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

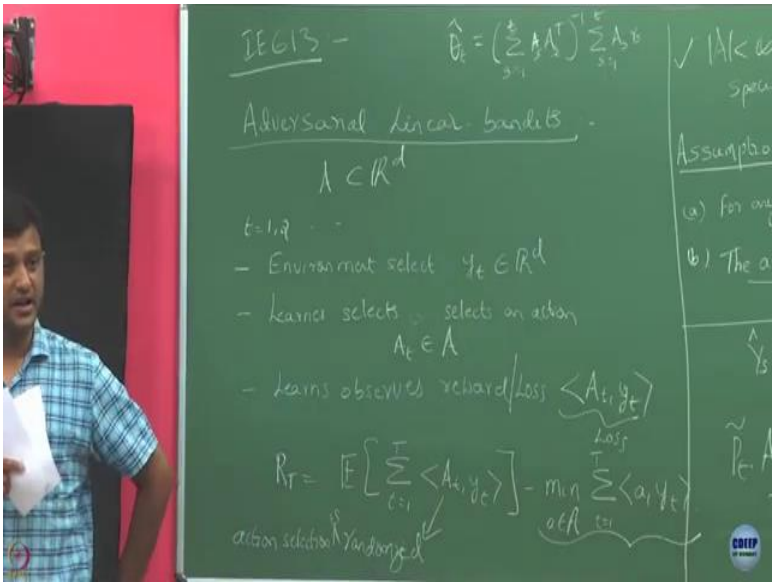
Lecture – 54
Adversarial Linear Bandits

So, when we looked into the contextual bandits initially; especially the stochastic version we made an assumption that, my mean rewards are all linear in the context right and we then try to solve that problem by building an algorithm that especially uses the confidence ellipsoids. So, the main problem there was, how to construct the confidence ellipsoid.

So, in that problem, what we assumed in the stochastic case? We assume that there was a fixed parameter θ^* and if you are going to if you see if you are play going to play an arm x. So, arms were there like vector. So, if you play some x, you said the mean reward you are going to get is x transpose θ*, right. So, we said that the rewards were linear in some unknown parameter θ*.

Now, we will move to the adversarial version of that, in which that this θ* need not be fixed and unknown. Yes, like it is unknown; earlier in the stochastic case it was fixed, but now we are going to assume we are going to consider set of where, this θ could be selected by an adversary in an arbitrary fashion. So, then how what could be the learning set up, ok.

(Refer Slide Time: 01:53)



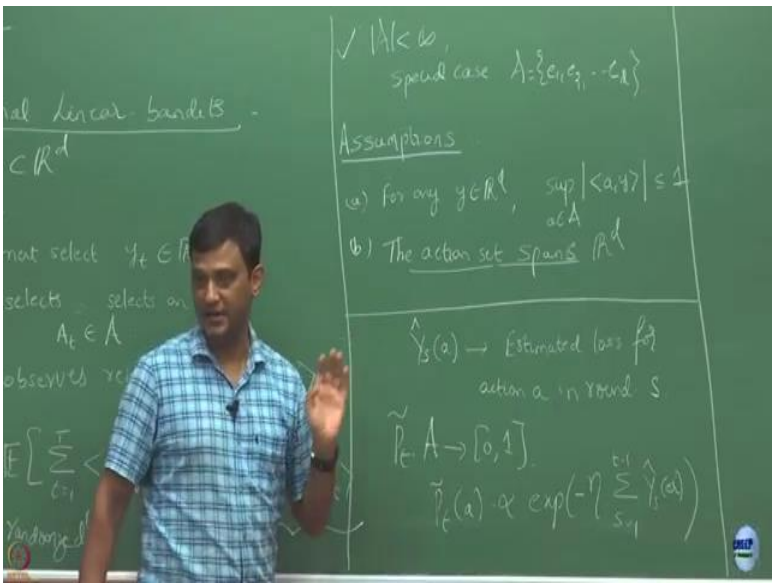
So, we are going to study adversarial linear bandit. So, what we are going to assume is like let say you have an action set A which is a subset of R_d and this is how the game setup in whatever the in each rounds, is that fine? So, the sequence of y_t is we selected by the environment, and now you have to select an arm in that round and by selecting an arm A_t you are going to get a reward, which is the inner product of this A_t and y_t ok.

So, just to compare and of and naturally your goal is to maximize your reward; cumulative reward. So, I am just going to consider the loss setting here let me call this as loss. So, this is like loss. So, this is what? If you know the sequence of let say y_1, y_2 … y_T, what does this give you?

So, this will give you the best loss you would have incurred in hindsight right, if you have known all y_1, y_2 you are now looking for, what is the action I should be playing.

So, that over this sequence I get the smallest loss and what is this part? This part is like you are playing some A_t in round t and this is the loss actually you are incurring, and this is the total loss you have incurred ok. So, this is for a given sequence y_1, y_2 all the way up to y_t. So, then I have why this expectation here? So, the learner could select this A_t's in a random fashion ok. So, the randomization this could be randomly selected. So, now we are interested in this set up, where I want to minimize this regret, ok.

(Refer Slide Time: 05:02)



So, let say initially my A is finite, that is my set of actions is finite and let say further as a special case, let say I am going to set A to be e_1, e_2 … e_d. If I have this set up, what is this set up then? In every round, the adversary is going to assign a reward. So, here I am saying this is the vector right like in the k arm, adversarial setting we had we said that the adversary is going to assign reward or loss to each of the arms, whichever you are going to pick you are going to observe only the loss from that arm others you are not going to observe.

So, here we are going to pick a particular A_t, that is one of this unit vectors then you are only going to observe that component of y_t and the and that is the loss you are going to incur.

And, this is simply which is the single best arm you want to pull in hindsight right, because A's are all coming from that unit vectors that is just an which is the single best arm I should be.

So, this in that way this is just like a generalization of your k arm adversarial setting ok, but now you are allowing the environment anyway it is going to choose a vector y_t, but now I am allowing the learner to play this actions A_t which are not just unit vectors. It could be any subset of R_d.

So, for time being henceforth, I am going to only focus on the case when $|A| < \infty$; this is finite, then we will discuss what happens when this is not the case.

So, before we continue we are going to make the following two assumptions: one for any $y \in R^d, \sup_{a \in A} |< a, y >| \leq 1$. So, what this says? Take any y that is y that is selected by the adversary or the environment. For playing any action, the reward you are or the loss you are going to see is bounded by 1.

So, this is we are just making sure that, their losses are the rewards they lie in the interval [0,1] ok. So, this is like equivalent to like when we did stochastic case, we assume that the means are in the interval [0,1], right. So, the, are the supports are in the interval [0,1], so, we are making just the same assumption.

If this is not the case, then you just scale everybody appropriately, so that you bring down the rewards at every any round to be in the interval [0,1]. So, if this is not the case your

algorithm will just scaled by whatever is the maximum loss you are going to incur in any of the rounds ok, this is fine.

Other thing, I am going to assume is, the action set spans R_d or I am going to assume that A is the basis for forms a basis for my R_d, ok. So, I will see that we are going to use this assumption when we when we are going to have a derive the regret bound for the setup. So, this will just like this assumption also make sure that, since it spans R_d I want to like.

So, this helps me to explore all possible directions of my y_t vectors whatever I am going to see. Like, so in each of the terms I want to explore well, that is why that will if this is the case I can achieve that target, fine.

So, now is the set up clear for this adversarial linear bandit? This is the setup we have, under this assumption now what is a good algorithm to minimize this regret. Now, what is that? So, can you think of any algorithm, any generalization of the algorithms we already know? We already know when this special case when my action sets are all just the unit vectors, we already know how to solve this problem right. I am going to simply use like EXP 3 or EXP 3 IX.

Now, we have just a generalized version of this, what could be a good algorithm? So, when we studied this EXP 3 there the main thing for us was how to estimate the losses of each of the actions in every round, right.

Like the for some actions which I actually played I observe the reward or loss for the once which I did not observe, I did not have any information, but still in that round I want to estimate the losses for each of the arms in that round.

Now, the arms have been replaced by my action set here, similarly I want to do like the same thing in every round I want to estimate the loss I would have observed for each of my actions. And, that is possible for me, if I can estimate what is the y_t that would have occurred in round t.

So, if you somehow figure out what is possibly potentially the y_t in round t, you could just go and find out for each action, what is the loss you are going to incur, and from that you can go to play a one which has the smallest loss right.

But, the question now boils down to how you are going to estimate that y_t in any given round ok. So, for that if you want to estimate we also want to ensure that whatever we are estimating is unbiased, the way we did it in using the importance sampling method. So, now we are going to now discuss about how to do this.

Suppose, you could figure out. So, let this denote the loss the estimated loss for action a in round S. So, this an estimated value in round S, for the loss incurred for action a. Suppose, if I can estimate this for all the actions a; so, then what would be how you are going to play an arm in that round? So, we can do that exponentially weighted distributions right.

So, in that case we know that it has a better properties right like we have been using it many times in our EXP 3 algorithms. So, then we want to construct a function in round t that will give me a probability distribution and we know that one particular way to do this is.

So, this is what like the exponentially weighted probability distributions we have, provided we could estimate, the losses I am going to incur for each of my actions in that round.

Student: Sir.

Yeah.

Student: Sir, it is action set with span r d.

Yeah.

Student: We can do a; we can do a transformation such that the like a space of this vectors.

Yeah.

Student: Basically, they will get transformed into like orthogonal vectors like yeah basically, basis orthogonal basis functions and then we can the reward this y_t what we have whatever we are getting we can separate it out into components in these directions, and then would it be, would not be this because similar set of (Refer Time: 15:35) ok.

So, it again so, like saying if you have this action set if that can span?

Student: They may be correlated these actions may be correlated.

Now actions is could, but you are saying I can always come up with. I will transform them and try to get this basis vector.

Student: Yes.

But, then you can map the losses one to one in that case.

Student: Yes (Refer Time: 16:00).

So, suppose let us say I have some action a and I. And, let us say this $\hat{A}$ is your transformed basis.

Student: Sir, cardinality of the actions it is d or is it more?

It could be more.

Student: It could be more.

Only dimension is d.

Student: Fine.

It could be large.

Student: Ok.

I only said A is a subset of R_d. I did not say.

Student: (Refer Time: 16:25). It has exactly d elements in it. This is just a special case.

Student: (Refer Time: 16:31).

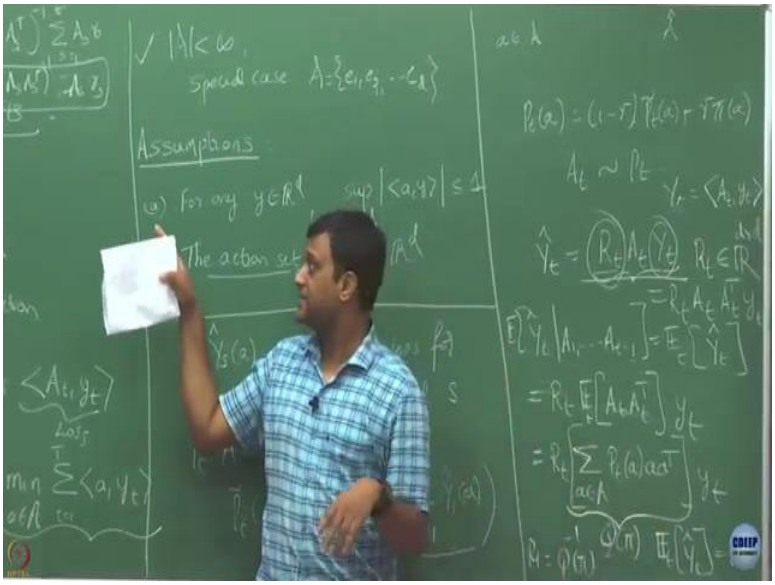
Where it had exactly d elements in it, ok. So, I do not see like how you can just restrict any of these action space to a particular to a basis and then, you still be able to do a one to one mapping between the loss for a particular action from this and loss and map it to a particular action in the transformed space, ok.

So, I do not see that, but let us see how to just do with whatever we are given. So, we are given an arbitrary subset of R_d and we this is this constitute our action set. So, now we know that, if we go with such a probability distributions, we have already seen that even

though it gives us that unbiased estimation like in the in the EXP 3 setup I am talking about, but we saw that its variance could be very bad.

So, the way we handled that bad variance how did we handle that? In we introduced γ otherwise we also deliberately include an exploration term there right, ok.

(Refer Slide Time: 17:54)



So, let say so, one thing we can possibly do is I can come up with instead of just going with like this we can go like. So, π_i is going to be my exploration distribution we will specify, how it looks like.

So, once we could do this then I am going to once I can construct like this I am going to simply pull an action A_t which is drawn from this distribution play it and observe whatever the reward in that corresponds to that. So, fine, now what remains is, how to estimate this? If I have this, I can do everything now, how to do this.

One possible way estimate my y_t in round t is that, is in this I am going to estimate and y_t and what is $\widehat{Y}_t$? y_t is the loss ok. So, let me just write y ok. So, let see this suppose I am saying that let say in in round t you played an arm A_t and you observe this reward y_t, ok.

So, once you play going to A_t whatever been observed, whatever been the selected by adversary y_t you are going to observe the inner product of this let us call this A_t.

So, if I want to make estimate my $\widehat{Y_t}$ based on this, now what is unknown for me is this R_t is a thing which is not specified here. Now, how can I make this $\widehat{Y_t}$ unbiased? What how should be I choosing my R_t?

Suppose, now let say let me take expectation of suppose $\widehat{Y_t}$ given all the things we have observed all the thing should. So, A_1 …. A_{t1t-1} is the action you have already played till t - 1. This we already know. This we have done before time t. Conditioned on this.

Student: Sir, what is A_t? A_t arm we are taking?

Yes.

Student: So, what is A_t Y_t?

This is a scalar right.

Student: Yes.

This is a vector and now, I am want to construct a matrix R_t.

Student: So, A_t is the learner selection (Refer Time: 21:12) ok.

A_t is what? The learner selected in round t right.

Student: Sir, distribution or the d (Refer Time: 21:22).

It is the actual action, the actual action played in round t.

Student: (Refer Time: 21:26).

Yeah, the d dimensional action. So, he is going to select an action A_t from A right, according to some distribution.

Student: Ok.

He has selected, but he has selected that.

Student: It is just a scalar.y_t is just a scalar because it is just a inner product. This is scalar that is fine; that is fine. What I am saying is, now this R_{tI} want to come up with this is going to be what y_t is d dimensional right let me say $d \times d$. How should I select my R_t

vector in round t, so that this guy conditioned on what have been I have observed so far this becomes an unbiased estimate of my y_t. See, $\widehat{Y_t}$ is what? This is an estimate for y_t, I am just trying to estimate this quantity y_t. This is the estimate for entire y_t.

This is for each action right. This is $\widehat{Y_s}(a)$. So, this is for one particular a, this is the probability of selecting a, with this; this is proportional. So, if you want to exact you can just normalize which happens to be the just sum of all these quantities.

So, now suppose I take this R_t, now what is random here? A_t ok. So, before I write this. So, if I have just simplify this, this is going to be $R_t A_t A_t^T y_t$. So, I just replace this whatever y_t that I am going to observe in round t, but by it is definition, ok. So, now this quantity is nothing but, R_t.

Now, expectation of whatever random quantities at this point is $A_t A_t^T y_t$ right. So, right now I am not yet specified, I am just whatever it is I am going to choose this R_t deterministically in round t. We will see what that is going to be and the expected value of this $\widehat{Y_t}$ this estimate condition on this is simply going to be R_t into this because the condition so, ok.

So, this condition this quantity is conditioned that I am so, when I say t this means this I have already conditioned on all the quantities I have observed so far. So, this quantity here R_t I am going to write it as $R_t[\sum_{a \in A} P_t(a) a.a^T] y_t$.

So, conditioned on A_1, A_2 all the way up to A_{t-1}, I already know what is my quantity this probabilities $P_t(a)$ and then, this quantity is then nothing expectation of this quantity is nothing but, you just take their values and multiply with the corresponding probabilities, right.

So, I am just now for this $A_t A_t^T$ I am just placing for particular $a.a^T$ and I am now going to consider an expectation with respect to the distribution P_t (a). So, a is what? a is the action one of the action in my action set A. I am just looking for all possibilities of $A_t A_t^T$. I am going to pick them according to this distribution $P_t(a)$.

This is conditioned on that because P_t of a depends on all these observation, but this does not y_t and as I said, R_t I am going to choose deterministically in that round, I have not yet specified. Now if I want to make this estimate here $\widehat{Y_t}$ and I unbiased estimator of this

quantity y_t, how should I will be choosing R_t? So, then if I choose R_t to be just an inverse of this matrix, then, this estimate becomes unbiased estimator for y_t right. So, let us choose this quantity to be I am going to denote this as. So, as we said this P_t quantity here could depend on the exploration distribution I have set.

So, let that is why let me call this as this whole quantity as $Q(\pi)$, then if I set my R_t to equals to $Q^{-1}(\pi)$ inverse then, my expectation of $E_t[\widehat{Y_t}]$ is going to be t.

So, when I say expectation means subscript t; that means, conditioned on this quantity. So, I am just. So, now you have kind of have built up an estimator for your y_t, if you have kind built an estimator for y_t we have done this job now. Now, you can just repeat the process, right ok. So, now let me write down the exact algorithm now. So, this is the general idea.

This particular form, this is basically coming from the intuition of what we did it in the least square regression, right. How did the least square regression work what was the estimate? We estimated $\hat{\theta}$ to be what v inverse of? So how did our $\hat{\theta}_t$ workout in the stochastic case?

$\hat{\theta}_t = \left(\sum_{\{s=1\}}^{t} A_s {A_s}^T\right)^{-1} \sum_{\{s=1\}}^{t} A_s r_s$. So, this is our estimator in the least square regression right. So, this was what A_s is the action you played in round s. Now, we know that in this case what we have a fix θ^* from which all these rewards the, are correlated through, ok.

So, that is why we all these rewards are correlated. So, we use all the information we will have till round t all from S 1 to t. So, now, what? When I have going to use adversarial 1 there may not be correlation across θ_t that the adversary is selecting. So, the θ^* need not be the same, right? It could be changing in every round.

So, I may only focus on one round that is what we I do not have summation here and for that just try to use this idea. So, now if you just ignore for all of them then for a single one it is going to look like $\left(A_s {A_s}^T\right)^{-1} A_s r_s$ right, if you have to deal with only one term here that is exactly what we have used.

But, instead of directly writing it like this we just wrote like this and just saying that to get an unbiased estimator of my $\widehat{Y_t}$ the way you have to choose R_t is like this, ok.