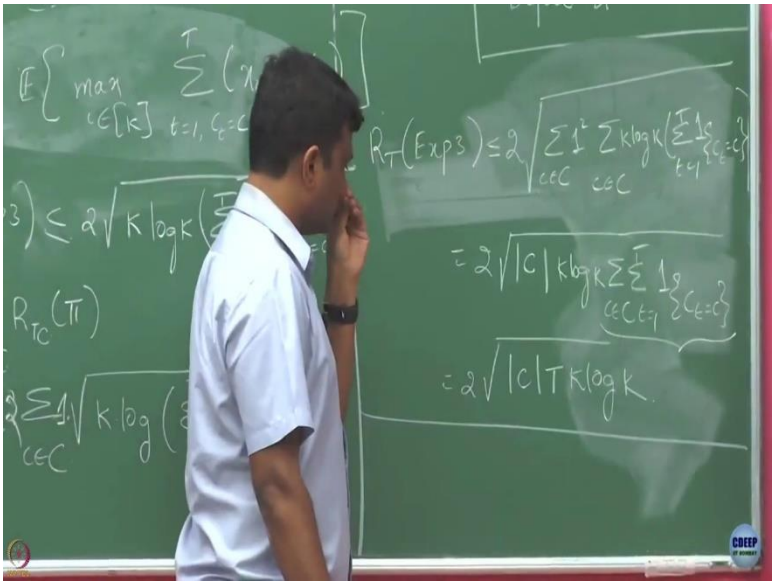
Bandit Algorithm (Online Machine Learning)
Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture – 51
Adversarial Contextual Bandits II

So, what we are next going to see is, is it possible to instead of focusing all the context can we do some restriction on this context or like group these context, and then we can see whether we can get a better regret. So, what could be the possibilities?

(Refer Slide Time: 00:46)



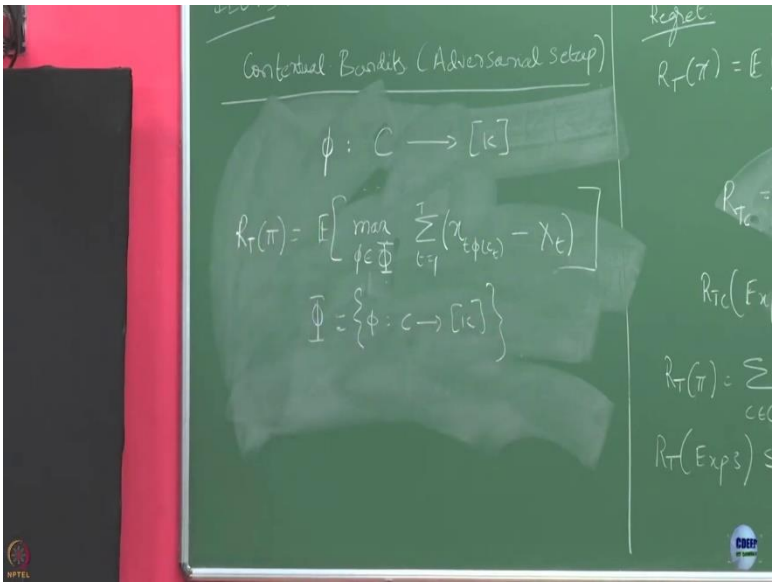
So, naturally we are seeing that if this context is very large, right. I mean just applying, maintaining a separate EXP3 algorithm or one algorithm for each context is a very costly affair, ok. So, we have to think a bit more here, ok. So, for that let us look for some special again, like the way we did it in linear bandits where we assume that by rewards mean rewards are linear, right. So, let us look at for some special cases where we will be able to bring down this dependency on the size of cardinality, cardinality of my contact set.

So, one possibility is; so, what learner is doing in all the setup is, given a context he is trying to identify which is the best arm for it, right. So, in this in the standard bandit setup where we ignore totally contextual information. What we did? We always try to look for a single best arm, right. We always try to see which is the single best arm and that was our benchmark. Against that benchmark we compared our algorithm.

But now we are saying this. See, I it is not always the case that there is always going to be a single best arm. The single the best arm is context dependent. We have discussed this, right last time. So, if you want to recommend a movie to the people who log into your recommendation system it is not that there is a single best movie that everybody likes, right. It depends on who is that person is and what is the his profile or I mean when I say profile is things like his age, his past activities and all these things.

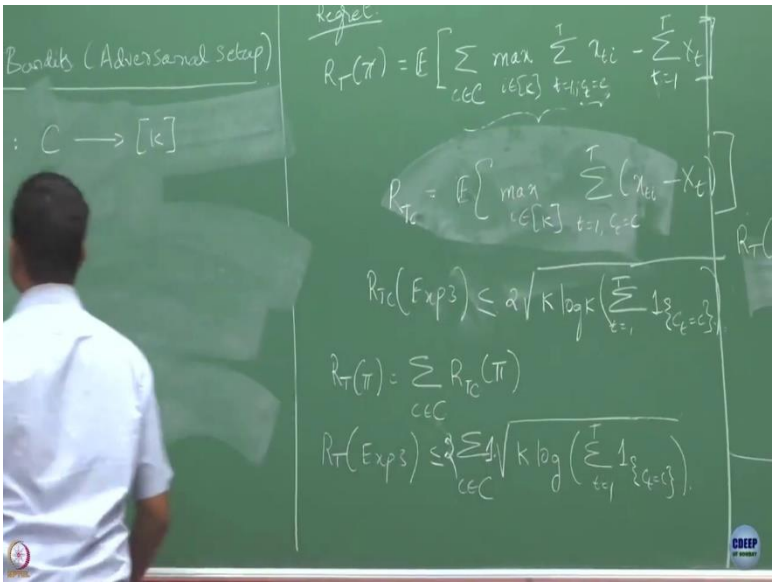
So, now what we are looking is find for every context the best arm could be different. Now, instead of what we are now looking is we are trying to do is we are trying to learn a function in a way which gives given context what is the best arm to play. Earlier we are just interested in one arm finding one arm, now for every context we want to find the best arm.

(Refer Slide Time: 03:36)



So, the problem bonds down to looking for a function which is given my context it want to tell me what is the arm to play. My search is basically over such functions, right. For every given context, I want to basically identify which is the best action.

(Refer Slide Time: 04:04)



So, I could rewrite this regret in a form. And what is ϕ here? This ϕ is collection of all ϕs where which maps C to k.

So, can I think of this regret more generally like this? I am now interested in finding a function which maps each context to particular arm. So, what is this saying? If you are going to, ok, you take a function ϕ if you are going to use that function ϕ; that means, you are going to basically select the arm ϕ_{C_t} in round t. What is C_t here? C_t is the context that appears in round t and this is the total reward you would have got if you are going to use the function ϕ.

And now what you are interested is you are trying to get a best ϕ, that means, the best reward you would have got over time period t and that is your benchmark. And now you are going to compare that against what is the total reward you would have got yourself using a policy π your policy π, right.

Student: Sir (Refer Time: 06:15) all function (Refer Time: 06:16).

Yeah, all functions.

What is basically learner is doing? Learner is basically mapping a context to an arm. We are now just looking what is the best you would have got, what is the best map that maximizes this regret, maximizes the reward. So, you are a learner for every context I give you have to decide which is the arm to play and obviously, you want to like to play and

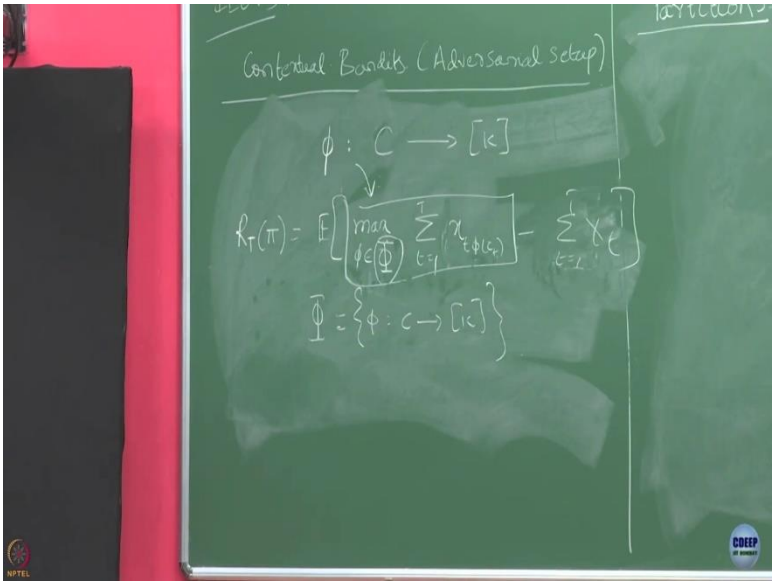
you like to map it in such a way that the reward you are going to get over time period t that is going to be maximized.

Student: Sir.

Yeah.

Student: So if that minus X_t was not inside the max if it was because outside then that if it a (Refer Time: 07:20).

(Refer Slide Time: 07:26)



Yeah, yeah. So, in fact, this does not depend on this ϕ, right. You could as well write it as outside.

Student: So these both are the same now.

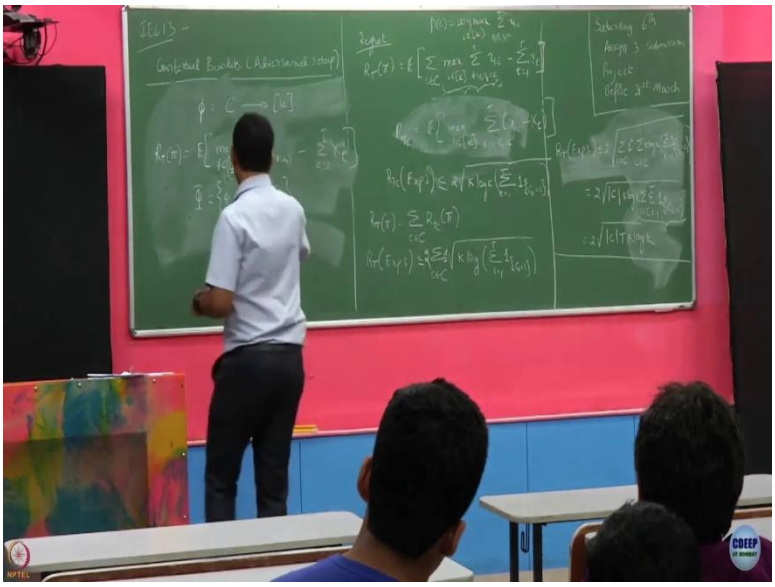
The both are the same because this is just a generalized version of this. Here you are looking at yeah; I mean, right now I have let us say this is all possible functions, ok, all possible maps; yes, in this case this should our identical.

Student: Ok. If you do their particular (Refer Time: 07:51).

They may not be. But if I allow all possible ϕ, then I could this is exactly saying what is ϕ of x here? Φ x, x is nothing but exactly arg max of this.

Student: Ok.

(Refer Slide Time: 08:11)



You are allowing every possible maps. What in that you are looking for such a maps which maximizes this, and I can always define such a map, right. So, this is provided, you know already all these x_{ti}'s in every round, that is this is the thing in hindsight, what you would have got in hindsight, ok. So, again here when I am doing this maximization, I am doing this maximization assuming that I know all the reward assignment that has happened in every round and to each of the arms. Ok.

And naturally, as you see here posing yourself. So, what is this? You are competing. This is what you would have got. And you are trying to compete yourself against the benchmark which is trying to get the best possible thing and it is going to get the best possible thing by knowing all the information. And you are trying to compare yourself against this by knowing none of this information, right.

So, you are doing, I mean in the sense that you are only learning it and trying to do, but this guy is trying to get this value by knowing all the values, right. So, this is the, so this is the oracle here. The oracle here is knowing all the values that has been assigned in to each arms in each round, and then it is trying to see what is the best you could have got.

Now, you competing against such an oracle is may be very you are asking for too much, right like because this guy the benchmark you are setting is such that it knew all the

information. If you want to get a similar performance like this case here, you what you can you could guess guarantees this and this regret bound is reasonable if this ϕ is small sorry this cardinality of c is small. If this is small; this is large then you may not be able to guarantee a good regret performance. So, for that what we will do is we are going to weaken this benchmark, ok.

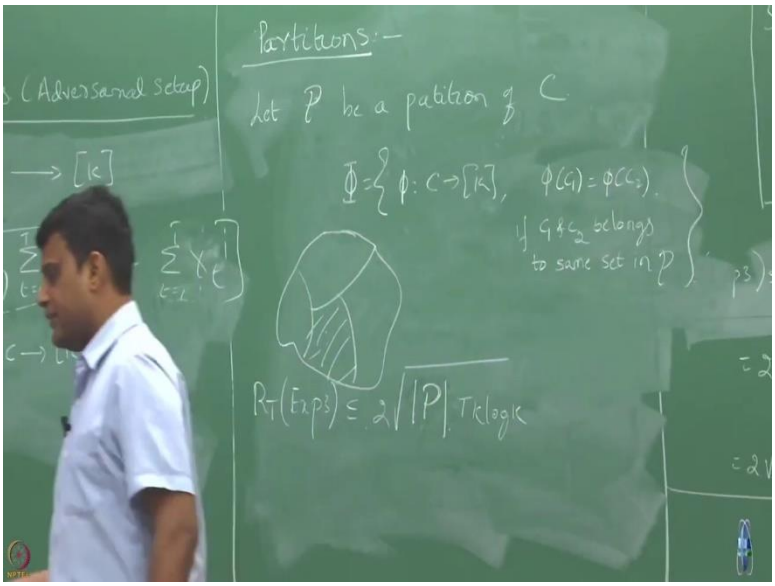
So, it this is always the case like when we are going to do such kind of performance, right. There are two things here. What is the benchmark you are going to set? If you are going to set yourself a weak benchmark, yes your actual algorithm may come as close to that benchmark and in that case your regret may be 0, but the actual reward you got maybe poor, ok. But you may try to set a very high benchmark, like as we have done here. This is like a this is what the oracle would you have got and you are trying to get try to get something as close as oracle. So, you are making yours your life very difficult, right. It is very difficult to achieve that.

So, it is always important like whenever you are going to set up such a things what is the benchmark you want to compare against. You should not set your benchmark either to be too easy, so that it is much easier for you or you do not want to set it too difficult, so that you are never going to achieve.

So, for example, in our class we do not want everybody to score, simple, give simple question and everybody score 100. And actually it would have happened that none of you knew anything, but you should got 100 because this questions were too simple. Or otherwise like we just set too hard questions, so that none of you got high scores I just made your life difficult, right. You may maybe you have learnt, but it just that you could not score too well, because the benchmark was too difficult.

So, what is that benchmark we are going to set? So, for that now we are going to see few possibilities and then we focus on a particular benchmark.

(Refer Slide Time: 13:38)



So, couple of things we are going to do is, these are some potential things we are going to talk about partitions. So, in this we are going to assume that yes we have many possible context, but some context are very identical. That means, if you know something on a particular context you kind of already know the same thing happens on another context which you believe is going to be the same as this context, ok. So, let us try to formalize that.

Let, so P be a partition of C. So, understand what I have mean by partition, right. So, let us take P to be a partition of C. And then we are going to look for this ϕ and then I am going to further enforce a constraint that $\phi(c_1) = \phi(c_2)$, if c_1 and c_2 belongs to same set in P. You understand this?

So, what you are doing here? Let us say this is my context sketch. I just assuming take one partition, let us say this is one partition. If let us say if these there are two contexts from this they will have the same value. Everybody in this region we are going to have the same value. That means, if I know something about one context in this region I already know everything about the other context in that region. So, in this case they are need to maintain one EXP3 algorithm for each context.

Student: (Refer Time: 16:29).

So, then I need to maintain one EXP3 for each partition, ok. So, I can just maintain one for each of this and in that case how does this bound is going to change.

Student: Number of elements.

So, it is going to be simply number of elements in the partition, right. So, in that case if I am going to with if I I am going to restrict myself to such a ϕ, then my if I am going to use EXP3 this is going to guarantee that 2 times instead of this I am going to write it as $2\sqrt{|P|Tk\log k}$.

So, it is the number of sets in this cell, ok. So, that is fine. So, and depending on how big how many elements are there in this partition accordingly you are going to get the regret bound, ok. So, for example, if you are you this, you are going to make it like a single partition this guy is going to be 1. If you have 2 sets in this this guy is going to be 2, ok. So, it often happens that.

If we have the prior information that the set; so, we have basically weakening the benchmark, right. So, we are weakening we were saying that we are looking for such a ϕs where such a thing already is happening. So, when I am going to (Refer Time: 19:04). So, here there is no constraint on ϕ. So, this is like the hardest benchmark I could get. But when I put such a constraint on this ϕ it is like a which less competitive benchmark for me.

So, can you think of other possibilities? What kind of other how can I further weaken my benchmark? And I want to weaken my basically benchmark and I do not want to be like a too simple one. Like I do not want like very tough benchmark which hard to crack and trivial one which anybody will crack, so what could be other possible good benchmarks, ok?

So, in all this recommendation system the way; so, you might have all this seen this movie recommendation systems, right. How do they recommend you? Even though you might have logged in into the recommendation system for the first time they will still able to show something to you. How they would have done it?

Student: (Refer Time: 20:24).

Yeah.

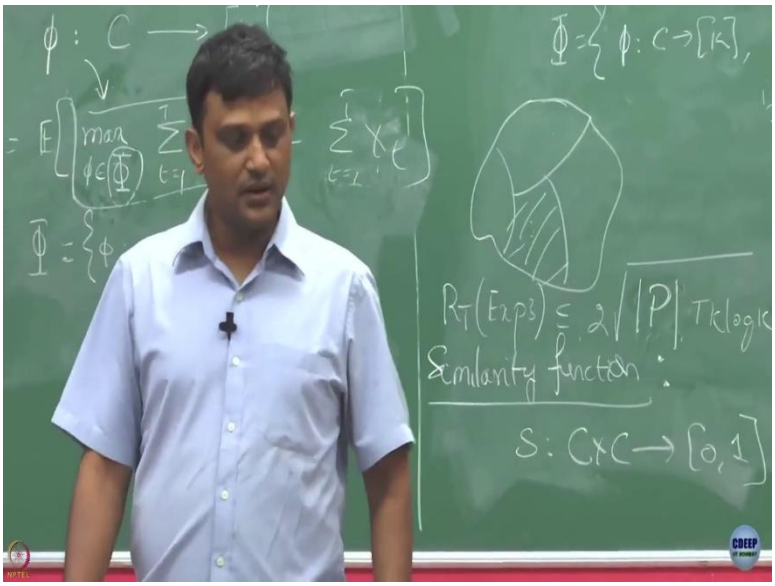
Student: (Refer Time: 20:26) it is like a delta (Refer Time: 20:29).

Yeah. So, they might have seen that, ok. So, your age is xyz, you come from this geographical area and you have done this that is your profile. But before you are not the only one who are going to entering their system, right. Many many people have would have also logged into there to see watch movies or download movies. Other people who have done it they might have already rated them; because every time you would watch a something they will ask you to rate, right, ok. So, that guy whatever he has liked that may provide some information about what you like. Is there a possible how can this this system can derive this information?

Student: These from metric can be differ between the context of the profile of the individual.

Right. So, some similarity score can be defined between the context, right. Like suppose if this is the feature and this guy has this given these rankings to these, now this guys is feature. And, I will going to compare this guy's feature with all the possible feature vectors I have that is profiles. And I see that where this this guy's feature matches closely with already the things I have in my database. If it matches maybe I will feel that, this profile looks very similar to this profile and maybe this guy will also like the similar things that the guy already with the similar profile liked and then it may show you. So, in that way we can come up with a similarity score here and based on that try to further restrict my ϕ function here, ok.

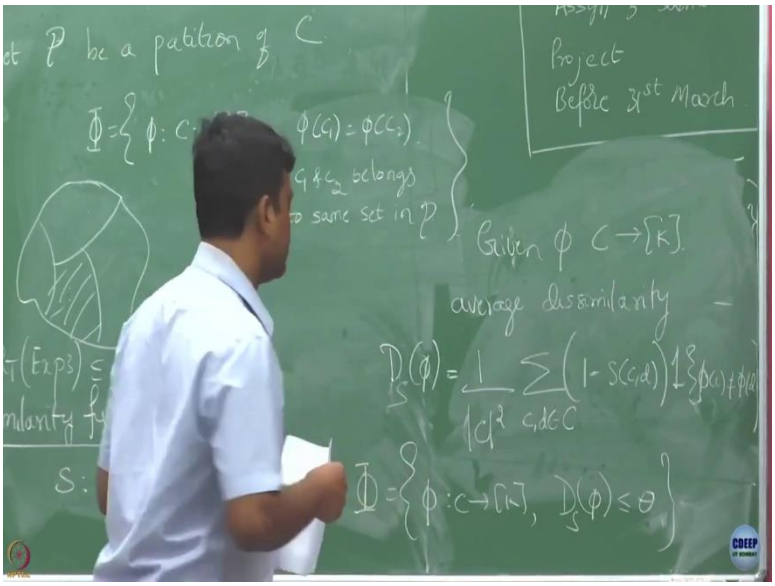
(Refer Slide Time: 22:26)



So, based on similarity function. So, for this obviously, we need to define a similarity function. We are going to define it as simply. So, let us say there is some function you already come up with given two context, it will say give you a number between 0, 1. So, this similarity score could be 1, if the two profiles matched very much, and it is 0, if they are not matching, ok.

So, in this case you can think of only those ϕ functions for which the average similarity score is or average dissimilarity score is less than some threshold, ok. So, let me formalize this. So, what is the average? So, let us say I want to now define for a given ϕ function, what is the average dissimilarity score, ok.

(Refer Slide Time: 23:51)



So, given ϕ, I am going to define average dissimilarity as $\frac{1}{|C|^2} \sum_{c,d \in C} (1 - S(c,d)) 1\{\phi(c) \neq \phi(d)\}$. So, can you all read this? So, what did is doing is, so, it is going to take a pair of context and on those points which your function c do not agree it is going to compute what is the dissimilarity score on those two and it is going to sum it your all possible pairs and normalize. So, if my ϕ is agreeing on some c d, I do not need to worry about that. I will worry about all the points where my ϕ function does not agree and now I will see that how much is the dissimilarity. So, I have define S as a similarity 1, so 1 - S is the dissimilarity score here, ok. And now, yeah.

I am not worried about partition here, like I am taking all of them. So, this is this this is for the previous part. Now, I am just defining my similarity function over all pairs. And now I am only looking for the dissimilarity scores where they do not. It could be, right. Like you just take a ϕ that is given to you. So, there are so many ϕs, right, so many assignments can be possible. You take a ϕ and you form some pair it is and it is not necessary that if it is true for every or it may possible that take any two pairs, the ϕ could be assigning different values or it could be possible that some pairs it could be assigning the same value. It is just worrying about the cases where that ϕ is not assigning the same value to that pair and now taking the dissimilarity score over all possibilities. Now, you want to make sure that this dissimilarity score is not too high. Yes.

So, this is define for a you take ϕ and S is already you have chosen that is whatever favourite you have. You took it and for that you are going to define. So, what I want to do is the way I set my ϕ here like that, so let me call this quantity as another thing, so dissimilarity score, dissimilarity of ϕ, ok. And also let me write it as S, because this depends on this function S also, ok. So, now, what I will do is here I am going to look for all ϕ such that my $D_S(\phi)$ is say less than some θ. So, I am now basically not looking considering to optimize this function or are all ϕ, but only if those ϕ's, where the dissimilarity is not too much.

Student: How do you want to inbuilt similar thing.

So, we are not trying to reduce this, right. We have defined it in this way and only looking for those ϕs who do not have too much of dissimilarity.

Student: (Refer Time: 29:17).

So, this is as I said if you just let this θ to be like arbitrarily large; that means, it will incorporate all possible ϕ's and it will be like same as this ϕ. But now as I am telling you I want to like weaken this benchmark and by and I want to say that it is only going to look to optimize over those ϕ's, which will try to group this context based on that similarity scores.

It is like; so, you are this is up to you, right. Like see you are going to either search your all possible ϕ's or you can search over only those ϕ's where the dissimilarity is not too high, ok. So, other way of thinking it is let us say like in the recommendation system we are going to have, right. You can their algorithm the recommendation system, basically when it is going to search and give the recommendations for you, right, it will have certain set of policies like this, ok.

Now, if you have to find the best one or all this it may be too much big task for it, so that is why I am going to restrict only those ϕ's assuming that this already captures a big chunk of all ϕ's. So, there will not be, so whatever the profiles I am going to create among the users, right for every users I am assuming that the profiles of users like will not vary drastically, ok, fine. Let us take some people are visiting a recommendation system, right.

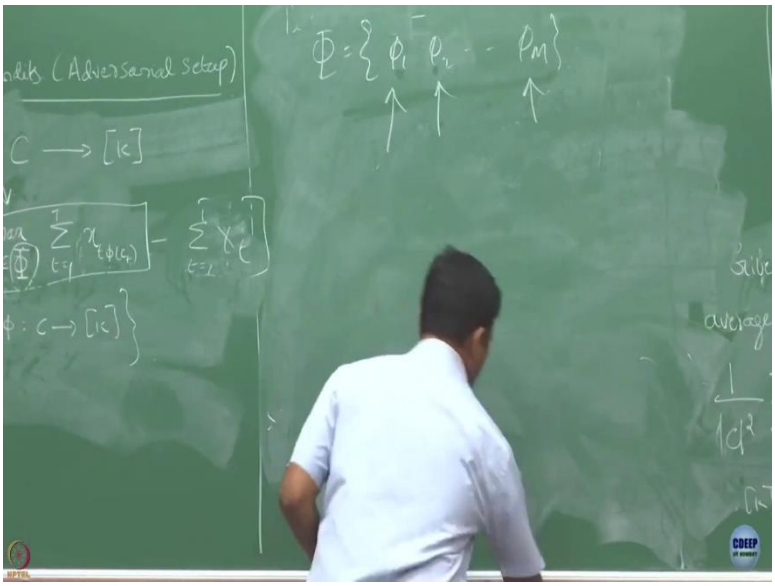
So, when I am going to design this θ I am going to assuming that at least they will have something in common because of that their dissimilarity will not be too much and that is why I am going to restrict my search phase to smaller ϕ, ok. And if I further know that all these guys are coming from IIT, Bombay, I will still set this θ to be smaller because all these guys are, young guys who is going to have a similar interest, right. So, this is based on my prior information.

And if I know that; so, all this like, suppose if it is a sports site it already knows that like most of the guys who are coming to it are like young guy, so some of the features are going to be common there, right. So, then they will have already some similarities based on that you can restrict, ok, fine.

So, this is just another possibility, right and indeed these are actually used like depending on your prior information what kind of customers you are going to get. We are going to because you do not want to arbitrarily search over a big space here, ok.

The other possibility is; what could be other possibility? Yeah, other possibility is as simple as this. Like, like how many hour are there in this I do not care about them, but I am going to only restrict myself to few of them. So, you are going to further have some kind of a priori information. All this mappings are not possible, only few mappings are possible. We are going to initially short list them, but you do not know among them which is the best, you want to still figure out, but at least initially itself you will narrow down your search and by restricting yourself to certain number of ϕ functions. Should that be fine or that is a very bad thing to do?

(Refer Slide Time: 33:48)



So, I am just going to take this $\phi = \{\phi_1, \phi_2, \ldots, \phi_M\}$. Some M set of functions and I want to just to find which one is best among them, right.

I mean this is most of the case, right. Like you do not want to deal with, yeah that is always the case we will not always have infinitely many options, right. We have few options and from that we have to pick the best one. And in that sense we can think of the number of possible mappings are finite, which you have you would have narrow down like which; like which you are going to select among ϕ? I do not know. That comes from your expertise for that particular application.

Like for example, in the recommendation system you may be already knowing that like; so, you maybe you may already knowing that if you just go to a some e-commerce site and if you already know if you are a young guy I already know that it is unlikely that you are going to look for some insurance policy, right. So, that mapping that given a young guy looking for an insurance policy that option I already rule out and that will already eliminate certain number of policies. So, based on that I will rule out certain policies and narrow down to certain number of policies and then I am going to search which one is the best among them, ok.

So, now, fine. I have narrow down. Is it any simpler? Can you think of what is a good way to search on this?

It is not still necessary that these two function should assign the same value to but any particular context. It is not necessary, right. They could be all they assigning different.

Student: From the original, from the original site you have narrow will down so.

Yeah, how we did? Right now we are not telling you. I am just saying somebody did it for you. We will always have some sales team, some big PR teams and all these things, right. Like they will narrow down, but they will already like they will say rule out, ok, so this is not making sense rule it out like. Then, what is possible then you are going to narrow down and from that available options then you are going to optimize, what is the best you could do. On that you are not going to make any assumption.

Student: That is the same he next will be applied (Refer Time: 36:38).

Is it? So, what is this ϕ function? Φ function is still doing with dealing with different context, right. Φ is again has to see if it is for which context which arm you are going to select? Ok. So, now. So, then what is a good policy here? So, one possibility is like each is like a bandit problem in its own, right. Like if you just take a policy ϕ, ok. So, what is this ϕ is basically? We want to identify the best arm, ok. So, not this, ok, sorry, ok. So, let us go back to. So, do you recall our weighted majority setting?

Student: Yeah.

What did we do there?

student: sir we gave equal weights to all the arm (Refer Time: 38:10)

Yeah. So, what we would say said that set up as? What we said that set up as?

Student: (Refer Time: 38:22).

Yeah. We said that as prediction with expert advice, right. So, prediction with expert advice. What we said there? The expert is going to, so there expert is going to tell us which arm to play. And what we did there? We go basically assigned a way to each of the expert and based on that we are going to pick an expert and we are going to observe the loss or reward accordingly.

Now, can we think of this ϕ functions as different experts and I want to identify the best one. But what is the difference here?

Student: Not a fully (Refer Time: 39:17).

Yeah when, ok, fine. Let us think fully information, ok. Let us compare EXP3 only. EXP3 we had only bandit information, right. We only observe the expert we played there. Then let us say just simple EXP3 here. So, EXP3 what we did? We actually maintained a distribution over the arms, right. In EXP3 we actually maintained a distribution on arms based on the cumulative loss or reward we have observed so far. And then try to play the one which has like according to some distribution which we updated in every round.

Student: (Refer Time: 40:12).

So, now is it worthwhile to maintain the distribution over these experts?

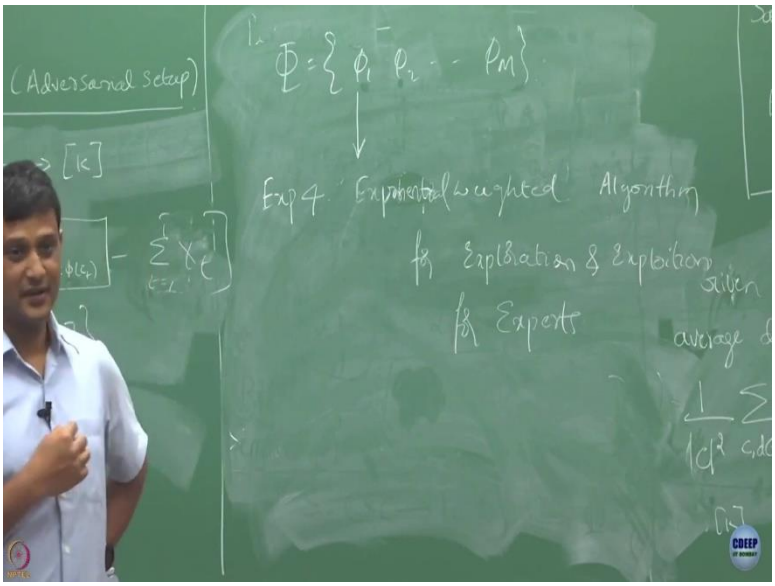
Student: How do you?

Yeah.

Student: How do you update the?

Yeah, how do you update we will come to that later that is an algorithm, and in terms of the broad structure of the policy if I want to just, ok. Now, we have this narrow down to this. Now, I want you to just think about an algorithm, at least the broad steps. So, is it like ok, like if you can now think these as different experts on which I want to maintain? So, what these guys can do? Each guy is going to basically tell this is a mapping, right, for each context what should be the arm. These guys is going to tell, ok, you use this this this arm.

(Refer Slide Time: 40:53)



Now, I want to whatever that guy tells me to do in each round, based on that I can maintain a distribution over him. And if I pick this particular expert then whatever that guy tells me to play in that round I will play that, and I observe the reward, and based on that if I get a good thing or bad thing I am going to update my weight on that expert and I will continue this, ok. So, based on this idea we are going to see an algorithm called exp 4 next time. We have seen EXP3, right. So, what is EXP4? So, what is EXP3?

Student: Exponential weight algorithm for exploration and exploitation.

EXP4 is Exponential weight algorithm for exploration and exploitation for experts.So, this is the exp 4 algorithm which will study in the next class. We will do a version of that a b testing only, right, what EXP3 and we are all doing there. We are just playing them and getting information about how they are doing and then we are updating about the weights. Just like we have more experts here and we want to identify an expert, ok.