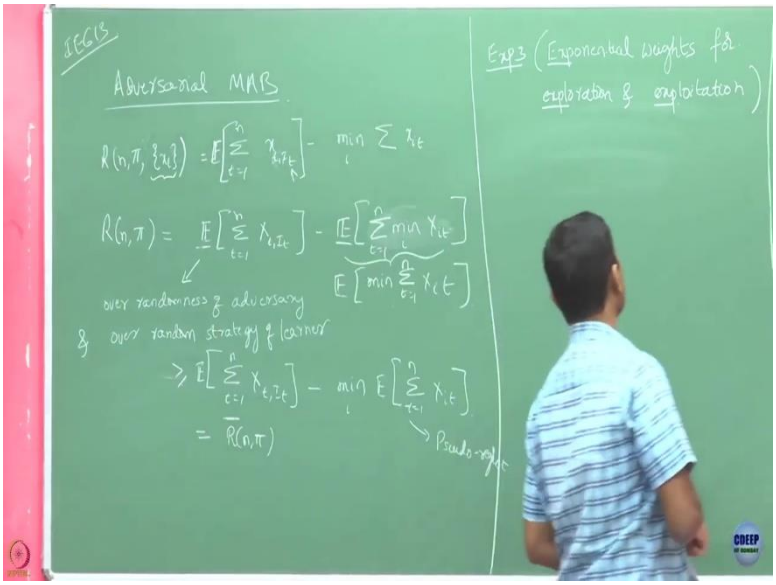
Bandit Algorithm (Online Machine Learning)
Prof. Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Lecture - 14
Exponential Weights for Exploration and Exploitation Algorithm

Now, in the last class, we started discussing about adversarial multi armed bandit right.

(Refer Slide Time: 00:29)



So, there was a slight confusion in the last class about the way we are going to define regret. So, let us revisit that part. Well, I say MAB; it is stands for multi armed bandits ok. So, we said that given n rounds and a policy π, and if I also tell you the sequence with which you do not know, but this is the sequence that you will be faced with.

We define this regret to be what? So, this is the regret where I_t is random, because the learner can randomize his choice of I_t ok. So, now, this is the regret you are going to face you are going to incur when you have played against a particular sequence that is generated by adversary, of course, you do not know the sequence a priori. But now we also said that instead of this, since we are allowing the learner to randomize this selection, I am also going to look at expected regret which is going to be this.

But when I wrote this expectation this expectation is with respect to the randomness of the learner, but why I will necessary that I am going to worry about one particular sequence that I

will be faced with. The adversary may himself be generating this sequence in a random fashion in which case I want to account for both the randomness in which the sequence is generated as well as the randomness with which learner is playing his actions right.

In that case I said ok, now I will not worry about a particular sequence that I am faced with I am going to the adversary can generate the sequence in arbitrary fashion. So, in that case, I will be interested in the expected regrets which is defined now in this fashion. So, notice that now I have allowed the adversary to randomize the sequence as well.

Now, what is this expectation with respect to, now what on the things I am averaging about on? Whereas, when I did this expectation it was with the randomness of the players strategy, but now I when I write this expectation, it involves two random quantities there. What are those? One is over randomness of adversary, and another over random strategy of learner. So, this expectation here involves expectation over these two quantities ok.

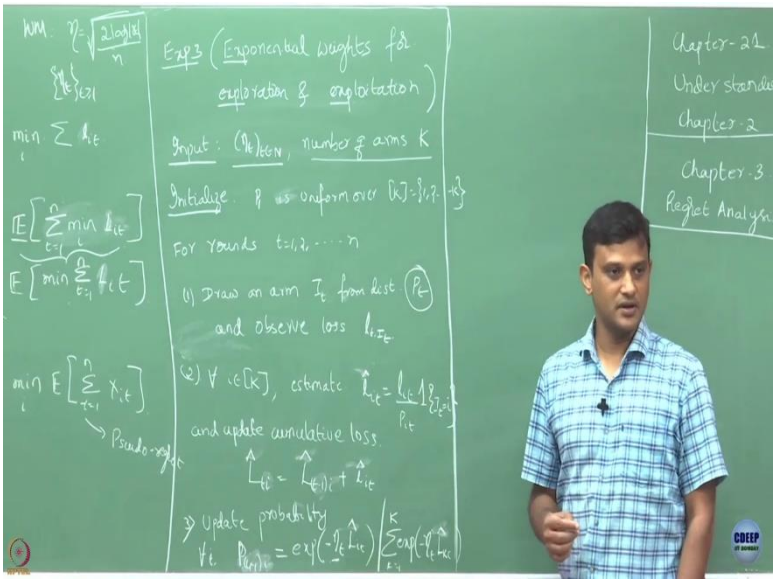
Now, looking into this further, we said as this quantity here, here this benchmark here against which I am competing, here we are asking for whatever the so I could always decide that x_{it}, and this is over this quantity here, and so here. So, if I am going to look into the minimum quantity in each round, this is going to be a too demanding task, because in every round I am looking what is the smallest round.

So, instead of that we said that I will be looking at this quantity, but here we said that this quantity is going to be what; this is a lower and upper bound, we said that this is going to be smaller than this quantity right. And here again what when I look into this reference here that is my competitor, now I am again looking at the single best action over my expected total loss I am going to incur. So, now this one we called it as pseudo regret, and I am going to denote it as with the bar here. Now, henceforth we are going to give bounds on this pseudo regret not the actual regret that I am looking at here ok.

Today, what we are going to now look at is it possible to bound this quantity. What is a good algorithm for me? When I am in an adversarial multi armed bandit setting where we are saying that in each play I am only going to observe the loss from the arm that has chosen and not for the other arms. In the last class, we briefly discussed about this notion of importance sampling right. What does it did? Yes, in this bandit setting, you are only going to observe reward of loss of the action you played, but not of the others, but we could estimate the loss of others in each round.

And we come up with an estimation strategy which is called as important sampling and we said that that estimation strategy was unbiased. So, in each round, I am going to estimate the losses of all, and I am going to pretend these estimates are the true loss that I have observed from all arms. So, then I am going to use that to update my weights ok.

(Refer Slide Time: 08:38)



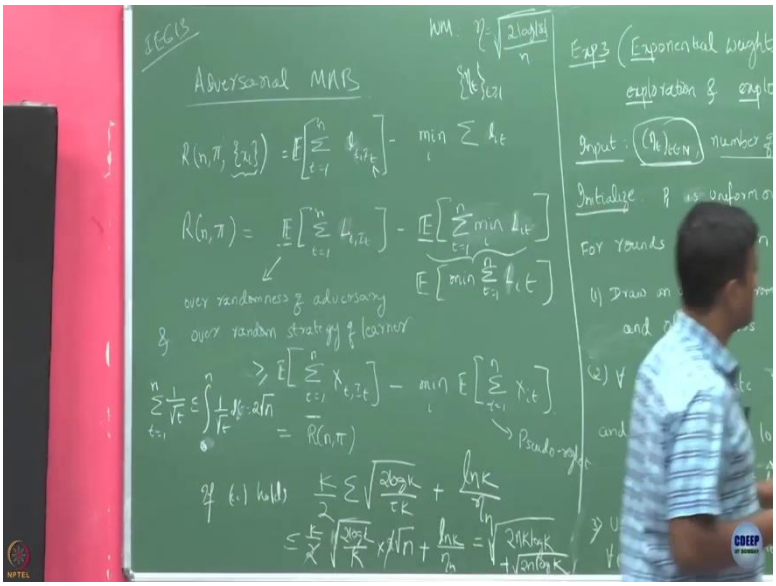
Based on that, let us write down this algorithm called it is called. And why is this called exp 3, because there are three, exp terms here, that is why this algorithm is abbreviated as its name is abbreviated as the exp 3. So, I am going to use which as even if I am going to use minimum of. So, just like this part right when I just applied expectation over here simply on both side instead of this like actually this is the expectation term I am going to incur right. And just take expectation on both side this is going to the and replace this x_i by the random quantity this is the expectation.

And now even if you interchange this expectation of minimization, you still get this lower bound. So, and we can consider either of these benchmarks, but these are difficult to handle that is why we are going to consider this lower bound as a benchmark ok. Now how does this algorithm is going to look like? Let me first write down this. So, as you see I am switching bit notations here, because I am also switching books, but let us try to be consistent with our notations.

So, here I am now using P1 here like if you recall when I was using the weighted majority algorithm, for weights I was using W's there. Now, instead of those weights with the weights

which we I finally, converted to probability, here I am also using probability, but instead of w I am going to use the notation P now. So, this is a probability vector now. And one more notation change I am going to make henceforth this, this x earlier I said this is a loss right, for loss again I now going to switch to small l ok.

(Refer Slide Time: 13:44)



So, here, so against every just the notation just to say that I am in a loss set up here like I want to minimize the total cumulated loss I am incurring ok; then observe loss l_{tIt}. ok. So, this is the whole algorithm. So, this algorithm has basically three steps. First step is you are going to play an arm according to your current distribution p_t. And then once you plan arm, then you are going to update estimates the loss of all the arms. After you update the loss of all the arms, then your third step is to update your strategy itself.

So, what is strategy here? Strategy here is to decide with what probability I am going to play each of these arms fine. So, I am saying to for this algorithm to make work, we have to tell how many arms you are dealing with, and also you have to pass on a sequence $(\eta_t)_{t\in N}$, this is defined for everything t, you have to pass you have to tell for t =1, what is this η_1; for t = 2, what is η_2 like that. So, this η_t is defined for every t. So, recall that when we are doing weighted majority algorithm for the expert predictions right, there also we had a parameter η ok. So, in the weighted majority, we had an η we set it in a specific fashion. How did we set that?

Student: $\eta = \sqrt{\left\{\frac{2\log d}{n}\right\}}$

Divided by n. What was n?

Number of rounds there. But you will see that in the assignment that we are going to see that it is not necessary that that eta has to be fixed like this. One can take an arbitrary sequence I mean some specific sequence and try to get a better bound then what we got in the weighted majority algorithm. So, η we said it is fixed, but eta has to be can be come up with a better in a better way, and that can change in every round right.

So, with that we can come up with the better, so that we are directly bringing in this concept here like a priori we are not going to fix η to be 1 value. We are just telling we will you pass to this algorithm what is the sequence of η_t . And, again recall what was that eta was doing the eta in the weighted majority was telling how much importance you are going to give to the exploration exploitation in some sense. Because, it was basically controlling how much weight you are going to give to the past observation, but in a way it was also controlling exploration exploitation.

So, this sequence will do this here. Now, this algorithm we are going to start with P_1 which is basically uniform distribution initially we do not know anything. So, we are going to put equal likelihood on each of these arms. And then the algorithm in each round keeps on updating this P_t distribution initially for t equals to 1, this is uniform distribution.

It picks an arm and going to observe whatever it take the loss for that arm, but it is going to keep estimates for all the arms for all i in k, it is going to do this estimation. We discussed this last time right. This is basically compact way of saying if whatever the arm you are going to play for that your estimate is $\frac{l_{it}}{P_{it}}$ for the arms which you did not play, this $\widetilde{l_{it}}$ is going to be 0.

And after that you also keep updating your cumulative loss. So, maybe instead of tilde, I keep I will write hence fore and this thing to indicate their estimates. Now, you are going to again for all i, you are going to update the total loss for that arm. And using this total loss we have observed so far you are going to update the probability in this fashion, and this is where your eta t is coming into picture ok.

So, what it is basically saying that it is going to take this quantity; and this is just dividing the same quantity, but after adding them over all arms ok. So, it is just this is the exponential, exponentiated cumulate this sum of arm k, here you are adding all arms. And then for each arm, you are just taking one component of this and dividing it by the sum, you know now it is

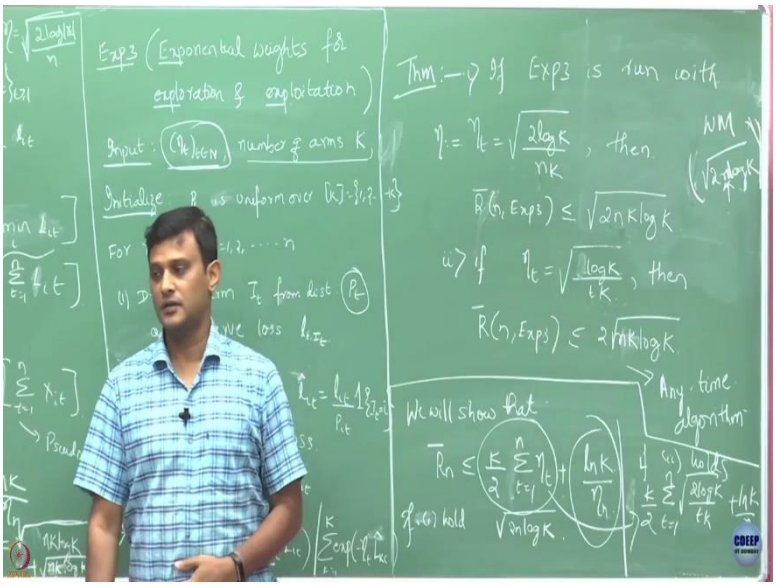
easy for you to see this forms a distribution right, because if you just add it over all as it adds up to 1, and each sum is a positive number here ok.

So, we are saying that yes I observe loss for only the arm I played. If you have picked I_t for some arm and this I_t equals to i happens that arms gets not simply l_{it} it gets a scaled value $\frac{l_{it}}{P_{it}}$. If I am going to observe loss for some arm, I am going to divide it by P_{it} and then going to take that as my estimate. But for the arms which I did not observe they will be assigned zero values ok. Even though we are being assigned zero values, we discussed last time that if you look it in expectation, it is going to be the true loss for that action ok.

Now, fine we have an algorithm like this fine we can have an algorithm anything you want now, but what is the guarantee that this algorithm has ok, so that is what we will now see. Of course, this is one particular strategy in which one algorithm is where you have specified the way you are going to come up with your distribution. To come up with the distribution, you have used a particular estimator.

Tomorrow you can go and come up with an another distribution here, and maybe that will have a different performance. But now let us say once we are going to update in this fashion, what is the performance we are going to get. Also I said this is an input to this right. So, depending on how I am going to choose the sequence, the performance can differ ok. Let us see what is the performance I am going to get; I will get rid of this part.

(Refer Slide Time: 24:04)



I am going to give, if this is first part the second part, if η_t is. So, suppose instead of this n_t I only pass on the number of arms and the number of rounds n. One can set this $\eta_t = \sqrt{\left\{\frac{2\log k}{nk}\right\}}$ for every t. So, this becomes constant in this case where it does not depend on what is the t here. So, if we are going to choose this may be let me just call it as constant here.

If you take this eta to be constant like this, then one can show, that the regret of this algorithm is upper bounded by $\sqrt{\{2nklog\ k\}}$ here ok. Now, if you do not know, so you can do this if you know n, how many rounds you are going to run it a priori. But if you do not know how many rounds you are going to run it a priori. But, you can stop it at some time and if you stop it sometime, and at that point you are going to ask the regret right.

Now, what then you do not know a priori n. So, in this case, we are you are you set in every round η_t to be like this. In round t, you know what is value of t accordingly you set that. And if you do that the regret you are going to get is $\sqrt{\{2nklog\ k\}}$. So, how much this regret is worse by than this regret $\sqrt{2}$ times ok. Now, let us; so, I said thus this regret is guaranteed to you if you a priori tell me what is the number of rounds.

And, this regret is guaranteed I am when I said is it should happen to stop at nth round you did not know a priori your nth round is the number of rounds you are going to run. If you stop at n round, this is the regret you are going to get. So, this set up here when I do this you kind of knew already the horizon how many rounds you are going to give. But when you did this, you did not know to know what is the horizon. When you do not need to know the horizon and you can give you regret bound at any time, then this is this kind of bounds are called any time.

So, sorry in this case if you are going to set η_t like this without knowing k, then we are going to call this set algorithm as any time algorithm. So, anytime algorithm basically is telling that I do not know a priori how many rounds I am going to stop, so I am I do not have the luxury to set η in this fashion. So, I am going to set I have to do without knowing n, I have to set and if that is the case then you are going to call in that setup anytime algorithm right.

So, this algorithm right now it do not need to know what is n right if you do not know n you can go and set up like this, in this case it becomes an anytime algorithm ok. So, now let us compare this algorithm with this bound what we got for the exp 3 with the bound we got for weighted majority. What is the bound we got for weighted majority $\sqrt{2n\{log\ d\}}$.?

Yeah. So, here d is there what number of experts right, I could treat them as number of arms. So, if I take this d to be k, what was the weighted majority bound it was like.

$\sqrt{2n\{log\ k\}}$. So, compared to this how much this guy is bad by what factor?

Student: (Refer Time: 30:07).

By square $\sqrt{k}$ factor right. This guy is larger than this quantity by $\sqrt{k}$ factor ok. Now, let us compare this with the amount of information that the weighted majority algorithm had and my exp 3 algorithm had. So, in the weighted majority, in every round, I get to observe the loss of all the k arms; whereas, this algorithm is working with the restricted information where it is going to use only loss of one arm in that round.

So, in terms of the information available to this algorithm, it is 1/k factor of what weighted majority has right. Is that clear? But whereas, in terms of the regret bound, it is only was by factor of $\sqrt{k}$. So now, let us see for the weighted majority, I used to I got $\sqrt{2n\{log\ k\}}$, so where I got k amount of information in every, every rounds.

But suppose let say in this weighted majority, let us pretend that to get k number of that information I have to wait like k rounds ok. So, in that case, what is basically I am to do the same amount of information in a weighted majority instead of running it n rounds, I have to run it for n*k number of rounds right. So, that is what happening here like if you replace it by n*k.

So, that that is like if you are getting only information of one arm instead of k arms in each round, it is like you are elongating your time by n*k number, and that is why that n is getting replaced by n k here. And this regret bound is worse by a factor of square $\sqrt{k}$ factor here ok. Now, to prove this algorithm, what we are going to do we will show that this bound $\overline{R_n} \leq \frac{K}{2}\sum_{\{t=1\}}^{n}\eta_t + \frac{\log K}{\eta_n}$. We will show this ok.

Now, if you go back and plug in this bound, n η equals to this bound, you will get this. If one holds you just plug in the value of η_t like this, you are going to get whatever this bound is ok. And when you are going to set η_t like this, and let say you are stopping at time at some n at which you want to measure regret, at the last time then you are going to take that last t to be n.

So, that last t will be same as this value because there t you are going to replace by n; but for the other t's, it is going to change like this ok.

Now, suppose, ok, so again to put it in a different way. Suppose, let say you are running this algorithm and you stop at nth round whatever that nth is, now what is the regret bound on that? So, in each time, you are going to be using η to be in this fashion. So, just plug in those values here ok. Now, if one holds, now let say if two holds what is this bound, this bound is going to be what? You are going to use this bond this is going to be $\overline{R_n} \leq \frac{K}{2} \sum_{\{t=1\}}^{n} \eta_t + \frac{\log K}{\eta_n}$

This is when you stopped at nth round right that is η_n this is the last round that quantity is nothing but $\sqrt{\{2nklog\ k\}}$ that you can compute. But what I am just going to see is only this factor now, the first part ok. Let me just write. Now, here if you look into this, I have $\sqrt{\left\{\frac{2log\ k}{tk}\right\}}$; in this only t is varying which is in the denominator. You can show that may be I will just write it here, whatever that bound. What is that? $\frac{K}{2}\sqrt{\left\{\frac{2log\ k}{tk}\right\}} + \frac{\log K}{\eta_n}$.

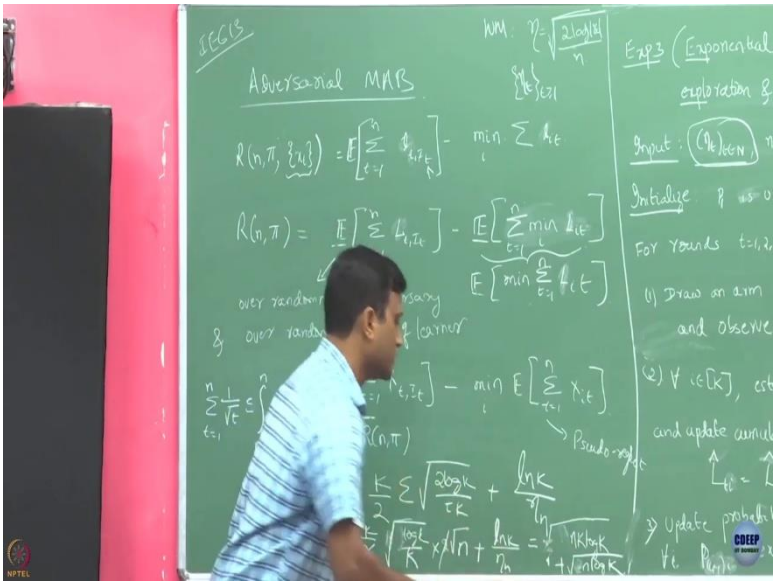
Now, we are going to use this inequality. So what, can anybody know what is a up how can I upper bound this quantity? So, I if I am going to integrate this, this is always an upper bound right; instead of summing I am just integrating for everybody possible value. And what is this integration? And I could as well take 0 also in this or may be in (Refer Time: 36:59) zeros is going to be the, what will be the upper bound for this, $2\sqrt{n}$ right. Maybe, maybe let us take a 9. So, this is going to be upper bounded by $2\sqrt{n}$. So, this will dt $2\sqrt{n}$.

Now, let us use that bound here. If you are going to use this, this is going to be k/2, everything I am going to just 2 log k by k into, I have this summation of square root of 1 by t which I will going to replace it by 2 square root n and this quantity. Now, if you are going to now everything is in the form we want, we have basically got rid of that if you just do this what you are going to get, there is a k here and there is a square root k here, you are going to get $\sqrt{2nk\{log\ k\}}$ here.

And there is also one more term, you can see that that will, but I want it 2 into square root. Just check that if you are going to simplify. What is this η_n? η_n is going to be log k/ η_n. So, it is going to be let us write that factor that is going to be plus. I am just taking with I am going to log k by this quantity, I will have square $\sqrt{nk2\{log\ k\}}$. So, with this will I get this right term.

So, there is something missing here right, some two factor is missing. Just let me write it there I did not take, ok, sorry, I missed up this is we are going to take it $\eta_t = \sqrt{\left\{\frac{\log k}{tk}\right\}}$. We are going to set it in this fashion.

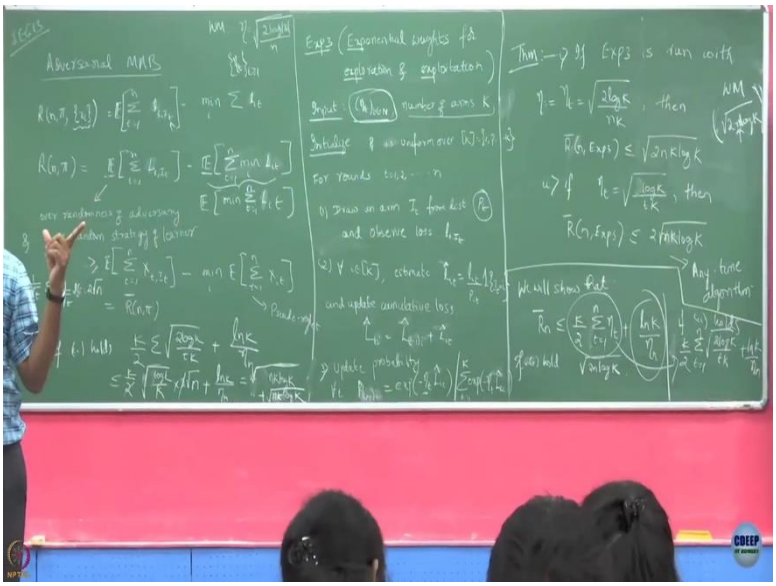
(Refer Slide Time: 39:44)



If we are going to set it in this fashion, I will not have this 2 here, I will also not have 2 here, I will not also have 2 here now, but this is exactly. So, I will end up with $\sqrt{nk\{log\ k\}}$ plus another $\sqrt{nk\{log\ k\}}$ and it is exactly 2 $\sqrt{nk\{log\ k\}}$ ok.

With that is that fine, finally, the bound I am going to get. So, fine if now we have shown that if at all I can show that the regret pseudo regret can be bounded like this now these are all true ok. Now, next let us try to say why this is true. So, this is going to be bit involved in whatever the remains in time, we will just going to write down whatever the proof steps possible.

(Refer Slide Time: 40:45)



So far any doubts in this algorithm? So, understand what is the difference between any time algorithm ok and the difference between the full information setting and the bandit setting, and how much one kind of what factor we can expect in the regret bound when we go from full information to bandit information right. So, all of these literature in bandits, they kind of study this.

It is not necessary that in every round, you are going to get only the action of the arm you are going to play, we may get something more than that. And also it is not necessary that you are going to get information about all the arms in each round, these are like two extremes right. One getting information about only the arm I play and getting information of all the actions. Even though I played one, but I am saying I am getting information, so these are two extreme.

There could be something in between also depending on how these actions are related with each other. So, people study lot of varieties of this, but right now we will be focussing on these two extreme cases – full information and the bandit. In fact, for the subsequent course we are only focussing on the bandit case, and at some point we will touch up on something in between these two.