Bandit Algorithm (Online Machine Learning)

Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture – 05
Consistency Halving Algorithm

So, in the last class, we briefly just touched upon the supervised learning set up right. We just said what all the things, what kind of settings we consider in supervised learning. We said what is the sample space, and how the labels are generated, and the samples are generated.

We then said the process of generation, we said they are going to be generated IID, then we said we set our metric of evaluation as the one which minimizes the we are interested in a hypothesis which minimizes the test error which we also called as risk.

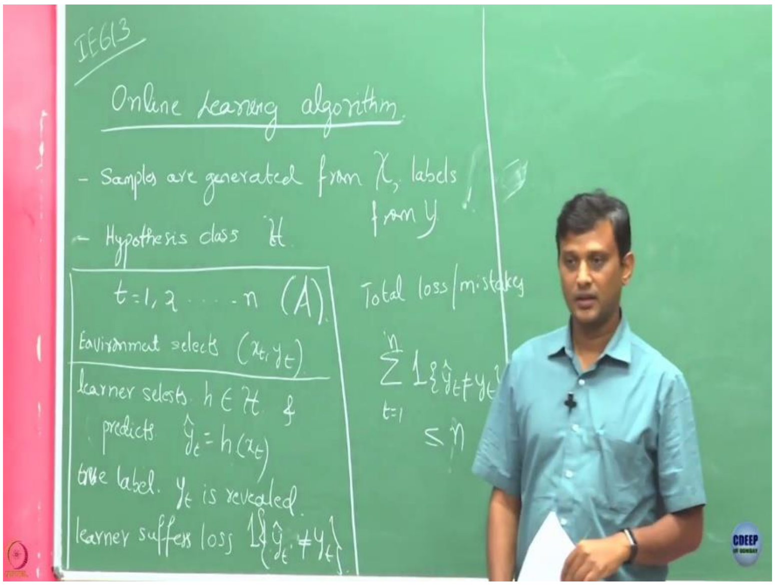
Then we looked at how to come up with a good hypothesis which gives me small test error right. We did it through empirical risk minimize. So, that was all in the path setting given me a bunch of data points, how I will cover with a whatever given to me, how I will come with a good hypothesis that performs better on a test point which I have not seen, but still generated from the same process on which I have trained myself.

Now, we will move on. And we will now consider a scenario where I do not have the luxury of collecting all these data points a priory to train, but what happens in my setting is the data points come one at a time. And once I say a data points, I have to make a decision.

So, for time being, we will restrict our self to binary classification like when a data point comes I have to say whether it is going to be plus or 0 or -1. And I goal is to make sure that I strictly I very fast making small errors, or eventually I do not make any error, and I want to make sure that I do as quickly as possible.

So, in this setup, there is no separation of training and testing. You get a point, you have to decide. You see you get to see whether you made a good decision or bad that is what we call as feedback. Then in the next round, you get an another point. Then you take an you apply an let say hypothesis, then you get to see whether we made a good choice or bad, then the process continues.

(Refer Slide Time: 03:06)



IE613

Online Learning algorithm

- Samples are generated from $\mathcal{X}$, labels from $\mathcal{Y}$
- Hypothesis class $\mathcal{H}$

$t = 1, 2 \ldots n$ (A)

Environment selects (x_t, y_t)

learner selects $h \in \mathcal{H}$ & predicts $\hat{y}_t = h(x_t)$

true label y_t is revealed

learner suffers loss $1\{\hat{y}_t \neq y_t\}$

Total loss/mistakes

$\sum_{t=1}^{n} 1\{\hat{y}_t \neq y_t\}$

$\leq n$

So, we introduced this general notion of online learning algorithm. So, as usual we are going to say that samples are generated. So, this is the point where the features are drawn. And then the let say this and we have an hypothesis class H, this is given to you ok. And let say labels are also from so Y and so.

So, the environment is going to draw a sample, and it is going to also associate a label to that sample, but that label is not going to reveal to you ok. So, let say, now I am talking about rounds right, round 1, round 2 because my data point is coming one at a time. So, I am going to now call time t = 1, 2 whatever. In every time what happens a data point is drawn.

And when I say data point it, will have both x_t and y_t. What is y_t? y_t is the associated label with x_t. At the beginning of the round, you will get to see x_t, you are going to choose a hypothesis from hypothesis class apply on that, you make a prediction, so that is basically $h(x_t)$.

After you do this, you get to see what is y_t ok. Now, you get you can compare whether $h(x_t)$ is same as y_t. If they are same you know that you made a correct decision otherwise not, so and this repeats. And whatever you are going opt based on that you are going to choose your hypothesis in the next round. So, this interaction any, and how you are going to choose your next action that is a thing which is specific to an algorithm ok. Based on that, you can come up with different algorithm.

So, we are going to consider this simple setting as so in every round let say write only then after you do this true label y_t is revealed, and then maybe you suffer loss. So, again let us repeat this. This is going to run in rounds t = 1, 2 on the way up to may be let say for some fix number of rounds which I am going to denote it as n.

So, in each rounds, the environment has selected x_t, y_t pair. How environment assigned a label to x_t, we do not know; how it is going to draw this x_t, we do not know that is completely we do not know that is the environment.

In round t as a learner my action is to choose a hypothesis, and then to give a prediction in round t, and this is my exactly my prediction. After I do this, I get to know y_t. And if my prediction, so actually this is $\hat{y}_t$ not same as y_t, then I incur a penalty of one unit; if they are the same, then I incur no loss ok.

Now, this I do not have any control over how the environment is selecting this pair x_t, y_t, I do not have any control. But what is in my control is, how can I choose an hypothesis in each round right. Depending on this logic, how you are going to choose your h in each round the kind of loss you are going to make is going to differ ok.

Now, so this is how I am going to call any in the beginning any algorithm which is in this format where the interaction between the learner and the environment happening in this, I am going to call it as online learning algorithm, and denote it as simply by A.

So, if you have an online algorithm, what are the desirable features, what are the desirable property we expect from him from it? So before also let say the total loss or may be mistakes of algorithm A you have going to denote it as. So, over t periods, this is the number of mistakes my algorithm A made right. What you expect, what you expect what are the desirable properties of a good algorithm? You want these number of mistakes made by algorithm to be as small as possible right.

Suppose, you do not know anything about the environment right, you are a learner. If you start choosing your hypothesis in this way, is it possible this you can make this losses zero? Can you think an of a an algorithm that is going to make this total loss to be 0? Let say I forgive you, I will not count if you make a first one, can you guarantee on the second one you can make a correct one?

May be like right now I do not know right like whether even the second one I will get it correct, or third one I will get it correct, or from which point I will start getting correct. Why would I want what is all I am desiring now is the first one which is going to give me the smallest total number of loss. Now, we are going to see that what is the best we can what is the algorithms we have that gives me some bound on this like obviously, one natural bound for this is T this cannot be greater than T right, but that is an useless algorithm.

I want suppose if I can a bound here which is strictly less than T may be that is not so bad, when not I will may be I will not immediately reject. So, first we will look into algorithm what is the best bound we guess, what are the some possible bounds we can give, then we look at what is the best we can do. That means, I will not be able to bound this, this I can ensure always that this bound is going to be smaller than some quantity. It may be possible that it could be larger than that, but I will see that can I achieve that smallest bound ok.So, let us get into see how to do that.

It is allowed to, it sees maybe I should write environment selects this and x_t is revealed to the learner, x_t's sorry a learners is x_t and based on that it he is going to choose an hypothesis.

I mean you are saying if this happen like I had made all the mistakes, but you are the now you are now saying, I am you are going to make a decision in hindsight. After ensuring that I have made all the wrong decisions, I should have done something else, but I am asking you to guarantying me till n point, not like then it is same as batch setting right.

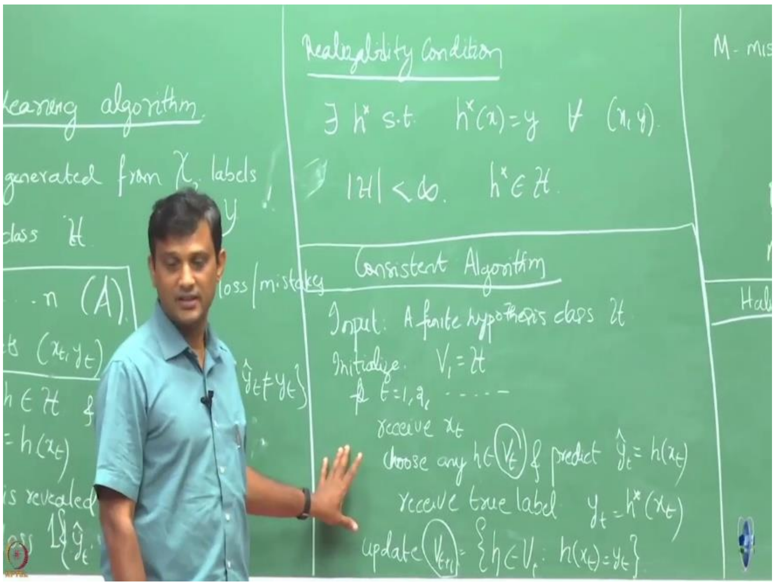
You have seen all what happened till n rounds, then you are saying what is best that is exactly the difference between batch setting and online setting. I do not have the luxury to see what all the points I got till round n, then make a decision ok.

Now, to kind of start thinking what is a good algorithm, let us try to follow the approach we did it in the supervise setting. So, when we talked about supervise setting to guarantee something on my risk or test error, I made one assumption right. What was that? In the last class, we called it by something. So, we said that realizability assumption. Let us make such an assumption here right.

Suppose, I do not have any assumption here, let us see what happens. Suppose, let us see I did not put any restriction on the way this samples and the labels are generated right. Suppose, let say you, I, so right now I am not specifying this, how this y_t is generated. Let say x_t the environment generated it gave it to you and you made a prediction. If the environment is let say it is an adversary your adversary.

Once you make this prediction, he would say the true label was just opposite of this, then you are you are always going to make a mistake right. If you are making this adversary or whatever the generation of the label very powerful, you there is no way you can do a good job here.

(Refer Slide Time: 16:01)



Realizability Condition

$\exists h^* \text{ s.t. } h^*(x) = y \ \forall (x, y)$

$|\mathcal{H}| < \infty. \ h^* \in \mathcal{H}$

Consistent Algorithm

```
Input: A finite hypothesis class $\mathcal{H}$
Initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
    receive $x_t$
    choose any $h \in V_t$ & predict $\hat{y}_t = h(x_t)$
    receive true label $y_t = h^*(x_t)$
    update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$
```

So, that is why we assume that, so we assume that this y_t label in each round, it is not like arbitrarily generated. There is a fixed rule that governs this label generation process. We do not know this (Refer Time: 16:50). If we it has been told to you, then we know which is the best hypothesis right. We just do not know this, and also we do not know how x's are generated what we are all assuming is, the labels are assigned to the sample samples in some fixed fashion.

By putting such constraint, we are restricting power of environment right; he cannot do the same thing which I said earlier. Like whatever I said $\hat{y}_t$ he cannot say the negation of this is the label because the negation of this may not satisfy this condition right. So, because of that the environment can have such adversarial role in that case ok. Let say such a reliability condition holds.

Now, can you think of some good algorithm in for this case what could be your hypothesis selection strategy? I in terms of the terminology, I may also sometimes call this hypothesis as actions ok. So, we have set of hypothesis class right. What I am doing is choosing one hypothesis, I can just think that each hypothesis is like a an action, and in each round I am picking one of the actions ok.

Student: (Refer Time: 18:12).

Let us assume that is also finite ok. So, this process this h is such that the true label generation process is coming from my class itself. So, by this I am ensuring that at least there is one hypothesis class, one hypothesis is in my hypothesis class which I happen to apply in each round, this should be 0.

It is just that I do not know which is that, I need to identify that. Now, the question is, then the question alter putting this question alternatively is how quickly I identify that hypothesis? Once I identify the hypothesis, I am no more going to make any mistakes right ok.

So, to begin this, let us start looking into some algorithms. One simple thing I can do is I will let say in some round I pick some arbitrary hypothesis, I do prediction. At the end, I got a label. Now, does it make sense once I get this I keep only those hypothesis, which is give me a label y_t on that x_t and throw everybody else.

So, if I throw everybody else whatever remains in the remaining one h^* should be there right. So, I am not losing it. So, I can do keep throwing that those guys were making mistakes on that particular y_t, then I can narrow down on my remaining ones. So, let us see how that algorithm. I am going to call it as consistent algorithm. So, this is how my algorithms is going to look like.

So, anyway now as I said once I make this realizability assumption that h^* belong to H, the problem now boils down to identify which is the right hypothesis in my hypothesis class H right. So, initially only that is given to me. My input is hypothesis class, and my object is to identify the good one there.

So, what I will do in this case is I will keep updating my hypothesis class. So, initially I will take my entire hypothesis class as my current hypothesis class. Then after I receive x_t in round t, I will just choose one hypothesis arbitrarily from my current hypothesis class.

What, so ok. So, notice this I am maintaining a new set V_t here which is getting updated in every round; and this V_t is the remaining set of hypothesis. So, in this we are eliminating hypothesis whatever remains that is going to remain in this set ok.

So, let say whatever the hypothesis is remains I have in round t, I am going to choose one hypothesis from that, and get a label as given by that hypothesis. After I do this, I will receive the true label which is generated according to this h^*. Now, what I will do, I am going to return only those hypothesis that are in V_t which are consistent with my true label right.

So, other hypothesis has gone has been thrown out. V_{t+1} cannot be bigger than V_t, because they are selecting hypothesis from that only ok. Now, the question is how much a mistakes this algorithm can make, can we say anything about it? Why is that?

Student: (Refer Time: 24:15) at least one of them may be thrown away.

Right.

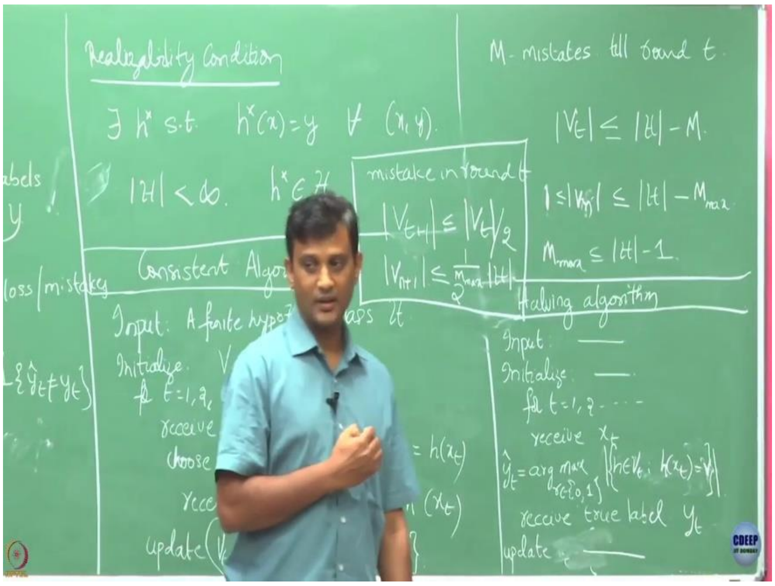
Student: Unless (Refer Time: 24:19) like finally, (Refer Time: 24:23) if there will be one thing remaining that has to be h^* (Refer Time: 24:26).

Right so.

Student: (Refer Time: 24:29).

So, can we write that formally? So, the intuition says that if there are h hypothesis, one of them is good, the remaining ones get eliminated if I make |H|-1 mistakes right ok. Let us see. No, this why is this all?

(Refer Slide Time: 24:56)



Let say in some point I have made M mistakes sometime t. I do not know what is that t, let say till some round t, M mistakes have been made ok. Now, we know that from this update logic, if a mistake is made at least the guy the hypothesis which made a mistake that will definitely going to get eliminated right.

So, my size if a mistake happens I know for sure that V_{t+1} will be smaller than V_t by at least 1. So, if right side till round t, M mistakes have happened, then what I can expect? I can write as this quantity how I can upper bound this, because M elements have been thrown out of my |H| ok.

Now, let say till V_t number of rounds, some number of whatever let say whatever let say max number of rounds, the maximum number of errors are happened. I want to bound this maximum number of errors right. And notice that this bounds here whatever writing ok, let me just call this n this bounds whatever I am writing they are independent of what is the sequence I am going to see ok, the sequence has nothing to do the way I am bounding this errors fine.

Can I say something about this also the lower bound on this in every round? It is going to be at least what 1 right, because the true hypothesis belongs to my class H, and that will never going to be eliminated right because that never going to make a mistake. So, because of that this is going to be this.

Student: Hypothesis at nth round final round.

Final round number of ones remaining. If when it remains let say if M_{max} number of errors have been this is the bound right. So, now, from this we know that max number of bounds I am going to get this that is what he also said right this is the number of mistakes ok, fine. Now, the question is can we do better than this?

So, is this is natural like at least from this 1 to if you want to improve that you would like to do that or not? Let us see. First let us write that algorithm and see why it is going to be better than if at all if it is better than this. So, we are going to call that halving algorithm. Why we are want to call halving algorithm, because in each round according to his logic if we are going to make a mistake, I am going to throw more than half of the hypothesis ok. Let me write it.

So, the algorithm is the same as this except it differs the way I am going to choose the label ok. So, let see. The input is the same; initialization is the same. Now, instead of choosing this hypothesis here, I am going to in this step directly give up label which is as follows. So, wherever I have written this bar, that means, this step is going to be the same as that step in the consistent algorithms.

So, initialization is the same. We are going to still give me the same hypothesis you are initialized with V_1 equals to H. Now, in each round, after you receive x_t, what you are going to do, can you all parse this statement here what I have written? What I am doing is I am going to do is looking at its arg max over this variable r which is taking two values 0, 1. So, 0 for label 0; and 1 for label 1.

What I am looking at here maybe I should write a set notation here. I am looking at I first step one r let say r equals to 0, I will looking for all hypothesis in V_t which are giving label this is r here value 0 on my x_t. And now I will count cardinality means I will count how many hypothesis are there with the label 0 on my point x_t. Then I will can go and do r equals 1, and I will take and I will do the same thing, and what I will get the number of hypothesis which are giving label 1 on that point x_t, and I take that cardinality.

So, this is basically if you look into the, this set is one set is all hypothesis which is giving label 0 and another set is all hypothesis which is giving label 1 on the same point x_t. And now I am looking at arg max here right; that means, just I am looking at this set which has larger value. And whatever the value they are telling maximum number of hypothesis saying I am going to take that as my prediction $\hat{y_t}$.

If the set which told 0 has the larger in number, then I will take 0 as my prediction. And then what I do, I will receive my true label y_t, then I will just do the same updation step here.

Student: if we are selected by $\hat{y_t}$ as 0, then all the hypothesis that gave one we will be rejected? (Refer Time: 35:38).

We do not know, we do not know yet whether they are going to be rejected it depends on y_t. So, right let say you have taken let for time being assume that in round t it so happened that all the hypothesis that said 0 label, they are larger in number they are in majority.

So, you gave 0 as a prediction. After that, now assume you received zeros the true label, then those guy remain the guy who said one they got kicked off. But let say it that y_t happened to be 1, then all those guys who have said 0 which were in majority they will get kicked off ok. So, it depends on finally you are going to make elimination only after you seeing the y t label right.

Student: So, sir even if the number of zeros are more right, we get y_t, do we still eliminate the larger set?.

The larger set get kicked off, because it makes sense right, because by kicking them off I am not loosing h t. What my focus is on keeping my true h t ok, just ok. Now, with these can you see when you made a mistake, are you ensuring that more than at least half of your hypothesis are getting eliminated? When you made a mistake here you could only ensure that one of the hypothesis got eliminated, but whereas this half of the bad ones got eliminated.

So, because of this if you made a mistake in some round t give me a minute just me take some space here. So, suppose let say mistake happens in round t. So, what is the relation between y_{t+1} and V_t in terms of the their size, can we say something about this?

Student: Is equals to 1 /2 (Refer Time: 36:40).

Is equals to half of this quantity right ($|V_{t+1}| \leq \frac{|V_t|}{2}$). And if I keep on iterating this where t periods what kind of bound I will get? So, over n period let say over n + 1, whenever there is a mistake is going to happen, this is going to get halved. And eventually if I keep repeating this, if let say whatever 2 to the power some max number of mistakes are happened, I will get and then I will going to get this bound. Is this correct? We gave many times that that example.

So, let say at some round t, you are left with 20 hypothesis ok. Now, an instance x_t came, and you notice that 15 of them said 1; and 5 of them said 0 ok. What will be your prediction? Your prediction will be 1. Now, after you did this prediction as 1, you saw that your true prediction is 0 right. Then what you are going to do?

Student: (Refer Time: 38:02).

You are going to kick out all the 15 ones.

Student: Can also happen that of one who is the (Refer Time: 38:08).

It (Refer Time: 38:09).

Student: Like fifteens are 1, and so it can also happen that the original label was one.

Yeah.

Student: And our current hypothesis were true, so the other five could have been kicked off.

So, in that case you did not make a mistake. What were only (Refer Time: 38:24) if mistake happens, then only this relation is true, this relation is true for every t.

Student: Only in (Refer Time: 38:31).

If mistake happens in round two this, this relation holds. If I am if the mistake did not happen in a round t, this relation is not true. So, that is why I am saying like if the maximum of number of mistakes is M_{max}, then this is the bound we are going to get. And if you just invert this, we know that this is going to be this. And what you are going to get the $M_{max} \leq \log|H|$.

So, you see that if I now go for a halving algorithm I will get significantly better bound right compared to $|H|$, I am going to get $\log|H|$ that means, this bound is exponentially better than the consistent algorithm.