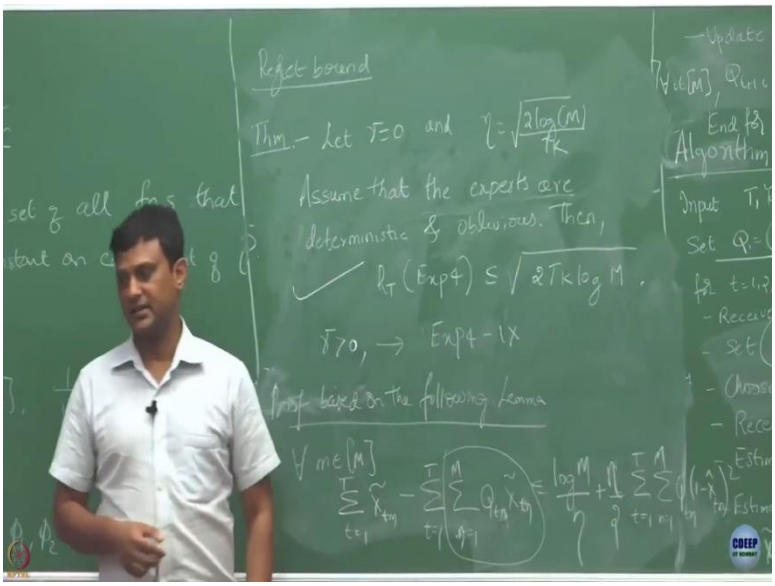
**Bandit Algorithm (Online Machine Learning)**
**Prof. Manjesh Hanawal**
**Industrial Engineering and Operations Research**
**Indian Institute of Technology, Bombay**

**Lecture – 53**
**Regret of Exp4**

(Refer Slide Time: 00:25)




So, now quickly discuss, what is the result we are going to get for this? So, we are going to state this for a particular case. So, let $\gamma = 0$ and $\eta = 2\sqrt{\frac{2\log M}{Tk}}$ ok. So, notice that like we are setting this γ to be 0 in this statement.

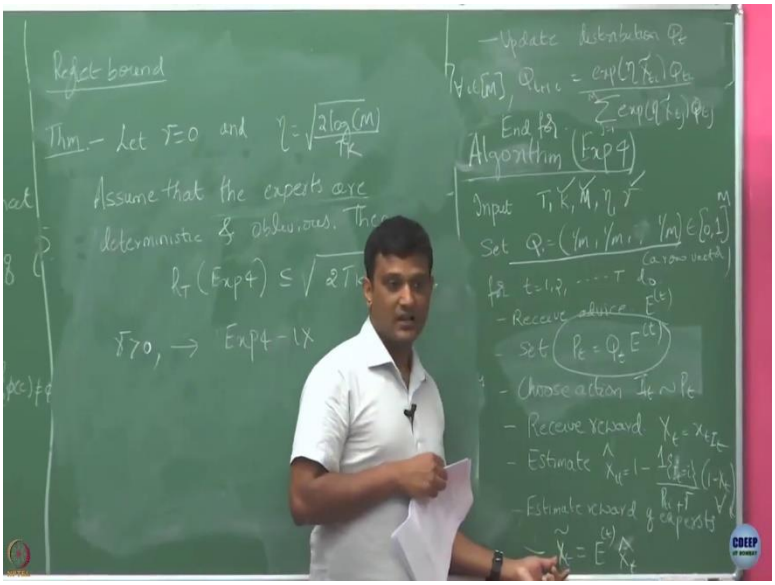
So, when we have this γ = 0, so we are just going to call this EXP4. But when under the case when γ > 0 ok; when we are this algorithm we are going to call EXP4.IX ok, when γ >0.

But the result translated here for the case γ = 0 and you can work out that for γ > 0 also similar bound holds, ok. So, now, let us revisit this theorem, what we are saying.

Student: (Refer Time: 03:20).

Take γ equals to 0 and you set your $\eta$ to be in this fashion and then assume that experts are deterministic, ok. What I mean by deterministic here is, as experts are actually giving me

a probability vector; but it is going to be the same every time I give a particular context, ok. And we are also assuming that these experts are oblivious. What does that mean?




So, that whatever based on whatever the rewards they are going to get right, they are not going to change their distribution accordingly. So, you in a way you can. So, this means like you can assume that each expert has already come up with what is the distribution he is going to put on each, the arms for each context and he is not going to change with.

Student: Answer.

What is the reward he has been observing so far? So, that is his, what is the context he observes that is irrelevant to him. You just tell me what is the context and he will just tell you what is the distribution. In that case then the regret we are going to get is $R_T\ (EXP4) \leq \sqrt{2Tk \log M}$.

So, this looks very similar to what we had gotten for EXP3 right; except the fact that this is now M instead of k. So, the regret bound you have EXP3 as $\sqrt{2Tk \log k}$ right, but now that k has become M now, where M correspond to the number of experts, ok.

So, the proof we are going to skip, when I am just there is one lemma which you need to get to prove this; I am just going to state this lemma, even that proof you can work out.

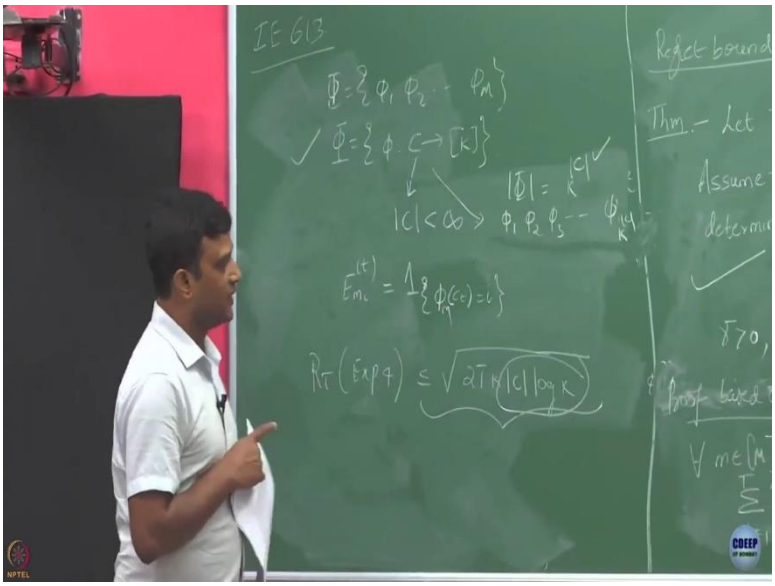
So, what this lemma saying is. So, what is this term giving you here? So, let us take an M an expert; what is this term giving you here?

So, if you look into this $\widetilde{X_t}$ right, what it give basically give?

The expected estimated rewards for that expert M, right. So, in that way we can treat it as the total reward that has been obtained by expert M, and then we are comparing it against; what is this quantity over here? So, what is $Q_{tn}$? This is the probability with which I am going to choose expert n in round t. And what is this sum is going to then give? The expected rewards that I would have got from the experts, ok. So, it is basically going to say how is the reward gotten from one expert compares with the mean that would have gotten from all the experts and it is going to bound and eventually this earlier I mean this theorem uses this fact to bound it. So, just look into the book like, how to get this. So, and using all these things, finally we are going to get this is upper bound, ok.

So, we will just skip it, I mean it most of the proofs are very similar to what is there in EXP3 again, ok. The ideas are all similar expect, except for the fact that, now we have to take into account two level of randomization into account; one with respect to the experts selection, and another with respect to the arm selection, ok. So, now, let us discuss couple of special cases.

(Refer Slide Time: 09:56)




Earlier one we started this was by saying that, I am going to take my $\phi$ to be $\phi_1$, $\phi_2$ some $\phi$ or $\phi$ could be all mapping from C to k. So, that final assumption we make that, the rewards are in the interval 0, 1 this X, $X_t$ vector right or $X_{ti}$ is in [0, 1] , ok. Suppose this C is finite. I mean the cardinality of C is finite and there are k arms. What will be the

cardinality of this set ϕ? So, it is going to be $k^{|C|}$. So, in this case if I took a particular ϕ, it is going to assign one value in k to each context, right. So, in terms of if I am going to think that as a probability vector, it looks like a unit vector right; where for that particular arm it is going to be one, for others it is going to be 0, you got this point. So, if I take a particular map here, it is going to assign a unique value to each value each context here; but I can think that assignment as a probability vector.

As how? I can assume that $E_{m_i}^{(t)} = 1_{\{\phi(c_t)=i\}}$, ok. So, what I want to write here is; let us call my functions here as $\phi_1$, $\phi_2$, $\phi_3$ ... $\phi_{k^{|c|}}$ .ok, I have these many maps.

Then can I map it the value recommended by expert m or the value recommended; if I am going to treat it as a map is nothing, but this right.

Student: (Refer Time: 13:25).

So, let us say each of these experts there are k to the power cardinality of c experts. And I am going to treat them; these experts are nothing, but these functions, ok. So, now, experts are nothing, but these functions. So, what are going to they going to; if I am going to give a particular context, what they are going to return me? It is they are going to return me the arm that should be selected according to that ϕ in that round right or based on that. But in terms of the probability vector, I can write like this right. For the m'th expert I can just write it as like this probability vector. So, this is just going to put value one on whatever the value this function assigned to that context t and everywhere it is going to put to 0, right.

So, in that way I can treat all the all the maps here is as one different different experts and we are already in this setup right, where I am going to set my $E_{m_i}^{(t)}$ to be exactly like this, based on what is the map I am going to use in that round. So, now, then even if I have all my if I am going to define my regret in terms of these policies all possible ϕ, is this regret bound still applies?

It should apply right; it is just like the experts are nothing, but these maps here and there also even though they are not giving distributions that i can take that as a distribution. Now if I am going to apply this bound on this, what is that I am going to get? So, in that case I am going to replace M by $R_T\ (EXP4) \leq \sqrt{2Tk\ |c| \log k}$.

But what is this regret bound like? We had gotten this regret bound by other method also right; what was that method? By applying EXP3 for each of the context; if I maintain my EXP3 for each of this context, I would have obtain this. Then what is big deal about this algorithm. Then why is this, why this I should; why this is of any interest? So, this say, when if I have to write this bound, it depends on |C|. When I applied all these algorithm, when I was in this setup; I only cared about the number of experts, I did not care about how many possible context are there. Whatever the context I have, I am just going to give it to an expert and that expert is going to give me a distribution on my arms, right.

So, this bound only worries about how many experts are there. It I do not care about how many contexts are there. As long as I have finitely many experts, this bound works. Right and often I may not as I said, often we will not be working with all the possible maps; we will be working with a restricted set off experts as I said like that as I discussed in the beginning of the class that, restricted class could be based on the partition or similarity. Or you are going to just only fix the number of $\phi$ is to be finite; like $\phi_1$, $\phi_2$ … $\phi_m$. So, in that case they you have only finitely many m and irrespective of how many context you are going to deal with, this bound is valid, fine. So, number of context is finite. This number of context is finite yes; but it could be arbitrarily large right, which I do not worry about here. I am just giving you the worst case, when you are going to deal with all of them. Because see, when I worked out with a maintaining one context Exp algorithm for each context right; I did not care about how many policies I will be competing against, it was just like maintaining one EXP3 for each one of them and that blindly that gave me this bound. But here I am well; I am deriving it based on how many experts I am going to deal with.

So, with this we will conclude this discussion on adversarial contextual bandits; we will not going to the lower bound proof and for all for this. As you see, do you expect this bound to be optimal?

You expect it to be simply $\sqrt{Tk}$ or M should come into picture.

Student: M should be there.

M should be there or at least order wise it you feel ok; square root $\sqrt{T}$ we know that we cannot do better than that, fine. May be like k also I cannot get rid of, because from the

adversarial setting I also know that it is like $\sqrt{Tk}$, right. So, for us now the new term that has popped up is log(M). Is that log(M) is the best or we can do better? Actually I also do not know maybe.

Student: I mean in the, it is similar in some sense to the weighted majority instance.

Yeah.

Student: So, in that it is that log d term is there. So, (Refer Time: 21:53) log M is somewhat similar to that log d term, where d is the number of experts in that. But we do not know weighted majority is the best; I mean whatever the bound we got, that is the best we could get.

Student: That is for (Refer Time: 22:06).

That is full information whatever it is.

Student: k is of.

Yeah.

Student: k is of or that is there any $X_t$.

k would not be there; because it is a full information case, where you are dealing with the bandit case here, right. So, k will come into picture definitely, because we are having only one kth of information compared to the full information case here.