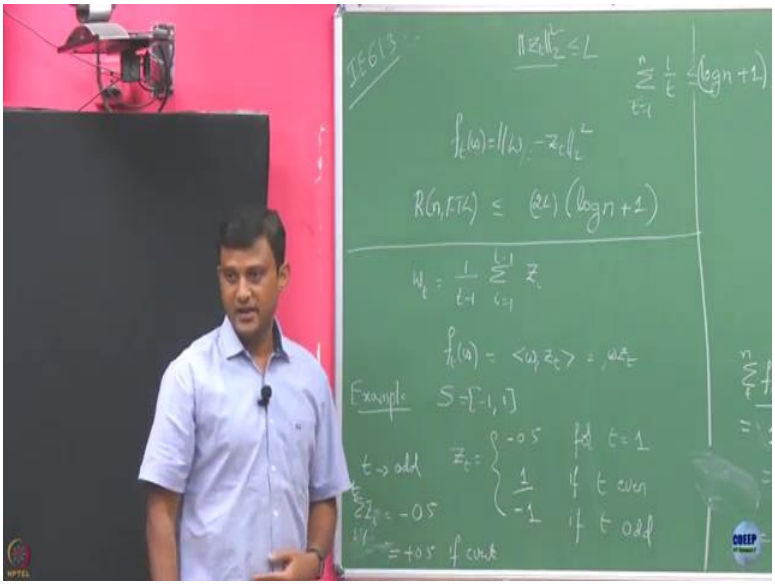
Bandit Algorithm (Online Machine Learning)
Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture - 20
Follow the Regularized Leader

So, we have been discussing what in the last class, we started talking about follow the online convex optimization problem; and in that we talked about we follow the leader algorithm.

(Refer Slide Time: 00:39)



So, in the last class, we notice that when my convex functions are of the form like my convex functions in particular if I take it to be a quadratic loss function, we saw that my FTL algorithms give me, what bound we got?

Student: (Refer Time: 01:00).

So, my regrets of my FTL was 2 L whole square times you got it to be log n, I think some constant we got it of this order ok. So, we just wrote it as log n, but I think we will also allocate this one term here ok. So, what did we assumed? We assumed this z_t square we assumed this to be upper bounded by L or we assumed this to be upper bounded by L? We assumed the Ls this squared ok. If this is the case, then we will not get a square term here ok.

So, let see we said this log n term came from what summation 1 by t t equals to 1 to n right. So, this we said as log n, but I think the right upper bound is this, so that is why we will write it as log n plus 1 here it is just this is a problem fine.

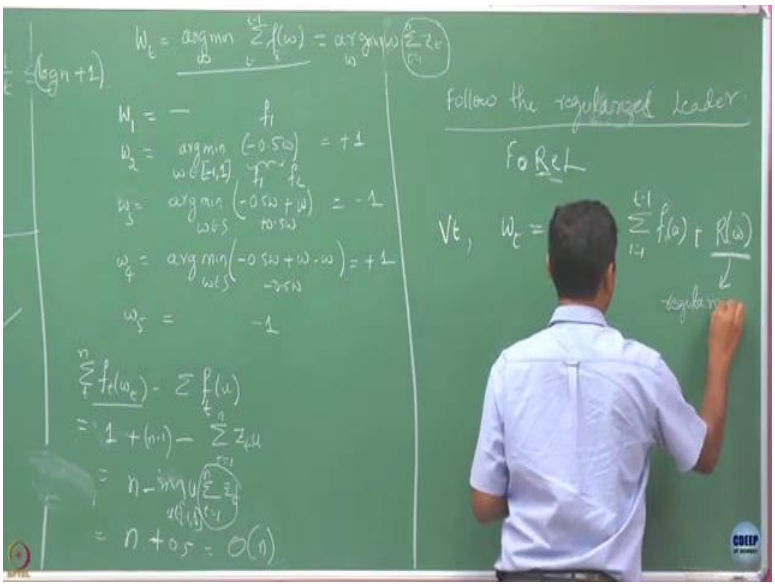
So, good, if you have this quadratic function, so, what was this we are defined it as this. And based on this what was the algorithm gave us, every time it gave us w of t is simply one z_i i= 1 to t-1. So, it basically said take the average of all the previous losses that we have observed.

Why then let us take this kind of loss function, so that is my linear loss functions right. And we said that this linear loss function is nothing but we can interpret as this is the expected loss right once the learner randomizes this strategy ok. How about applying this algorithm FTL algorithm on this setup? So, we will now argue that if you are going to use this setup, if you blindly apply your FTL algorithm on this, you may end up with a very bad performance ok. Let see why is that.

So, for example, take your S the convex set to be -1 and 1. And I am only going to interested in the scalar values now; I do not take them to be linear I will just take I am interested in a scalar case. So, the dimension is all 1 here. Now, in this case I am going to treat like if their dimensions are 1, I could simply write them to be <w, z_t >, both w and z are scalar. So, it is like this instead of some d dimension I am setting d to be 1 ok.

Now, suppose this z_t is that are chosen by the environment are such that, so z_t is this chosen by environment right, it is up to him in whatever way you want he can choose them. So, let us consider some specific case let say he chooses is to be 0.5 for the first round, and for the subsequent rounds he is going to choose it as 1 if t is even. And he is going to choose it as -1 if t is odd, odd and greater than 1 ok. So, now let see what happens if I am going to apply my FTL algorithm on this ok.

(Refer Slide Time: 05:15)



So, let us keep on computing. What is my W_1 is going to be? So, what is my FTL algorithm will do FTL algorithm will do arg min over w sum_{i=1}^{t-1} f_i (w) this is what my FTL algorithm will do right in round t. So, for W_1 this is empty. So, you play something, I do not care anything, I am not worried about it.

Now, let us take W_2 . What is going to be W_2? So, W_2 is going to be minimum of f_1 (w) but f_1 (w) is nothing but f_1 function is going to be -0.5w right because z_1 is defined to be -0.5 in this ok. Now, keep on doing this what is going to be W_3 . So, W_3 is going to be 0.5w + w yes.

Student: There can be a W_2 .

So, W_2 means I am looking at f_1 right. So, what is f_1 (w)? f_1 (w) is w in z_1, z_1 is that quantity. And what is this quantity is going to be, it is simply going to be w right. So, this quantity is like f_1 function, this is f_1 , and this is f_2 , and f_2 we have this one here. And now if you are going to like if you continue to do this, let us do this one more time. What I am going to get 0.5w + w what I will get?

Student: -w.

Is going to be -w. So, if you can continue to do this, you see any pattern in this right. So, this guy here is going to be what -0.5w.

Student: (Refer Time: 07:45).

Plus 0.5w, and whereas, this guy is minus?

Student: (Refer Time: 07:52).

0.5w. So, because of that what will be the arg min of plus 0.5 w. Where w is coming from? So, this w is coming from plus and we are taken s to be minus 1 right. So, what is this quantity?

Student: -1 (Refer Time: 08:19).

It is going to be -1 right. And what is this going to be?

Student: Plus.

If you do like this, what is that you are going to get for W_5?

Student: - 1.

You are going to get -1 like this, and like that it keeps on. So, you see that what is happening your w_i 's are alternating from one round to another ok. So, now, let us compute what is the loss incurred by we are interested in this quantity right f_t (w_t) minus f_t (u) ok.

And now what is my f_t (w_t) over n rounds ok, what is the first one is going to be? In the first round, it is going to be what? It is going to be +1 ok. So, let us ignore the first term, f_1 something I will get it as f_1 (w_1) . So, what is going to be $f_2(w_2)$? f_2 (w_2) is $w_t z_t$ right? What is that going to be?

Student: w_t (Refer Time: 09:40).

When it is z_t , when t is even I have 1; but whereas, when t is even here what I am going to get?

Student: 1.

1. So, w_t into z_t is going to be 1 when t is even. When it is odd?

Student: -1 into -1.

-1 into -1, we get again 1. So, all the other terms here is going to be 1 1 1 1 right, so this is going to be let say n -1. And for time being assume that my this is also like my first loss is also one whatever you incurred I mean I can do anything, but I have simply putting to be 1 here.

Now, what is this quantity? So, this quantity is nothing but z_t into u, where t equals to 1 to n. So, the total loss incurred is almost 1, almost n like I incurred loss in every round. And what is this quantity is, this quantity is nothing but u times t equals to 1 to n of z_t right. And what is this is going to look like plus minus minus whatever it is going to look like.

But I am interested in the smallest value of this right, because I will also be interested this is for a given v. But if I am going to look over the minimum value of this, I can take this u to be between 0, -1 to 1, but how can I make this is the smallest value like upon let say n is an odd number, what will be this sum?

Whatever it is. Let say for time being let us take an n in which this quantity happens to be like after cancelling everything only one term remains that happens to be a positive number. Now, to minimize this, all I need to do is said this u to be 0. So, I can I make this quantity to be 0, by choosing u to be 0 irrespective of what are these quantity is. Had it been a negative quantity, I would have chosen something, but let us assume this turned out to be my n is such that this is a positive term.

Now, because of this if you look this over if you now minimize over u, this quantity is nothing but n. Is this clear? So, what I am saying is the losses incurred by your total n, but the smallest you could have incurred is 0 by choosing your u to be 0. So, maybe like instead of this, let me just make it like min over u ok, just let us look into this. Let us write down what happen. Let say t odd number, what is the sum of Z_t say odd number and greater than 1. So, let us take 3, 5, 7 like that. If it is 3, what is this number?

Student: -0.5.

And if it is 5, it is always -0.5. And if it is if this ok, and if the t is even.

Student: +0.5.

It is going to be + 0.5. So, this whole quantity here depending on my n is odd or even, it is going to be -0.5 or +0.5. So, when it is +0.5? How can I make this the smallest value? By choosing u to be 0.

Student: (Refer Time: 14:02).

Yeah.

Student: (Refer Time: 14:05).

When u is what is that we said when it is let say in this case it is in this case when it is 0.05, right, I can make it we can make it just -1, in that case it will be -0.05 ok. Let us do that, -0.05, and that minus with this minus, it become some 0.5. And if it is the other case, when I have minus of this I am going to choose my u to be +1, it is still going to be.

Student: (Refer Time: 14:50).

It is going to still this right. But still you if you you see that this is still like order n right, this is still order n. So, because of this, if you are going to use this follow the leader on a linear function here, you will end up with a very bad regret ok, you are. So, what is the issue here? It works so well for this quadratic loss functions (Refer Time: 15:28) quadratic functions, but it is doing so badly for the linear functions.

So, what, so what is that like can we say something about this. So, what happened basically when we are trying to do this minimization right, this is arg min. If I write simplify this, this is going to be simply w times z_t t equals to 1 to n if this function.

So, depending on summation, the values the arg min here, it was changing in every round right, it has becoming plus minus as we argued here, it is changing in every round. Whereas, in the quadratic case the change was not so abrupt. Why was that? No, I am asking about this loss function. So, in this loss function, what was my w_t , we wrote it (Refer Time: 16:53) some we already written here.

So, in this case, the w_t we found by this method where my f_i (w) was this function, my w_t turned out to be average. So, because it is an average, every term here will be influenced by what happened in the large samples. So, in a way the change will not be that abrupt here right, because the past everything is kind of getting accumulation,

because that in the next round the things will not change that drastically. It is kind of a running average right running average usually do not change suddenly.

But whereas, this in this function, the things were changing very rapidly right plus minus, plus minus like that. So, in a way the kind of past what has happened, it was kind of getting nullified it has no impact on the current ones. So, such abrupt changes were making this algorithm kind of unstable, it is rapidly changing.

Whereas, there is kind of stability here because your average your updates are kind of getting averaged based on the past observations. So, in a sense, in a way what this is telling is for your FTL algorithm to give good performance, you were weights should not be getting changed abruptly, and that was happening in my quadratic loss function, but it is not happening with my linear function here.

So, now the question is it possible to make that even if I am going to observe a sequence like this, my w_t 's will not change so abruptly, so that the updates are in some sense are stable. Maybe if you can do bring a such kind of stability in this even for my linear function, my FTL algorithm may do better ok. And usually the way to do bring that kind of stability is through bringing our regularizing functions ok. How many of you heard about regularizing functions? So, what kind of regularizes they use there?

Student: (Refer Time: 19:25).

LASSO so for what?

Student: (Refer Time: 19:29).

But what kind of what was the difference between loss and (Refer Time: 19:36) and the L_2 regularizer.

Student: So, (Refer Time: 19:40).

So, one of them was trying to make your weights kind of sparse.

Student: Sparse.

But doing that did you see any advantage? Right, I am asking whether the fine, if the weights are very close very small or 0, those features are no effective, you can better not use them. But the question is did this improve a performance?

Student: Yeah, (Refer Time: 20:02).

Ok.

Student: (Refer Time: 20:05) not in the training set, but in the overall they are.

So, I mean in the test case you said that, but then did we did you realize what was the reason?

Student: (Refer Time: 20:23).

So, you are saying it kind of avoided over fitting. Anybody else came across this, but anything related to stability there?

Student: What is the average stability, but adding this is better than is saying the (Refer Time: 20:44).

Student: (Refer Time: 20:49) we want to make our models in some sense smaller.

Student: Smaller model would be (Refer Time: 20:52).

So, how is that (Refer Time: 20:58) was ensuring by making sure that some all the weights are not big enough or not all the features are relevant. So, some of them need not be given importance, but why is that squared norm?

Student: (Refer Time: 21:09).

Why, why smaller value, why you are saying it is smaller values?

Student: (Refer Time: 21:13).

You are adding it and then you are adding it to the loss and trying to minimize it.

Student: Yes, (Refer Time: 21:18) the weights can only will large if they are actually having a good (Refer Time: 21:24).

Weights are large?

Student: So, if we are adding w (Refer Time: 21:28).

Student: So, w can only be large (Refer Time: 21:32).

If it is really important.

Student: Really important and relevant.

Ok.

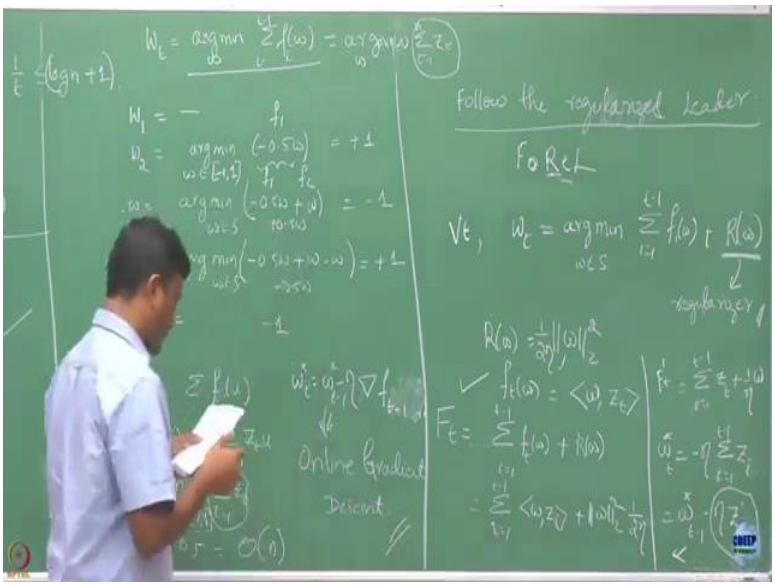
Student: If sometimes they making this loss go down (Refer Time: 21:40).

Fine, the other way of looking it is you are adding this L_2 norm to the loss itself. Instead of other way of saying that you are you are trying to make sure that loss that norm of L_2 is not large right, we can think of its like a constraint on a norm of two you are constraining it to be within some value, fine it also kind of.

So, there your kind of restricting your weights, there the goal was to make sure that you do not over fit to a training data, but here in this online version our goal is to make sure that the updates are not becoming too erratic in the sense the updates are not changing too much from one (Refer Time: 22:25) to another.

So, we want them to be kind of stable like we do not if it is abruptly changing too much in a way that means, that we are started kind of ignoring the past. So, if we stable means we are completely not ignoring the past like we are taking it and slowly allowing it to vary, fine. So, then what is my the regularized version of my follow the leader algorithm ok.

(Refer Slide Time: 22:58)



Now, we are going to study something called follow the regularized. So, what we will do in this case is instead of simply maxim simply finding the sum of the minimum value of the sum of the losses we have observed so far we will try to do this minimization after adding a another regularizing term to this. So, the what this algorithm we are going to we denote it as FoReL. So, Re here stands for regularized leader.

So, this algorithm is going to do for all t w_t equals to arg min over and this term we are going to call it as regularizer. So, we are going to see different regularizers as we go on in the class, and we see that how they are going to affect our performance. So, let us take a specific example of L_2 regularizer, where I am going to define this R(w) to be this quantity. So, I am adding this quantity to directly to the loss right. So, I want this to be kind of small and this is the normed version of this.

So, if I am doing this, I do not allow my the minimizer to be kind of too big right, because of that I will in a some way I am controlling its variations ok. Now, with this suppose if I take this, and let us take my f (w) to be simply my linear function where f_t (w) is defined by <w, z_t> is the parameter in round t. So, if I plug in this, can you tell me what is the w_t I am going to get?

So, just plug in you have put it here and put R norm of w here, and then differentiate and find out what is the w_t you are going to get. Can you differentiate and tell me? But I am

going to also defend this to be some parameter here. This is going to be, now tell me what happens?

So, if I am going to differentiate this with respect to w, so this is going to be. So, do we get this? The minimizer w_t* to be minus eta times summation of z_t ok. So, there should be. So, can I write it as? So, this is a summation from i equals to 1 to t -1 right. I can split this make the summation i running from 1 to t-2, and separate out -eta.z_{t-1}. But the first part is nothing but w star of till t-1 right ok.

So, what I have basically done is I have iteratively written this updates. If my previous update I have gotten in round t-1 was w_{t-1}*, and I got this z_{t-1} my new value is expressed as like this ok, ok. Now, if you look into this function my linear function, what is z_t here? So, my this is my f_t (w), my variable is w, then what how can I interpret this z_t to be here?

Can I take this to be the slop of this function z_t ? So, what I am doing here in this in this update to get my w_t* from this w_{t-1}*, what I am doing is basically subtracting the gradient or slope of my this function, and with this coefficient of, but scaled by this coefficient eta, where is the eta is a term which is coming in my regularization function.

So, what I am doing to get the new update I am subtracting this gradient of my function at t-1 from my previous update. So, you see that my weights are not drastically not will not change here right, because they depend on the previous update, and they are going to change as per the change in the gradient of my function.

So, because of this, so what I am doing, my weights are going to change. In fact, they are going to decrease based on they are going to change based on this slope gradients. So, because of this nature, if we are going to use this regularization like this and I am going to use a slope like this, what I will get my update rule to be a rule which is which we call it as gradient descent right. Because this is my gradient and I am reducing my weights by that. So, we are updating as per this gradient descent method.

So, relatively just to this what you have being basically doing we are saying that this is nothing but w_{t-1}* minus eta times gradient of my f_{t-1}. So, z_{t-1} is the gradient of this function right.

So, I am just going to write it as f_{t-1} as a gradient of my f_{t-1} ok. So, because of this, we are going to call it as. So, any question about this? So, we have (Refer Time: 31:52) with a simple update rule right which is going to decrease, they have going to decrease the weights according to the gradients.