STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Engineering

Indian Institute of Science, Bangalore

Week 12

Lecture 44

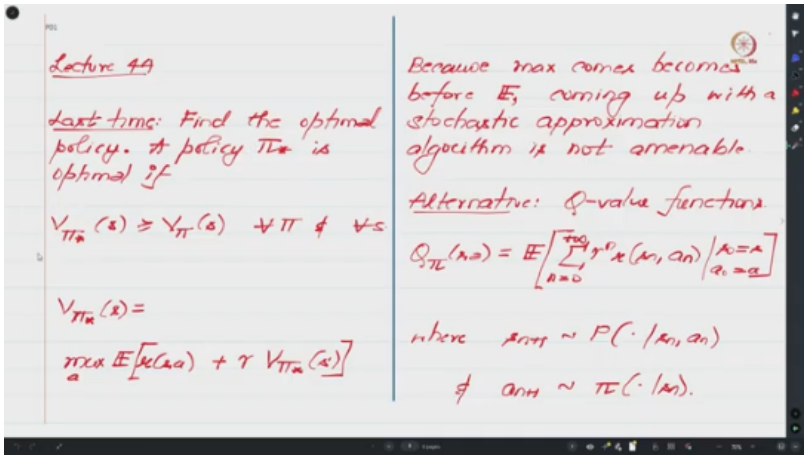
Asymptotic Analysis of Q-Learning Algorithm

Hello and Namaste everyone. Welcome to lecture 44 of this NPTEL course on stochastic approximation. In this week and also the previous week, we have been looking at various applications of stochastic approximation in the context of reinforcement learning. In the previous week, we looked at the problem of policy evaluation with function approximation, that is, given a policy, can you estimate quantitatively how good that policy is? We came up with the update rule for the TD0 method, right? And in this week, we have been looking at the problem of control, which is, can we find the optimal policy, right? Towards that, in the last lecture, we looked at

the Q-learning algorithm, right? And we said that somehow if we manage to find the value function, the Q-value function associated with the optimal policy, then we can estimate the optimal policy from this Q-value estimate itself. Of course, the challenge is that we do not know the optimal policy. Hence, we have to evaluate the optimal policy's Q-value without knowing the optimal policy. Right, and we sort of designed the Q-learning algorithm as an approach to estimate this quantity. So in today's class, we will begin our analysis of this Q-learning algorithm. Over the next couple of lectures, we will continue the analysis and eventually show that indeed the Q-learning algorithm is good enough to find Q-star, which is the Q-value function of your optimal policy. We will also see how, given the Q-value function, one can estimate the optimal policy itself.

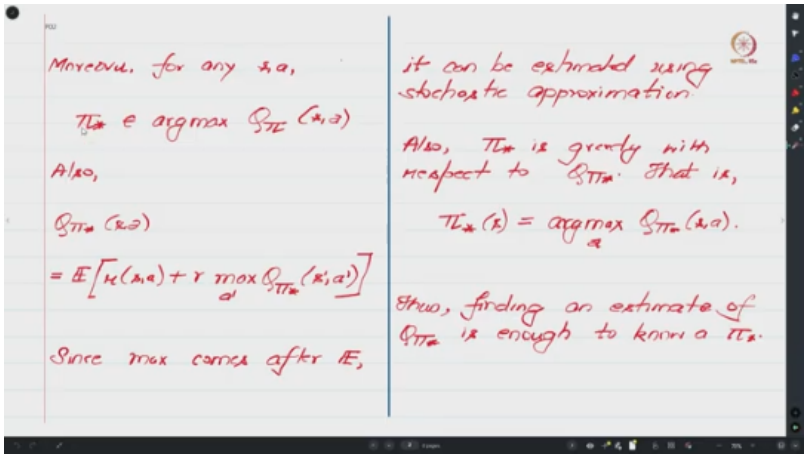
So now, let us begin the formal discussion. So as I said, the goal that we are focusing on this week is to find the optimal policy. We said that policy pi-star is optimal if V-pi-star S is greater than or equal to V-pi S for all pi and all S. Right, and we saw that the value function, the state value function associated with the optimal policy pi-star, satisfies a

relation of this form. The challenge, using this relation to find V-pi-star, was that the max was outside the expectation, right? Hence, we looked at this concept of Q-value function. So, for a generic policy pi, your Q-pi is defined to be the expected sum of discounted rewards. But the difference between V-pi and Q-pi is that the conditioning over here is different.



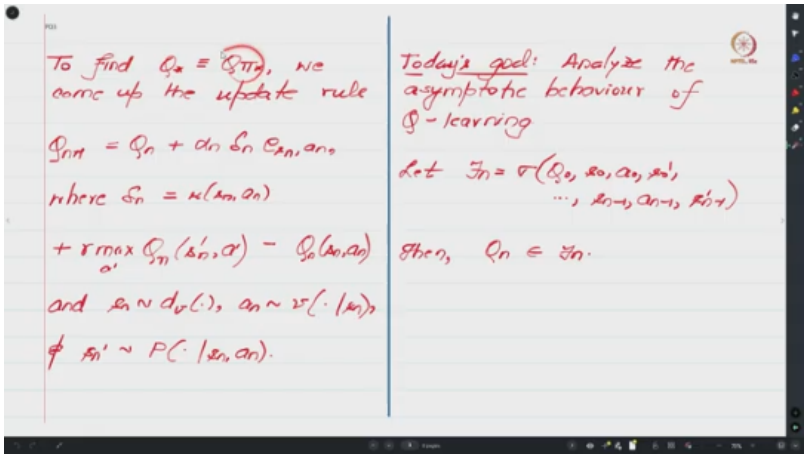
In particular, we force that the initial action be A and that need not be chosen according to your policy pi. But every subsequent action should be chosen according to your policy pi. And in this expectation we presume that Sn plus 1 is randomly chosen from this transition kernel and An plus 1 is chosen according to policy pi but this is true only from you know for any greater than equal to 0 which means that this is true only from A1, A2 and so on. The A0 action is forcefully chosen to be A which is the input that is given over here. And then we said that the nice thing about working with Q value functions is that the policy that optimizes the V value function is also something that optimizes the Q value function.

So that is what I have written here. So pi star which is optimal with respect to V pi is also optimal with respect to Q pi. In that Q pi star SA is bigger than equal to Q pi SA for all pi and all SA. Furthermore, the Q value of the optimal policy satisfies a relation of this form. And as you can see here, the expectation and max appear in the right order.



And because of this right order, we can use your stochastic approximation principles to estimate Q pi star. right and furthermore this is I believe something that I did not mention in the previous class but the nice thing about this Q pi star additionally is that pi star which is the optimal policy right that is greedy with respect to Q pi star which means if I can somehow estimate Q pi star then I can figure out what pi star is by doing this operation that is you go to every s and right? And pick the action that optimizes this value and whichever action optimizes this value, you know, you set pi star of s to be that action. If there are multiple actions, you can either pick any of those actions or you can pick maybe some distribution over these actions which maximize the q pi star values, right?

So, what this tells us is that if you can somehow find an estimate of q pi star, that is good enough to find pi star itself. So, you know in the previous class, you know we denoted Q pi star by Q star and then we came up with the update rule for the Q learning algorithm and let me recap that over here. So, the Q learning update rule is Qn plus 1 equals Qn plus some step size times delta n times Esn An and recall that this Esn An is the standard basis vector in the cardinality S times cardinality A Euclidean space. So ESA is basically the standard basis vector whose SA coordinate is 1 and all other coordinates are 0. And this delta N is known as the temporal difference error because this quantity is one approximation of your Q star.



This quantity is another estimate of Q star. In particular, this is an estimate of Q star NAN and this is another estimate of Q star SNAN and we are sort of looking at the difference between the two. And that's the reason, you know, we refer to delta n as temporal difference. Right. And, you know, for simplicity, we had presumed that this SN is sampled from the stationary distribution associated with some behavior policy nu.

And this AN over here is chosen according to this behavior policy itself. So behavior policy is a policy with which you interact with the environment, collect data and using that data you try to learn the optimal policy itself. So that is the idea of this behavior policy. And this Sn prime over here is generated according to your transition kernel. So at every time step n, you will pick Sn, An, Sn prime in this fashion.

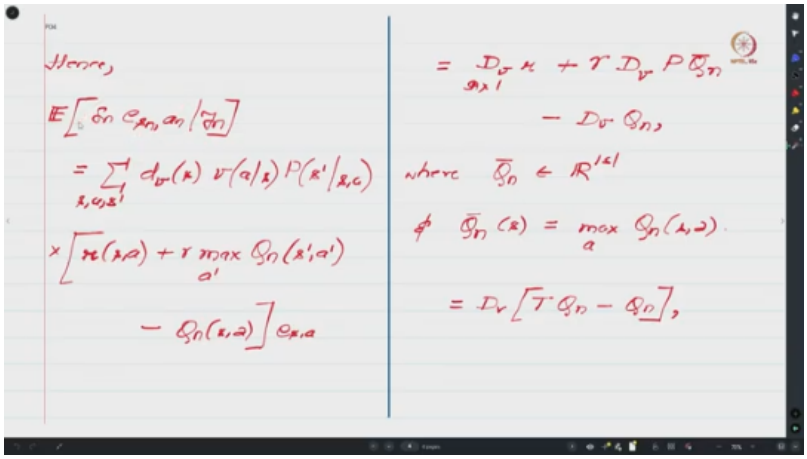
And as I told you, you know, in practice, we may not want to do this. We may want to, you know. you know start at Sn go to An and whatever next state you get that is Sn prime that you will think of it as the current state and you know then sample another action from that Sn prime state and continue going on. So the difference between whatever I said now and whatever is written over here is that at every time step n your state Sn is sampled afresh from this distribution. The reason why we limit ourselves to this

distribution is to keep our analysis simple right and as I told you in a practical way this you know sampling from this distribution can be enabled by you know running your Markov chain under your behavior policy mu to evolve for some time right. you know, whatever state you get at the end, that will roughly be distributed in this way. However, we can also analyze the more complicated version, but wherein, you know, you sort of

take Sn prime to be the current state of time step n plus 1 and continue your you know algorithm from there but that will introduce what is known as a Markov noise which we have not discussed in this course but you know once you understand this concept of Markov noise we can you know also analyze that algorithm right. But let us keep things simple in this course so we are presuming that SN is sampled in this fashion.

And the goal was or you know the goal that we started off in the previous lecture and which will be continuing through the next couple of lectures is to understand the asymptotic behavior of this algorithm. That is as n goes to infinity what happens to qn and the claim is that under certain conditions qn will actually converge to q star and now we are going to see how to validate that statement. Now, to understand the asymptotic behavior, what we do is we first define the information that we have at time instance n in the form of a sigma field. So, this is the sigma field that is generated by all these quantities. It is generated by Q0, S0, A0, S0 prime, S1, A1, S1 prime and all the way up till Sn-1, An-1, Sn-1 prime.

And one can see that if you know all these values, you would know the value of Qn, and that is the intuitive justification for this claim. But one can also formally show that since Qn is a function of all these quantities, Qn is actually measurable with respect to this sigma field Fn. And hence, if you take the conditional expectation of delta n times E SN AN, right? Which is what you have over here, which is the expression that you have in the Q-learning algorithm. So, if you take the conditional expectation of this, you will treat your QN as a constant, right?

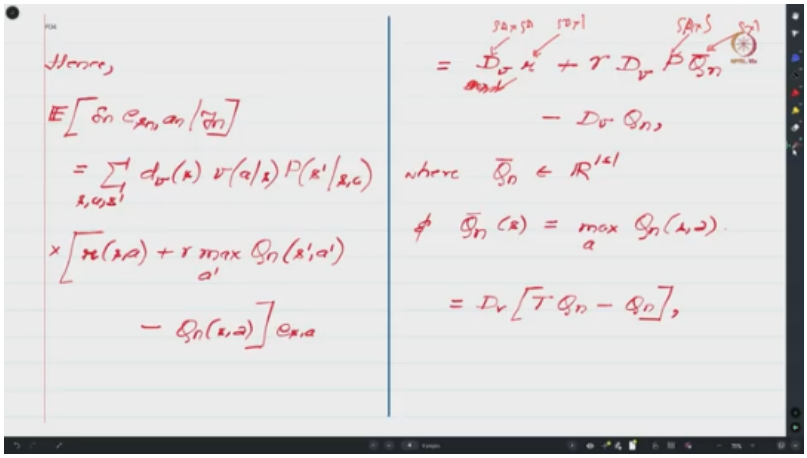


And then you would get this term over here, right? So, this is the probability of choosing your state S, right? This is the probability of choosing your action, and this is the probability of choosing your next state. So, you take their products, and then you sum over S, A, S prime, and then you substitute the different values. So, instead of S N A N, you substitute S comma A, and instead of S N plus S N prime, you substitute S prime, and then you take the max over here and then subtract it with Q N S A and multiply it with E times S A, right.

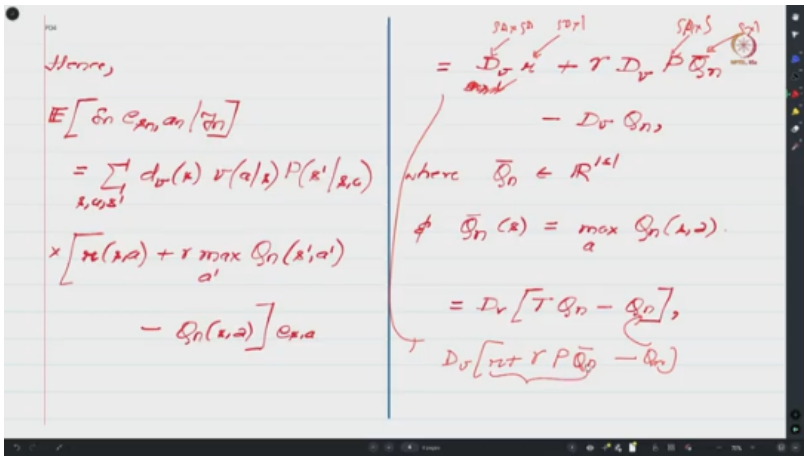
So, this is the conditional expectation of delta N E S N A N. And in compact form, one can see that this whole expression can be written as a diagonal matrix. So, let me just erase this part. So, a diagonal matrix D nu times R plus gamma times D nu, and you can see that here there is no S prime, and hence there is no P here. But here you have S prime, and hence this expression times this one can compactly write it as D nu times P times Q bar n. I will tell you what this Q bar n is soon, right?

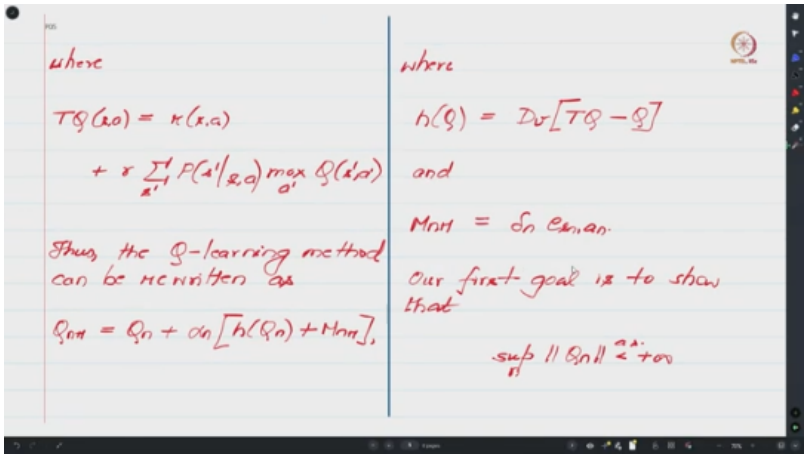
This expression times this and this again, you do not have any S prime here, and hence this times this can be compactly written as D nu Qn. So now, let me explain the different notations. So, D nu over here is the SA cross SA matrix. R is your SA cross 1 vector, and P here is basically your SA cross SA. Matrix and your Q bar n is your S cross 1 vector.

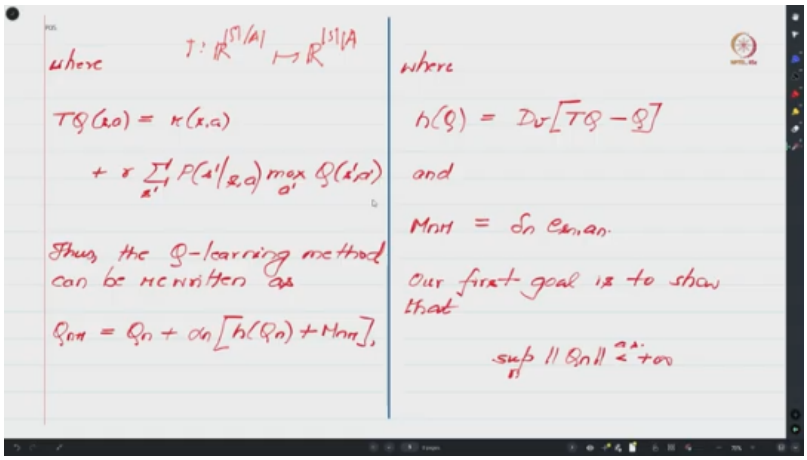
So, I will explain what these things are. So, Q bar n in particular, if you look at the asset coordinate of you know Q bar n right, this will basically be max of Q n SA. So, this is the interpretation of Q bar n. So, Q bar n is basically an S-dimensional vector—I mean, sorry, cardinality of state space-dimensional vector—and the little S-th coordinate of Q bar is basically the max over A Qn of SA. Now, this quantity over here, if you pull D nu outside right.



So, you will end up with D mu times R plus gamma P Q bar n minus Q n. So, this expression I have written as it is, and if you go back and look at the definition of your Bellman operator, one can see that the definition of the Bellman operator is indeed this expression, and hence wherever you have this, you can replace it with T Q n. Is this okay? So, in this way, whatever is the update rule that you have over here, we can write it in this fashion, alright? And you know this T operator is defined over here. So, you know T is basically your operator which goes from RSA to RS. SA right, and if you give Q as input, TQ will be a vector in this space, and if you look at the SA coordinate of this, it will basically be RSA plus gamma times sum over S prime P S prime given SA times max over A prime Q of S prime A prime. So one can see that this is precisely—so if you sort of put it all together, one can see that TQ is precisely R plus gamma.

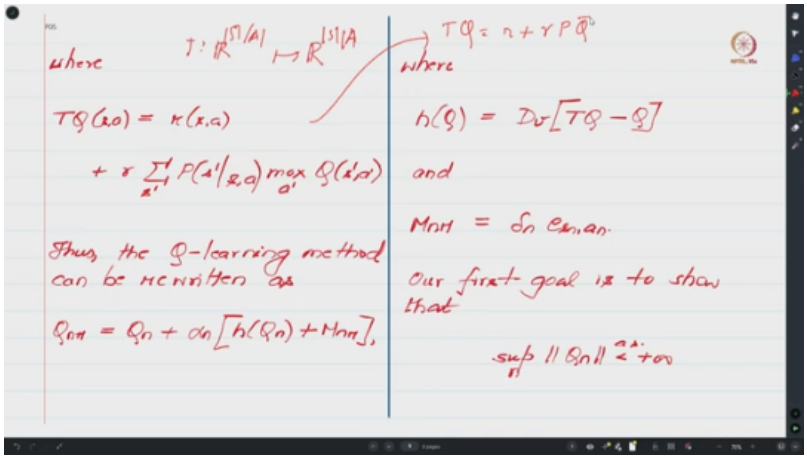




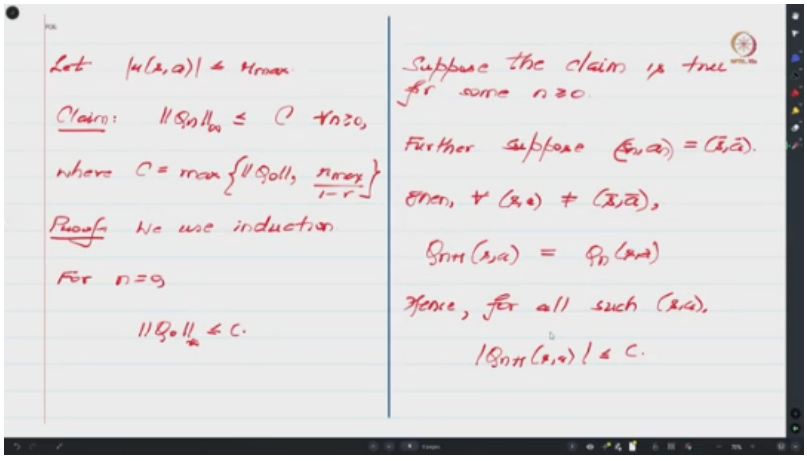


p q bar q okay bar q sorry q bar okay q bar so this is what we have right and hence the q learning method can be written in the following way that is qn plus 1 is qn plus alpha n times h of qn plus mn plus 1 where h of qn is this expression right and mn plus 1 is basically delta n e s n a n minus H of Qn right and again one can see that if you define Mn plus 1 in this fashion if you take the conditional expectation of Mn plus 1 then one can see that that conditional expectation is 0 which makes this Mn plus 1 as a martingale difference noise okay so sort of that mirrors the analysis that we have right and we have this H of Q which is defined in this fashion. So, now our goal would be can you somehow look at the limiting ODE that is associated with the Q learning algorithm and see if we can say something about the behavior of the solution trajectories of this limiting ODE. And in today's class what we will do is we will instead focus on an alternative goal which is to first show that the iterates generated by your Q learning algorithm are almost surely bounded. So if you recall you know in one of the discussions that we had during the

convergence analysis of general stochastic approximation algorithms was to you know show that



You know the iterates remain almost surely bounded so that the limiting behavior of your stochastic approximation iterates mirror those of the solution trajectories of your limiting ODE. Now that was one of the assumptions that we had made and in today's class or this class what we will do is we will establish this condition and in the next class we will focus on the you know asymptotic or the behaviour of the limiting ODE that is associated with your Q-learning algorithm that is given over here. So, let us see how we can establish this. So, what we will do is we will presume that the rewards are bounded, right?



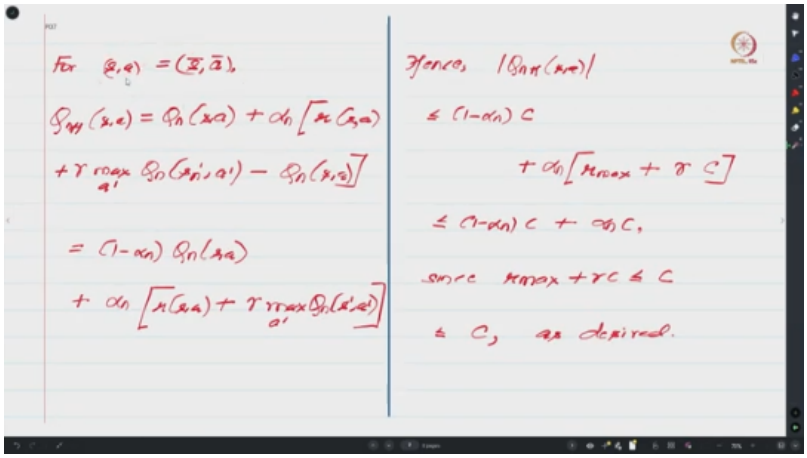
So, we will presume that there is a value R max which is an upper bound to the absolute value of R SA for every SA, right? And our claim is that the L infinity norm of QN is

upper bounded by C for all n greater than 0 where C is defined to be the max between the L infinity norm of Q0 and R max over 1 minus gamma. And to prove this claim, we are going to make use of induction. So, let us see how we can prove this claim.

So, for n equals 0, observe that here there is a max between the L-infinity norm of Q0 and R max over 1 minus gamma. So, since C is a max between the L-infinity norm of Q0 and this quantity, it trivially follows that the L-infinity norm of Q0 is upper-bounded by C. Now, suppose this claim that we have made over here is true for some n greater than 0. And as I said, we are going to use induction, and since we have made this hypothesis and proven the base case, the goal now is to show that the claim is also true for the n plus first term in the sequence of qn's. That is, we are going to now show the L-infinity norm of qn plus 1 is also upper-bounded by C. So, towards that, let us suppose that your SN AN takes this specific value S bar A bar.

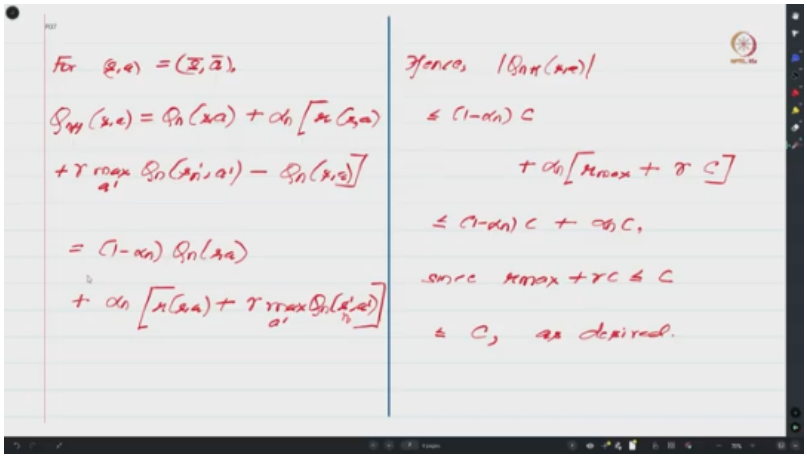
So, SN AN is actually chosen in a random fashion. So, let us say, you know, when you draw SN AN, it turns out to be S bar A bar, right? And because of the presence of this ESN AN, I hope you can see that only the S bar A bar coordinate of QN plus 1 will differ from the same coordinate of QN, while every other coordinate will— of Qn plus 1 will equal the corresponding coordinate in Qn. So, more formally, because SnAn is S bar A bar, for every Sa, which is not equal to S bar A bar, we have that Qn plus 1 Sa equals SnAn. Qn SA, right? And hence, for all such SA—by that, I mean all those SA's which do not equal S bar A bar—right, your absolute value of Qn plus 1 SA will be equal to the absolute value of Qn SA.

And we already know that the claim is true for some n, and hence the L-infinity norm of Qn is upper-bounded by C, which implies that the absolute value of Qn plus 1 Sa is less than or equal to C. So, what we have managed to show is that for every SA which is not equal to S bar A bar, it is trivially true that the absolute value of Qn plus 1 is less than or equal to C. So now, what remains is: what is the absolute value of Qn plus 1 SA for SA equals S bar A bar? So, this is the condition we will now look at. So, in this case, one can see that the SA-th coordinate of Qn plus 1 will equal this expression. So, notice that here you do not have ESA because you are only looking at the SA-th coordinate of Qn plus 1.



So, one can see that the SA-th coordinate of Qn plus 1 equals the SA-th coordinate of Qn plus this step size RSA plus gamma times the max of this quantity minus QnSA. And one can take this expression and combine it with this. So, we would end up with 1 minus alpha n times QnSA plus alpha n times the square bracket where you have RSA plus gamma times the max over A prime QnS prime A prime. So, I should write this as Sn prime. So, let me just write this also as Sn prime.

And now if you take absolute values on both sides, so this is the absolute value of what you have over here, and then here 1 minus alpha n, we will presume that your alpha n is less than 1. So, 1 minus alpha n will come out as it is, and if I take the absolute value here, From our induction hypothesis, it follows that this absolute value will be upper bounded by C. That is what I have written here. And since alpha n is between 0 and 1, I have my alpha n as it is. And here, if I take the absolute value, because we have presumed that all the rewards are upper bounded by R max, I end up with R max over here.

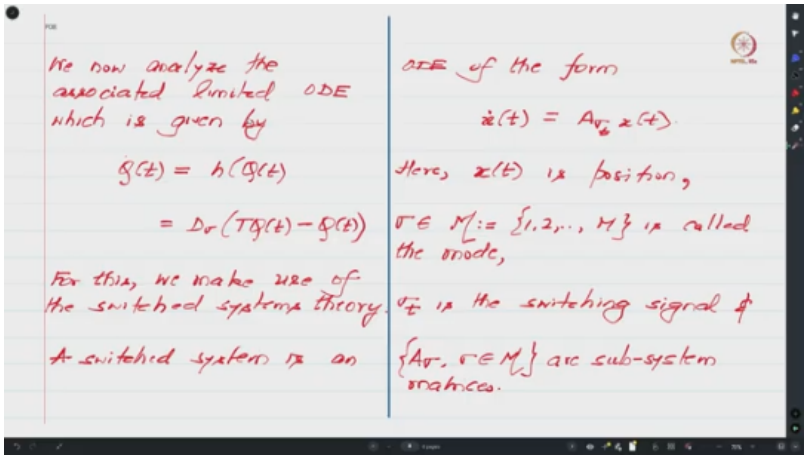


And then gamma times the absolute value of this, and one can show that because these quantities are upper bounded by C—I mean, the absolute values of these quantities are upper bounded by C—the max of this is actually upper bounded by C as well. Hence, one can see that the absolute value of Qn plus 1 Sa is upper bounded by 1 minus alpha n times C plus alpha n times R max plus gamma C, right? And, you know, one can then see that from the definition of C, in particular, we had shown that the C is the max of this quantity and this quantity, so from the fact that C is greater than or equal to R max over 1 minus gamma, one can see that, you know, C will satisfy some relation like this. I mean, this is precisely saying that C is greater than R max over 1 minus gamma, right?

So, if you use this relation and substitute over here, one can see that this expression will be upper-bounded by C. And since you have 1 minus alpha and C plus alpha and C, this quantity will be equal to C. This is the quantity, or this is the claim that we wanted to establish. So, the L-infinity norm means that the L-infinity norm being less than or equal to C is equivalent to showing that the absolute value of every coordinate is less than or equal to C, which is what we have established over here. From this, one can see that the claim holds. So, what was the claim? The claim was that, for every n, the L-infinity norm of Qn is upper-bounded by C, where C is this fixed quantity. So, one can see that at least in the Q-learning algorithm, where we are not using function approximation, one can see that the iterates can be very easily shown to be upper-bounded.

So, notice that in establishing this result, we did not have to go to the scaled ODE or things like that, right? So, in different contexts, one can perhaps use different properties

of the algorithm to establish that the iterates are almost surely bounded. So, one can see that in the Q-learning algorithm, without actually having to go through this limiting ODE, we have directly managed to show that the iterates are almost surely bounded. So, now it becomes clear that because the iterates are almost surely bounded, we can



So, conclude that the behavior of your Q-learning algorithm will be dictated by the behavior of the solution trajectories of the limiting ODE. And in the next class, we are going to formally discuss the behavior of the limiting ODE's solution trajectories. In particular, we will show that if you consider the limiting ODE, then every solution trajectory of this limiting ODE almost surely converges to Q-star, which is basically the Q-value function of your optimal policy. Hope you will join me in the next class. Until then, goodbye, thank you, and namaste.