STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Engineering

Indian Institute of Science, Bangalore

Week 13

Lecture 49

Analysis of Limiting Dynamics in Q Learning with Function Approximation

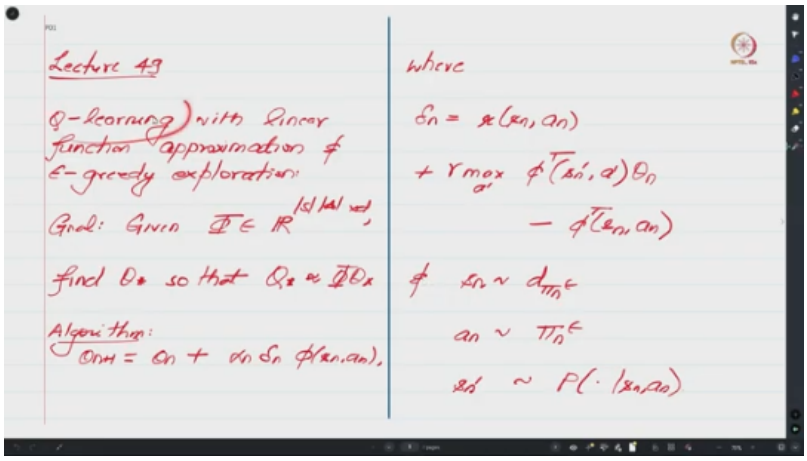
Hello and Namaste everyone. Welcome to lecture 49 of this course on stochastic approximation. So, let us do a quick recap of what we are doing so far. We are trying to use our tools that we have studied under stochastic approximation theory to understand the behavior of Q learning with linear function approximation and epsilon greedy exploration. This algorithm actually shows very interesting behaviors.

In particular, we saw that when you run this algorithm for this special case of a two state, two action MDP with two dimensional linear function approximation that we had chosen, we saw that the behavior of this algorithm is different on different runs. In particular, the iterates can potentially converge to different places, and furthermore, you know, along some runs we saw that the greedy policy that is associated with the phi theta n iterates does not necessarily stabilize, right? And this is what is known as policy oscillation in literature, and one would like to know why do we see this policy oscillation and is this good, bad, and so on and so forth, right? So, towards that, you know, we began our formal analysis in the previous class.

And we saw that the behavior of the Q learning with linear function approximation and epsilon greedy exploration can be viewed as if it is being driven by a piecewise linear function. That is, we define these greedy regions associated with different deterministic policies. And we saw that whenever your phi theta n lies in those greedy regions, then we have a specific linear function that governs the dynamics. But once you move from this greedy region to the other, the dynamics discontinuously changes, and then you have a different linear function that governs the behavior. And in this class, we will see what are

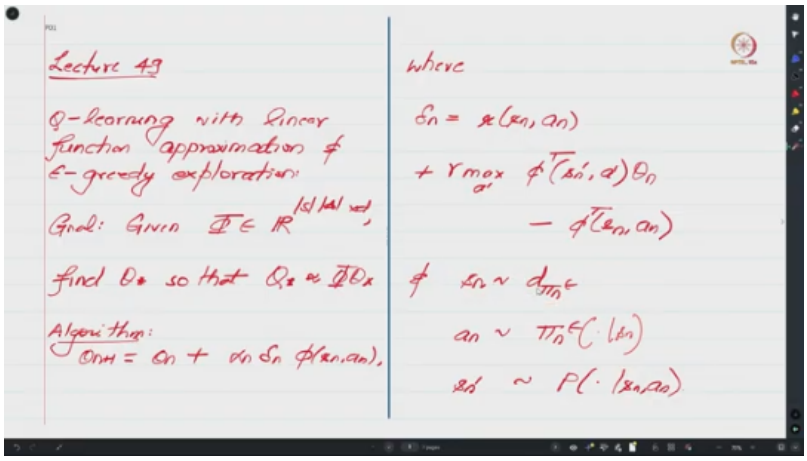
the consequences of this discontinuously changing linear function on the behavior of your Q learning with linear function approximation and epsilon greedy exploration algorithm.

With that in mind, let us begin our formal analysis. So, as I said, our goal is to study Q learning with linear function approximation and epsilon greedy exploration. And just to give you a quick recap, you know, in this problem, the goal is that we have been given some matrix phi, right, whose number of rows equals the product of the states and actions and number of columns equals D. And the goal is, given this matrix phi, can we find a theta star such that Q star, which is the Q value function associated with the optimal policy, is approximately equal to phi theta star? So, one can see that phi theta star is a vector that lies in the column space of phi.

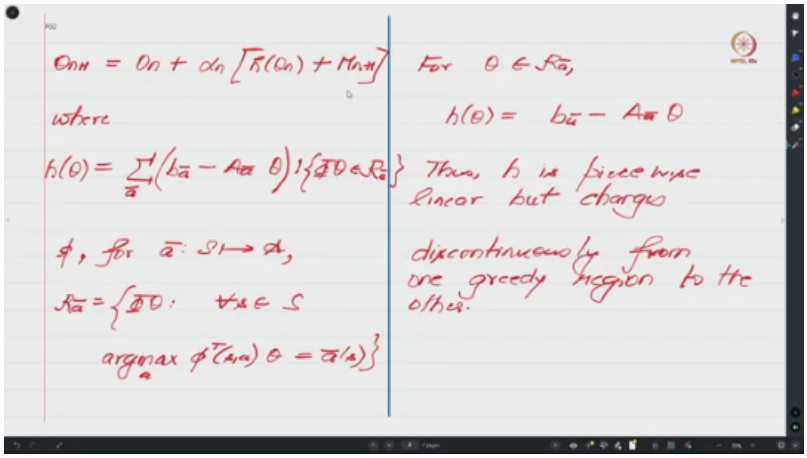


So, in this way, one can see that we are searching for a vector in the column space of phi, which is D dimensional, and hence D. If D is small, we are searching in a smaller dimensional space for an approximation to Q star. So, in this way, we are trying to handle the problem of large states and actions, and you know, the Q learning with linear function approximation with an epsilon greedy exploration algorithm has the update rule that is given over here, where the delta n is your TD error and is defined as shown here. And in particular, this algorithm differs from your TD(0) algorithm, which we used for policy evaluation, in the way in which S N, A N, S N prime are sampled. In particular, in this case, your S n is sampled according to the stationary distribution associated with the current epsilon greedy policy.

I say current because pi n is basically the policy that is greedy with respect to phi theta n, and pi n epsilon is the epsilon greedy version of that policy, and d over here is the stationary distribution of the Markov chain that is induced by this policy over here. A n is to be chosen according to this pi n epsilon policy conditioned on S n. So, I should say that A n is sampled according to this pi n epsilon conditioned on S n, and this S n prime that is over here, this is sampled according to your transition kernel. So, S N is chosen according to your stationary distribution associated with your pi n epsilon policy, A N is chosen according to your pi n epsilon conditioned on S N, and S N prime is sampled from your transition kernel given the current state is S N and the current action is A N. And I would again like to highlight that pi n epsilon is the epsilon greedy policy that is, with 1 minus epsilon probability, you act according to the policy that is greedy with respect to phi theta n, and with epsilon probability, you pick actions arbitrarily.



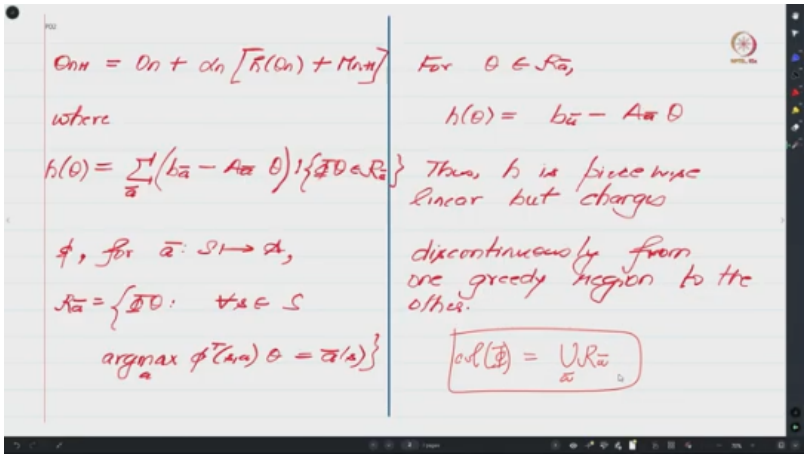
So that is the definition of pi n epsilon. And in the previous class, we saw that the Q learning with linear function approximation and epsilon greedy exploration update rule can be written in the following way where your H of theta is given as shown here. So, you have some linear function. That changes depending on what the value of theta is. In particular, this choice of linear function changes depending on, you know, in which greedy region does phi theta live.



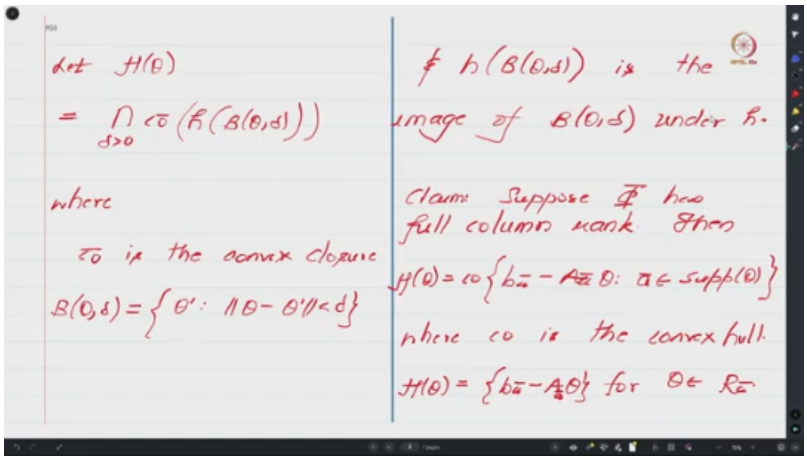
And if you remember from the previous class, I had told you that if you take the greedy regions associated with different deterministic policies, right, if you take their union, it will form the whole S A dimensional space, right, and It will form the whole S A dimensional space and depending on which greedy region you live in, the linear dynamics will change accordingly. And if you remember, R A bar is defined in the following way. It is the collection of all those phi thetas. So these are vectors within the column space of phi whose associated greedy policy is A bar.

So, for every deterministic policy A bar, you will have an associated greedy region and one can show that I mean I should be careful I think inadvertently said that the greedy region if you take their union it will go to RSA. I think the right way to say is that the greedy region if you take their union it will form the whole column space of V. I think I should be careful about that. So, one needs to say that column space of V. can be broken down into union of RA bar. I think this is what I wanted to highlight.

Sorry about that. That was a mistake from my end. So, your column space of phi can be written as union of A bar RA bar. And one can see that when theta lies within RA bar, H of theta is precisely this. I mean this notation and what I have written here mean the same thing.



That is, whenever theta lies in RA bar, H of theta is BA bar minus AA bar theta. So, in this way H is piecewise linear, but the definition of H changes discontinuously when you move from one greedy region to the other. So, if you move from RA bar to, let us say, RA bar prime, the linear function will change from BA bar AA bar theta to BA bar prime and AA bar prime theta, which could be discontinuous at the boundary between RA bar and RA bar prime. So, that is what is the challenge in understanding the behavior of this Q learning with linear function approximation and epsilon greedy exploration algorithm. Now, because of this, you know, discontinuous nature, the analysis of this algorithm becomes a bit challenging.



So, what we will do is we will define a set valued map. So, H of theta. Capital H takes as input a d dimensional vector and spits out a subset of Rd. And what is this subset? It is

defined as the intersection of the convex closure of the image of B theta delta under H. So let me try to explain the different terms over here.

So delta is some scalar and you consider all possible values of delta which are strictly bigger than 0. CO bar is the convex closure. So whatever your set is, you identify the convex closure associated with it. Basically, the convex closure ensures that if you take any two points, the line segment joining those two points will actually sit within that set itself. That is what convex means, and closure means that the set actually includes all its limit points.

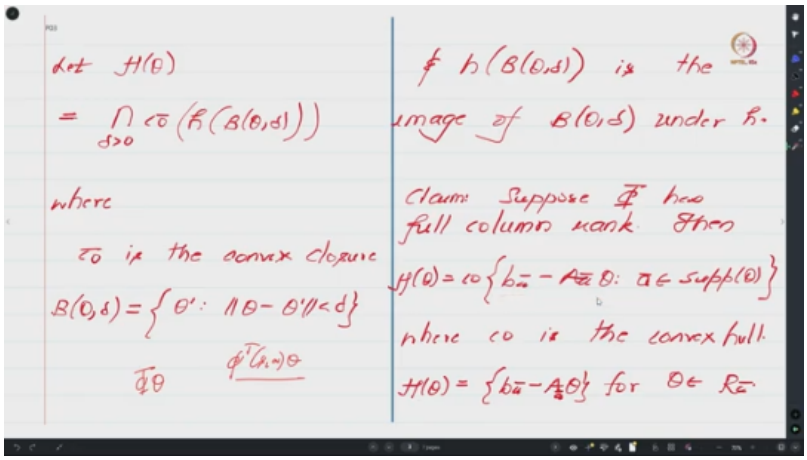
So, whatever this set is, you try to take its convex closure, and what is this expression over here? For that, first let us understand what B theta delta is. So, B theta delta is the ball of radius delta around theta. So, B theta delta is the ball of radius delta around theta. In other words, it is the collection of all theta prime such that theta minus theta prime is less than delta.

So you collect this set of points, and this one can see is the ball of radius delta centered at theta, and you take the image of this ball under H. So by image I mean that you take every point over here, that is every theta prime, apply H, and you will get a vector. So you collect all those vectors, and that is what this set H of B theta delta is, and so once you get all those vectors, you take their convex closure, so that will be another set, and then you take the intersection of all such convex closures for different values of delta to get this capital H of theta. And one can show that if your phi matrix has full column rank, then your capital H of theta, in fact, is the convex hull of vectors of this form where A bar lies in the support of theta.

So let me explain what I mean by that. So whenever theta, for any theta, if you consider this vector phi theta that is associated with theta. In certain circumstances, one can see that, for a fixed S, the action that maximizes phi SA transpose theta is phi transpose SA theta. So for a fixed S, there will be some choices of phi theta where if you try to find the maximizing action, there will be a unique action. However, it is also possible in certain cases that there are multiple actions for which phi transpose SA theta have the same value.

In which case, if you try to find the maximum, there will be multiple answers to that. And what we are trying to say here is, When I write support of theta here, support of theta is basically how many such policies can be viewed as greedy, how many deterministic policies can be viewed as a greedy policy that is associated with phi theta. So, as we saw before, whatever phi theta you give, there will at least be one greedy policy. However, there is also a possibility that there are multiple greedy policies associated with the same phi theta vector, and these multiple greedy policies can arise when this quantity has the same value for different actions, in which case there could be multiple greedy policies associated with phi theta.

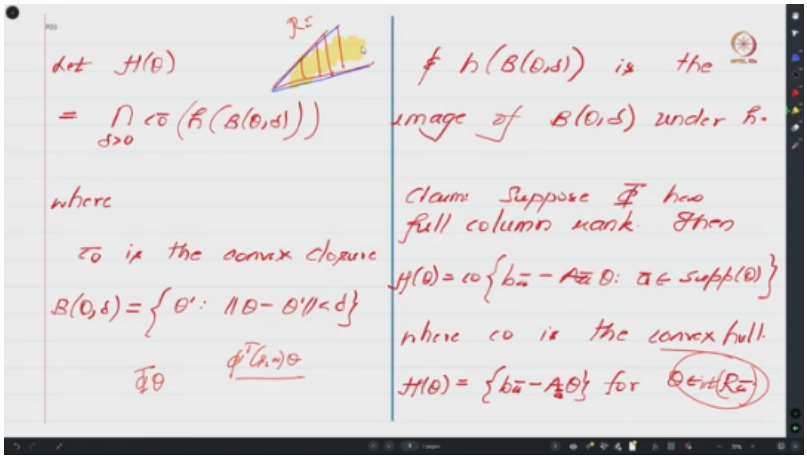
And one can show that this capital H of theta is the convex hull, not convex closure. It is the convex hull of this BA bar minus AA bar theta for all A bar in support of theta. In particular, if your theta lies in the interior of RA bar. So let me just explain what is the interior of RA bar. So the interior of RA bar is basically the collection of phi theta for which the only greedy policy is A bar.



One can show that I mean more geometrically if this is your cone RA bar, right? You sort of throw away the boundaries, right? So the blue lines are the boundaries associated with this cone. You remove them and whatever is inside which I am shading right now, okay? This is the interior of RA bar.

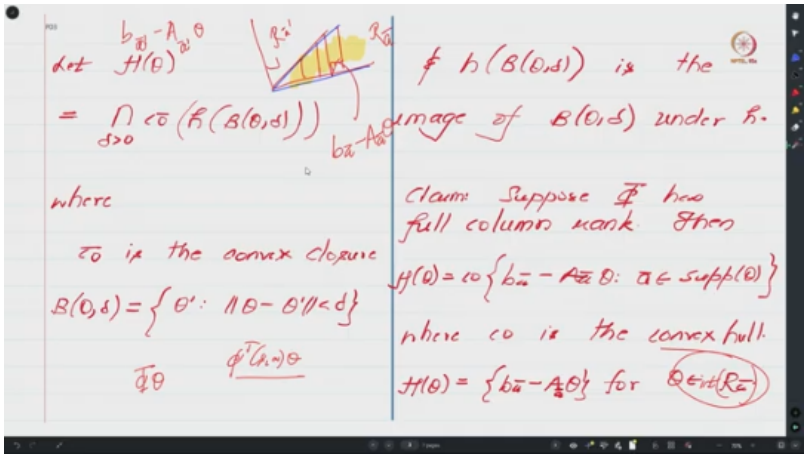
You take any theta which sits in the interior of RA bar. One can show that in that case, your H theta which is defined in the following way is actually made up of a singleton which is BA bar minus AA bar theta. So, in other words, what we have done using this

formula, H theta definition is that if you take two different cones, okay, so let me, since I am defining these cones, let me name them appropriately. So, this let us say is RA bar and let us say this is RA bar prime, right.

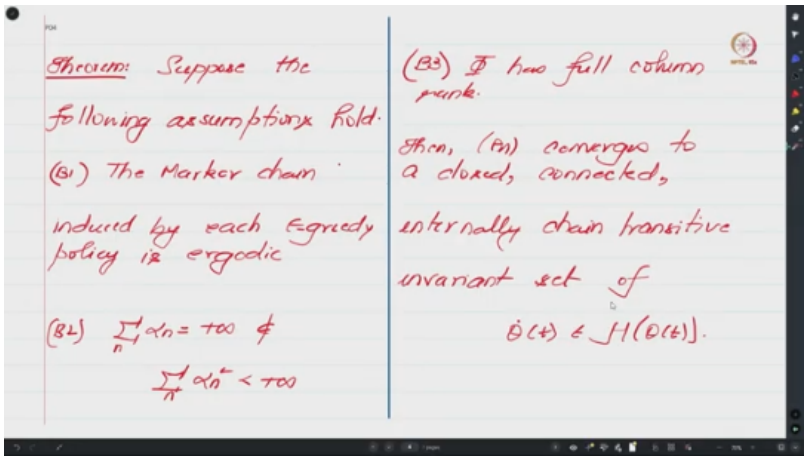


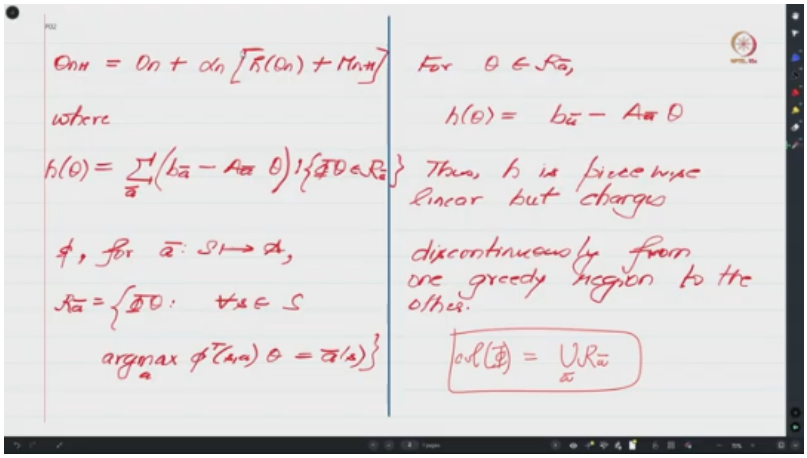
So, when your phi theta sits here, right, it would, I mean sits in the interior of this, your capital H will be H. And the capital H function here that is in the interior would basically be BA bar prime BA bar prime minus AA bar theta and when it is in the interior here it will be BA bar minus AA bar theta. So, it should be BA bar prime minus AA bar prime. So, let me write this properly.

So, this will be a bar prime theta, and in this case, whenever you are in the interior, it will be b bar minus a bar theta. And what we are saying over here is when you are sitting on this boundary, the boundary between two greedy regions, your capital H theta will basically be all possible linear combinations of this update direction and this update direction. I should be careful. Instead of saying linear combination, I should say all possible convex combinations of this vector and this vector. H of theta will take all possible convex combinations over here, whereas as soon as phi theta sits here, it will take this singleton direction, and when phi theta sits in this interior, it will take this singleton direction.

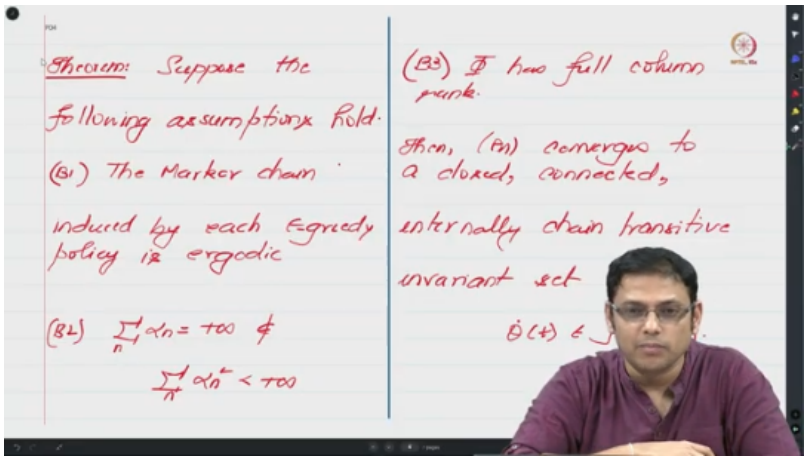


So in this way, you can define H theta, and one can see that H theta has this expression over here. So what this means is that Now, the update rule that we had seen previously, the update rule that we had seen previously, can now be studied from the lens of stochastic recursive inclusions. That is, instead of viewing this as being a singleton map, What we can think of is that there is some higher, you know, like apparent set valued map, and whenever you are at a particular theta n, you are taking a direction from the set that is specified by capital H of theta n, right?





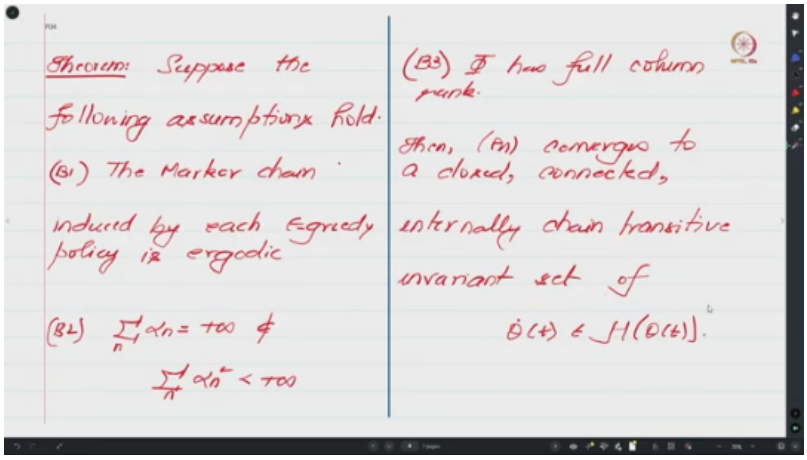
So if you sort of take that interpretation, one can view the update rule that we have over here as a stochastic recursive inclusion, right? And with this in mind, we have proved the following theorem in that paper that I mentioned, whose screenshot I had shared in lecture number 47. So there, what we had shown was suppose the following assumptions hold. That is, the Markov chain that is induced by each epsilon greedy policy is ergodic. This is needed to ensure that your stationary distributions and all those things are well defined.

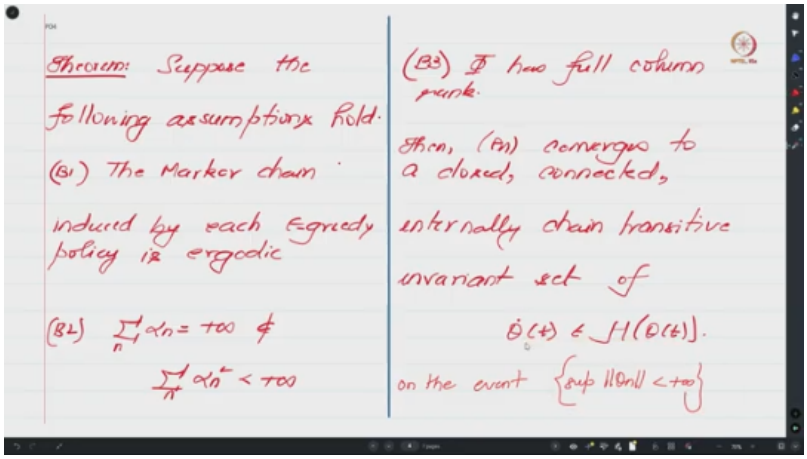


Further suppose your step sizes satisfy the usual Robbins Monro condition, which is that the step sizes add up to infinity and the square of the step sizes is less than infinity. And finally suppose that phi has full column rank. So if you make these three assumptions, then what our result says is that the theta n, that is the iterates generated by your Q learning with linear function approximation and epsilon greedy exploration. If you

consider the iterates generated by that algorithm, then that converges to a closed connected internally chain transitive invariant set of this differential inclusion that is given over here. That is theta dot of t belongs to capital H theta of t.

So what is nice about this result is that you know we have a stochastic algorithm which is this Q learning with linear function approximation and epsilon greedy exploration. What we have now managed to show is that the limiting behavior of this algorithm is actually mimicking the limiting behavior of the solution trajectories of this limiting DI. I think I missed one point here I should say then theta n converges to this on the event sup of norm theta n less than infinity. So what this point over here means is that when you run this Q learning algorithm, we are not guaranteeing that the iterates will be bounded.





So either the iterates, you know, their norm explodes to infinity. On the other hand, if the iterates do not escape to infinity, then the limiting behavior of this theta n will be similar

to some solution trajectory of this differential inclusion. So, now let us go back to this picture that I have been showing over the last two lectures and you can see that when you start the solution trajectory very close to this optimal Q star even then we see three different behaviors. So, let us try to identify the limiting differential inclusion associated with this story over here and let us see if we can interpret these different behaviors. So, that is given over here.

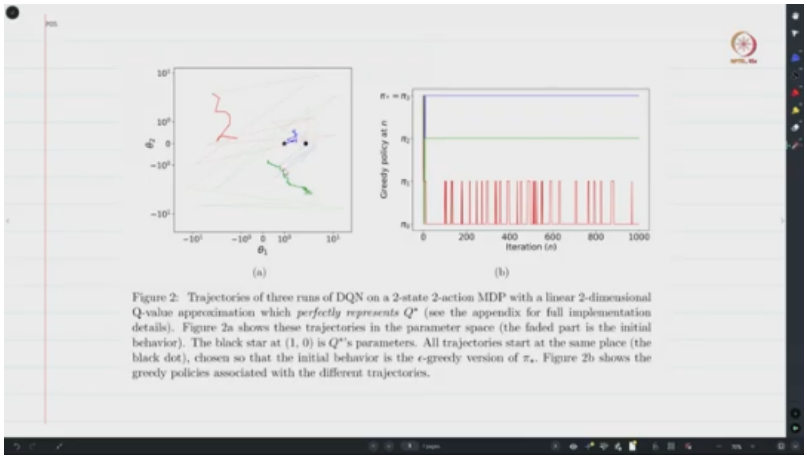
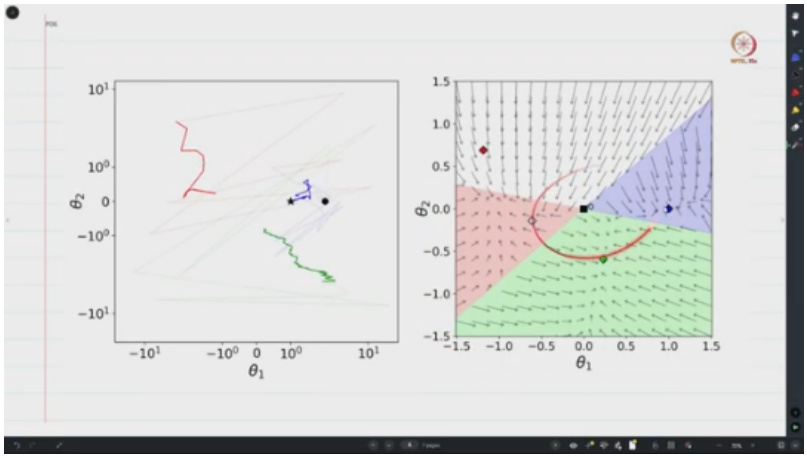


Figure 2: Trajectories of three runs of DQN on a 2-state 2-action MDP with a linear 2-dimensional Q-value approximation which *perfectly represents* Q^* (see the appendix for full implementation details). Figure 2a shows these trajectories in the parameter space (the faded part is the initial behavior). The black star at (1, 0) is Q^*'s parameters. All trajectories start at the same place (the black dot), chosen so that the initial behavior is the ϵ-greedy version of π_*. Figure 2b shows the greedy policies associated with the different trajectories.

So, this is the behavior that we saw in the previous slide, and this is the behavior of the limiting differential inclusion that is associated with this algorithm over here. So, you can see that we have used four different colors for four different greedy regions that are present over here. So, there are four different greedy regions that are present over here. So, what you know, the arrows basically indicate B A minus A A theta, that is, you know, there is a linear dynamics that is present in this greedy region. There is another linear dynamics that is present in this greedy region, and there is another linear dynamics that is present in this greedy region.



Finally, there is a different linear dynamics that is present in this greedy region. And one can see that here the dynamics sort of pushes you in this direction, whereas the dynamics here pushes you in this direction, and the dynamics here sort of pushes you in this direction. The dynamics here, on the other hand, pushes you in this direction. So what we have done over here is, in this example, you see there are some diamonds that are sitting. These diamonds basically correspond to A inverse B for the dynamics that is present in each greedy region.

For example, for the dynamics that is present in this white greedy region, we look at A inverse B, and whatever that thing is, we have marked with a white diamond. And surprisingly, you see that the white diamond did not necessarily sit in this white region. It, in fact, sits in some other region. On the other hand, the A A inverse B that is associated with the dynamics in this greedy region, which is denoted by red diamond over here, this sits in the white region. And here the A A bar inverse B A bar, which is associated to the dynamics in this region, that actually sits within the greedy region itself.

And finally, even for the dynamics that is present in the green region that also sits within the green region itself. So now let us try to understand how the solution trajectories of this differential inclusion will behave. So if you start over here, this dynamics will sort of push you towards this red region. So as soon as you come into the red region, you know the dynamics changes, right? So when you are in the white region, you are pushed towards the, you know, this slightly pinkish region, and then as soon as you enter the pinkish region, the dynamics will actually force you towards the white region.

So you can see that you will be forced along like this, and whenever you go back to the white region, you will again be forced towards this pinkish region. So one can see that the dynamics in the white and this slightly pinkish region will push the iterates in either direction. And one can show that because of this nature, there is some point that resides on the boundary where if you take a convex combination of the dynamics that is enforced from the white region and the dynamics that is enforced from the red region, the convex combination turns out to be 0. So it is that point which one can show is actually an asymptotically stable equilibrium point that is sitting over here. So one can see that any solution trajectory that starts from this white region will try to go towards that intermediate point that is sitting over here.

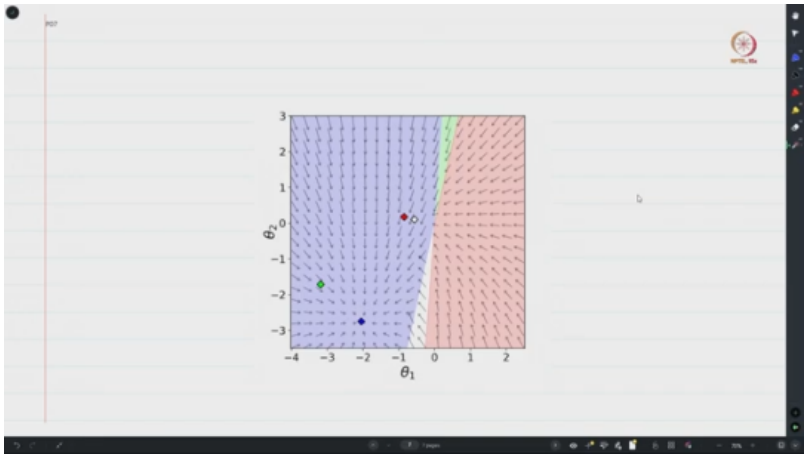
And if you remember this behavior, and if you remember, I said that if you look at the greedy policy associated with the iterates that are present over here, then the greedy policy keeps changing. So the reason is that when you sort of start the solution trajectory in this region, I mean you start over here. So, the initial noise pushes you, and eventually you land up over here, which is somewhere over here. So, if you now try to, you know, behave according to the limiting differential inclusion over here, you sort of go here and then try to sort of settle along the boundary that is present. Now, when you, because of the noise, you will often get pushed up and down, right, and because you are getting pushed up and down, the greedy region sometimes will be the deterministic policy that is present here, and sometimes when your iterates are over here, the greedy policy will be the deterministic policy associated with this cone.

So that is the reason when you see the behavior along the red trajectory it sort of settles somewhere over here right whereas I said there is a convex combination of the dynamics governed by this region and this region which is equal to 0. So you will settle along that point right and you know when because of noise you will keep fluctuating along these two regions and the greedy policy will keep jumping between the policy that is over here and the policy that is associated with this greedy region. On the other hand the explanation for this behavior can be shown as follows you start over here initially the noise pushes you around and eventually you land up in this region And from here on you sort of try to follow the dynamics that underlies this cone over here.

So you start over here and you try to move towards this green diamond. So you can see that here you sort of try to mimic the behavior of the solution trajectory dictated by this vector field. So you sort of move and try to reach this green diamond. And since this green diamond sits within the interior of this cone, if you take the greedy policy associated with the iterates along this trajectory, one can show that they will always be the deterministic policy that is associated with this cone. And one can similarly explain this blue trajectory over here.

You start from this region, sort of the noise pushes you around and eventually you land up over here, somewhere over here and this sort of takes you slowly towards this blue diamond that is present over here. And this is your Q star. So in some sense, the explanation for the different behaviors for Q learning with linear function approximation and epsilon greedy exploration in some sense mirrors what we have seen in this picture that is governed by some vector field which is piecewise discontinuous. So in other words in different cones you can have different local dynamics and different local dynamics can push you in different directions and some of them need not necessarily force you towards this blue diamond which is associated with Q star. So we are in the perfectly realizable case but you can see that the local dynamics need not push you towards this and this is the reason for this very diverse behavior that you see with Q learning and function approximation in practice.

And in our paper, we also identified a strange MDP linear function approximation combination which is described over here. So we took some phi, we took some environment, and we looked at the limiting differential inclusion associated with that. So here again, one can see that there are four different cones and each cone has some dynamics. But the nice thing over here is that all of them seem to be pushing you towards this blue region. So, you see this white region over here; this is also pushing over here.



If you see this green region, this is also pushing you towards the blue region, and the dynamics in this blue region seems to be pushing you towards the red region. So one may guess here that okay, if all the dynamics is pushing you towards this, maybe this diamond that is present over here, or the phi theta associated with this diamond over here, maybe is associated with a good policy skew value function. And surprisingly, what we observed is that although the dynamics across this differential inclusion all of them push you towards this blue diamond. This blue diamond, if you take the phi theta corresponding to this blue diamond and take the greedy policy associated with that phi theta, then one can show that that greedy policy is the worst among all the four different policies. So, in this case, one can see that even though you have tried to mimic the behavior of Q learning tabular Q learning in the function approximation setting, if you know in this particular MDP and function approximation choice, if you run that algorithm and if you try to rely on the limiting policy to be an estimate of the optimal policy, you will actually be wrong because in this particular case the greedy policy associated with your limit will actually be the worst possible policy.

So, in this way, one can see that by using the tools that we have studied in stochastic approximation, one can actually analyze very, very interesting and popular algorithms that have been used in reinforcement learning and try to understand their behaviors, and hopefully, you know, you have learned something from this behavior and you know, you can come up with a way to fix some of these issues. So, I hope I can see you later. Thank you and goodbye. Thank you.