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

Lecture – 17
Exp3.P and Exp3.IX

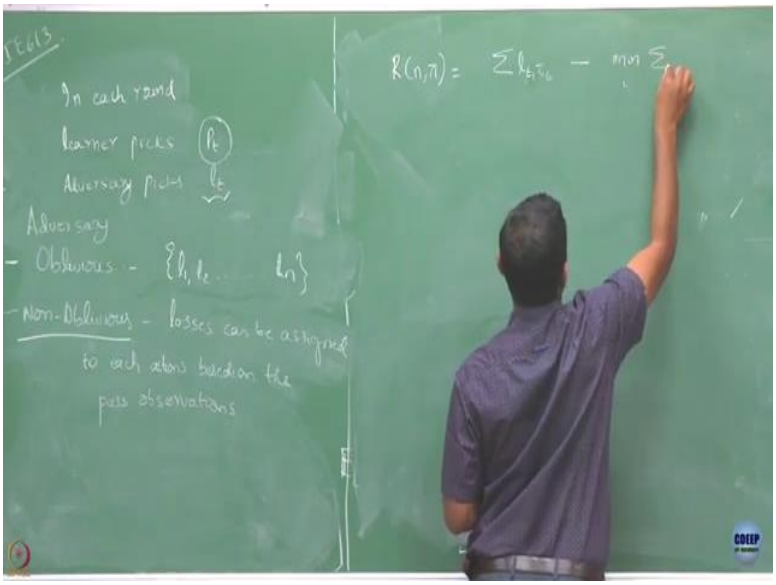
So, one more thing I want to make a distinction between the type of adversaries we are going to see. We said that, the adversary cannot see what is the loss value predicted by the player in each round, he would have observed it in the past, but in the current round he do not know what is the loss value predicted by the adversary.

So, had he seen, had he is been able to see the loss value predicted by the adversary he can always make the loss value incurred by that adversary very high right he if we predicts the adversary the loss could be 0, then the learner predicts it to be 0, then the adversary can assign one to it and he can make him suffer more losses.

So, because of that we said that we allow the leaner to randomize his strategy and because of that adversary and we did not allow the adversary to know what is the strategy that the player is going to I mean whatever the final action that the player is going to select in that round because he is basically randomizing it.

Now he is not able and see what because of this randomness what is the action that the learner will be selecting, but once the learner selects in that round he knows the adversary what is the action that the learner took and because of that he cannot manipulate what and he is going to assign the losses before the leaner selected an action in that particular round ok.

(Refer Slide Time: 02:37)



So, suppose let us say the learner is going to randomize, but at the end of the randomization he picks a particular arm so, what we said I can revisiting that in each round learner picks distribution P_t and he picks loss vector l_t. The adversary may know might have observed what is the actual action that has been selected by the learner, but in the and based on that, he may have some information about how the learner is going to come up with this new strategy.

But suppose the learner is using a deterministic strategy, then the adversary can what he do? He can see what is the next action that the player is going to play and give a very high loss to that, but given that in the t^{th} round we have made the learner to pick these actions according to some distribution the adversary his is no more having that capability because you are choosing this actions in according to some distribution he cannot particularly makes some actions to have a very high loss and others low loss right because he a priori do not know which is the action you are going to select because you are randomizing.

But once you select an action, he knows that in that round you selected that action and maybe in the next round, he may get some sense of what you are doing, what is your strategy and accordingly he can come up with his losses, but still in the next round you are going to randomize your actions he do not know precisely which is the action you are going to pick and he cannot penalize you on that itself.

So, because of this even adversary can update the way he is going to pick his losses in every round because adversary we are not assuming anything about the way he is putting losses right.

So, we can consider a worst case and see that, this adversary he is choosing this losses in such a way that, he will make you incur maximum penalty or maximum loss. So, based on this we are going to consider two types of adversaries called oblivious and non oblivious. So, we will come across this terms in the literature when you read about this adversarial setting.

What does oblivious says, oblivious says that the adversary is not adopting his losses by observing what you the learner is doing ok. So, for example, what we can seen, the loss vectors for the t rounds before the game starts, the leaner can decide a priori this is how them I am going to assign the losses in each round without knowing what the learner is doing.

So, he will come out with his losses he fix that and you are going to play against that. If this is the case, then we are going to say adversary is oblivious you still do not know, the learner do not know what is the sequence it is just that the adversary has come with this sequence before the start of the game. But in the non oblivious, the adversary can in each round can decide what is the loss he is going assigned to the actions based on the past observations.

So, understand the difference this two now oblivious and non oblivious adversary. So, which is the more difficult adversary?

Student: Non oblivious.

Non oblivious right because he can in each round, he can see what you are doing and try to incur maximum damage to you. But the analysis we did so far, does it hold for both the cases or on any other specific cases.

Student: (Refer time: 07:55).

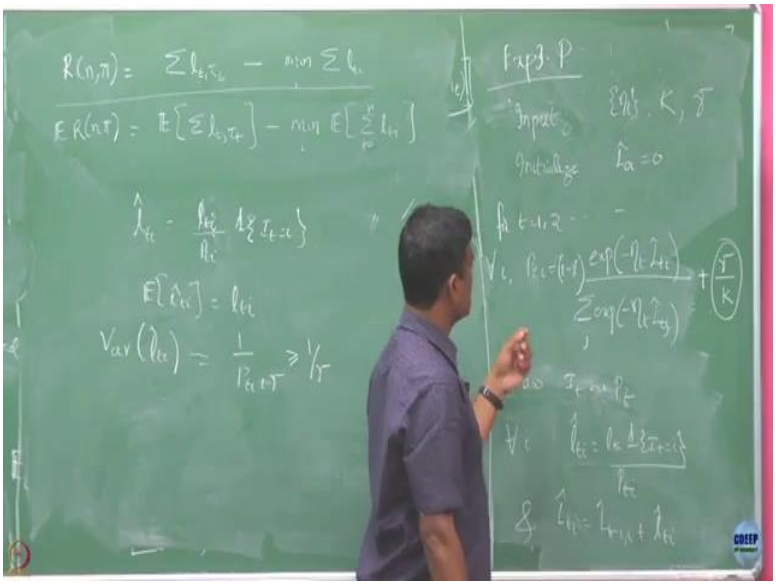
So, we did not assume whatever the strategy of the adversary we say this right like even he may be updating his losses based on the observations made so far so, our analysis still holds because that is why we have two expectations right like one expectation is over the randomness of the adversary.

So, in each round based on this observation the adversary can come up with the new loss according to some he may be updating his distribution in every round ok. So, you adversary seeing that you have playing some strategy by seeing what is the actions you are going to select, based on that he may be updating his strategy in every round.

May be one possibility he may have his own P_t to say I am going to assign high values to this actions and low values to this action whatever and accordingly he can update his strategy in every round ok. So, whatever we have done the analysis can hold for the non oblivious adversary, but of course, this also holds for oblivious because oblivious is a simpler case I really.

So, in that case, if I take like this I really do not need to do the expectation on the strategy of the adversary pick right because adversary is already picked a sequence and I am facing that sequence whereas, here the adversary can keep on updating his strategy and there could be a randomness in the way he assigns the losses and that randomness can keep change from one round to another.

(Refer Slide Time: 09:39)



The way we defined regret of a policy π to be this and we have also said I am interested in the expected value of this. Now in this case, what I am seeing is second case what we said that this is going to be with which we called it as pseudo regret. So, this we are trying to guarantee the

regret in expectation whereas, this is like a when you come through a sequence of losses, this could be your true loss incurred.

So, let us say that I want to give instead of a bound on this expected regret; I want to give a bound on this regret where you are going to not take the expectation over both randomness of the adversary and the randomness of the learner. See when you are going to run this algorithm for n round, you are going to incur some regret based on the sequence of losses you have observed right.

But when we gave this bounds in expectation whatever the bound we get this need not hold on that particular realization because the expected bound can be different from the bound you are going to give on a particular realization right.

So, because of this what happens it may happen that in expectation this bound would be smaller, but if you look into one particular realization this bound could be pretty large. For example, this is like if you have a random variable X and its expectation. So, this X let us say it takes 100 values, when you draw one sample from this X that some value of that random variable could be much higher than this expected value right.

So, in this sets we may want to guarantee itself when you are going to look into when you draw one sample of this, what is the probability that this is going to be actually small or that the value you have gone got from this sample, it will have a bound that will have this that will be have the same bound that you gave on the expectation.

So, what now basically I want to ask this question or whatever the bound you have here let me call that as regret value may be just write may be I want to see that whatever the value of X you got it is going to be smaller than the expected value itself then if you have a bound on expectation, that bound also holds on this X with that probability right. So, same thing you may want to do here.

It is basically saying that if you observe one sample how does it concentrate about its mean value. Can we say something about these things ok. Now, but when we do this, we will now going to consider only the oblivious cases where we say that this sequence is fixed, but the adversary a priori, but I am only now looking into the randomness of the learner. Can I say anything about what bound this regret has you may be interested in that right.

So, now let us see if you can give a bound for this quantity rather the expected values ok. So, quickly one thing is when you want to give bound on the sample itself rather than the expected value, you want to make sure that the estimates we are going to have there of course, they should be unbiased and what is the other quantity that I would like.

So, what are the properties you look at when you are having an estimator? One is it should be unbiased, another one is its variance should be small right if its variance is too large than it may it fluctuates too much right.

So, let us now quicken into see that what is the variance of the estimates we have in the Exp3 ok. So, what is the you we have estimates to be of this format right. We know that in expectation this is equals to l_{ti}, but what about this variance? You can verify that this variance is like again of the order 1 by P_{ti} because like you this term gets squared and you will have P_{ti} here, but you can compute this to be like of the order 1 by P_{ti}.

So, if P_{ti} happens to be small in a particular round, then you see that the variance can be very large right. So, which the variance is very large, your estimates are not good enough and because of that you cannot expect like you are values to start concentrating well soon.

So, because of that, even though when we use this kind of estimator in the Exp3 algorithm, it gave us a good bound on the expected regret, but if you are going to look at the this quantity itself directly it may end we may end up getting a small a very bad regret. Because with some probability, this quantity can be very high that is like this may not happen with some probability that because every time we are going to draw a sample it need not be smaller than its expectation it may be larger than this also.

If at all I want to give a good bound in probability, I need to make sure that my estimators have good variance ok. So, how to get a good variance? Do you see any easy way? So, when is the variance is going to be bad man? When these guys are small when P_{ti} is very close to 0 this guys are going to be very bad that is when some quantity P_{ti} is very slow I am not picking it often right because the probability of selecting that term become small so, I am not observing it enough times.

One possibility is somehow if you can increase all of these by some factor γ right because of this, even if P_{ti} is arbitrarily small, this quantity will not be very large right what is that in that

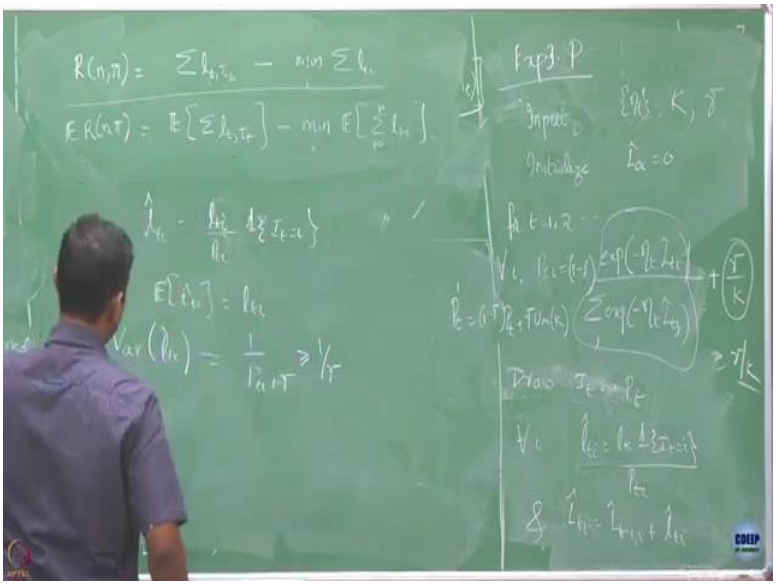
case it is going to be at least going to be greater than 1 by γ right you have ensuring that this score.

So, one way to ensure this is kind of do some exploration, let me write that now what this I is called as Exp3.P ok. So, again this algorithm has initialize P_t, then k then sorry this is input, then initialize this $\widehat{L_{0t}}$ equals to 0 than for t equals 1, 2, 3 you do the same thing P_{ti} equals to what for all i.

Now, you are going to do, now for all i you have the estimators. So, now, you are going to say $\frac{\gamma}{K}$ and for all i you do the same thing. So, what is the difference between the Exp3 and Exp3 algorithm which step you are differing?

P_{ti} right. So, I should be also adding 1 minus γ here and also γ is an input. So, now, tell me what is the difference between Exp3 and Exp3.P here what is that I am changing. So, γ is between 0 and 1. So, I am weighting this was there already right I am assigning a weight of 1-γ to this and weight of γ to this, what does this mean? I am going to choose an arm i which is a convex combination of these two.

(Refer Slide Time: 22:49)



So, this was my earlier probability and now this is my new. What does 1/ k means here what does that 1/k implying it is basically saying that, uniform weight. Now what basically it is

saying is I am taking uniform distribution with weight γ and $1/\gamma$ weight the other distribution I have right.

I alternatively what this saying is this is like P_t is whatever like the new P_{tI} am going to have 1 minus γs P_t, P_t was the earlier one with γ uniformed or k you understand this. So, earlier I was only up taking this probabilities with this, now I am perturbing it by a uniform distribution, but I am weighing them accordingly.

So, the earlier algorithm you had it was a kind of assigning the probabilities based on the losses you have observed. Even though there was some discounting factor here which control basically exploration and exploitation, but here you have deliberately bringing another term which is biasing it towards somewhat uniform exploration also.

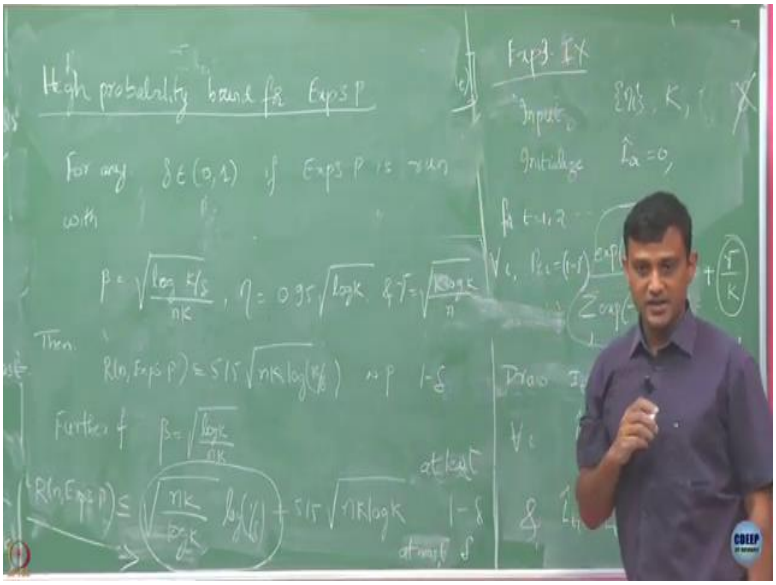
Suppose let us say you said γ equals to 1. γ equals to 1 means this term is not valid right no more effecting, what effects is only 1/k; that means, you are uniform distribution you are selecting. If you are going to set this γ equals to 0, it is exactly the Exp3 algorithm we had earlier it was only looking into this exponential weights, but now we are giving taking linear combination of this uniform weights and the probability defined through this exponential weights.

Now, with this kind of things, you see that P_{ti} there will be always one constant term we are adding right γ by k because so that, you see that this P_{ti} here will be never 0 very close to 0 it will be at least γ / k always right, this quantity is going to be the γ /k because if do like this I am controlling my variance right.

So, because of this, this P_{ti} is not going to be arbitrarily 0 it will be at least γ / k and because of that, this will be at least lower bounded by some positive quantity it cannot be arbitrarily 0 close to 0 fine.

Now, the question is, what is the bound I am going to get for this ok. I am just going to state this and you are not going to the proof you see already the proof of Exp3 that was already pretty regret so, you can go through this proof yourself, but the steps are largely the same except to take into account the factor, the additional factor γ / k.

(Refer Slide Time: 25:47)



I think, I made one more thing I had to add not only this, but there is also a β terms there which should called as an input to this algorithm there is another term β there for any.

It would not be, it would not be. Anyway we are basically biasing it. So, any so, we can come demonstrate this kind of bound for Exp3.P. So, now, we are saying that like suppose if you give δ as an input factor, another factor here is δ here. So, you see that we have so many input parameters here. So, if I am going to set β like this, so, maybe what you should do is only this are inputs you can set β in this whatever the way you want, but I am saying that, this is enough based on that you set your β like this and your η to be a constant, all this $\eta's$ to be like this and your γ to be like this.

If you can do this, you can show that the regret is upper bounded by some $5.15\sqrt{nklog\left(\frac{L}{\delta}\right)}$ and this bound holds with probability 1 - δ. So, because we are giving on the sample regret right it is not an expected regret it is a sample regret. So, we can only say this holds with some probability and we are saying that this holds with probability 1 - δ.

Suppose let us say you want this to hold with very arbitrary precision; that means, you want to set δ very small. If you set a δ very small, this β is has to be large right and you also see that if you set your δ to be very small, this bound is also going to be larger, this bound is like function of δ.

So, if you wanted with a arbitrary precision, you can expect this bound to be also worse larger right because if you want it to be like very high precision this is going to be bounded by this may be then the one can only get into that in that case, I will give you a larger bound like this.

And similarly one can suppose like here I wanted you to give δ as input. Suppose you do not give δ is input if this δ is not there then I will going to said β to be like this and suppose if I am going to stop after n rounds, I am going to claim that my regret is going to be this and this is going to hold with probability again 1 - δ.

If I knew δ a priori, I would have set like this and then I would have guaranteed this, if I do not know the δ, I am going to give you guarantee with high probability guarantee with 1 minus δ, but this bound is I can only guarantee this much you see that compared to this I have ended up with an extra factor here right so, this is the penalty I am going to incur if I do not know this δ a priori.

Any doubts about do you understand difference between expected regret bound and high probability regret bound. Which one is desired expected regret bounds are desired or high probability regret bounds? So, this is called high probability bounds because you are going to give the regret to hold with high probability and that is the with what precision you want it is given through this δ parameter.

So, any questions about or any confusion about difference between high probability bounds and excepted bounds or which one you prefer for algorithm. Let us say you want somebody to give an algorithm for your problem, if he gives expected guarantee you would be happy or if he gives to high probability guarantee you would be happy?

Student: High probability.

You would like high probability right. Why is that?

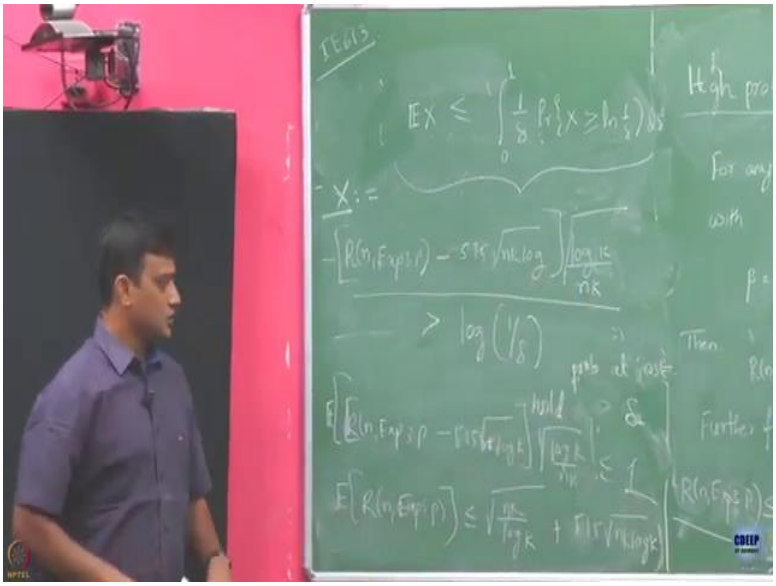
So, the one you actually face not what would you happen if you kept on running it multiple times or in expectation ok. So, these are desired, but these are also very sensitive in the sense you want this whatever the regret sample regret you are going to observe to be bounded.

It may happen that with remaining this is true with probability 1 - δ, with another δ probability it may happen that this will not satisfied this property and that value could be very large and

that is often the case ok. So, to control that with whatever probability you are giving this that is going to be well behaved like this it requires bit more efforts by tweaking your algorithm in certain fashion. So, that you control your variance ok.

I will just take couple of 2-3 more minutes to just introduce one more algorithm which you are going to see in the assignment. By the way, if you have an high probability regret bounds like this, one can always converted this to expected bounds you know how? Suppose let us say so, this is just I am going to write this here.

(Refer Slide Time: 34:18)



Suppose let us say X is a random variable, its values is upper bounded as $\int_0^1 \frac{1}{\delta} \Pr\left\{X \geq \log\frac{1}{\delta}\right\} d\delta$. Suppose let us say if you can tell for some random variable X what is the probability that it is going to be larger than 1/δ for a given δ if you know this, then you can translate that bound as a bound on your expected regret you understand this?

So, let us quickly see how to use this result in this case. So, here now random quantity is what this regret. I know that, if I am going to rewrite my regret, I know that if I am just manipulating this take it that side then this guy being greater than $\log\frac{1}{\delta}$.

So, I am saying that, this bound holds with probability 1 - δ; that means, that the opposite inequality holds with probability δ right if I am saying that this relation holds with 1 - δ, if I say that this if I look into greater than this holds with probability δ right.

So, suppose I know that this relation it holds probability δ. So, now, I if I am going to treat this quantity as X, I know that this guy being greater than 1 - δ that is log(1 – δ) this holds with probability δ. So, than this probability I know is at least δ this is with probability at least or at most because this is at least here. So, that is the case this is at most; that means, this term I can write it as upper bounded by δ right.

So, I am saying that this holds with at least 1 - δ. So, the negation of this should hold at most δ right. So, because of that if I am going to define this quantity as δ this to be X, then I know that this (Refer time: 38:20) holds by this propagation with probability δ. So, if I plug it in here, I am going to get this to be δ by 1 by δ this happens to be 1.

And now I know that this quantity in expectation is upper bounded by 1. So, that is $(E[R(n, Exp3.P)] - 5.15\sqrt{nklogK}\,)\sqrt{\frac{logK}{nk}} \leq 1$,then you can translate this to expected bound on this quantity which will turn out to be. So, you have through this and this relation we will end up with expected regret bound of Exp3 P.

So, a homework for you is go and compare the regret bound of expected regret bound you got this on the Exp3.P with that of the bound you got for Exp3 you have also regret bound there right just compare.

So, one last remark I want to let us say, there is one more algorithm that you have to implement in the book that we called as Exp IX. What that algorithm does is, instead of getting alpha by k here it adds a γ here explicitly in this that is the only difference.

So, earlier what you are doing by adding γ/K you are having an explicit exploration right we are forcing uniform exploration but now, by adding here you are not forcing that explicit information, but you are still enforcing that your P_{ti} this denominator is at least γ even this P_{ti} is 0 by this you can see that I am not having an explicit exploration, some kind of explicit implicit exploration happens with this addition and you will see in your assignment that the case we have specified this kind of algorithm actually performs better than your Exp3.P and Exp3 algorithm ok.