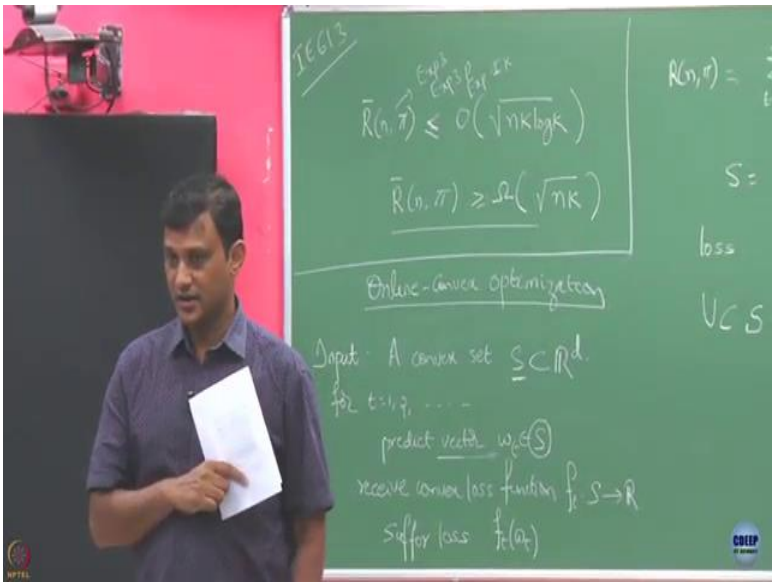
Bandit Algorithm (Online Machine Learning)
Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture - 18
Online Convex Optimisation

So, till the last class we were discussing about is adversarial settings where we went from the prediction with expert advice where we had full information to the Bandit setting. And now we in the bandit setting we came across different different algorithms right. What was those? Like Exp 3 and Exp 3 p and Exp 3 IX.

(Refer Slide Time: 00:53)



So, all of these algorithms if you run it over n rounds. So, pi here is the policy. What is the regret you got? If you just ignore the constants there the expected regret I am talking about maybe like I should write a bar here this I am talking about a pseudo regret here. We got it to be O(√(n K log K)) right. So, there are some constant like sometimes for Exp 3 we got a constant of 2 here right and then for Exp3 IX we got an another constant there.

So, I am just ignoring this constant other than that in terms of the parameters, number of rounds, the number of actions this is how it look like. So, of course this regret is sub linear, but then the question is whether this regret bounds are kind, are they good or how far they are from optimal? Ok. So, for that we need to know what is the best we can do

with any algorithm. Ok. So, there is a one can show that or one can come up with a lower bound on this setup saying that this pseudo regret when I say pi. So, here pi is first specific it was like either Exp 3 or Exp 3 P or Exp 3 IX.

Now, one can show that for any policy pi adversary can choose a distribution and come up with a sequence such that irrespective of what policy you are going to use we can make sure that you incur a regret of $\Omega(\sqrt{(nK)})$. That is no matter what algorithm you use adversary and or the environment can come up with a scheme in which are basically our kind of distribution which will make you incur this regret. So, how much is the gap between $O(\sqrt{(n\ K \log K)})$ and $\Omega(\sqrt{(nK)})$?

Student: (Refer Time: 03:48).

$\sqrt{(\log K)}$ right. But, if you ignore that $\sqrt{(\log K)}$ factor, K is fixed right K is the number of arms. What we were interested in mostly is how it varies with the number of rounds. Because if you run more rounds we want to see that how quickly we start doing good compared to the other hindsight strategy. So, if you just ignore for this bounds are like for most identical right maybe up to constants.

So, because of that these 3 strategy we already saw they are like optimal, if you just ignore this log K factor. So, we are now not going to show this lower bound proof we will revisit it at a bit later point where I want to now show this bound once I complete the stochastic bandit also. Right now we are only talking about adversarial setting right.

So, after some time I will also complete stochastic bandits, then we will going to revisit this lower bound. Because the lower pool proves uses some stochastic argument. Ok, now in a sense all these 3 algorithms they are optimal ok. So, now what we will do is we will stop our study of this special this adversarial bandit setting and today we move on to something called online convex optimization. And we will show that the convex the algorithm we are going to generate or discuss for online convex optimization they already captured what we have already studied for this adversarial bandit settings ok.

Also notice that when we started with initially the full information setting right that is prediction with expert advice. For that case, what is the regret bound we had? For the expert advice case when we use weighted majority algorithm.

Student: (Refer time: 06:15) √(2nlog d) right.

So, d is like K there. That is their number of experts or number of actions. So, the difference when we had a full information setting we already discussed this we would have gotten through weighted majority square root. So, O(√(n log K)). Even for that case, one can show that that is even your full information case that is the best one can get. The lower bound one can also show that you can that lower bound will be also order O(√n) and also there will be some logarithmic term there.

So, that also for the full information setting I did not discuss the lower bound proof. We will not do that. We will only eventually do this part, but the algorithm we studied already weighted majority for the full information case and this Exp 3 based algorithm for the bandit case they are optimal ok. Now, today we will move to something called Online Convex Optimization. Ok, before I move on to this topic any questions about this adversarial setting we studied so far. Yes?

Student: (Refer time: 07:57).

Yeah.

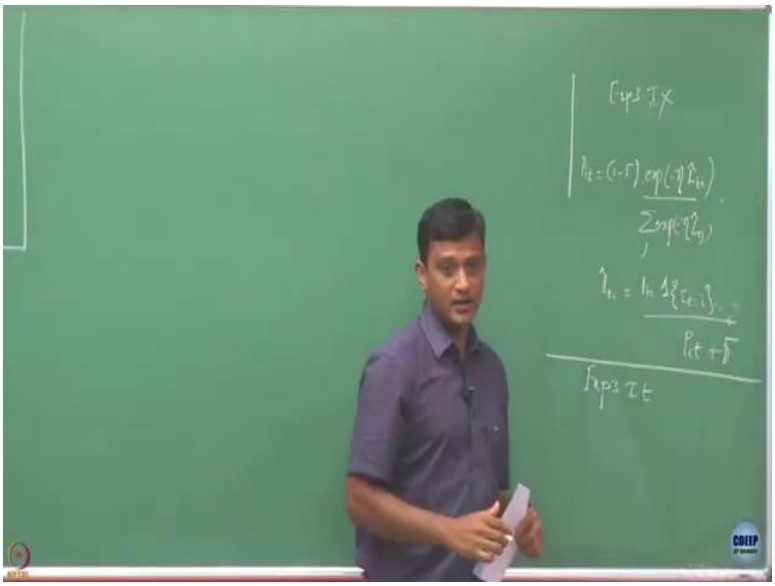
Student: (Refer time: 08:01) are if the (Refer time: 8:04).

Right. We already discussed that if you look into the estimators we have right. How are the estimates? There were some indicator divided by P_{it} and how was that P_{it} was divide in terms of the exponential weights right that was defined to be distribution. And we also argued that the variance is of the $O(1/P_{it})$ that we already discussed right.

So, if this P_{it} is very small, then this variance can be very high. But, once you add this P_{it} by adding that γ now we are variance is inversely proportional to $1/ (P_{it} + \gamma)$. So, the variance would not be bad right even if P_{it} is very small. It will be restricted by this γ fact γ term there. Exp 3 P, but the same logic also holds for Exp3 IX right.

Student: (Refer time: 09:13).

(Refer Slide Time: 09:20)



Same thing let me revisit that. In Exp 3 P, how did you define P_{it} to be 1 minus some gamma, then there was this exponential factors right divided by the summation and there was γ/K. And when we defined the estimators, the estimators were. How did we define? For my Exp 3 P, this was my estimator.

Now, if you look into this estimator, this estimator is already this probability is here there will be at least γ/K. Because of that it is variance would not blow up if this quantity is small because it is always making it at least this much. This P_{it} will be at least γ/K. Now, in the Exp 3 IX, P_{it} was just this there was no this term, but what we did? We just added γ here.

So, the effect in both the cases what to make sure that if you look into the various this term do not become arbitrarily small. There will be capped by this gamma term here. But, so then why it is called implicit exploration here. Like earlier you are forcing exploration, right. By adding gamma by K here, you are forcing some uniform distribution there, but now it is not there. You can just bit if you look into bit carefully that kind of exploration is already being done through this gamma, but not actually forcing that kind of uniform exploration. So, that is why it is called implicit exploration here.

So, adding γ here or here both are kind of doing kind of same job of reducing your variance. Anything else about at anything about the setup we discussed, the algorithm or

the analysis part, Ok. Now, how when we started with our method first we started assuming that my labels are generated by a hypothesis in my hypothesis class right. We called it as a realisability case. For the realisability case, we showed that we can get a regret bound of what. If you have a finite hypothesis class to learn from and it is a realizable place, what is the bound what is the algorithm and what is the best bound we showed?

Student: (Refer time: 12:54).

Yeah, what was that bound? So, further at least one thing we did is we had discussed halving algorithm and if we showed that its bound, it gives me what bound?

Student: (Refer time: 13:11).

Lock to cardinality of h, right. So, we show we are able to argue that if realizable case is there I can give guarantee that much. But, then if I remove that realizable case then we argued that adversary can make you incur linear regret by arguing the covers impossibility result right. But then we said if you. So, this is too bad then we restricted adversaries power by what? By allowing the learner to.

Student: (Refer time: 13:52).

Randomize his predictions and when the learner did this randomization then we are able to show that one can still come up with a sub linear regret; that means, we can do on an average as good as an I recall asymptotically. So, basically these 2 things realisability and the randomization helped us to come up with good algorithms.

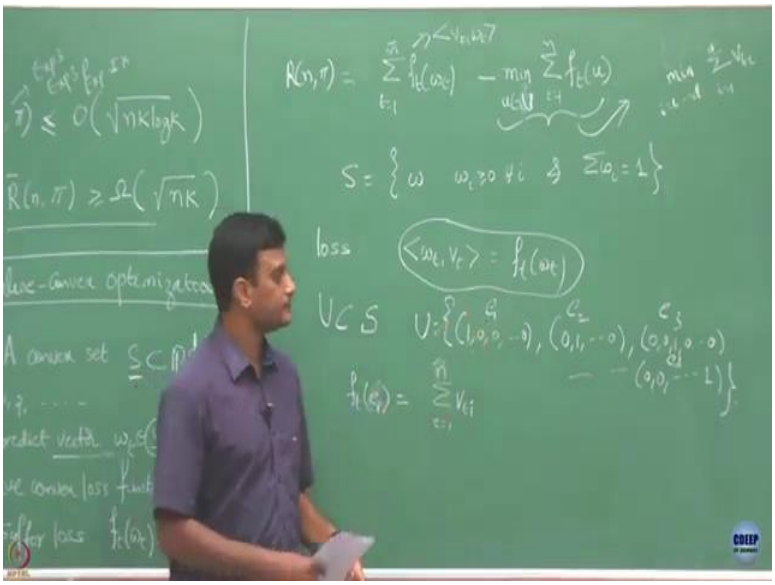
So, as we go along now we will show that these both case of realisability and randomization, allowing the learner to randomize this is basically coming from we are doing the convexification of the problems. So, this is nothing, but we are looking at some convex loss functions.

So, we will come to that, but before that I am going to just define what is the convex, online convex problem we are going to set up and then we will see that how all of those things will fit into this set up ok. So, input. So, this is the setup we are going to consider. So, in the online convex optimization setup we will assume that adversary is choosing convex function in each round ok. I do not know that function before hand, but in round

adversary is going to choose a convex function. My goal is to choose a vector in each round and I want to choose my vector such that, that function at the point that I have chosen is the smallest value ok fine.

So, we will see what is my objective, but this is the how the interaction happens. In every round, learner predicts a vector w t from a set S this set is assumed to be a convex set and after that the environment of the adversary generates a convex function. So, it is not a one value, it is the function it is from it is defined on the entire convex set. So, and then whatever the vector you predicted you are going to incur loss of loss computed at that w_t on this function f of t ok. This is what we are going to incur. Now, regret for the setting is defined as.

(Refer Slide Time: 18:03)



So, depending on how you are going to choose your w_t, how you are going to predict these algorithms are going to be different ok. So, let us say pi is your thing. Now, the total loss you incurred is f t w t. Remember, w t is what you are predicting in each round, f_t was generated by the environment, $f_t(w_t)$ is what you incur in t-th round and this is the total loss you incurred.

Now, you want to compare it against the smallest loss would have gotten by playing one point. So, what is that? Suppose, so, we this is the loss we incurred right I am going to compare it with the hindsight strategy. Suppose, you knew all the f_ts' from t to n rounds,

and now let us take this U to be for time being to be S only. So, we will discuss like this could be different, but let us say take for time being this because it is f_t is from S to R ok.

You want to play an U in this S, so that in hindsight that would have given you the smallest value. We are going to call this as regret and your goal is to come up with an algorithm that name as just this value ok. So, what environment actually? Environment is choosing this f_ts', you are choosing w_ts' in each round, this is the total loss you incurred and this is the best you could have done if you know all the f_t sequence over t rounds ok. Now, does this setup map to our earlier setup where in each round? A loss.

So, you remember we were earlier discussing the online classification problem right where in each round a point x_t was detected and you decide what is the label to predict. After you predict a label the true label would have been revealed and you would have incurred a loss right. Now, let us say can that setup be fit into this setup. So, any guess how you can do that?

One thing is let us say what will be the convex set in this case. So, there what we are doing? Every time you were coming up with the probability vector on the hypothesis right and then we were taking the average of the hypothesis we have, ok. So, can I take my S to be all my w's basically simplest with for all i and my summation w_i equals to 1.

So again now I am going a bit backwards. Now, I am not recovering this for the bandit setting, but I am recovering for the full information setting ok. So, recall in the full information setting what we did when we applied the weighted majority algorithm? So, in the weighted majority algorithm, we were maintaining a weights on all the hypothesis and then when the loss vector and when and we played an expert drawn according to this probability vector and then we incurred a loss and there we were interested in the expected loss ok.

Now, I am going to set this S to be this ok. So, there what was happening in each round adversary used to select a loss vector right in each round and then if w_t was your weight vector and V_t was the loss vector what was the expected loss you incurred? w_t.

Student: (Refer time: 23:08).

The inner product of $< w_t X_t>$ right. So, your loss was. So, can I treat this as $f_t (w_t)$? So, it is now w_t is what you chose now this is parameterized by the loss vector chosen that is and is this a convex function in w_t? Yes, right this is just the in fact linear in this w_t. And now let us chooseok. Maybe now I am just I also want to write this to be in U. So, I will just let us say I will be allowed to select from a reference point which is coming this set U which need not be always the same as this convex set S. It could be a subset of my convex set ok.

So, I am saying this U could be some subset or maybe like even if it is U, I could choose this U to be some set S where U is set of all unit vectors. What I mean by unit vectors? U is like 1 if it is let us say some d dimension. I am going to say that this is the convex set which is a subset in R^d. R^d is what? d dimensional Euclidean space. So, this will have sum like this (1,0,..,0), (0, 1, 0,…,0) and (0,…, 0, 1).

I am going to choose this U to be this vector this set ok. Now, if I do this then what is this term is going to be? So, $f_t (u)$. So, let me. So, this is like e_1 vector right this is like e_2, this is like e_3 and this is like e_d. These are the unit vector in the d dimensional space. Is that clear? Now, I am looking at this quantity over this unit vectors ok.

So, suppose now I take this some particular let me call it let me take this to be e_1. What is this quantity is going to be? It is going to retain only the first component in this loss right. So, $f_t (u)$ is w_t into sorry u into V_t and if that u happens to be this vector it is only going to retain the first component of this or in general if I am going to take e_i it is going to keep the first component of this. And now what I am looking at? I am looking at the minimum over this quantity right. So, it is basically looking at that component of my V_t vector which is smallest. So, this is like basically this portion is basically saying as minimum over i={1,..d} sum_{t=1}^{n} V_{ti}.

And now we already argued that this portion is nothing but $<V_t w_t>$. So, you see that this is the same setup as prediction with expert advice right. This is the exactly the regret there we defined and try to solve it. Now, what we are trying to do is just like instead of. So, this was like a what we are basically is trying to now treat it as a convex function by convexifying what we are earlier did is we are did this convexification by allowing my learner to randomize over his action and then taking the expected loss is going to incur.

So, because of that t our expected loss was defined in this case. So, we basically made this to be a convex function there even though it was linear.

Now, what we are now we are saying is let us try to do not necessary linear function like this. We can just try to solve a bit more one, a convex function ok. So, as you can see here the randomization we allowed on the learner is basically translated our problems, in basically convexified our problem ok. Now, we are looking to solve a problem here where I am observing a convex function in each round. Earlier it was equivalent to the convex function was earlier parameterized by the V_t, the loss vector we used to observe in every round and if. So, this V_t defined my f_t there and that was what environment chose that is I am now calling it as f_t function and w_t is what you chose as a learner.

Student: (Refer time: 29:14) everything is that.

Right.

Student: (Refer time: 29:27) is not the (Refer time: 29:28).

Yes, it could be something else.

Student: (Refer time: 29:31) discuss.

Yeah. So, I am just saying what we studied earlier happened to this special case of this setup by choosing your u to be in this fashion. That is why I am saying, we can it is not necessary that u this u has to be the same as S we will be interested. This is like my benchmark right. I may be interested to customer benchmark in whichever set I am interested in. I may not always go for this entire set S, ok.

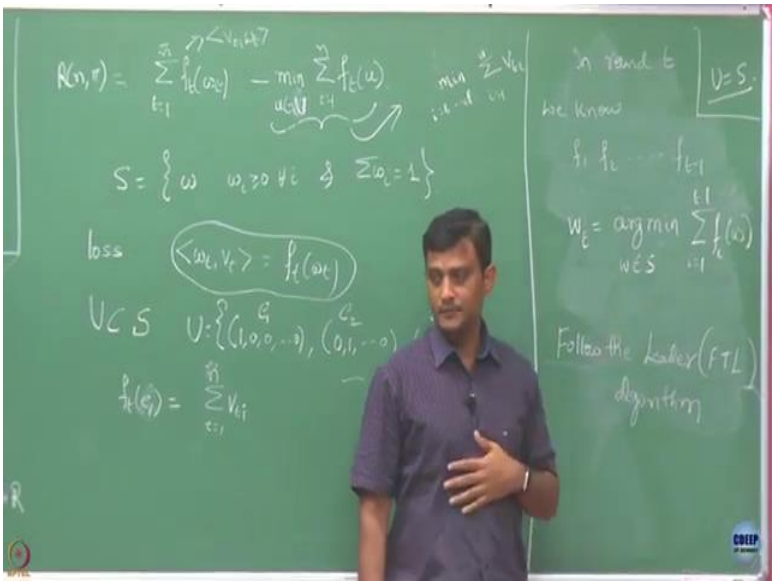
So, if in this case if you set this u to be in this fashion then you are already getting back your full information bandit setting fine. So, right now what you will do. We will simply focus on this set up now and try to see how we are going to solve by considering different convex functions. This was one convex function ok, but convex function could be more than just linear right we will consider the other aspects. What is the algorithm? The setup is clear, the goal is clear, now what is the algorithm? How we are going to approach this? Ok. So, if you recall the consistent algorithm we had, what we did in our consistent algorithm? Was that a halving algorithm or consistent algorithm?

Student: Halving.

That was halving ok. Just before that also like in the consistent algorithm, what we did? We pick then hypothesis from a hypothesis class and we checked how many of the hypothesis were consistent with the observed label. We retain them everybody else withdrew and whichever we picked in that round whatever that predicted we calculated our loss based on that. So, what we try to do is every in every round we try to keep the hypothesis which you are consistent with our observation and threw everybody else ok.

So, in a way what I am doing? We are doing is trying to see which are the hypothesis that are performing better at that time and we know that by the realizability assumption there the good one has to be there ok. So, drawing similarly, there what we did we tried to be consistent with our observation every time. Now, what we can do? We can also be in when we try to do this we can try to be consistent our base try to find my prediction that best splits my past observations ok. So, I am saying in round t after we made a prediction f_t is revealed, but before round t you have seen f_t s' of all the previous rounds right ok.

(Refer Slide Time: 32:46)



So, let us say consider round in round t we know f_1, f_2, are till f_{t-1}. This function is revealed to be what I do not know is what is that will be selected in the t-th round. Now, I have to make a choice of w_t in the t-th round. So, what is the best thing I can do?

Student: (Refer time: 33:18).

Yeah.

Student: (Refer time: 33:20).

Why you want to do that?

Student: (Refer time: 33:23).

Yeah. So, this is so far we know and I kind of want to do this, but I do not have full all the n observation, but at least I try to be consistent with what I have already seen so far ok. So, then one thing is to do is you select a w t which is going to be I henceforth we will just to take this U and S to be same like just to I am what every time writing U and S let us take U=S. Then I am going to do is argmin_{w ε S} sum_{i=1}^{t-1}f_i (w).

So, what I am trying to do is in round t I am just trying to see which is that value which is consistent like when I say consistent that is basically minimizing my losses that have been observed so far and this algorithm is called as Follow the Leader algorithm. So, what I am doing? I am just specifying that how I am going to choose this w_t and the way I am choosing w_t is as per this minimization function. Ok, fine. Now, the question is a fine you are trying to be consistent with your observation why does this work good?

Student: (Refer time: 35:38).

We are not making any assumption.

Student: (Refer time: 35:41).

Yeah, but we are just assuming they are all convex in this setup currently ok. They did not be any relation between the current like it is the same right like when we said when in earlier case when we said w_t is the loss vector that has been selected by that environment we did not assume anything like how V_1, and V_2, V_3 are related they could be arbitrary. So, here that is why we are saying this f_1 f_t it could be arbitrary. What I am interested is I want to minimize the setup ok.

So, next let us discuss what kind of guarantee I get if I am going to use this kind of algorithm Follow the Leader algorithm. So, any questions or doubts about the setup or

why is there anything else we can do in this setup, why to just Follow the Leader, is this intuited one or this is the most natural one. Anyway, so let us see how to get a bound for this. Now, why is that? Suppose let us say first in the first round you have to select w before you knowing the function let us say you did something. So, w_1 you has to select without any knowledge right basically.

Student: w_1 (Refer time: 37:14).

The w_1 earlier like if you are starting with a weighted majority right w on like a uniform distribution. I did not have any knowledge. So, I am going to put equal mass on all of them. I started with that and after that I observed the loss after I did that and then in the second round.

Student: Update.

I updated my weights.

(Refer Slide Time: 37:37)

So, you start with some w_1 because you do not have any information you play it you are going to incur some loss you would not have any control over there. Then you observed f_1 then w_2 how we are going to do? You are going to argmin over w of $f_1(w)$. You got this and you played it. Now, comes f_2. Now, how we are going to do w_3 is argmin over w of $f_1(w)+ f_2(w)$. This w_3 whatever you are get, why it has to be same as this w_2? It could be potentially different right.