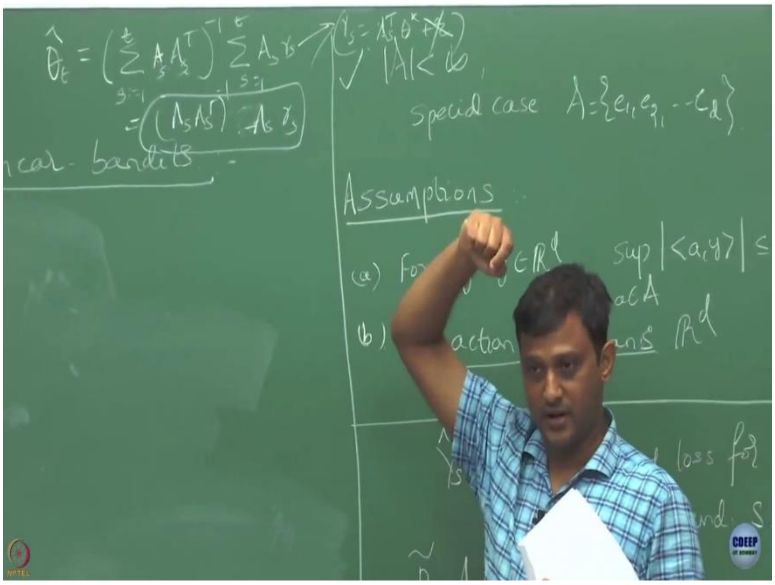
Bandit Algorithm (Online Machine Learning)
Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture – 55
Exp3 for Adversarial
Linear Bandits

(Refer Slide Time: 00:19)



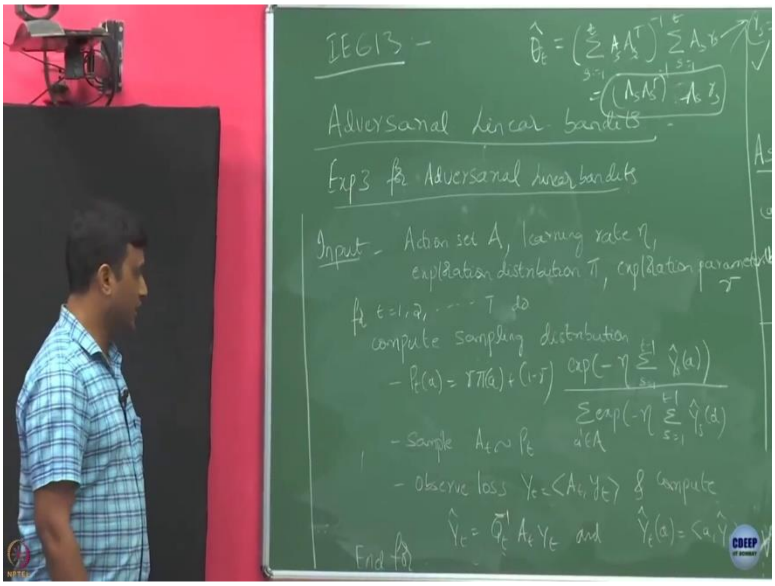
So, by the way; what was r$_s$ in my stochastic case? So, how was that? It was like $A_s^T\theta^* + \epsilon$ right in a stochastic case that was the model. We are going to observe a noisy reward in every round.

The noise we assumed some sub Gaussian noise, but what remained fixed throughout is; this θ^* because of this my rewards are all correlated across. But here when I come to adversarial setting that need not be the case what we allowed is; so, in adversarial setting you have actually get rid of this noise yes there is no noise.

But this θ^* could be adversarial selected by the environment, it is under environments control and unknown also it could be in adversarial it could be changing.

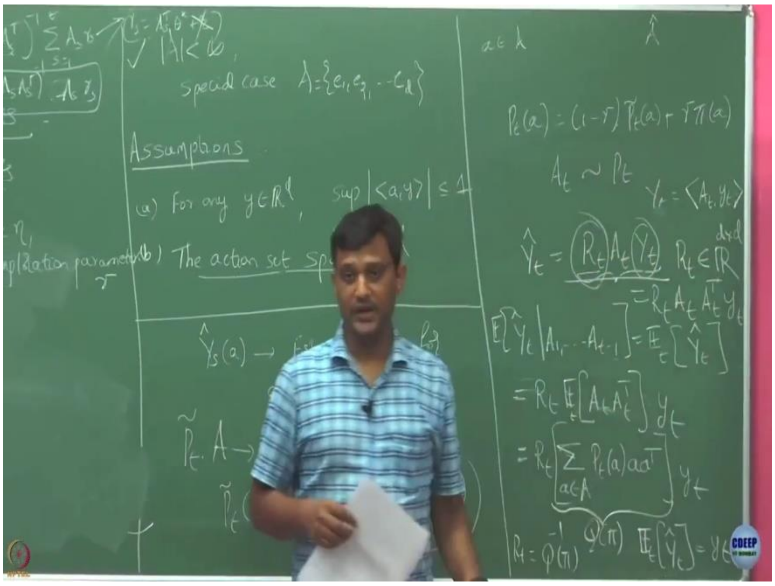
In the stochastic case this one also θ^* ar also selected by environment, but it held fixed throughout. What we are observing only the noisy versions of the rewards ok.

(Refer Slide Time: 01:37)


IE613 -
Adversarial Linear bandits
Exp3 for Adversarial linear bandits
Input - Action set A, learning rate η, exploration distribution π, exploration parameter γ
for t = 1, 2, ..., T do
compute sampling distribution
- P_t(a) = γπ(a) + (1-γ) exp(-η Σ Ŷ_s(a)) / Σ exp(-η Σ Ŷ_s(a))
- Sample A_t ∼ P_t
- Observe loss Y_t = ⟨A_t, y_t⟩ & compute
Ŷ_t = Q_t^{-1} A_t Y_t and Ŷ_t(a) = ⟨a, Ŷ_t⟩
End for

Now, let see how to adopt EXP3 for adversarial case for ok. So, the algorithm goes as following oops. So, input are action set A, learning rate η, exploration distribution π and then exploration parameter γ .

(Refer Slide Time: 03:00)


|A| < ∞
Special case A = {e_1, e_2, ..., e_d}
Assumptions
(a) For any y ∈ ℝ^d, sup |⟨a, y⟩| ≤ 1
(b) The action set sp...
P_t(a) = (1-γ) P̃_t(a) + γπ(a)
A_t ∼ P_t
Y_t = ⟨A_t, y_t⟩
Ŷ_t = R_t A_t Y_t   R_t ∈ ℝ^{d×d}
= R_t A_t A_t^T y_t
E[Ŷ_t | A_1, ..., A_{t-1}] = E_t[Ŷ_t]
= R_t E_t[A_t A_t^T] y_t
= R_t [Σ_{a∈A} P_t(a) a a^T] y_t
R_t = Q(π)^{-1}   E_t[Ŷ_t] = y_t

So, γ and π where we are used here; when we try to construct these exploration distributions ok. Now, you do the following for t = 1.

So, when I wrote here I just say like this is it does not depend on time right; it is like one fixed which I am using throughout.

So, this algorithm takes the actions at and the learning rate η we will see how to set this learning rate η and then the exploration distribution and the exploration parameter γ. So, in each round it is going to have this distribution defined for each the probability distribution, probability defined for each of this action in this fashion which is nothing, but the linear combination of this exploration distribution plus, what are the exploration the distribution we have through this estimates ok.

So, notice this I did not explicitly mention the case what happens initially. Like because initially I will not have these terms right when I start with t = 1 round; this set is this summation is empty. So, we will just assume they are all uniform in the first round the way you usually do ok; for t = 0 round this quantity is nothing, but 1/|A| that is the uniform distribution. Then subsequently we are going to sample an arm A_t occurred for from the distribution P_t; you play this action A_t you are going to observe a loss for that action A_t.

Once you will observe the loss we are going to compute this $\hat{Y}_t$, the loss vector you are going to compute. So, sorry $\hat{Y}_t$ is what this is the estimation for the loss.

Student: (Refer Time: 05:21).

I mean the vector that the adversary or the environment would have selected. Once you estimate it in this fashion we are just discuss that this is going to be an unbiased estimate are of that vector Y_t.

Now, you go back and see what happens what is the loss you would have incurred for each of the possible actions. So, notice that for this particular A_t you have already know that this is the loss you are observed, but you are not going to take that loss for action A_t. Whatever that this is going to give for the estimated value of Y_t that is what you are going to take; like the way we did it in EXP3 and then you are going to repeat this process..

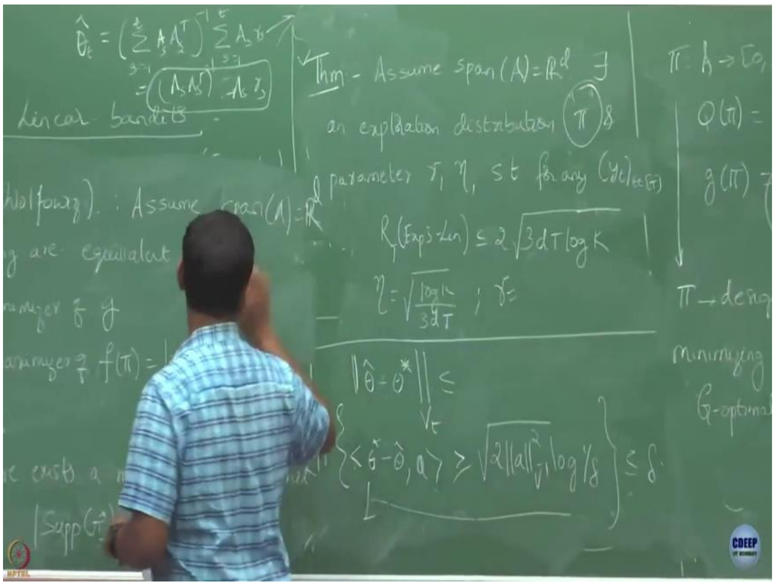
We are going use it in the statement of the regret for this algorithm is generic it works for anything right. Like I mean even if it is this algorithm as of now only if you can compute this for A even if my capital A is uncountably many that is fine this algorithm as of now. So, where is the issue in this algorithm if a happens to be uncountably many terms actions?

Student: (Refer-Time: 06:52) from the.

This I cannot define properly right because this sum could be summation over infinitely many terms. So, we will look into that aspect how to handle the case where a is uncountable or it could be a continuous set, but as long as my a is finite this is fine right everything works here. And now let us see and coming to this exploration distribution that is given to me π and the same one I am going to use it in every round ok.

Now, the question is; how to choose this exploration distribution? Right. Naturally if we are going to change the exploration distribution for that may be the performance of this algorithm may change ok. So, now, first we are going to say that; there exist some good exploration distribution that will help us give a sub linear regret. And then we will see how whether indeed such an exploration distribution exist and if at all how to get it.

(Refer Slide Time: 08:14)


Thm - Assume span(A) = $\mathbb{R}^d$ ∃
an exploration distribution π &
parameter γ, η, s.t. for any $(y_t)_{t\in[T]}$
R_T(Exp3-Lin) ≤ 2√(3dT log K)
η = √(log K / 3dT) ; γ =
Linear bandits
$\|\hat\theta - \theta^*\|_{V_t} \le$
CDEEP

So, this is the statement I am going to say there exist an exploration. So, just I do not know there is the name no name is do not say I am just going to call this EXP3-Lin this is just like our notation. We are going to say that regret of EXP3-Lin is upper bounded by $R_T(EXP3 - Lin) \le 2\sqrt{3dT \log K}$. We can specifically set $\eta = \sqrt{\frac{\log k}{3dT}}$.

So, what this saying that see the now this is the regret this theorem requires that to state the it requires that your action space spans your R_d and it says that there exist an exploration distribution π. And if you are going to set η to be like this and γ have not specify how we will see then the regret that your algorithm EXP3 Lin is going to achieve is going to be of this form. It is going to be what? This is going to be sub linear in T and it looks very similar to what we had for EXP3 algorithm, but what is the difference now?

What is that? In just the regret bound.

So, the d coming into picture right like the dimension. Earlier it was just like square root of T.

So, it is not that number of arms actually mattering us, but the number of dimensions that we have to figure out because, now we have linearized the rewards. So, now, what matters is in what dimension the unknown parameter lies. So, once I know that how many are number of arms I can figure out the reward for everybody right. So, as long as I can find out those d dimension the d parameters I have knowledge of all the arms.

Now, how is this distribution? π right while we just said is there exist a π such that this holds. So, for that it is bit involved we are going to just again this comes from some other result which guarantees existence of such a π ok. So, let us briefly discuss that part.

So, I am not going to the proof of this part you can just look into the book. This is again most of the time it is going to be very similar to what means they we have done it in EXP3, but of course, with little jugglery of the estimations the new kind of estimations we have bought into picture here.

So, to understand this existence of how does this π looks likes. So, we had to look into some design of experiments. Have you any of you gone through in any of the courses you have taken the design of experiments is covered? Ok. So, design of experiments could be like as simple as; if you want to estimate some parameter with high confidence and every time you are going to play an action let us say the reward is going to be linear in this unknown parameter.

So, for time being assume that the reward is going to be $X^T\theta^*$. If you are going to choose x the reward the thing you are going to observe is $X^T\theta^*$, but plus noise added. So, we already noticed that when we had; this is exactly the set up of linear bandits right we had. Now, there what we wanted we wanted to ensure that how to quickly get a good bound on this.

So, what would we say for this? We say this we have v_t here and what would you say we had. So, we said this upper bound by some quantity with high probability right what would we say.

So, we actually said that before this we said if I am going to take θ* and $\hat{\theta}$ and some and some arm a. So, we did this argument right like we did not exactly show this under very generality, but under some assumptions we did show such a thing is possible right the probability that the projection of this error on a particular arm is upper bounded is it.

Yeah this being larger than some number is going to be very small a probability δ. So, now; so we want to achieve such a thing; suppose we you want to achieve such a thing that I keep on observing my rewards by playing a particular action that reward is going to be simply let us say $A_S^T\theta^* + \epsilon$ I am going to get. Now, the question how should I be choosing a sequence of as I am going to play such that as quickly as possible this is achieved?

So; that means, I have been able to estimate my $\hat{\theta}$ good very fast right. So, what is this v inverse here?

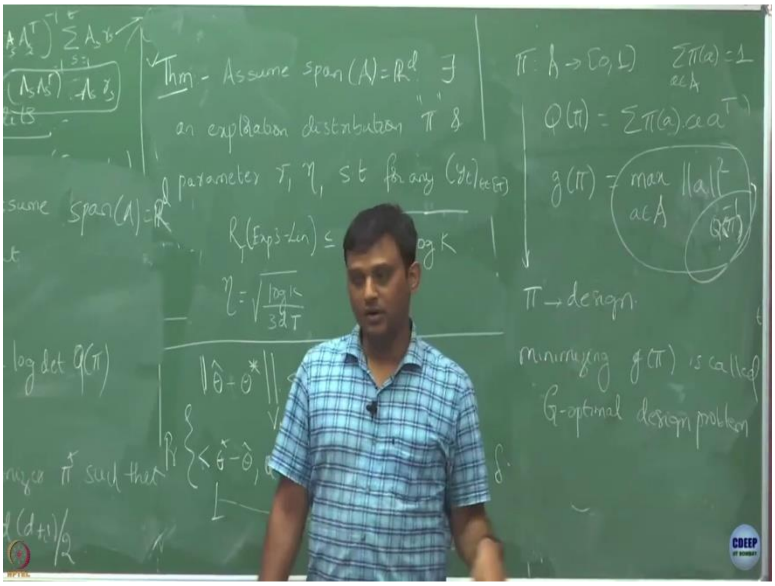
Student: (Refer Time: 15:27) data we have.

This is the depends on the data we have been gathering. How should I be using my data to make the observation such that my estimation error quickly falls down ok. So, this is then just this is the question about how should be I designing my experiments so that I quickly able to estimate my under lighting parameter well right. Now, if you just going to randomly select in every round some s may be that is not a good idea. What you want to do is; you want to always select some actions such that the all the dimensions all the directions in this θ* are well expert.

So, if you just happened to randomly places an actions may be you may end up only exploring certain directions. And also if you just do randomly you may happen to explore all the directions, but on none of the directions you have good information. But if you are going to design your experiment may be in adaptive fashion such that as you go on you feel that some directions are not explored well, I will choose my action such that in those directions I get better information so whatever.

The point is the way you are going to select actions in each round matters and linear bandit exactly did this they try to select actions in each round such a way that you get a better your estimates improve from each step to the next step ok. So, to exactly to come up with this distribution π we are going to state a result from this design of experiments that tell that indeed such a π exist I mean some good π exists and how to compute that.

(Refer Slide Time: 17:32)


Thm - Assume span(A) = R^d ∃
an exploration distribution π &
parameter γ, η, s.t. for any (y_t)_{t∈[T]}
R(Exp3-Lin) ≤ ... log K
η = √(log K / 3dT)
π: A → [0,1)   Σ_{a∈A} π(a) = 1
Q(π) = Σ π(a) a a^T
g(π) = max_{a∈A} ||a||²_{Q(π)^{-1}}
π → design
minimizing g(π) is called
G-optimal design problem
span(A) = R^d
log det Q(π)
(d+1)/2

So, let say I have this π which is a map from my A to [0,1) and such that my $\sum_{a\in A}\pi(a) = 1$. Now, let me define this Q(π) = $\sum_{a\in A}\pi(a)\,a.a^T$. And then I am going to define $g(\pi) = \max_{a\in A}||a||^2_{Q(\pi)^{-1}}$.

So, if you now map it to our scholastic bandit problem. So, let us say this is about I want to come up with a sampling distribution π such that my confidence bounds become tight.

So, if my confidence bounds are tight can I do a curve with a better algorithm? Yes right because we already know that the confidence bounds really play important role right. If you have a tighter bonds then ah the when I select my action when I am going to order them may be like optimistically based on just the what are the estimation plus the confidence term I have. So, if my confidence terms are tight may be the probability that I make error is also smaller yeah.

Let us say I am just going to now I want to sample my actions and observe the loss in the linear setting and now I want to estimate quickly what is the confidence I have in this arms about the reward. This quantity g of π here in a way corresponds to that.

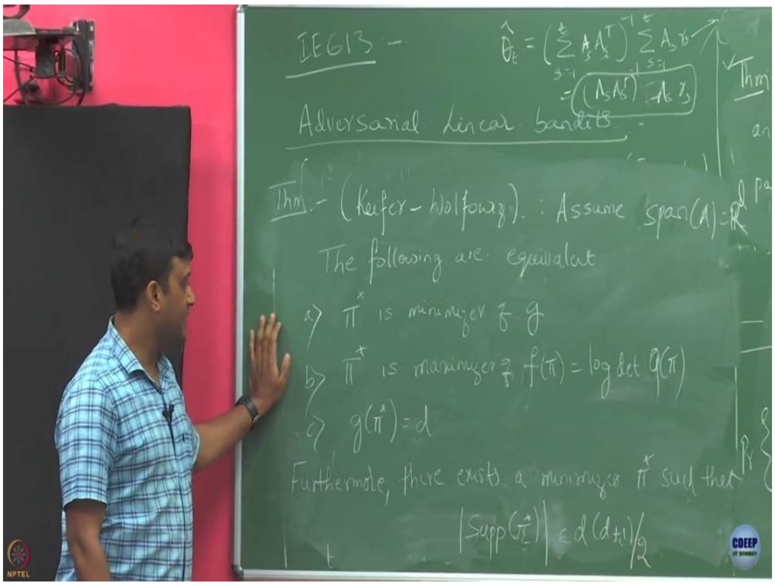
So, it tells you how much confidence you have or like how much confidence or how much amount of the information you are going to gather by playing a particular action a anyway. What you want eventually is as you keep on playing the actions from this particular sampling distribution π; you want this quantity to be get smaller and smaller.

So, what is this let us say if I am going to think this as a confidence term and this is the largest confidence among all. Now, if I want to have a good experiment like if I want to do a good selection of my sampling through this particular π I want this to be eventually fall start quickly falling down right. So, in the design of this experiments in that terminology this usually this π is called this design and then minimizing g(π) is called G-optimal design problem.

So, just think as this as a separate problem. So, you want to now come up with a you are looking for a distribution π such that it minimizes this. And what is this $Q^{-1}(\pi)$? $Q^{-1}(\pi)$ is just defined like this it is nothing but the Q(π) $= \sum_{a \in A} \pi(a)\, a. a^T$. So, this is just a problem like we will see how it connects to what we want to do. Now, the question is what is a good π that minimizes this? Right.

So, this is like as of now there is no iteration here right round one round two like I am not going like this. It is just think of like one shot I want to do I want to sample my arm such that; whatever this quantity here which I am calling the g of that sampling distribution is minimized. As I said this quantity I can interpret it as the confidence term; corresponding to that action a and I want this to be smaller. So, this is called the minimizing g(t) or G optimal design problem in the statistics or in general in the optimal experimental setup problem.

(Refer Slide Time: 23:48)



IE613 -
$\hat{\theta}_t = \left(\sum_{s=1}^{t} A_s A_s^T\right)^{-1} \sum_{s=1}^{t} A_s y_s$
Adversarial Linear bandits
Thm:- (Kiefer-Wolfowitz). Assume span(A) = $\mathbb{R}^d$
The following are equivalent
a) π^* is minimizer of g
b) π^* is maximizer of $f(\pi) = \log\det Q(\pi)$
c) $g(\pi^*) = d$
Furthermore, there exists a minimizer π^* such that
$|Supp(\pi^*)| \leq d(d+1)/2$

Now, for this we have one result called Kiefer Wolfowiz ok. So, we are going to the theorem uses the result from this theorem he says that; assume a span(A) = R_d. So, this result may be just of independent to interest to you may be you may want to use it in some other analysis also; the following are equivalent ok.

So, what this result says? It says these three statements are equivalent ok; it says that if π* is the minimizer of that g here. So, we said that we are interested in minimizing this g function right it is same as saying that at that π* the g(π*)is exactly equals to d. If this is the case it says that this π* can be also is the maximizer of this quantity here. So, this algorithm.

Student: Ok.

Actually tries to construct such a π because as I said this π here is the one which is going to minimize which is this term here which is an equivalent of a confidence term for me. And in every round as I go on from one round to another I want this confidence term to be smaller.

So, it is going to start choosing in every round it want this to be smaller and it is going to try to come up with an π such that it kinds of minimizer this term. And this is as we said this is for one round here, but we have multiple rounds. How we does that? It is going to use that using the set whatever I have a a transpose and using the knowledge of $Q^{-1}(\pi^*)$.

So, how exactly that overall π is come up; it is based on this idea. As of now I am just telling you this exploration exist, so it need to be exactly constructed. And I will leave it to you to look into the proof what is the exact π that is the algorithm is using. So, what it is? What they make sure is at every this π is such that it is trying to minimize this quantity every time ok. So, you know right now you see that the way this minimization problem is different it is only in terms of the what are the actions that I have. And now it is going to define based on the norm of this.

So, I am not sure like this whatever this is happening this is the π exactly the one this algorithm is claiming which exist, but I think it is some tweak question of this; its not necessarily whatever. So, fine if that is the case how to compute such a π? The π computation is already given here right it is the maximizer of this quantity which is easier to compute all you need to do find the determinant of that and take the log and try to see which is that π it minimizes this sorry that maximizes this.

So, we know such a π there are some good π’s which was going to give us this G-optimal design problem or in a way they are trying to try to shrink your they are going to give a tighter confidence terms. So, the exact π is going to be based on whatever the π* we are getting here based on that; exactly how it is I will just leave it to you to look into the proof ok.

Ah so, the last point what is it says is whenever such a π* exist it support is bounded by this quantity; supp stands for support. So, we understand what is support π star means?

Student: (Refer Time: 29:14).

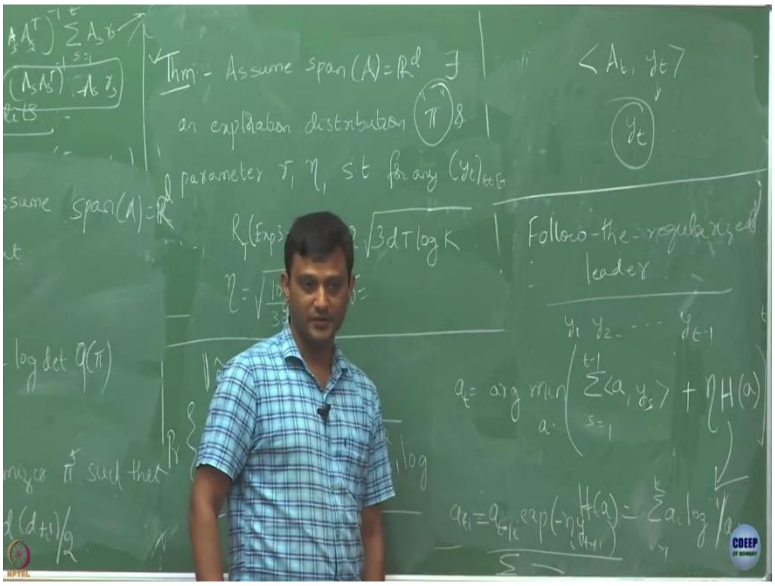
So, the number of places where it is going to put non zero values. And that is going to be at most d(d+1)/ 2. I mean this term is used only the analysis it is to upper bound this. So; that means, what? It is actually not putting mass and all the actions right as we said the number of actions could be much much larger than the dimension ok; like my dimension could be 10, but the number of actions could be 1000, but what it is saying is, but this π* is has to be defined on all possible actions right; yeah this π is defined on all possible actions it is saying that.

So, suppose if you just take d equals to 10 what are these quantities 10 into 11 by 2? 10 into, so some 55 right even though you could have 10 into 11 yeah even though you could have like 1000 actions it is only going to put some mass on some 55 actions. So, its not necessary that it has to put it will ask you to act go on exploring all the actions ok.

So, fine so it is saying that as long as you can come up with a good exploration distribution. And this is the standard tweak like right like the η parameter to be set like this and also I think you can I can look into that I do not have the exact value here what is the value for γ I think it is somewhere submerged in the proof. So, some γ I think it should be some function of η like once you have this η, γ could be set in terms of that.

So, once you set it you can get a regret bound of this form and we know it is you know we have a sub linear regret ok. So, before we conclude on this part I just want to highlight one more aspect of the studies. Suppose in this setup what if the entire y_t that has reveled to you in every round.

(Refer Slide Time: 32:05)


Thm: Assume span(A) = R^d ∃
an exploration distribution π & γ
parameter γ, η, s.t. for any (y_t)
R_T(Exp3...) ≤ ... √(3dT log K)
η = √(...)
span(A) = R^d
log det Q(π)
π such that
d(d+1)/2
< A_t, y_t >
y_t
Follow-the-regularized leader
y_1 y_2 ... y_{t-1}
q_t = arg min (Σ_{s=1}^{t-1} <a, y_s> + η H(a))
H(a) = Σ a_i log 1/a_i
q_{t+1} = q_t exp(-η ...)
CDEEP
NPTEL

So, what I said right now in every round if you are going to take action A_t what is reveal to you is $<y_t, A_t>$ from this you are going to estimate.

Suppose let us say the environment is nice to you and it reveal to you exactly y_t itself. You play whatever A_t action you wanted to play and you actually incur this much loss after this the environment actually reveal to you what is y_t; would you have been in a better position like you can come up with a better algorithm?

Student: (Refer Time: 32:39).

Yeah.

Student: Now, we want (Refer Time: 32:41).

Now, its a full information case like once you know y_t you have information for loss of all the arm all the actions you have. So, this is exactly what we started in the first class first of couple of classes right. So, this is like when a label is revealed suppose. So, if you are if you look into the classification problem we have a set of hypothesis; if a label is revealed at the end of the instance you already know how what is the loss you would have incurred by applying any of the classifiers. So, that is exactly this. So, in this case what algorithm you would have like to use if y_t is revealed at the end of each round?

Student: Weighted majority.

You would have like to use weighted majority right. In general there are both class of algorithms which we will not have time to look into that, but they called follow the regularized leader. I am not talking about this case where we have this full information case; what they do is in every round try to play an action that is the best so far. What I mean by that? So, if you y_t is revealed in every round right; let us say y_1, y_2 … y_{t-1} till round t - 1 this you have been revealed to you.

So, what you are going to do in the next round? One possibility you want to do is; you want to would you like to do this?

Student: (Refer Time: 35:01).

And play an action in round t you want to play the action that is like arg min of this. So, what is this? y such that reveal to you in every round so far, but y_t is not reveal to you in round t yet, but you have to make an action for that round; what you are going to do? You are going to see that for all the things I have observed so far which action would have given me the best possible loss. And then whatever that is you may want to play that.

This is I am regularized. But you can show that even if you just do this case there are some instances where you may be get stuck to some bad actions. But the general one could do this to avoid that you may want to make this a bit smooth, but bringing in the regularizer terms and one possibility for that is you may want to include a regularizer here and which is now the action.

Student: (Refer Time: 36:16).

The action; for each action you are going to define a function H. Now, you want to minimize this and play the action. And one can so you can make sure that, but properly choosing this regularizer you should be able to get a good performance. So, one particular choice of this regularizer which is often used is the entropy function.

So, I am right now assuming that this action sets are now probability vectors for me ok. So, in that case we already know that this is nothing, but a i log 1 by a i whatever how many are there ok.

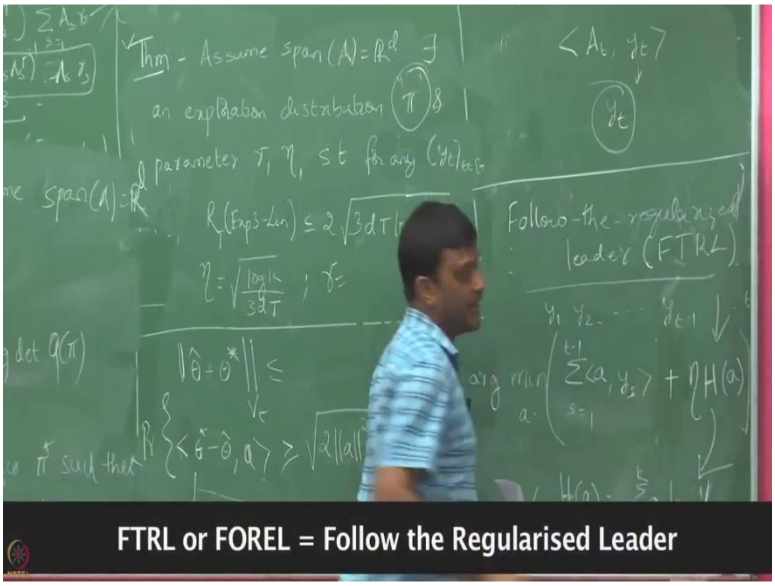
So, did you notice did you realize this formulation like where did you see this. So, now, if you take this entropy exactly like this what is that you are going to get? What is this a star is going to be like; this is a distribution right how does a star is going to look like?It is going to look like my exponentiated weights further. $a_{t_i} = \frac{a_{t-1\,i}\exp(-\eta y_{t-1\,i})}{normalization}$ component exponential what are the loss you have observed for that particular i in the previous round and divide it is just a normalization. This is just what we had observed in the weighted majority algorithm right.

So, if you have this entropy we are going to get this weighted majority correspondence. And in that we already know if once we have this kind of distribution we already know what should be the good value of η right. Like because when we use the weighted majority we started with we will start with such a distribution we use this distribution and then further optimize my regret by tuning this parameter η. So, once my distribution result in this I know what is already from my knowledge of weighted majority algorithm how should be I tuning this parameter η.

So, there are class of algorithms based on this idea this regularization entropy is just one function you could think of other functions ok. And like people have use something like a divergence I mean not the (Refer Time: 39:56) divergence, but there is another notion called Bergman divergence and all. So, by using different regularization you will get different performance ok.

So, you may also want to look into that one chapter one such regularized. So, this is called follow the regularized leader FTRL algorithm.

(Refer Slide Time: 40:17)


Thm - Assume span(A) = $\mathbb{R}^d$ ∃
an exploration distribution π &
parameter γ, η, s t for any $(y_t)_{t \in [n]}$
$R_n(\text{Exp3-Lin}) \le 2\sqrt{3dT \dots}$
$\eta = \sqrt{\frac{\log k}{3dT}}$; γ =
$\langle A_t, y_t \rangle$
Follow-the-regularized leader (FTRL)
$y_1\ y_2 \dots y_{t-1}$
$\arg\min_a \left(\sum_{s=1}^{t-1} \langle a, y_s \rangle + \eta H(a) \right)$
span(A) = $\mathbb{R}^d$

FTRL or FOREL = Follow the Regularised Leader

So, there are this itself because such FTRL or FOREL algorithm give a very good performance. So, people have been using different playing with different different regularizers and coming up with different different bounds. So, you may also just want to look it. So, I am not going to be for to define other regularizers we need to kind of a divergent to some other topics. So, we will not going to that.

So, as you see that we started like looking into specific cases, but thing can be studied in more generality. Like what we stated as weighted majority algorithm is nothing, but this regularized. So, why is called follow the leader?

Student: (Refer Time: 41:07).

Because we are trying to play the leader right till that point. Till that point whoever is the leader you just want to play it. But it is just like taking a regularized version of that.