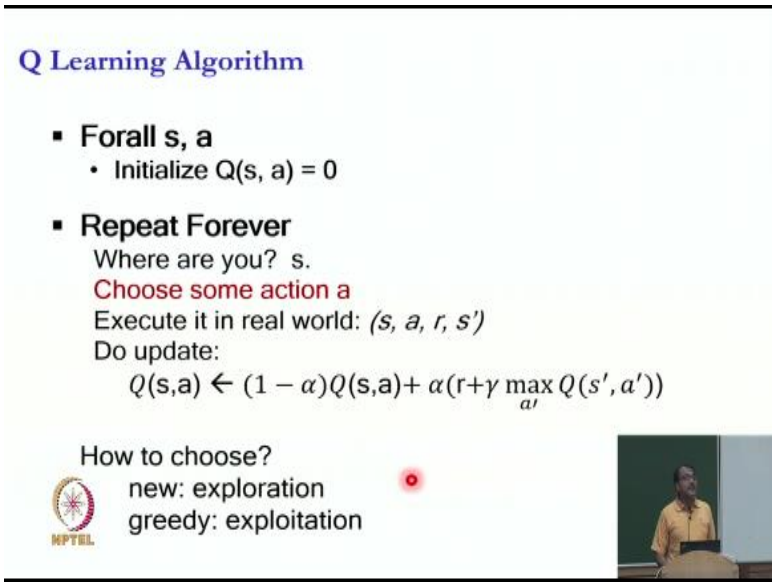
Artificial Intelligence
Prof. Mausam
Department of Computer Science and Engineering
Indian Institute of Technology-Delhi

Lecture - 82
Reinforcement Learning: Exploration vs Exploitation Tradeoff

Okay. So in the last class we talked about Q learning, right and we say that Q learning algorithm is a very simple algorithm where we are maintaining for each state the value of each action.

(Refer Slide Time: 00:32)



As we gather experience by performing actions in the environment. So the template algorithm for Q learning is that you initialize Q values to something let us say 0. And then we repeat this forever because it is an agent which learns forever. So we repeat this forever. It checks where it is in the state space.

It is in some state s. And then it chooses some action a and it executes in the real world action a in state s reaching a specific transition s prime and a specific immediate reward r. This s, a, r, s prime was is your sample. So we execute a in state s and we reach state s prime and we get some immediate reward r. And now I need to somehow back this information up so that I am able to improve my value estimate, right?

So because I have taken action a in state s I am going to improve that value estimate and my initial value estimate what it was Q s, a and my new value estimate is now r plus gamma times V of s prime. But we do not have V of s prime. We are not maintaining it explicitly. So we will maintain it by max over all a primes Q of s prime a prime, which is equivalent formulations.

And this way we were able to express the whole set of equations in the form of Q function itself. You can in fact do this for value iteration also. However, we did value iteration with the V function. You can do the value iteration with the Q function and that is what we are approximating because that is an expectation over something and then we are taking an incremental version of computing it by averaging, okay.

So this is where we were and the part that was missing for us is how do we decide which action do we execute. Okay? Now let us think about and this is the problem we also faced in model-based learning. Because in model-based RL I said that yes, I can execute a given policy but it will only explore some part of the state space. It will not even explore the other part of the state space.

So I will not learn about the other part of the state space. Therefore, I will not be able to improve my policy in the direction I want to improve it necessarily. Now there is a trade-off and this trade-off is very important to recognize. What should be our reason to take an action a? What are we trying to do when we take an action a? What are we trying to do? Ultimate goal. Divyanshu says it is to maximize the reward.

Maximize my long-term utility, maximize my value function. So our goal is to our objective for an agent is to maximize whatever it is that it is trying to maximize, the reward function right. On the other hand, there is a secondary goal for taking this action. And what is the secondary goal? It is to do not use the technical term tell me what is it that we want this action to do in the context of this particular algorithm.

We want to take every action from every state so that we can update all the Q s, a values so that in the limit we converge to the optimum, right? If we were doing model-based learning you can say that I want to learn the right model, right? So therefore there are two competing objectives here. I want to always try to find a new

action to do or an action which I have not done very often in the hope that I will probably discover something I have never discovered before.

Maybe a large reward state, maybe a transition that leads me to a good reward state and so on so forth. On the other hand, I have some information about my world right now encapsulated by the Q function and in that situation what can I do? Well, I also need to maximize my reward and I will also always try to take the greedy action the action that gives me the maximum possible reward.

(Refer Slide Time: 05:05)

Now these are fundamentally at conflict with each other or almost because the action that looks greedy to me the action that looks the best action to me is an action I know a lot about. I have explored it. I have figured out that you know this particular action leads me to good outcomes. It leads me to good reward. This is a good action to do. So therefore I want to do it in order to maximize my reward.

On the other hand, there is the alternative exploration term, which says you must take actions that may look bad to you right now but that is because you have not explored them very much and maybe if you explore them often enough, you might discover new rewards and in the long run, it may lead you to taking those actions and they become your optimal actions.

Now first question If I am an agent who has which has just born which is just starting their life, do you want more exploration or more exploitation? More exploration. If on

the other hand, it is an agent which has gathered a lot of experience in life. Do you want more exploration of more exploitation? You probably want more exploitation. Sort of done learning as much. You do not want to learn as much anymore.

Now think about you know the little babies and think about old people. We have not yet brought old people into the mix, but this is our time to bring them. Have you seen such old people who just do not want to change their ways. I am sure each one of you have come across one such member in your family. You know it took me five years to get my mother to start using a cell phone.

And another three years to start using a smart phone much later than the actual curve. Now why would she not do it? She is always she is not explainable. Absolutely she is explainable. She knows a certain way of living. She has optimized it. That is her greedy action, right. And because here has learnt the world in a overtime in a certain way now you are asking her to say oh take this exploration action.

She is like I have never done that particular kind of action it has you know whatever flaws. And you say no there are flaws, it can cause cancer but it would you know lead you to more optimized life and you know she is used to a certain life. So she says my life is optimized enough. And it takes a lot of effort. This is also true for think about the early adopters versus the late adopters versus the middle adopters, right.

iPhone comes up with iPhone 12. There will be some early adopters. Let us say we are talking only about the rich people right now, not people who cannot afford because that is not the point. Well, a new software comes a new website comes into the market or whatever it is, right. Many things come. There are always some early adopters. These early adopters are the people who are willing to do more exploration actions.

They are willing to try and learn new skills, right in the hope that that helps their life in the long run. On the other hand there are people who are sort of content with where they are. They do not know what they are missing, right. If you know what you are missing you can find it. But they do not know what they are missing.

They do not know what they do not know in some sense in that case at least and they choose not to go and expose it. To them uncertainty something that is that you have to learn new is not going to be very helpful. In other people's case something is new, it is probably going to be very helpful. So this is the fundamental trade-off. This happens every which where right?

Why are you taking the AI course? You did not know whether the AI course is going to be good or not, right. You explored it. But IIT allows you an add drop deadline. So you take it for a week and then you can drop it, but then the you can have a adversarial instructor who is very nice to you in the first week and then completely becomes like me in the weeks afterwards, right. And so then they can play with you, right?

So your exploration action was good in the beginning you thought this is my greedy action and later you recognize this is not a greedy action. A new restaurant opened up. You go there you try your favorite dish. You try whatever chicken tikka masala or paneer butter masala, whatever it is. That is your favorite dish. And it sucked. What would you do? You may never go back to that restaurant anymore.

Is that the right thing to do? We do not know it depends right because it is possible that they make everything else awesome. Just make the worst paneer butter masala in Delhi or that their cook had a fight with somebody in the morning and they were in a bad mood and they actually make great paneer butter masala or just that you had a bad outcome low probability event that you were.

So you may say now it depends you may say, oh, I have just tried this restaurant once. Let me try it again. Let me give them some chances. Let me work with them for you know 5, 6, 7, 8, 9, 10 times. Let me try paneer butter masala. Let me also try the kadai paneer. Maybe they make the better dal makhani . And then after some point you say I do not like this restaurant.

Or you could say I have gone there once. They did not give me good food. I am never going to go there again. Now who do you want to be? The first or the second or somewhere in the middle? What is the right thing to do forget who you want to be?

You can be whoever you want to be. But what is the more intelligent thing to do? The more intelligent thing to do is somewhere in the middle, right?

It is possible that if you take just one action you get a you know a low probability outcome. However, if you have taken a lot of samples, then you can be sure that your average is close to the real expectation, right? So on so forth. So I mean this is this happens to us again and again and again. I know people who love to say oh, I am learning guitar these days.

And then you ask them, you know three months later oh I am learning the drums these days. Then you ask oh I want to do bungee jumping. I am or I am preparing for the marathon. Two years later you ask them, you know these days I have picked up painting. So these are people who love exploration. Is that a wrong thing to do? No not necessarily. I mean that is life, their life.

On the other hand, you will find some people within your group who you can ask any math question or a computer science question, and they will know an answer to and you ask them anything about cricket or movies or anything that is going on in IIT as a hostel or anything they would know nothing about it. They are extremely narrow extremely focused. They are doing a very high amount of exploitation, right.

They are they are not just doing exploration. These choices become extremely complicated when they cause you to make life decisions like you are in IIT studying computer science or electrical or something like that, okay. This was one of the better things to do because you had the best rank. But do you know you like computer science? Did you know you like computer science at the time you chose computer science?

Many people did not. There might even be some people who got to see or got to work on the real computer after coming to IIT. If that is not the case in your generation that was absolutely the case in our generation, right. People had not ever seen a real computer physically in our times, right. These days there are laptops and things are easily available. And there is a you can practically call your mobile phone or computers or things like that.

They did not know what branch they want to do. They still do not know what branch they want to do. If I force you to raise your hands asking how many of you love computer science? I bet you that half of you will not raise your hands. I do not want to put you in a spot. It is okay, by the way to not like computer science not like engineering. We are in it because we came through a certain route.

Once you graduate you will now be in a position to make a call. Do I get this high paying salary job in computer science, which is what I have been trained for not necessarily because I wanted to train for it, but just because I was a good student. And do I take this job and forever do this job sitting in a you know in my cubicle doing Python Programming or whatever it is and make a lot of money for me and my family.

Or do I explore my passion and you know go start playing music with a garage band or start playing football. And so you will always have that conflict. Do I take an exploration action? Do I start working on an exploration which is my passion which is not what is exploitation today for me because this is not what I have been trained for and so on. It is possible that an and forget your passion, forget you knowing that you like football.

How many of you know that they do not like ocean marine engineering or oceanography or linguistics. How many of you do not know this? I mean know that you do not like. You do not know. It is possible that today if you leave computer science and start becoming a philosopher you might do really well, right. So there is no limit to exploration.

We cannot be sure that if we did this we will not lead to an outcome which is so much better than computer science. That it is a great idea to leave computer science and do that, right? We would never know until we try. But how many things can we try? Now the good news is you and I have some model of the world.

You and I also can say okay we will split a little bit of time doing some exploration while we split more time doing exploitation and if exploration starts to become better,

we can organically balance our time better differently and maybe later that would become my exploitation action and this will become exploration and this will happen to you.

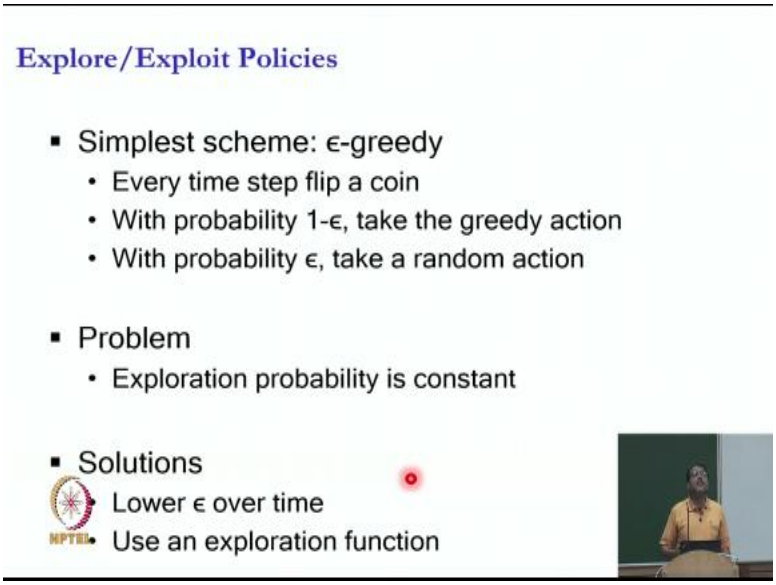
This will happen to you. Some people from here will take a computer science job leave it. Some people from here will take a finance job to make money and then leave it, they will hate it. I mean, they will hate it nevertheless. They might still do it for the money, right. And some people from you will you know go from finance to computer science, computer science to finance, computer science to starting a start-up, doing something social.

Some people may become something fundamentally different. I remember one of my friends from computer science in IIT days is now a sports manager for some sports team in England. Not bad. You are smart people, you will figure out your path. But your path will require some notion of exploration and exploitation.

You should be mindful of what it is that you are optimizing and what is your threshold between the two because those things are important and then there is the optimal threshold and optimal way of doing this which may or may not be practical because we do not have infinite time, right? We do not have infinite samples in life. We have some limited time that we are working.

Okay. So this is the fundamental trade-off that a reinforcement learning agent has to deal with. And the simplest scheme that they can do, so we are trying to figure out which action do we do and in this action, we try to balance exploration and exploitation.

(Refer Slide Time: 16:55)



And the simplest scheme we can do is Epsilon greedy, extremely simple with probability Epsilon do something completely random in the hope that tomorrow this will be the greedy action and with probability 1 minus Epsilon take the greedy action. If you do this, very high probability you will be making you know taking good actions, going, exploring. I mean getting good rewards, the better rewards.

But with some probability when you expose yourself to a new region that may open up a new possibility in the future. Now what is the problem with this approach? What is the obvious problem with this approach? Epsilon is constant. And what do we want? Slowly we want exploration to move towards exploitation. So we want Epsilon to slowly decrease. We want Epsilon to slowly decrease, right obviously.

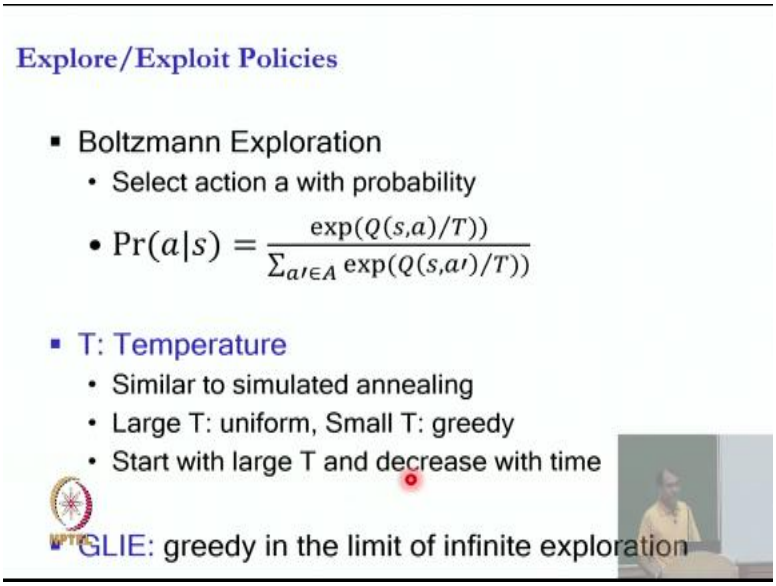
And so a simple solution could be just reduce Epsilon over time, right or a second thing that we will study called an exploration function. And the second problem is that my exploration action is completely random, uniformly random, right. When I say random I have not written which how random but let us say uniformly random. So we are not saying that one exploration action is better than the other exploration action.

So let us think about which exploration action is a better exploration action. There are two intuitions there also, two intuitions. One is the more obvious intuition. Let us say I am maintaining my Q values. So which exploration action is a better exploration

action? One which has a higher Q value. Why because higher Q value tells me that it has better reward.

So let us say I order all my actions by their Q values then the topmost action is my greedy action, but the second best action is also an important action. Maybe if I explore it a little more its Q value will increase. It will go even beyond the first action. It will become greedy. So it has a higher chance of being an important exploration action, right. It is more important. Maybe the one lower down the line I can do once in a while, not very often, right.

(Refer Slide Time: 19:22)



So I can probably start to take my actions based on just the current Q value estimates. And this equation will probably look familiar to you. So I say that I will take the probability of an action as, so I will not even do epsilon 1 minus epsilon. I will always sample from this distribution. And I will sample from this distribution using e to the power Q s, a over T as my estimate of the probability.

And of course, I will normalize it to 1 so I will sum over all possible actions for this particular thing. So the numerator is normalized to 0 and 1 and everything is a probability distribution. Now what is T? T is the temperature. Have you seen temperature refer simulated annealing right? It is the same intuition. If my temperature is too large then everything is divided by infinity.

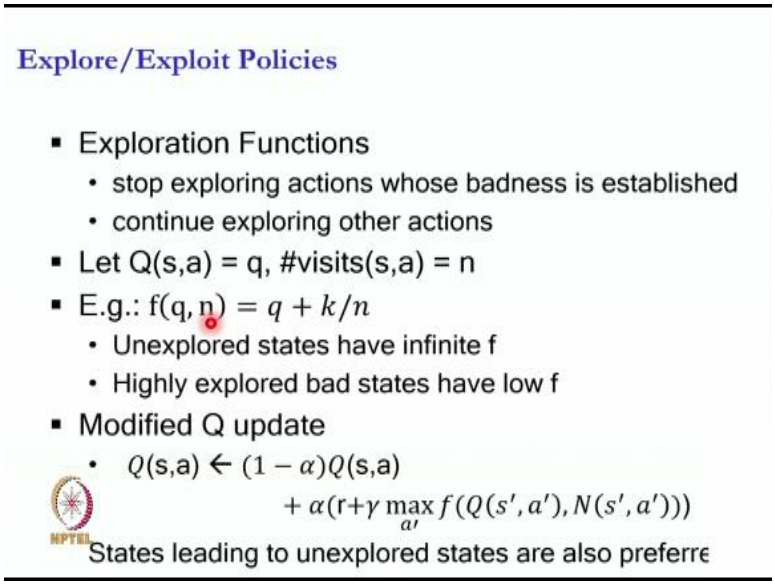
That means everything is sort of close to 0 and if everything is sort of close to 0 then it is sort of uniform. If on the other hand, I am dividing by epsilon a small number and normalizing then what is going on is that a larger number is becoming much larger and a smaller number is not matching up. And so when you normalize it then it will become very close to greedy.

So therefore I will start with a very large temperature and I will decrease the temperature with time and this is called the Boltzmann Exploration function and these kinds of exploration functions are called GLIE. Greedy in the Limit of Infinite Exploration. So when I do infinite exploration, if I keep doing it slowly my temperature will go to 0 and slowly my whole probability distribution will become completely peaked and will become greedy, okay.

The other intuition for how you should explore. What is the other intuition for how you should explore? So one intuition is the higher Q value the more important it is. What is the other intuition? There is a natural intuition. The one which is less explored is more important for exploration. Suppose in a state s I have never ever taken action a but action a 2 I have taken at least twice.

So then which is more important for exploration? Hey, independent of or let us say I have explored a once and a 2 five times then a might be a better action to do for exploration.

(Refer Slide Time: 21:59)



So how do we operationalize this? We operationalize this by what is called an exploration function. So in exploration function, we stop exploring the actions whose badness is established, but we continue exposing the other actions whose badness is not yet established or they have not been explored very much. So how do we do this? So let us say my Q function is q and let us say I have done n times of a with state s.

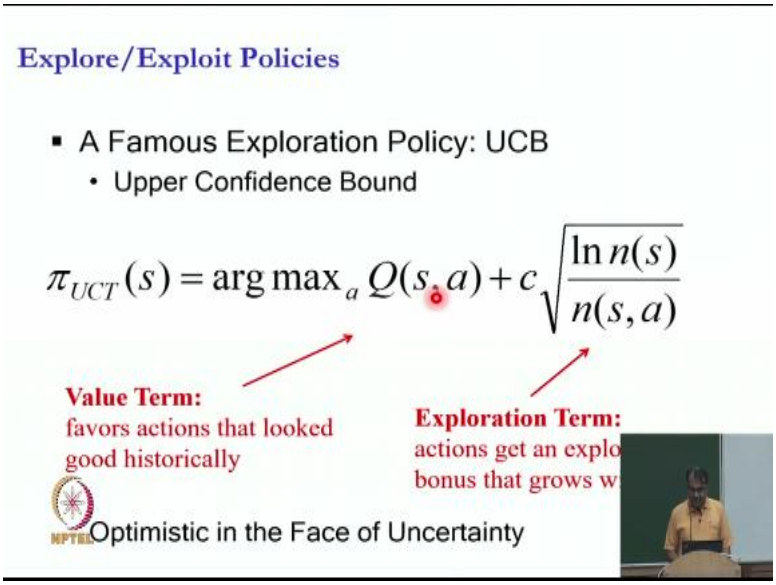
n times I have executed action a in state s. So then I can define a new function f and this new function f is like the exploration function. Let us say it is the q + k by n function. Now suppose in a state I have never taken an action a what happens? What is 0? n is 0. So what is infinite? f is infinite. So f says this action is extremely important for you. On the other hand if there is an action, which I have explored infinite times, then what happens?

It does not become 0. Its exploration term becomes 0. There is the exploitation term. So this is the exploitation term q and there is the exploration term which is how many times I have explored and reciprocal of that. So for unexplored states, they have infinite f. Highly explored bad states will have a very low f.

Now the beauty of exploration function is that when I am doing Q value update while learning not in the end while acting but while learning I will do something like r plus gamma max of not Q s prime a prime but f of Q s prime a prime n of s prime a prime. So instead of backing up the Q function, I will back up this exploration function f. What does that mean?

That means that if I have explored a state but their successes I have not explored. I might still prefer the action. So I will still prefer an action where I reach a state for which from which some of the states have not been, some of the actions have not been explored. So I am sort of giving bonus to other states for visiting or taking me to a state that is unexplored, okay?

(Refer Slide Time: 24:32)



One of the most famous exploration function is what is called the UCB function. Upper confidence bound and it was used in a very famous algorithm called the UCT algorithm. Now we will not talk get to talk about UCT in this class because there is enough to cover. But UCT was the algorithm of RL 5 years ago before AlphaGo happened.

So before AlphaGo happened in 2015 and 2016 and so on so forth what was the most important availed algorithm? It was the UCT algorithm. It is a search algorithm, but it is a search algorithm with exploration exploitation and sampling. So they are there were many bells and whistles to a search algorithm. So it was a tree search like traditional expectimax kind of an algorithm but not every branch was fully developed.

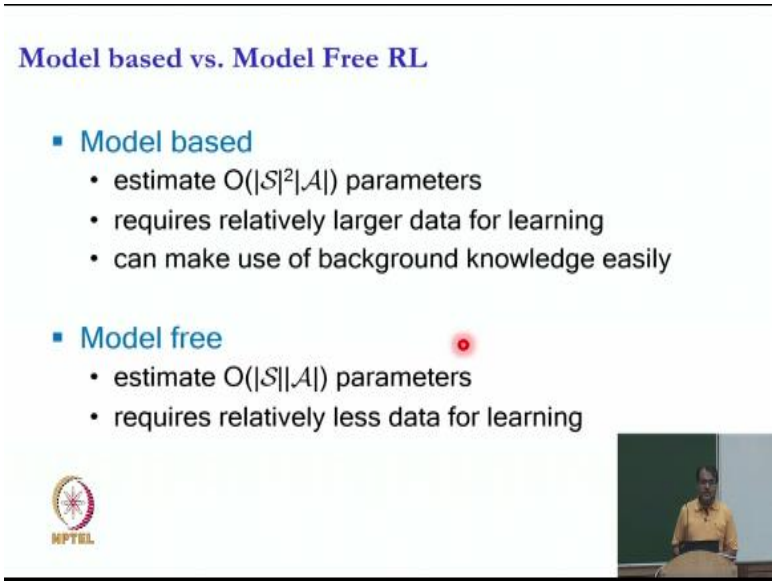
Some branches were highly exploited. Some branches were weakly explored. And in fact, even in AlphaGo this algorithm was used, okay the first version AlphaGo this algorithm was used. But we will not talk about it except to say that there they have a exploration function called the UCB function where you do arg max Q s, a plus c times log of n s divided by n of s, a.

So in other words if an action is not very highly explored then this term will still be very high, right. But we will take log of n s. So if I have explored this particular state many times then it is the state's importance reduces and that happens with the logarithm. And there is a lot of theory on why this is a good exploration function,

which we will not talk about except where to say that this particular term gives me a confidence interval bound on the value of the Q function.

So it is called optimistic in the face of uncertainty because it is a bound, right. So my Q value could be plus this or minus this. But we are saying that we will take the plus term because we it might be better. So we should try it outside. So this is being optimistic in the face of uncertainty.

(Refer Slide Time: 26:47)



So this sort of leads us to some level of completion on the topic of RL, at least the basic table based RL. So we have talked about model-based RL methods. We have talked about model-free RL methods. Both of these require exploration, exploitation, right. And we can use all these policies in whether we do model-based RL or whether we do model-free RL.

The difference is that in model-based RL I am maintaining I am learning the transition function and the reward function, which means I am learning order s square a parameters, right. Transition function is a function some s cross a cross s 2 0 1. So I am learning s square a parameters. On the other hand model-free I am learning only s, a parameters.

Because I am learning the Q function which is a function of s, a. I am not explicitly learning the transition or the reward function. Because I am learning larger number of parameters, it is relatively requires large amount of data. So it is not as sample

efficient as the model-free learning method. But both of these methods require a large number of samples.

So I should mention that RL algorithms require humongous amounts of data to train. So in normal applications, it becomes not very easy to train, okay. And then last but not the least if I have a domain designer who can tell me that oh in this state you cannot go to that state or this reward is high this reward is low. It can give me any kind of background knowledge.

Then it can easy, the domain design they can easily give me that background knowledge in a model-based setting. Because they can say, you know this reward is high or this reward is low. This transition function is zero. This transition function is 1. That requires me to model the T and r function which can only be done in a model-based setting.

Similarly, if I have learnt a particular problem and a policy for it and suddenly my transition function changes but changes a little bit, then which algorithm will be able to respond to it faster? Model-based might be able to respond to it faster because I have the full model. I have to only make small changes in the model and that will naturally lead me to the optimal policy.

Whereas model-free does not have transition function or reward function. It has memorized a policy, learnt a policy for a specific transition and reward function. So if that changes it will find very hard or very long to respond to the changes in the policy.