, this is another parameter right. So, now you can think this as a Bernoulli distribution with parameter mu hat for one Bernoulli, another Bernoulli with parameter q. So, now this is the divergence between two Bernoulli distributions; one with parameter this and another with this.

So, if this is for a, some arbitrary distributions p and q right. If it is a specifically for Bernoulli, let us say Bernoulli is just characterized by one parameter right. Let us say one Bernoulli has parameter p and another Bernoulli has parameter q.

So that means, if p means other probability is already 1 minus p and this is 1 minus p. So, then just for this divergence, I could write p, q this is nothing but just this p log p by q plus 1 minus p log 1 minus p divided by 1 minus q. So, I only need to specify one parameter, because the other one is already known.

So, here this divergence is between these two Bernoulli distributions, ok. Now, let us see. If q equals to this ratio, that is this quantity is equals to this quantity, what is this value of divergence?

It is going to be 0 right. So at least I know one quantity is there in theta which is going to satisfy this. I could like let us say q to be just this ratio, which is always going to satisfy. If t is a some value greater than t, this is going to be some non-negative number. So, for some q this properties already satisfied.

So, I will have some element. But now, what is, but what I am looking is the maximum one; maximum one such that this inequality continues to hold. Now, you can show that this divergence, I have defined for the Bernoulli case. If I am going to take this quantity q to be in the interval p to 1.

So, what I am basically doing? I am looking for all those q's which are larger than p ok. You can show that, if I choose q in this range, this quantity is going to be increasing in q. So, at p equal to exactly q this is 0; but if I start increasing q beyond that point p, this going is going to start increasing.

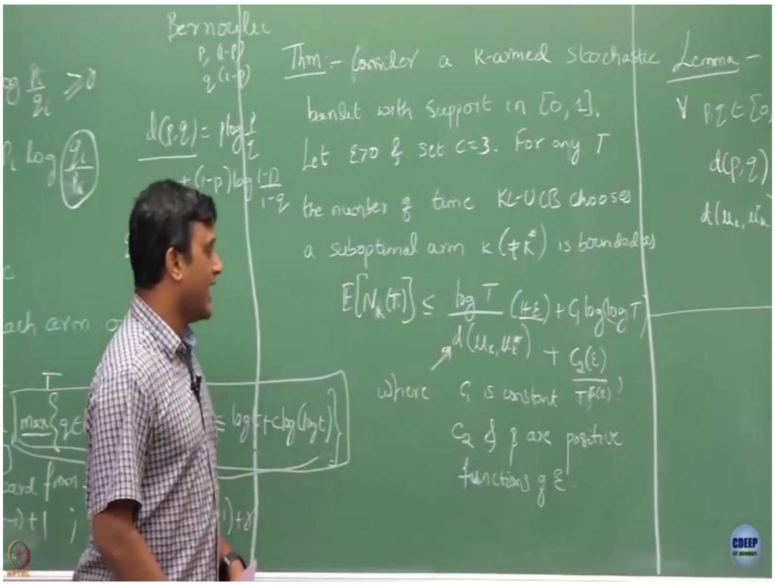
So, here at when you set q equals to this quantity anyway this is 0; but now you start increasing q, this guy starts increasing and it may happen that at some point, it may cross over this upper bound and for those q's this inequality is violated.

So, you are just going to look for that q, for which this inequality is violated and you are going to sorry you are just going to keep on increasing it and just see at that which point, this inequality is going to be violated and that is going to be your max value here. And, this is going to act as index for my arm a here; unlike we are just defining the index of the arm to be in a different way in this fashion here. Now.

Student: (Refer Time: 14:47).

Right. So, to say another input to the (Refer Time: 14:50), but anyway let us see. So, as I written here c right; you can choose any c, you want to use. But for I think the bounds, we are going to state it, we are going to state it for a particular value of c ok. First, let us I am now just going to state what is the bound, we are going to get for this algorithm.

(Refer Slide Time: 15:20)


Bernoulli
Thm :- Consider a K-armed stochastic bandit with support in [0, 1]. Let ε>0 & set c=3. For any T the number of time KL-UCB chooses a suboptimal arm k (≠k*) is bounded as
$\mathbb{E}[N_k(T)] \le \frac{\log T}{d(\mu_k, \mu_{k^*})}(1+\varepsilon) + C_1 \log(\log T) + \frac{C_2(\varepsilon)}{T^{\beta(\varepsilon)}}$
where $C_1$ is constant
$C_2$ & β are positive functions of ε
$d(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$
Lemma -
∀ p, q ∈ [0,
d(p, q)
$d(\mu_k, \mu_k$
$\le \log t + c\log(\log t)$

So, this algorithm has many terms; but finally, we will just see what is the final result, we want. Support in. So, you understand the meaning of this statement, I am just going to say consider a K armed stochastic bandit with support in 0, 1; that means, all the rewards are within the interval 0, 1. Beta are positive functions of epsilon. So, now let us compare this bound with what we had for the UCB algorithm.

So, for the UCB algorithm, if you recall let us compare the term involving log T. What is the bound you have for UCB, that had for the expected number of pulls of an suboptimal arm? If you recall that was like 6 log T divided by delta k square right and there were some other constants.

And now, here we have this another term log log T; but a log of log T is going to be much much smaller than log T itself right, so we will ignore this. And also, here also this is not n, this is T and this $C_1$ and we have another term, where this depends on epsilon and also something T to the power.

So, the T already appears in a denominator. So, we will also not worry too much about this. The largest term here is the 1, here is the log T term. So, now, let us compare what we had for the UCB with this. So, now the difference is ok, we have this 1 plus epsilon here which is the small quantity, I can choose epsilon any, this should hold for any epsilon.

But now, in the denominator, there I had delta k square, but now this one is divergence between mu k and mu k star. So, now how big is this quantity depends on this denominator? So, we can argue that this divergence between mu k and mu k star is actually smaller than delta k square ok. Let us see why is that? So, that follows from a results called Pinsker inequality. Anybody heard of Pinsker inequality? Ok.

Student: Sir?

Yeah.

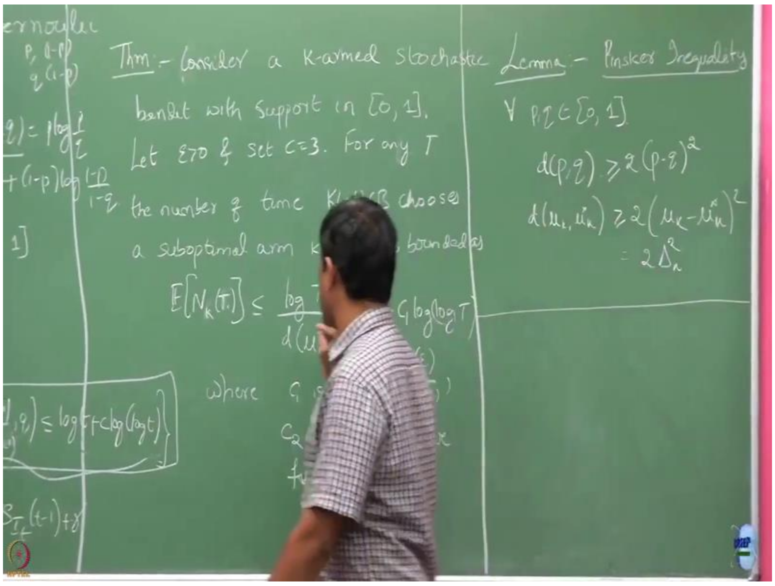
Student: Is the divergence is smaller than delta k square?

Yeah.

Student: Then, (Refer Time: 18:36).

Sorry. Divergence is larger, I mean sorry; so larger. So, this divergence is going to be larger than delta k square.

(Refer Slide Time: 18:48)


Thm :- Consider a K-armed Stochastic bandit with support in [0, 1].
Let ε>0 & set c=3. For any T
the number of time KL-UCB chooses a suboptimal arm k is bounded as
$E[N_k(T)] \leq \frac{\log T}{d(\mu}$
$c_1 \log(\log T)$
where
Lemma :- Pinsker Inequality
$\forall\, p, q \in [0, 1]$
$d(p,q) \geq 2(p-q)^2$
$d(\mu_k, \mu_k^*) \geq 2(\mu_k - \mu_k^*)^2$
$= 2\Delta_k^2$
$d(\cdot, q) \leq \log t + c\log(\log t)$
$+(1-p)\log\frac{1-p}{1-q}$

So, we have a lemma which is called Pinsker Inequality. It says that, you take for all p and q in the interval 0, 1 divergence between p and q, this is lower bounded by 2 times p minus q whole square. So, in terms of if I write this, so if I am going to take mu k mu k star, this is going to be then like mu k minus mu k star right, this is nothing but 2 times delta k square.

So, I will see that this guy here which is appearing in the denominator is larger than this quantity which was appearing in the denominator in the UCB. So, that is why you expect this guy this bound to be tighter than that of the UCB.

So, we will skip the proof of this. This is a very commonly used inequality in many information theoretic analysis. It is also straightforward; you can look in the book, ok fine. And we know like once, we have a better bound on the expected number of pulls, we also have a better bound on the expected regret also right. Because by the regret decomposition we know that the expected regret is nothing but delta k multiplied by this quantity and summed over all arms.

So, now the question is how to exploit this fact? So, how did this bound come up then? Yes, fine; I once I have a bound you can see that if I can apply this Pinsker inequality, I see that this bound seems to be better than that what have in UCB. But then, first of all how did we get this bound?

So, we are just going to discuss one result, which will give us a better concentration bounds than what we had obtained using Hoeffding’s inequality for the case of Bernoulli distributions. So, like see like this guy, he is dealing with a divergence which are requiring you to compute it divergence on the Bernoulli distributions right.

So, the Hoeffding’s inequality when we used that are the general case as long as your rewards are bounded that was applicable. But now, if you have a simpler case of Bernoulli distribution maybe you can come up with a better concentration bounds and that is what this KL-UCB exploits.

And now, we are going to see what is that better concentration bound we are going to get, if you have a Bernoulli distribution.