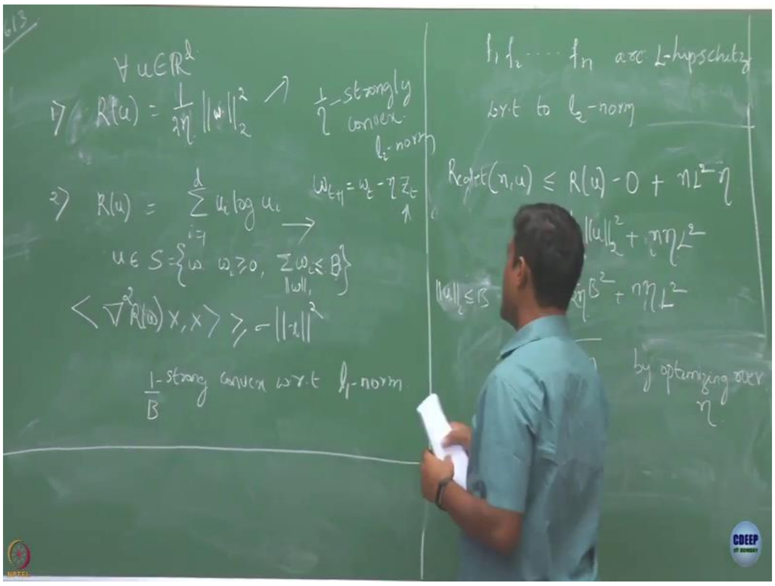
Bandit Algorithm (Online Machine Learning)
Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

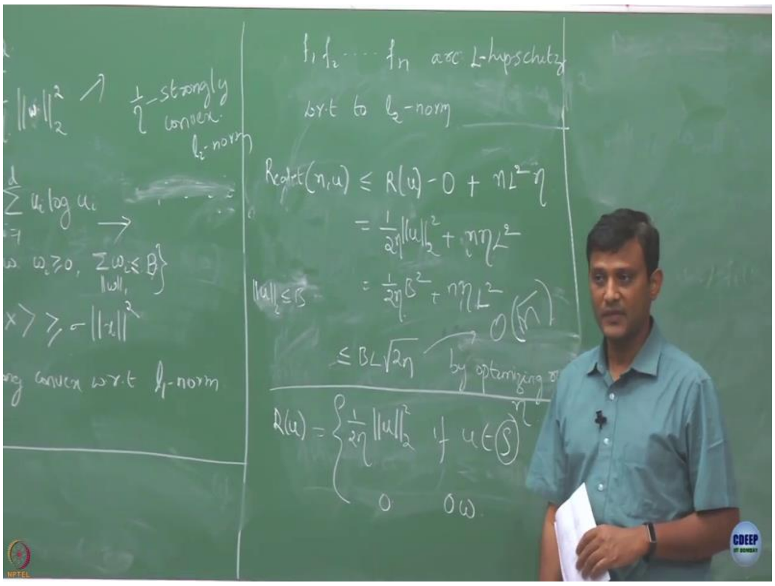
Lecture - 25
Euclidean and Entropy Regularizer

(Refer Slide Time: 00:23)



So, let us first compute the regret with this Euclidean Regularizer.

(Refer Slide Time: 00:35)



And we will also assume that all my functions f_1, f_2, whatever I am going to observe are Lipschitz, or we are going to assume that they are L-Lipschitz with respect to l-2 norm because now my Euclidean my regularizer is defined with respect to l-2 norm, so I will also take this parameter to be with the respect to the same norm ok. Now, what is this value? So, if I am going to apply my follow the regularized leader on this, it is going to be n, u is going to be what this is going to be R(u) minus, what was the second term? minimum of R(u) over u. What is that? If you are going to take all over the Euclidean space that part is going to be 0. And what is the third part? N L square and?

Student: Eta.

Eta, but eta value is. So, by sigma right, but sigma is what?

Student: 1 by eta.

1 by eta. So, this is going to be eta quantity ok. Now, I mean, I am ok, I should have directly, let me do this. And now suppose assume that we have been doing right norm of u, we have been assuming this right. Let us assume this u is such that this, but instead of this like I think to be consistent here let us remove this, and just assume that the l-2 norm of u two is upper bounded by B ok. So, if that is the case, this is like half of B square plus n eta L square. Oh, did I miss eta? There was an eta here right. Why did I remove eta? I removed eta just because I wanted to initially argue it is the one strongly (Refer Time: 03:24), but this is what I am interested in. So, there should be an eta also here.

So, do you say we had gotten this earlier as well, this bound? When we just said when we just right now we are obtaining this as a consequence of the theorem we stated, but earlier also just based on my bound I had with my follow the regularizer one and with the Euclidean regularizers, I have already gotten this, now which I am recovering by substituting the values in my theorem right. So, now, when we gotten this, what did we do? Eta. So, what is the value we got?

Student: (Refer Time: 04:16)

So, when we add earlier we had gotten it to be $\sqrt{(2 B L n)}$ right, but now if you optimize this you are going to get it as B L$\sqrt{(2 n)}$, this is just because we had. We are taking this to be a u norm B and so, but earlier we took it like this, but now we are just taking it to be this that is why this difference. But otherwise if you just do this, you are going to get this bound to be this by optimizing over fine.

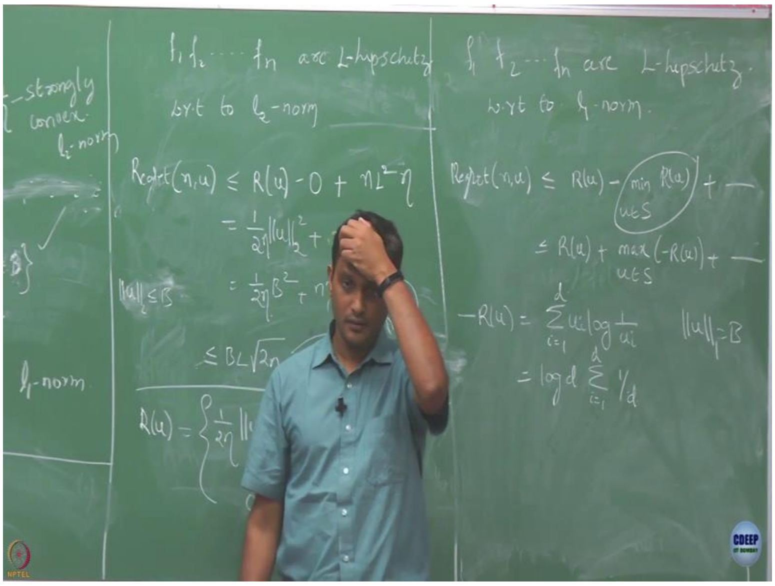
Now, while I am applying this Euclidean regularizer things, whatever I get w's through my follow the regularized leader, they need not be necessarily be a probability vector; they could be any vector in my Euclidean space. So, because of that, if you are using this algorithm on something like prediction with expert advice, where in every round my goal was to come up with the distribution on the set of x parts. So, every time my update used to give me a probability vector. But if I am going to use this Euclidean regularizer and with follow the regularized leader, I will not get that ok.

So, how to account for that case? I want it to be always taking, so I want it to always take value from a probability space or let us say more generally I want to take it values from a specific such not anywhere over my R^d space. So, here if I just going to let my follow the regularize leader, we show that the update rule we are going to get is what gradient decent right. So, w_{t+1} is w_t minus eta Z_t. So, this was the case when my last my f_t functions where linear with Z_t be the gradient in that round t.

So, here this need not be necessarily. So, when I do this update rule depending on Z_t, this can be anywhere in my Euclidean space ok, but how to ensure that my finally, my weights are all within some space which I am interested in. To do that one possibility is you can redefine your R(u) function to be if u is some S that you are interested in this at could be simply probability space and you could define it to be simply 0 otherwise.

Now, if you just do this where you want your u's to be coming only from particular set, then also the same thing holds, whatever we have done everything holds. It is just that we are not allowing it to take value something outside my space S ok. So, we are just defining, but even if you work out all these things again, this is what you are going to get again that says my regret bound is here of the order $\sqrt{n}$ ok.

(Refer Slide Time: 08:39)



Now, let us see what happens if I am going to use my entropy regularizer. Now, let us say my entropy regularizer when I am dealing with this is strongly convex with respect to l-1 norm ok. So, I am now going to assume my f_1, f_2,.., f_n are L-Lipschitz with respect to l-2 norm l-1 norm ok. So, notice that like if a function is L-Lipschitz with l-2 norm, and now if I am asking for is L-Lipschitz with respect to l-1 norm, this l and this l need not be the same ok, there could be potentially different ok.

Now, compute again what happens to the regret, and here also my space is defined to be like this ok. Now, let us compute what happens to my regret here. So, this is R (u). What is the minimum value of this function with under this constraint? What can we what is the minimum value of this under this constraint? Suppose, this is minimization over all ω's right which are satisfying, so let us for time being take it to be 1 ok. What I am now interested in computing? So, this is like R(u), I now I want to minimize over u coming from S, where S is defined like this.

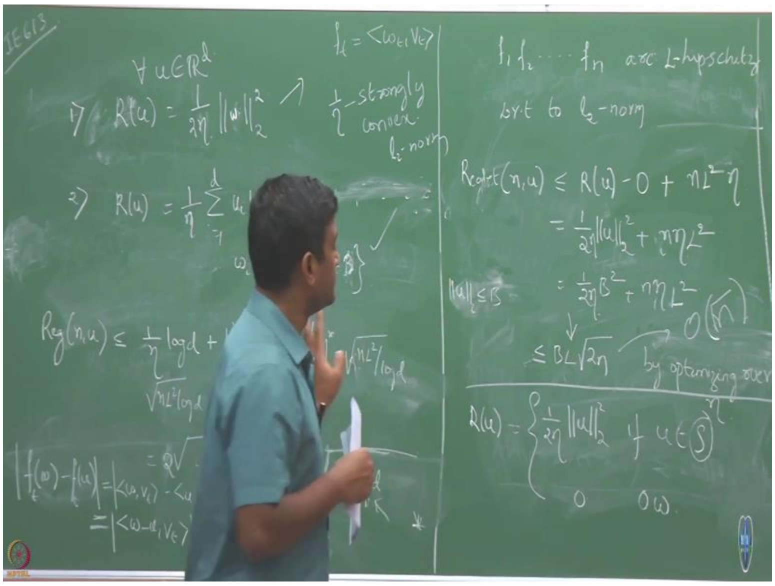
So, this is like a with a negative quantity right. So, if I just ignore this quantity, still I am going to get an upper bound provided this whole thing is not a negative quantity right. If it is whole thing becomes a negative quantity, then negative minus this, I will end up with something ok. So, what is the, so here I am basically interested in minus of this right. So, if it instead of minus, if I am take this inside this will like basically I am looking at the max value of this, max value of minus R(u) ok.

So, I let me write it as. So, there will be a some other term here which I am recovering now R(u) plus max over u of S of minus R(u) ok. So, what is minus R(u)? Minus R(u) is u_i log(1/u_i). I am just taking like minus log(u_i) to be log(1/u_i) ok. Now, I am interested in the maximum value of this over this space. And I am assuming all this l 1 norm to be less than or equals to 1. What will be the maximum value of this? Did anybody come across this, what is the maximum value of this? So, this is the actually the entropy function.

When will the entropy will be maximum? Did you come across this? 1/n, n is different, n is the number of rounds, d is the dissension you should distinguish between these two, so that that would come. So, u_i here is what 1 to d, not 1 to n; n is the number of rounds; d is the dimension of this space, how many components are there in vector u. So, when is this maximized? You are right like it is gets maximized when your distribution is uniform like you put equal amount of mass on each of the components.

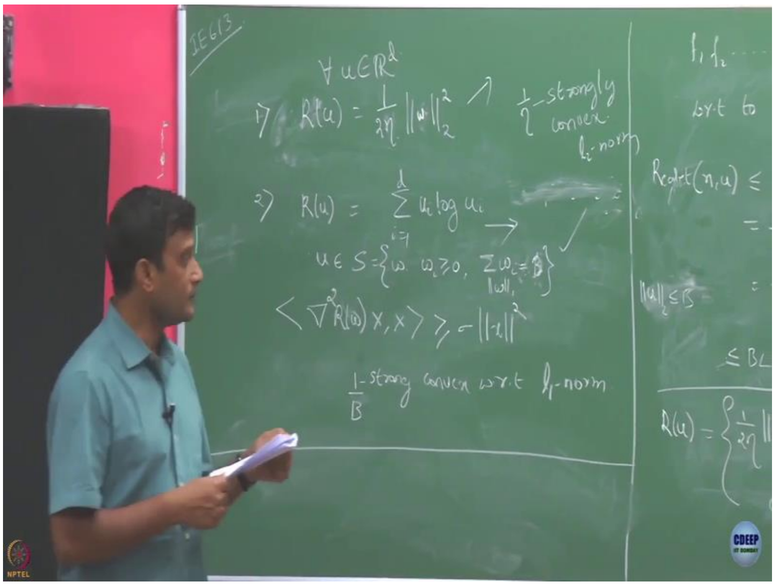
So, if you do that, so this is like if u_i is 1/d. So, this comes out to be log d, and then this is like u_i equals to 1 to d right, or sorry this is like 1 by d. And this sums to what? It sums to 1, because we are adding d times 1 by d. So, this will be upper bound is just log d ok. So, this whole thing here is log d ok. One more thing I am doing here is, so when I compute, I did this right like I am assuming that all this components sum to 1, that is the probability this w's are all probabilities, so that is that sum should be equals to 1.

(Refer Slide Time: 14:29)



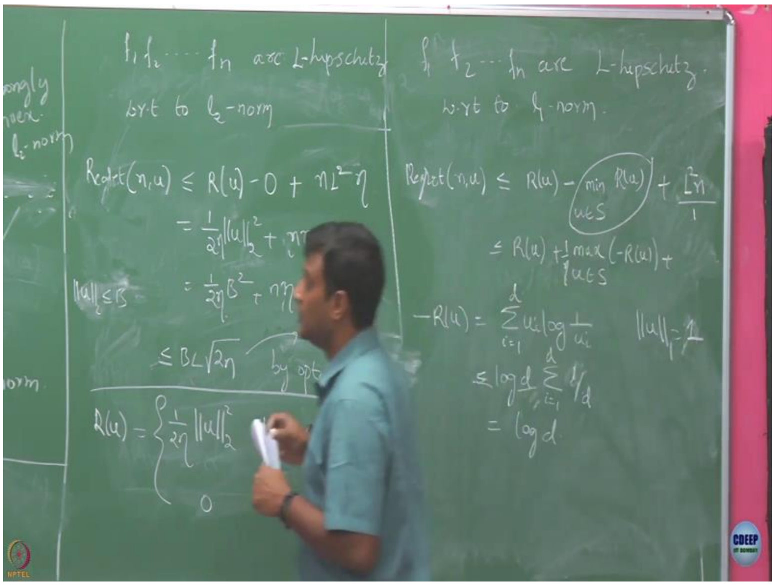
So, now, if you want to maximize this quantity over this probability, this is true ok.

(Refer Slide Time: 14:49)



So, now, suppose instead of 1, if you take B here, the summation of this components is B. What is that you expect that maximizes this? What is that u vector that should maximize this? If your constraint is that the 1- norm of u is B. So, B by d, why is that? So, all the elements are equal right. In that case, so this will be like becomes B by d. And what will this be u_i is B by d right. What this will be then, d by B, and you can similarly like simplify this and get it ok. So, for us we will just to be concrete we will just focus on B equals to 1, so where we are interested in this and for this everything is clean. What we will get is finally, this guy is upper bounded by log d fine.

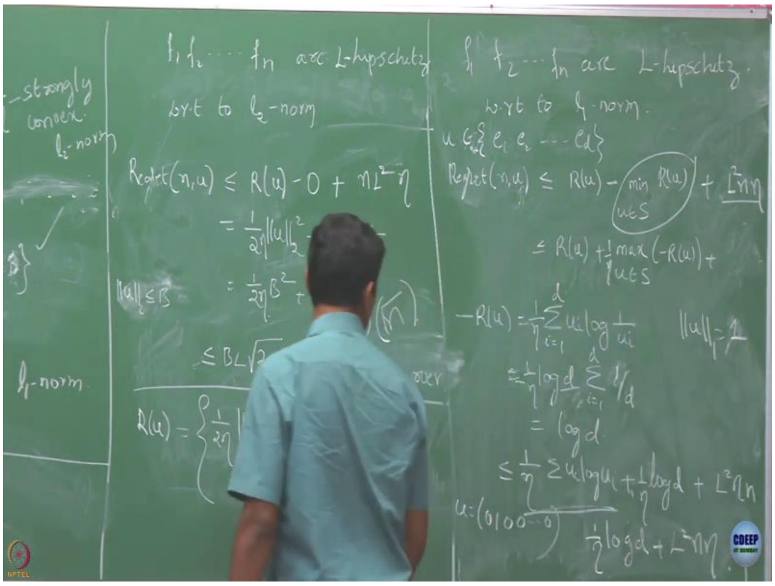
(Refer Slide Time: 16:00)



Now, what is this other part? This other part we had L this Lipschitz constant l square n and we had a sigma in the denominator. What is sigma for this case? So, when it was let say B equals to 1 that is like one sigma strongly convex right. So, it is simply going to be 1 in this case, it is going to be simply L square n in this case ok. I am going to take this to be 1 by eta, just to make sure that. So, what I will get is finally, this is all 1 by eta here. And why did I get this, finally.

So, if there is 1 by eta here, so this will make it what strongly convex? So, this is just a constant here right, it will just come out everywhere right. So, if I am taking B to be just 1, this is still going to be 1 by eta strongly convex ok. So, that I will that sigma I am going to replace it by 1 by eta, so that will give me L square n eta ok, fine.

(Refer Slide Time: 18:06)



So, finally, what we get? This is R(u) is written as $u_i \log(u_i)$. So, what we got this quantity as simply log d plus what is this term here, so I am also missing 1 by eta here right, there should be an eta here 1 by eta, 1 by eta plus 1 by eta and this come here is L square eta n, fine. So, this is what we get.

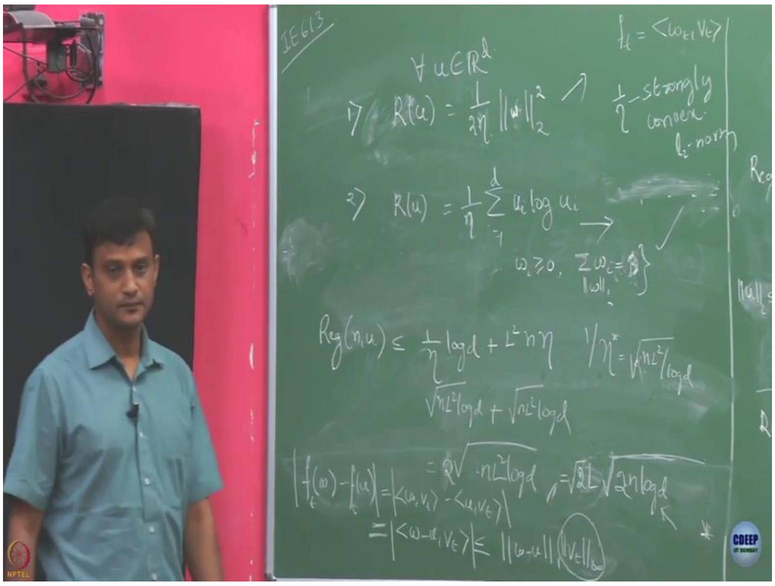
Now, this u's, what was u telling you here? The u was telling how competitive is my algorithm when I am going to use u throughout right. So, if I am going to use the same u throughout what is my regret with respect to that. So, here what I am doing is, I have to choose if you now let us try to put it in comparison with the prediction with expert advice right, because now I am interested in I have brought in distributions here. When I was trying to deal with prediction with expert advice, how did my I defend my regret? My regret was whatever the expected cost my algorithm is going to incur with respect to the single best expert right.

So, there, how to, so, I want to now bring that here. So, if I have to make what is the single best expert, then I can then this u's are like unit vectors right. So, this u is u comes from one of these vectors like e_1, e_2 like whatever e_d. What is e_1 here? It is a vector of all zeroes except for the first place; that means, I will be basically comparing my performance against playing expert one all the time. If I am going to use u to be e_2, it is like I am comparing my performance always with respect to playing second expert all the time like that.

So, let us take u to be anyone of this, then what I am basically doing I am comparing my performance with respect to one particular expert being played in the all the time right that. So, that is what also I was doing when I was computing the regret for prediction with expert advice. So, suppose you take u to be anyone of them, e_1, e_2 or whatever, now what is this term is going to be? So, when I take u to be anyone of them, so let us take u to be e_2, in that case, u_1 will be 0, u_2 will be 1, u_3, u_4, rest are all going to be 0. What is this term is going to be in that case? So it is let say u is 0, 1, 0, 0, all zeroes, it is going to be 0, right.

So, we are just going to define 0 log 0 as 0, ok, log 0 we do not know how to define, but let take the convention 0 log 0 is 0. If that is the case, when it is component 0, 0 log 0 is 0; when it is 1, 1 log 1 is also 0, everything becomes 0. So, the first term vanishes. So, what we will only end up with is 1 by eta log d plus L square n eta ok. Now, can you optimize this and tell me what is the regret bound finally I get for by minimizing this over eta?

(Refer Slide Time: 23:12)



So, what we have finally? 1 by eta log d plus L square n eta. Just minimize this with respect to eta, what is that gives you the value of eta that minimizes this optimal value of this is under root of n L square by. So, now, if you just plug it back here.

Student: log d by (Refer Time: 23:57).

It is reciprocal of this ok, 1 by n. So, if I am just going to plug this what I will get is n L square log d plus and reciprocal of this, it is again going to be n L square log d. So, this is nothing but 2 n L square log d. So, if you just simplify this, this will like 2 times 2 n log d.

So, you see that. So, here also what I had gotten same L to n, but in this case I have set B to be 1 that is fine. And I will get same √(2n), and the extra factor is simply log d. Also can you compare this regret what we have with the regret you got in the for the prediction with expert advice? What was the regret you have gotten for the prediction with expert advice? So, this is same as this right 2 n log d, but now I have this extra factor L here. But what is this L here? Lipschitz constant.

So, when I was dealing with prediction with expert advice, what was my convex functions there? There every time my convex function was like linear function right because I am looking at my expected loss which was like this some v_t ; v_t was the loss vector chosen by the advisory and w was your weight this was the case. Now, for this function f_t, what will be its Lipschitz constant with respect to l-1 norm, how you are going to compute its Lipschitz constant with respect to l-1 norm?

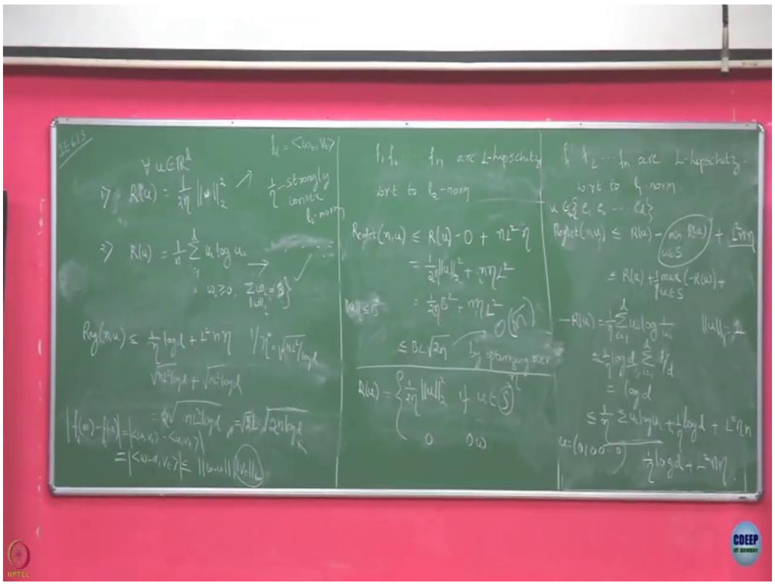
So, let us take f_t (w_t) and so let us take it w and let us take another value at u for some function t. So, this is going to be what by that definition this is w(v_t) minus u(v_t).

Student: (Refer Time: 26:43).

In this case, now let us try to see if finally if you are going to compute the Lipschitz constant of this function with respect to l-1 norm is this it is going to be √(2) ok, ok. Now, we have this 2-L square. So, this I can write it as u w minus u like this right. Now, I am interested in l-1 norm of this right. If I am going to bound it, is this bound true, we have discussed last time that if I take this is to be l-1 norm the next part is going to be the dual norm of that. If it is l-1 norm, this is l-infinity norm. And what I want is for Lipschitzness thing I want to now see, what is this value is going to be right.

So, what I want finally? I want yeah. So, if I can find what is this quantity with respect to infinity norm, then I will get the Lipschitz constant for this function ok. So, I would let us take even if I take the absolute norm of this. Is this correct? I am just applying what are the definition we have last time.

(Refer Slide Time: 29:11)



So, what is this? This v_t is what our vectors, these are we are assumed this vectors to be loss vector in the in the range 0 to 1. So, because of that what is the infinity norm of this

v_t, it could be maximum one right. So, if I am going to take my loss functions to be the linear functions of this form. This linear functions are happens to be Lipschitz with constant one right here.

So, what we have is all fine looks like what we will with by this analysis what we are going to get is finally regret boundaries $\sqrt{(2L)}\sqrt{(2n\log d)}$, but this Lipschitz constant for the linear functions that we encountering in prediction with expert advice is 1. So, if I want to compare this scenario for my linear function, what I am getting is $\sqrt{2}$ times $\sqrt{(2 n \log d)}$. When I applied my weighted majority algorithm, we are able to derive their regret bound which has only $\sqrt{(2 n \log d)}$, but here we got an extra factor of $\sqrt{2}$.

So, we have to bit go back and visit that there is a some error here, or is it that still there is no error here, and we are still going to get a factor of $\sqrt{2}$ here extra, but ok. So, if we just look into the order wise, just ignore about the constants, let us now compare it with respect to my parameters. What are my parameters number of the dimension here which is equal to number of experts and the number of rounds here.

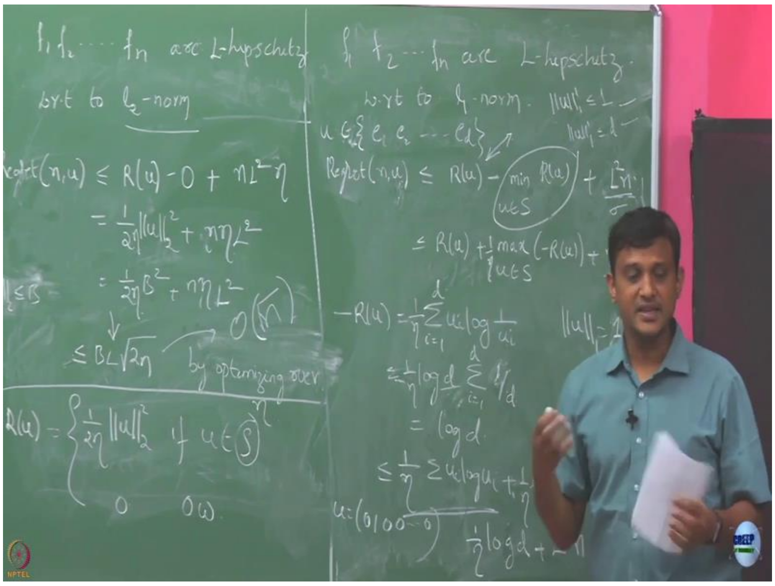
So, whatever we get this order it is the same as that in the weighted majority algorithm right. There also we had got an if you ignore the constant like $\sqrt{(n\log d)}$, here also we have the same thing ok. Actually if you workout all these things with these regularizer and find out what is the optimizer in each round, you will see that these weights turn out to be the same exponentially weighted values that we used in weighted majority.

So, if you just go back and simplify what is the w_t you get in each round for this regularizer function, you will see that this will actually turn out to the same one as that in the weighted majority, you will get the same update rule as that fine. Now, finally, when to use what kind of regularizers? Right. So, we said in one case Euclidian regularizer can also be used by restricting it to the domain of interest. If your domain of interest is only the probability space, then you could also gone and used entropy regularizer. Then the question is which is that you want to use?

You notice that the regret bounds are getting effected by this Lipschitz constants. And this Lipschitz constants are basically governed by which is the norm you are going to look at, or there will be a governed basically by first thing the norm with respect to which you are regularizer is strongly convex, and that the same norm is intern governing this Lipschitz constants ok. In your, when you took your regularizer in the l-1 norm that was when you use entropy kind of regularizer, there what will be your L order? What will be your Lipschitz will be of the order? L when you have entropy regularizer, also fine.

Let us let us come to the Euclidean regularizer. Here what will be the Lipschitz constant with which you are entropy regularizer is going to be Lipschitz with what constant. Did you work out that like yesterday we talked about something right we said what will be the Lipschitz constant of this equilibrium regularizer?

(Refer Slide Time: 35:06)



Did you check at least if even it was an?

Student: (Refer Time: 35:01).

Ok, fine. So, when you are looking let say here your competitor is this unit vectors right, each one of them if you are going to look at u_i here. So, u here which is coming from this space, what is will be its l-1 norm, it will be bounded right it will be at most one because only component is going to be 1. If you are going to let your u to be any value in your Euclidian space, what will be that value?

Suppose, let say you allow it to be instead of it to be any value, you allow each component to take only value in the interval 0, 1 not over the entire R^d space right, this is going to be what? d dimension, right. Whatever is the dimension, it would be upper bounded that it is possible that all the components can take one values.

So, depending on with respect to which competitor you are going to base, in this case when your u is going to be coming from this one of this unit vectors, you have a low value for the l-1 norm right. In this case, because of this you may you are also noticed that if you are going to use a regularizer which is strongly convex in the l-1 norm that is giving us may be somewhat you may want to use that one, because it is what is mattering is this constant right here it may be turning out to be smaller. So, when you are going to compare the regret bounds right fine this L, 2 n factor is there both of them may be lets for time being ignore d. What now worries for you is which is that which one has a larger Lipschitz constant now. And you may want to choose a regularizer which will result in smaller Lipschitz constant, right.

So, based on that, you may want to decide and what is this Lipschitz constant here is the Lipschitz constant here is for your loss functions that is covered actually governed by the norm of your regularizer. So, you see that how what is the regularizer you are going to use that is going to effect the Lipschitz constant and that is going to affect your regret bound.

So, depending on what with respect to not what is your regularizer with respect to norm and what is the corresponding Lipschitz constant it is going to give you may appropriately want to chose what kind of regularizers you want to go about ok, and that is also going to affect your regret bounds ok. Let us stop here.