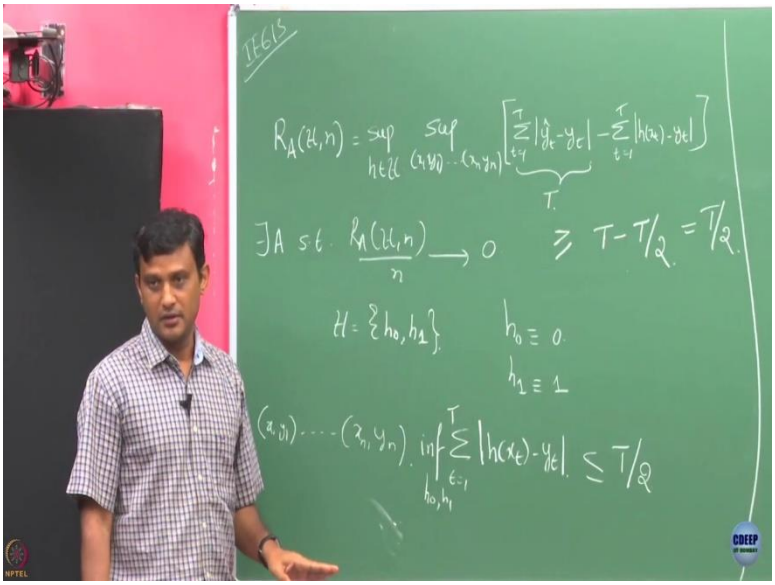
Bandit Algorithm (Online Machine Learning)
Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture - 09
Covers Impossibility Result

So, in the last class we started discussing about how to deal with the case when hypothesis just generates the labels does not belong to our hypothesis class right. We said we want to relax the realizability assumption and we started defining what is the notion of performance criteria when we do not have this realizability case.

So, we introduce the notion of regret which was basically said like how good you are in hindsight ok. How good you are in hindsight compared to a I mean how good you are algorithm is performing compared to an algorithm which knew all the information. So, just to reiterate that we have defined our regret as.

(Refer Slide Time: 01:14)



So, this was our notion of regret and we said that our goal is to identify algorithm which makes this regret sublinear. Or, alternately we said that this hypothesis class is learnable if I have an algorithm A such that $\frac{R_A(H,n)}{n} \to 0 \; as \; n \to \infty$).

Now, let us try to understand what kind of bound we can expect on this quantity. Is it possible to achieve this at all? Can I come up with an algorithm? Which makes my

hypothesis class learnable?. Now, that I have remove the realizability assumption, the power of adversary is much more now. Like maybe he can just look into the label what you predicted and then just give opposite of that as the true label right.

In that case what could be this value? If the adversary can look into what is that you predicted then what is the what value what is the maximum he can make this value?

He can make it T right by just flipping what you predicted he can make it, fine. So, he can force you to this. Now, we will try to understand what this quantity can be for the simple case after hypothesis. Let us say that this hypothesis class take it to be consisting of just two hypothesis h_0, and h_1 where, h_0 always says 0, h_1 always says 1. Let us try to understand what this quantity is going to look like for this specific example.

So, is this hypothesis class clear? This is the trivial hypothesis class ok. Now, given any sequence of labels let us say whatever the sequence, I have to look at this quantity over because since I am doing supremum over h; I will look into this quantity for only two h possible h values. One is h_0 and another is h_1.

Now, irrespective of what is the sequence, if I am going to look at this value. So, let us let us take let us take this sup inside here, instead of doing this let us say I will take here. I basically looking at infimum value of this quantity right. What will be the infimum quantity of this quantity, of this?

Student: (Refer Time: 05:23).

What is that? y_0. So, let us say; let us say y_t sequence right. So, if I have y_t sequence for T rounds. Either, if less than T/ 2 are 0 rest of them are 1 or other way around right; if less than T/2 or 1 others are all 0. Now, because of this if we are going to look at two possible things h0 and h 1, whatever this quantity can by choosing one of this h_0 and h_1 depending on what your sequence is.

Can this quantity be made at least, sorry at most T /2 the mistakes?. So, let me put it; so, let us consider this quantity. So, let us say you have been given this y_t sequence, let us say now you have to predict using your hypothesis over h_0 h_1; we have to match this label sequence. Suppose let us say in this half of them are 0 less than T/2 are 0 and the remaining ones are 1.

By choosing your hypothesis class to be h_1 what which is always predicting 1, what this quantity can be? Less than T /2 right, because they are the one only mistake remaining ones are always already 1. So, there will not be error.

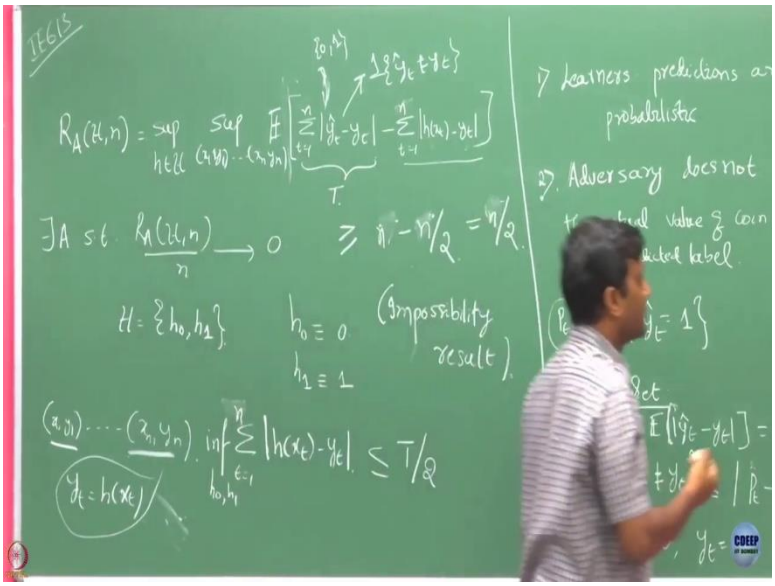
Now, look at the other possibility, where the first T /2 are 1 and the remaining ones are 0. In this case if we are going to choose this hypothesis h_0, what will be this bound? Again it is going to be T /2 right. Irrespective of what is this sequence is going to be, if you are going to loop minimize it over h_0 or h_1, this guys always going to be T/ 2.

If environment give all this y_t is to be 0; in that case you since you are minimizing you the best one is to choose 0. It is going to be upper bounded by 0, but we are taking the worst case. Irrespective of what is the sequence you are going to choose, if I have to choose one of them, I am then this quantity will not be more than T/2.

Now, if you go with this, this can be T ,this remaining quantity can be at most T/2. And, if I want to then can I get a bound on this like this for this problem and because of this, this is going to be like T/2. So, even for the simple case of two hypothesis here we have a case here, where this regret will be at least T/2 that is what right like this is I had did it over arbitrary sequence.

So, you do it for any possible sequence, still this is going to hold right. Like whatever the sequence this bound is going to hold right. Now, because of this we have a lower bound of T/2. So, instead of T, I am using and let me consistent with that.

(Refer Slide Time: 09:28)

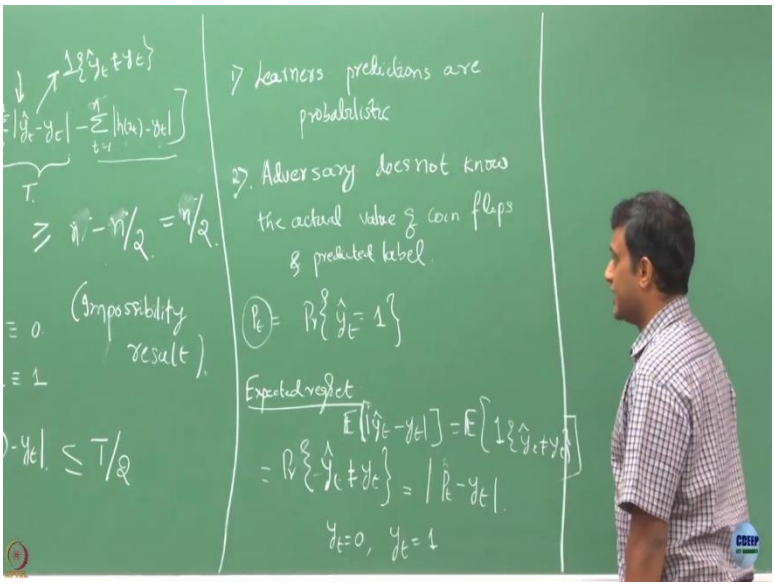


Now, if I get a bound like this, like lower bound is n/2; can I ever get a can this hypothesis class will be learnable? No, right because if I divide it by n and let n go to infinity this fraction is never going to be 0, it will be at least half right. So, because of that if we under this unrealizable case there is no way that I can come up with an algorithm which makes the even this such a simple hypothesis class learnable.

And, this is called like impossibility result ok. So, this is a bad case right like if adversary is so powerful that there is nothing you can make, like you will not be able to kind of get a sub linear regret here. Now, so what we will do is instead of going to compete against such a powerful adversary; we will restrict again the adversaries power here.

It is again in the unrealizable case only, but we will give bit more flexibility to that learner while we restricting the power of adversity.

(Refer Slide Time: 11:15)



So, what we are going to do? First we allow the learner to make probabilistic predictions ok. It is not necessary that he has to in any round his strategy has to be deterministic. He can may be in every round he can have his own strategy, where he is going to say a label 1 with some probability. Even though he knows it is he want to say it label 1, but he will only going to say its some probability ok.

So, even if I make this assumption will anything change here in this? Will if the learner is going to make a probabilistic decisions, can this become learnable? Not really right, because when I said he is going to make $\widehat{y_t}$ is the predictions made by the learner; I did not say what is the strategy actually he is going to use. He could have use randomize a strategy, but what was the case here like if the learner adversary can see the final predictions made by the learner, then he could force such things here.

So, that is why even though we have allowing the learner to make probabilistic predictions; we will further say that the adversary is going to come up with his label y_t before learner makes his prediction $\widehat{y_t}$. You understand this points? That is he is not first in round t learner this whatever let us say in x_n y_n in nth round the adversary comes this prediction before he says what is the prediction made by the learner in that round.

So, if we restrict that then maybe like this kind of situations can be avoided. At least we cannot make you to be wrong in every round. So, when I say this point second point, it is

only for that particular round. Let us say I am in particular round t, in that round adversary does not know a priori with what probability you are going to make a prediction of one.

And, also you he will not see what is the prediction that has been made in that round before he chooses his level ok. But, he might have known all the prediction you have made in the past. He has that history, but what he does not know is in that round what is the probability with you are going to make a prediction and what is the true label, what is the predicted label.

So, fine we are not making that assumption here right, we are not making that there is a relation like this. So, adversary is generating this according to his own logic. It could be doing this, but it is not necessary ok. The only restriction you are putting on the adversary is that he cannot see your prediction, because before he declares the label ok.

Now, how does things change with this? So, what we are making is we are kind of allowing the learner to kind of confuse the learner adversary also right. Because, you are making the randomized prediction maybe adversary even if we looks what is the prediction you made finally, he may not be knowing what is the actual value you would have predicted right.

And, now that you have also restricted this adversaries power maybe in this case learner can do something better than this bad case, where there is no way that learner would have been able to learn even the simplest case here fine. So, with this assumption $\widehat{y_t}$ in round t is no more deterministic right. Because, of that we are going to now start dealing with what is the probability that y_t equals to 1.

So now, this is the strategy of the learner. In each round he is going to come up with probability with which he is going to declare the label to be 1 ok. But, what you care is environment could be completely random. You also do random things maybe if environment is totally random, if you somehow figure out that random things you have now started doing as good as him right, that will be our aim.

Somehow you figured out the adversary is doing that with probability 1 he is flipping he is declaring label 1 and probability ½ is declaring 0; you also suppose you also did that. Do you think you will be able to do better? Like you have basically understood what the adversary is doing right; that means, you basically learn the environment.

If there is a hypothesis which makes the label that is fine for you right. It will be you are you will be able to you see that you should be able to figure out that. Now, the question is what happens if there need not exists a hypothesis class ok.

So, right now this is not obvious to you, but can we hold on this thought and see that without making this assumption if we can show that we will come up with an algorithm such that this sub linear regret holds. So, fine all your question now is you are not able to see an algorithm which will guaranteed this right. But, if we can eventually come up with an algorithm that should tell you why that is possible right. So, we will exactly do that ok.

So, we will try to show an algorithm such that this is indeed true ok. And, there we will see that is it necessary to have this condition that there exists a hypothesis or it is not at all necessary. So now, the strategy of the learner is to in every round come up with this probability with which he is going to declare the label to be 1 fine.

Now, that the learners is learner is randomizing his predictions what we have to now do is we have to redefine this regret to be a regret in an expected sense right. Because, this quantity is no more deterministically chosen by the learner ok. So, how to redefine this? So, we are going to take the expected regret which is going to take the expectation of this quantity over the randomness of a learner ok.

So, let me put an expectation here and this is what I am going to now call as this expected regret. And, now this quantity is no more we are allowing this to be random quantity right. So, because of this quantity anyway now it is not going to change, there is nothing random here right. Because, for a given sequence these are fixed and anyway you are your h is also fixed.

Now, anyway given this sequence this y_t is fixed, now what is going to change is this depending on your strategy ok. Now, let us write this. So, what how can you write this quantity? So, if I take the expectation here I will just interested in knowing what is the. This quantity here; can I write it as indicator $\widehat{y_t}$ not equals to y_t? And, now if I take expectation of this indicator what it will be?

Student: (Refer Time: 21:40).

Right. So, in that case it is going to be Pr $\{\widehat{y_t} \neq y_t\}$ right. So, I want to write this quantity, this quantity can be written as $|P_t - y_t|$. So, let us understand this ok. So, just let me write this bit more clearly; $E[|\widehat{y_t} - y_t|] = E[1_{\{\widehat{y_t} \neq y_t\}}]$.

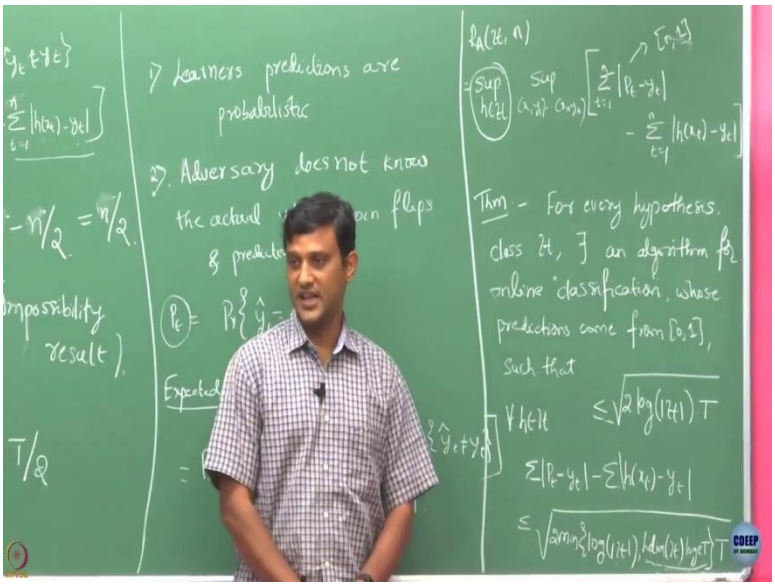
So, why this is true? Suppose, let us say here y_t is equals to 0, sorry let us begin with y_t equals to 1. So, y_t equals to 1 y_t equals to 1; that means, what? You are saying $\widehat{y_t} \neq 1$; that means, y_t is basically.

Student: 0.

0 this out of this ok, let us take the case y_t equals to equals 0. If y_t equals to 0 the right hand quantity is simply P_t right. And what is this quantity here? $\widehat{y_t} \neq 0$, that is same as saying $\widehat{y_t} = 1$, right and $\widehat{y_t} \neq 1$ is exactly P_t by our definition. So, this is true.

Now, you similarly if you take y_t equals to 1; so now, this quantity is like $P_t - 1$. Because, of this mod than I can think this as simply $1 - P_t$ .And what is what is $1 - P_t$? is the probability that you make prediction to be 0 and there is exactly this right. Like if y_t equals to 1, this is like saying $\widehat{y_t} = 0$, and that is $1 - P_t$. So, we have this relation.

(Refer Slide Time: 23:48)



Because, of that if you simplify this quantity, I have we are going to write our regret as simply $\sup_{h \in H} \sup_{(x_1,y_1)\ldots(x_n,y_n)} [\sum_{\{t=1\}}^{n} |P_t - y_t| - \sum_{\{t=1\}}^{n} |h(x_t) - y_t|]$

This is exactly that expectation we have removed and then we showed that this is exactly this quantity right, is it not that what we just now argued?

What is the difference between this quantity and that quantity there now? So, here this quantity $\widehat{y_t}$ was either 0 or 1, the prediction right. But, what is here? This quantity here is anything between 0, 1 right. So, we are just now when we move from this setting to this setting we are just saying that, everything remains same except that instead of saying that $\widehat{y_t}$ has to be either 0 or 1, we are allowing it to be any value between 0 1 0 and 1.

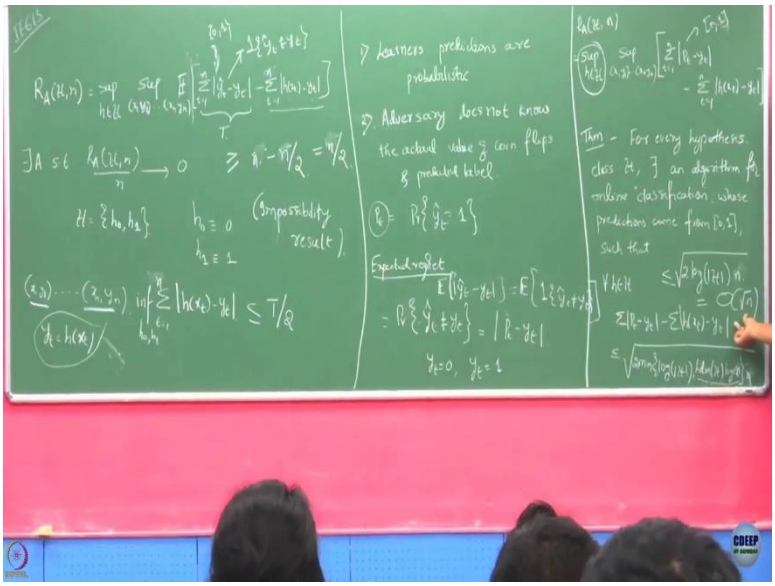
It is like any here we have this is like when p t equals to some quantities; this is just like probability of predicting label 1 ok. So, what now after you did this we have redefined regret like this. Now, am I interested in showing this to their existent algorithm such that this quantity holds. So, whenever this happens we are going to say my regret is sublinear or my hypothesis class is learnable.

So, the next theorem we are going to make this bit more formal for. So, what this theorem is saying you take any hypothesis class h, right now we have not mentioning anything like this hypothesis class is finite or infinite ok. You just take any hypothesis class.

There exists an algorithm for online classification, notice that we are still in this classification regime like 0, 1; we are our predictions are always 0, 1. So, we are just interested in classification so far and whose predictions come from 0, 1. Do you remember like instead of going a prediction to be 0, 1 we have translated it to predictions to be coming from 0, 1 which we interpreted as probability of giving label 1.

Such that you take any hypothesis then this quantity will be upper bounded like this, all of you can see this and read it? So, this quantity is nothing, but the quantity inside whatever; inside this square brackets and is going to be upper bounded by this.

(Refer Slide Time: 28:34)



So, what we are saying is we are not, you take any hypothesis, use that as a benchmark. And, now if you compare whatever the loss we are going to incur, this is going to be your regret bound. So, it is saying that if you are going to use like let us say this fixed hypothesis class ok.

Now, you compare the loss we incur with this fixed hypothesis class against the loss you incur by your policy. This difference in the loss is going to be upper bounded by this quantity. And, this h could be any h that is coming from a hypothesis class. In this like we already said that right if he uses this he can force to a loss of total T.

Yes, this theorem is valid like all the statements holds provided this condition hold. If you if this any of this are violated, this theorem is no more guaranteed ok. All of you read what is this bound here. So, this is the l dimension of h, this is the log of cardinality of h; it is a bit complicated, but I think you can parse it. This is the log(eT). So, this is one term here, l dimension of h into log(eT) and log h; the minimum of these two whatever it is you take this and this whole thing multiplied by T here ok.

Maybe so, this is the precise statement, but what we would be interested in a bit slightly weaker version of this. This is an arbitrary, this statement holds irrespective of what is the cardinality of your hypothesis class h ok. So, it may so, happen that even though our hypothesis class has infinitely many points in that, but the l dimension of that hypothesis can be finite.

In that case this in the minimum term that will only make impact like that has no meaning, I mean that has no importance there. So, similarly it may happen that for some hypothesis class even if it is anyway if the hypothesis class is finite, we know its l dimension is also going to be finite right. Because, we know that l dimension of hypothesis class h is upper bounded by what?

$\log_2(|H|)$; so, if h is the finite size everything is fine. So, if h is infinitely many then maybe like only thing that matters is this quantity, not this quantity ok. Now, suppose let us say because this is bit many terms here, suppose let us assume for time being my hypothesis class is finite ok. So, in that case can I write this bound like this? So, this is a simplified version of this, when my hypothesis class is finite.

Why this came? Because, if I am going to take this is the minimum of these two quantities right. So, if I only take this quantity, it is still an upper bound on this; is that clear? So, if I drop this and only retain this, it should be still an upper bound because here I have a minimize. In instead of looking at minimum of these two terms I am just take one of the terms ok.

Now, if I have a bound like this, notice that this bound is independent of the sequence; is that clear? It is independent of this clear and this point hold irrespective of what hypothesis you have chosen, because of that is this also bound of this quantity right. Now, can I say there exists an algorithm such that this holds, why is that? Because, here it is square $\sqrt{n}$ right. This is some constant, but it is like growing like order $\sqrt{n}$.

So, this we are going to denote it like order square root n, all of you understand this order $O(\sqrt{n})$ notation. So, this regret is order $O(\sqrt{n})$. And, now if you are going to divide square root of n by n and let n go to infinity, this quantity is going to be 0 right. So, this theorem is saying that hypothesis class h is going to be learnable as long as either it is going to be finite or that its little dimension is finite.

If let us say either hypothesis class is if its little dimension is going to be infinity, this upper bound is already vacuous; I cannot say anything. But, as long as this little dimension is finite; we can always guarantee that this there exists an algorithm which will make my hypothesis class learnable.