STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Engineering

Indian Institute of Science, Bangalore

Week 11

Lecture 40

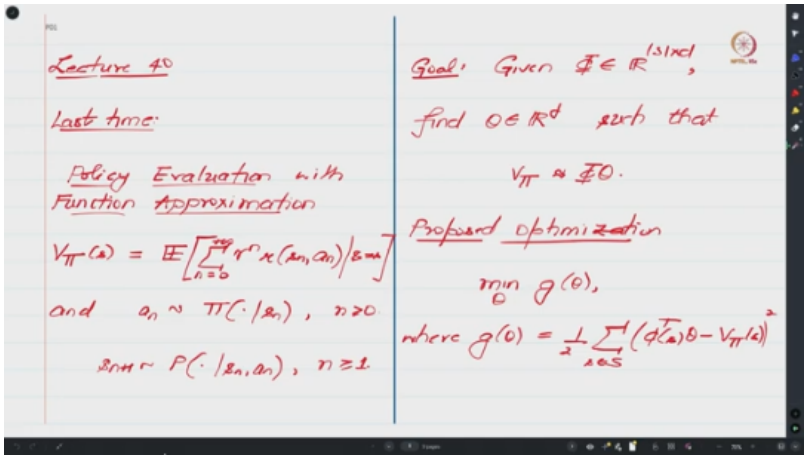
Temporal Difference Algorithm Through the Lens of Stochastic Approximation

Hello and Namaste, everyone. Welcome to this lecture 40 of this NPTEL course on Stochastic Approximation. So, in this week and the next few lectures, we are going to look at various applications of Stochastic Approximation. In particular, we are going to focus on applications from the reinforcement learning context. In the previous class, we looked at this problem of policy evaluation; that is, given a policy, can you quantitatively say how good that policy is? And towards that, we looked at this concept called the value function, and then we said, can we somehow try to estimate the value function of a policy pi when the state space is very large?

In that case, what one tries to do is make use of what is called function approximation. And in the previous class, we began our discussion by looking at what is known as linear function approximation. And at the end of the last class, we came up with an algorithm. And in today's class, we are going to recap that algorithm and begin our convergence analysis of that algorithm. So, let us do our formal analysis.

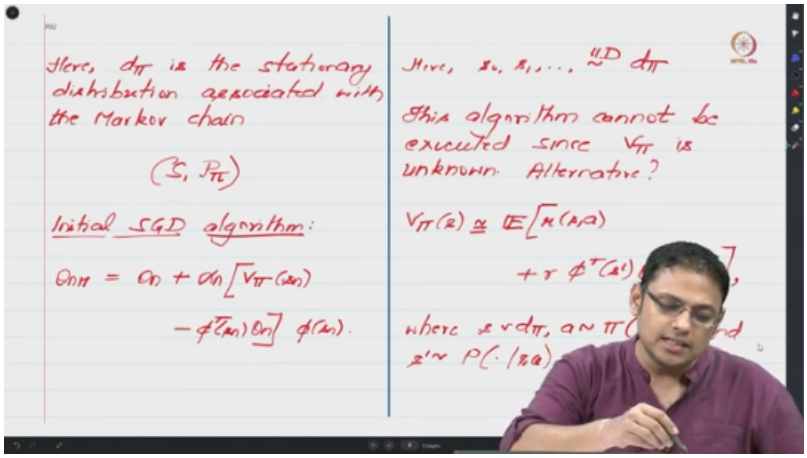
So, recall that we want to do policy evaluation in the function approximation context. What that means is that we have been given a policy pi, and we want to find V pi, which is defined in the following way, and this quantity over here tells us the cumulative discounted sum of rewards that we would get if we act according to this policy pi. Formally, V pi of S is the expected value of this infinite sum where the nth term is gamma to the power n times R of Sn An, and this An is presumed to be sampled from our policy pi, right? And this Sn plus 1 is sampled from your transition kernel, right? And in the linear function approximation setup, what we have been given is that we have this

matrix V whose size equals the cardinality of the state space times D, and typically this D is very small as compared to the state space.



And our goal is to find a vector theta such that V pi is approximately equal to pi theta. So, towards that we said okay let us consider this optimization problem g of theta where g of theta is so I have made a mistake here I should multiply this by d pi of s right. So, if we said let us look at this optimization problem over here right where the sum is over And the s-th term here is multiplied by d pi of s and this is the error between the s-th coordinate of phi theta which is phi s transpose theta and minus v pi of s. So we look at the difference. square it and then you know multiplied by d pi of s you can think of this as weight so if the stationary distribution of the Markov chain induced by your policy PI assigns more weight to a state s then we prefer that this error be small whereas if it assigns a small weight to a particular state s then it is you know reasonably fine if this error is large right

And as I said, this d pi over here is the stationary distribution associated with the Markov chain where the state space is S and the transition probability matrix is p pi. And you can look up the previous lecture to recap the definition of p pi. And then we said, you know, one way to minimize this, you know, g theta function would be to perhaps come up with an SGD algorithm, right? So the algorithm's update rule would be theta n plus 1 equals theta n plus some step size times the negative of the gradient. And in particular, we view the gradient as an expectation of something and we take the random variable whose expectation equals the gradient of g theta.

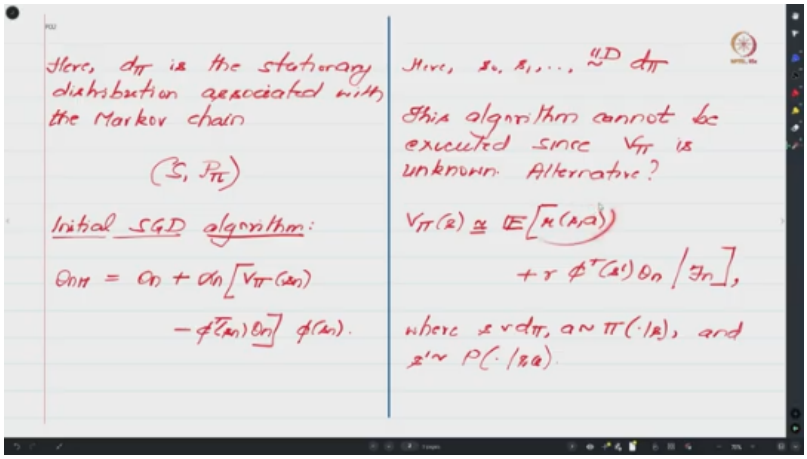


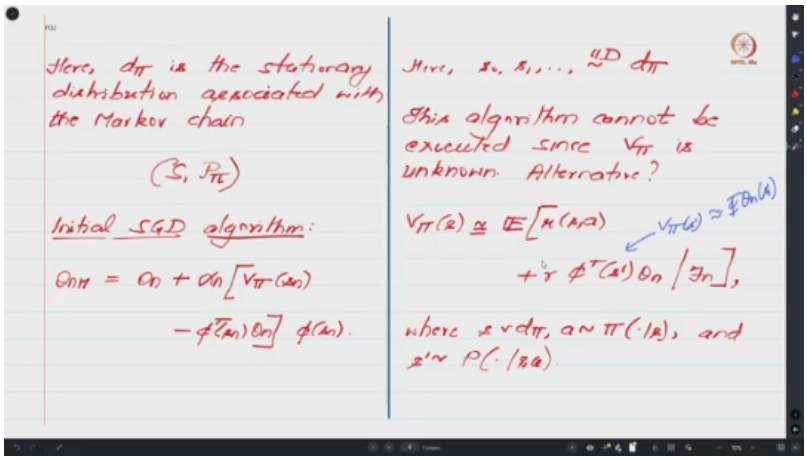
So from that perspective, we ended up with this update rule where we presume that S0, S1, S2 are generated in an IID fashion from d pi. So, as I told you d pi is the stationary distribution of this Markov chain and one has to figure out a way to sample from d pi and one way I said was you start from some arbitrary state S, run the Markov chain for a large number of steps and then take a state that you see at the end. And one can show that such a state roughly has this distribution. But you know later on in the next few classes, we will see how to get rid of this assumption as well.

But at this time, at this point in time, we will focus on the simple scenario where we presume that the states actually can be sampled from this d pi distribution. So even if this is allowed, observe that we cannot run this algorithm because we don't know V pi. I mean, this is the quantity that we are trying to estimate, and hence it is obvious that we do not know V pi. So now the question is: what is the alternative to this algorithm or what is an approximation to this algorithm that we can come up with that is implementable?

So, towards that, we observed that your V pi function, in particular the asset coordinate of V pi, approximately equals this quantity over here. So, if you see this expression over here, it's the expectation of this quantity where you know in the original relation we had V pi over here; in particular, we had V pi of S prime over here, okay? V pi of S prime. But since theta n is your current estimate, in particular phi theta n is the current estimate of V pi, right? So if you look at the S primed coordinate of this, this would precisely be this expression. And hence, one can, you know, think of V pi of S being approximately

equal to this conditional expectation, and this conditional expectation is taken so that this theta and variable becomes measurable with respect to Fn.



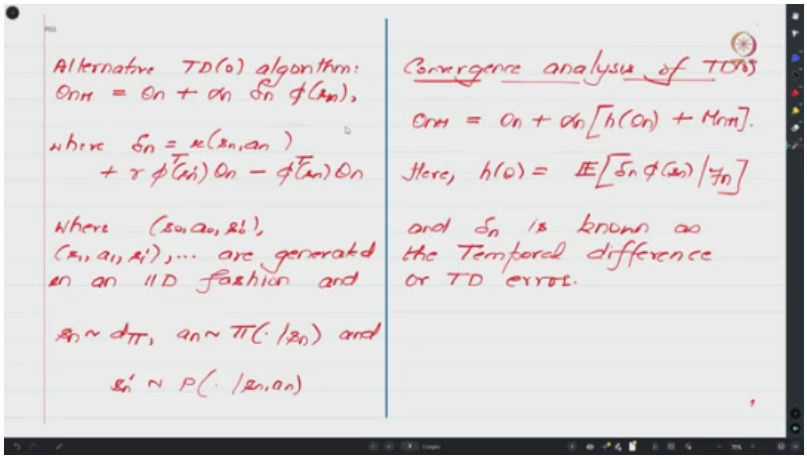


I mean, loosely you can think of you know, under this conditioning, the theta n becomes a constant, and the only thing that is random over here is S, A, S prime, and the expectation is, you know, under the assumption that S is sampled from your stationary distribution, the action A is sampled from your policy pi, and the next state is actually sampled from your, you know, transition probability function. So one can view this V pi S approximately as being equal to this expectation, and one can then use this fact and replace this V pi SN with a sample that you obtain from here.

$$\theta_{n+1} = \theta_n + \alpha_n \left[V_\pi(s_n) - \phi^T(s_n)\theta_n \right] \phi(s_n)$$

$$V_\pi(s) \cong E[F_n]$$

In particular, we ended up with this TD0 algorithm, right? Where the update rule now was theta n plus 1 equals theta n plus some step size, right?



And now, observe that the update rule that follows after the step size is delta n times phi of Sn, where delta n is given by this expression over here. So, you can see that in this unimplementable algorithm, we had V pi over here, right? And now, Wherever we had V pi of, you know, Sn, we have replaced it by an expression that looks like this. Is this okay?

Now in this update rule at any time instance n, right, you know, notice that the update rule includes SN, AN, and SN prime, and you know, in the analysis of this algorithm for the time being, we will presume that S0, A0, S0 prime, S1, A1, S1 prime, these you know tuples are generated in an independent and identically distributed fashion such that the state SN, which you can view as the current state at time instance N, is generated from your stationary distribution D pi, AN is generated from your policy, and SN prime, which you can view as the next state, is generated from your transition probability function. Is this okay? So, you know, assuming that your SN can be sampled from your stationary distribution d pi, the rest of the things indeed can be sampled, right? Like, for example, if you want to evaluate how good your chess strategy is against, let us say, a computer, you know, computer chess engine.

So you can imagine that the states would be the different configuration of the pieces, and your action would be making a move on the chess board, and the chess engine will make another move, and consequently you will end up in a new configuration of pieces. So,

you can see that once you presume that the current state is somehow sampled using the stationary distribution d pi, this is your move and this is the configuration of the pieces after the chess engine makes its move. So, these two things. Indeed, one can sample; the only challenge that remains is how to obtain Sn from this d pi distribution. So, as I said, you know, some practical ways would be to just allow the Markov chain to evolve for some time and then sample, but later on we will see how to get rid of this assumption as well.

So now the question is, presuming we can, you know, sample Sn, An, and Sn prime in this idealized way, you know, what can we say about the asymptotic behavior of your TD0 algorithm. And now what we are going to do is we are going to make use of whatever we have studied, in particular the ODE method that we have studied, in order to understand the behavior of this algorithm. So, towards that, we will try to rewrite this algorithm again in the form that enables our analysis, that is we would write theta n plus 1 as theta n plus alpha n, you know, times h of theta n plus m n plus 1, where h of theta. In particular, I should say H of theta n is basically the conditional expectation of whatever was there with respect to the information that you have at time n, and your Mn plus 1 is basically delta n phi of Sn minus H of theta n. Is this okay? And one can see that because of the way we have defined H of theta n, this indeed becomes a martingale difference noise, right?

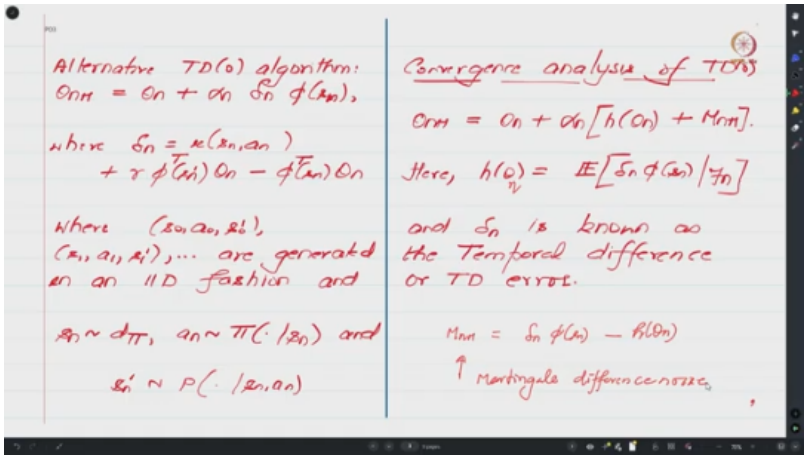
$$\theta_{n+1} = \theta_n + \alpha_n\left[h\left(\theta_n\right) + M_{n+1}\right]$$

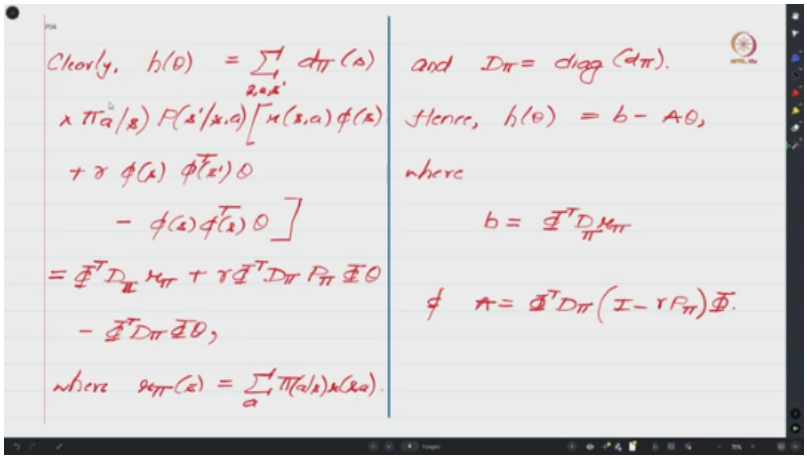
$$h\left(\theta_n\right) = E\left[F_n\right]$$

$$M_{n+1} = \delta_n\phi\left(s_n\right) - h\left(\theta_n\right)$$

So now what we are going to do is we are going to understand the structure of this H function, and as I told you, once we condition on this information that you have at time n, you can think of theta as being known or a constant. And the only random variables include your current state S, your action A, and the next state S prime. And accordingly, one can see that H of theta is basically the sum of, you know, over S A S prime d pi of S, which is the distribution with which you sample your state S. times pi of A given S, which is the distribution with which you sample your A, times P of S prime given S A,

which is the distribution with which you presume S prime is being sampled, right? And hence, you know, H of theta is basically the sum over these terms multiplied by the term that we had in the update rule, which is R S A phi of S multiplied



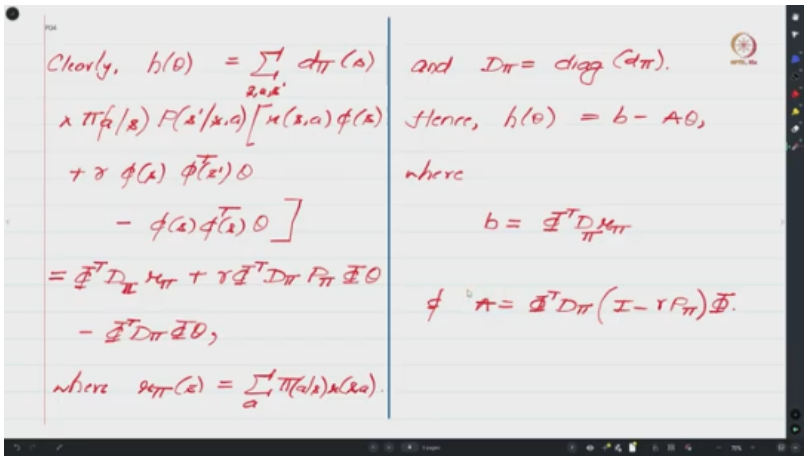


plus phi of S, phi of S prime transpose theta minus phi S, phi S transpose theta. So, what I have done is basically whatever is your delta n, I have multiplied by phi of Sn over here, right? So, if I multiply phi of Sn over here, you can see that this expression is what appears over here. And similarly, if you multiply phi of Sn here, then this is the expression that you will end up with. And similarly, this is the last expression that you will end up with.

And you know if you look at it in this form it looks quite complicated. So what we will do is we will try to write this expression in some compact form. So, towards that observe that you know this expression does not depend on S prime. So, if you sum over this term

you know over here you will see that the sum of this term will be 1. And hence if you only look at the first term one can see that it can be written compactly as phi transpose 1.

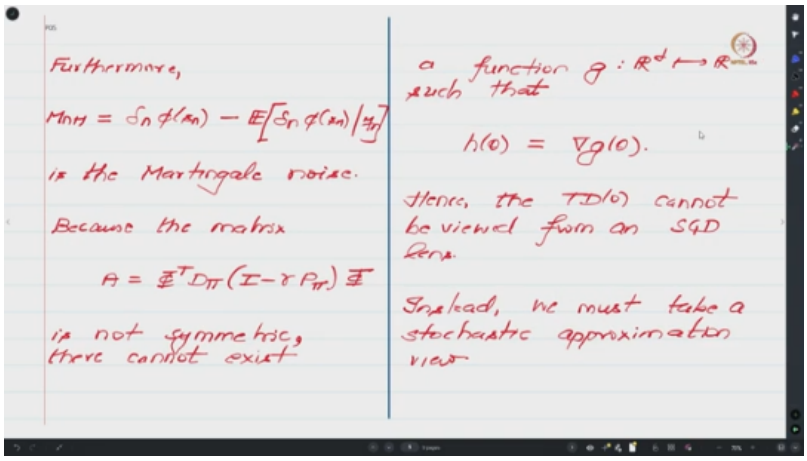
d pi r pi okay where your d pi is the diagonal matrix of size cardinality s times cardinality s and its diagonal entries basically include this stationary distribution little d pi right and your r pi is basically a vector of size cardinality s and its little sth coordinate is defined in the following fashion so one can you know just see that this expression until this point actually equals this And similarly if you take these quantities and multiply it with this one can see that that will result in a term of the following form that is gamma times phi transpose d pi p pi phi theta. right and this last term when it is multiplied by these expressions again notice that there is no s prime here hence we will end up with a term of the form phi transpose d pi phi theta right and one can consequently see that this h theta expression that we have over here can be written as b minus a theta where b is the term that does not depend on theta which is you know your phi transpose d pi r pi and A matrix is basically you know you consider all those theta dependent terms and pull out things that you know other than theta. So if you do that and because of this negative sign one can see that your A matrix will basically be phi transpose d pi you can take it as common and you will end up with I minus gamma p pi right times your phi.



So, this thing is what you have over here. So, in this way you have your A matrix and one can hence see that H of theta equals B minus A theta. So, you know through this algebra one now gets a sense of you know, the structure of your driving function. And hence, one

can ask now that we know the structure of this driving function, can we say something about the asymptotic behavior of your TD0 algorithm, right?

And as I said, this M n plus 1 has this form and one can see that this is a martingale difference noise. I should maybe add martingale difference noise. And now what we are going to do is we are going to understand a bit about this matrix A and then try to see if we can say something about the limiting ODE that is associated with your TD(0) algorithm. So we want to do that and based on the behavior of the limiting ODE, in particular the solution trajectories of the limiting ODE, we will see if we can extrapolate the knowledge of the asymptotic behavior of the solution trajectories of the limiting ODE to the limiting behavior of the stochastic TD(0) algorithm itself. I mean that is what the ODE method is and that is what we will try to replicate.



But the first thing you need to understand is that This matrix A has the form that is given over here. And notice that this P pi matrix over here is not symmetric. In general it need not be symmetric. It basically says under your action pi what is the probability of going from state S to S prime.

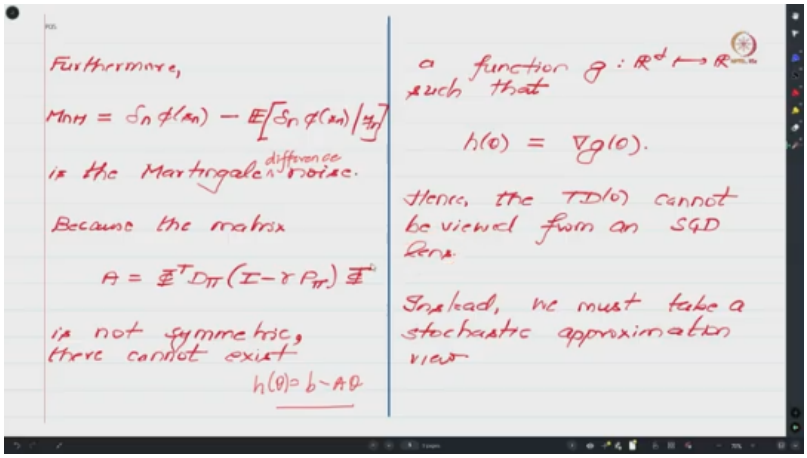
So, symmetric would mean that the probability of going from state S to S prime is the same as going from state S prime to S. So, in general, that may not be true, and hence this P pi matrix is not symmetric. Consequently, one can show that in general this matrix A will not be symmetric. And because this matrix A is not symmetric, and you know your H theta has the form B minus A theta. One can conclude that there cannot exist a function

g, right? No function g such that h of theta is the gradient of g theta, right? So, why am I saying this? Because your h has a linear nature, right?

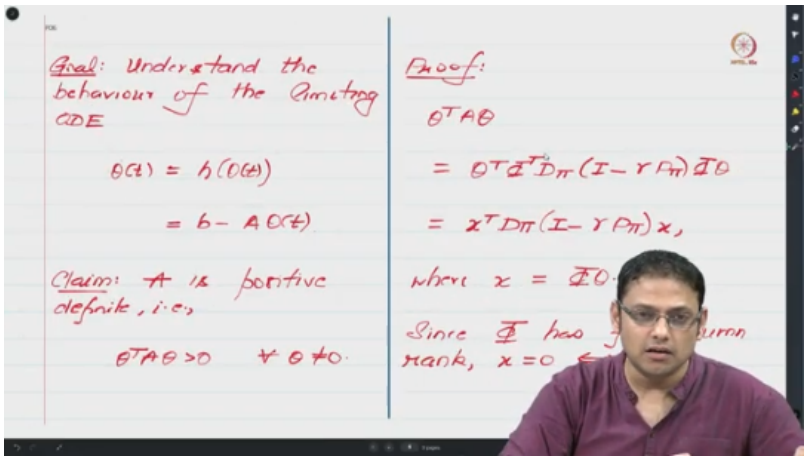
You can see that it is some constant plus some matrix times theta. So, this implies this is a function that is linear in theta, right? So, which means if this has to be the gradient of some function, then that function has to be quadratic in nature, right? And for that function to be quadratic, the matrix that is present in the quadratic term necessarily has to be symmetric. Is this okay?

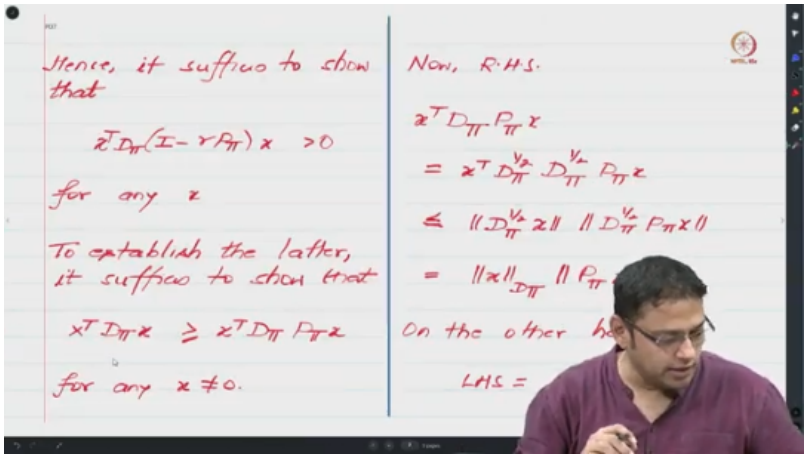
I mean, one of the requirements is that it be symmetric. The other requirement is that it be, you know, positive definite, right? So, because it is not, I mean, I should drop the positive definite, I only meant that it should be symmetric. Now, because it is, you know, this is not symmetric, one can conclude that there can exist no function g such that h of theta is grad g of theta, right? And because of this reason, your TD0 algorithm cannot be viewed from an SGD lens, right?

Instead, we must, you know, take a stochastic approximation view, right, which sort of allows you to consider algorithms which need not be, you know, whose update need not have a gradient descent nature with respect to any objective function, right. So, let us summarize what we have done, we wanted to come up with an approximation function for your Vpi towards that we started off with an objective function wrote down the gradient SGD method with respect to that objective function However, we later on realized that that algorithm is not implementable in practice and then we sort of came up with some variant of that algorithm and then we started looking at the driving function that is the H function for this modified algorithm and then we realized that this H function cannot be expressed in this form. Hence, this new algorithm is not an SGD method with respect to any objective function.



So now given this issue, what we would try to do is we will try to understand the behavior of the limiting ODE, that is theta dot equals h theta of t, and based on our understanding of the solution trajectories of this limiting ODE, we will make a guess about what would be the asymptotic behavior of the iterates generated by the TD0 algorithm itself. So, towards that, we will first make the following claim: that this matrix A, even though it is not symmetric, it is positive definite, which means that if you take theta transpose A times theta, then this quantity or this scalar will be greater than 0 for all theta which is not equal to 0. I would also like to emphasize here that this quantity is a 1 cross D vector, this is a D cross D matrix, and this is a D cross 1 vector. So, let us see why or how we can show that theta transpose A theta is indeed greater than or equal to 0.



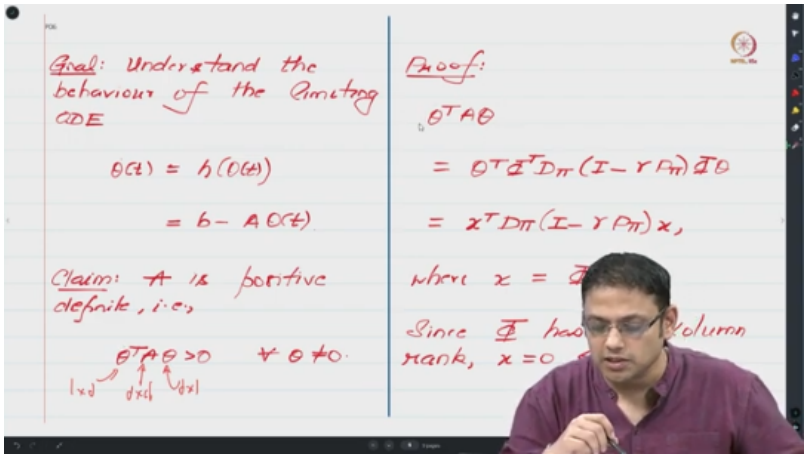


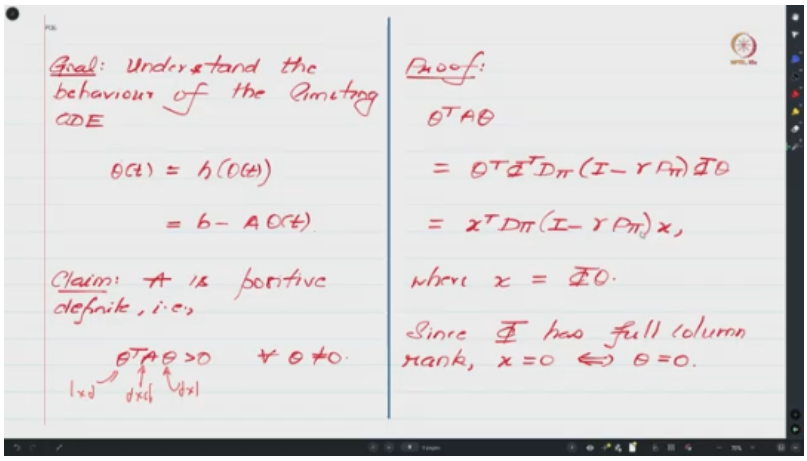
$$\theta(t) = h(0(t))$$

$$= b - A\,0(t)$$

$$\theta^T A\theta > 0 \quad \forall \theta \neq 0$$

So, let us first get an expression for theta transpose A theta. So, I have just substituted the expression for A. So, one can see that because A has this form, your theta transpose A theta will have an expression like this. Now, if you think of phi theta, you can see that at both places they are, and if you define x to be phi theta, then this expression simplifies to x transpose d pi i minus gamma p pi times x. And in our previous class, if you remember, I emphasized that we will presume that this phi has full column rank, which implies x is 0 if and only if your theta is 0. So, in order to show that this is greater than 0, it suffices to show that x transpose d pi i minus gamma p pi times x is greater than 0 for any x which is not 0. I mean, if you establish this, then this is equivalent to establishing the original claim, right?





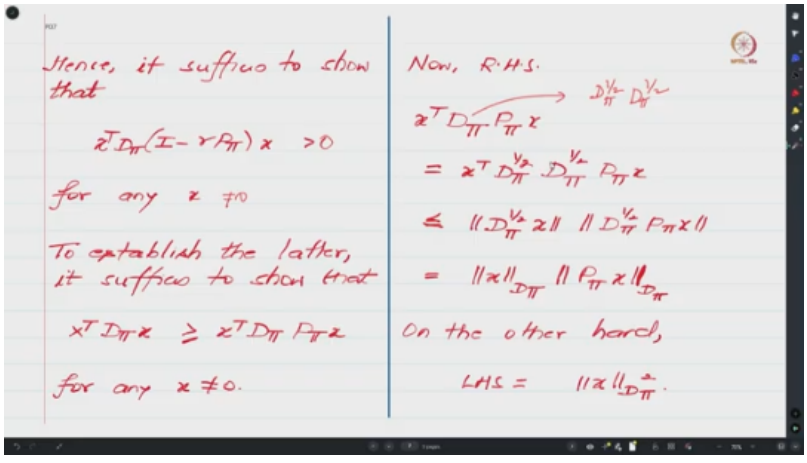
And to establish the latter, if I sort of multiply this x over here and this x over here and this quantity over here, You can see that this condition holds if we somehow manage to show that x transpose d pi x is greater than or equal to x transpose d pi p pi x. So maybe an alert reader may notice here there is a strict inequality whereas here there is a greater than or equal to. Now observe that here we have gamma whereas here I have not put in gamma. So, if we somehow manage to show that x transpose d pi x—which is the quantity that you will obtain over here, right—is greater than this, and since gamma is strictly less than 1 (an assumption we make in the discounted reinforcement learning setup). So if we manage to show this,

Because gamma is strictly less than 1, it would then follow that this inequality holds with a strict sign over here. Hence, to show this, it suffices to show this where here you have no gamma, right? So, in order to show this inequality, let us begin with the right-hand

side. So, we have over here x transpose d pi p pi x, right? And what I will do is I will take this d pi and recall that we have presumed that the Markov chain induced by your policy pi, right?

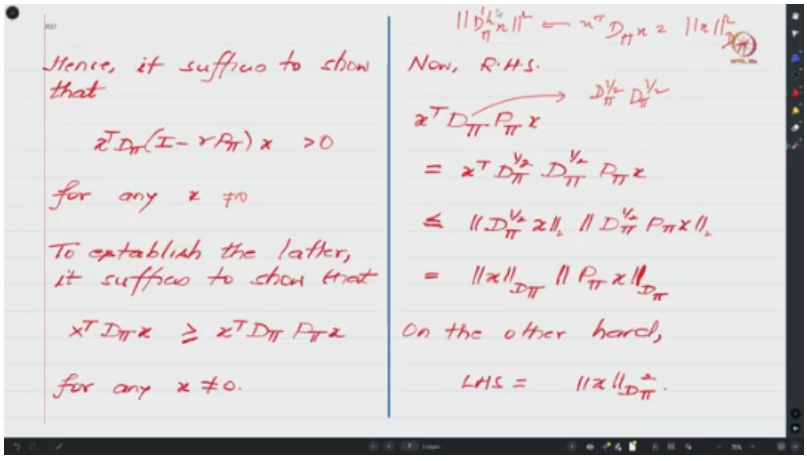
That is ergodic which means that your stationary distribution has all non-negative entries and hence this diagonal matrix is positive definite, right? Because it is a diagonal matrix, it is symmetric and it is positive definite. Hence, I can, you know, look at its square root and the square root will basically be the matrix, the diagonal matrix whose diagonal entries are basically the square root of, you know, the diagonal entries of your d pi. So what is the advantage of viewing it in this way? You will soon see.

So I write this d pi in this fashion. Now once I write it in this fashion, you can think of this as one vector and this as another vector. And hence, we can invoke the Cauchy-Schwarz inequality to conclude that this, you know, inner product between these two vectors is upper bounded by the norm of this vector and the norm of this vector. So notice that whatever was the row vector here, I have taken its transpose and written over here, and since your d pi is symmetric, one can show that the square root of d pi will also be symmetric, and hence even though I have taken the transpose here, whatever was there I have written it as it is, and that is how we get this expression.

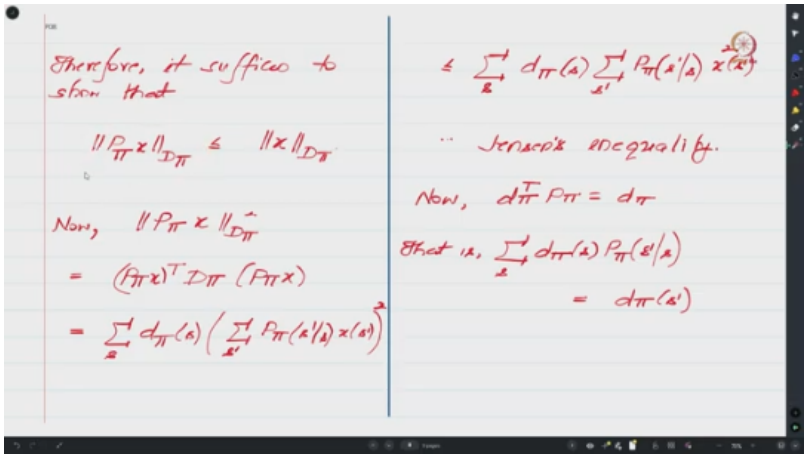


And here the norm is the Euclidean norm. So this is the Euclidean norm. Now because of the presence of this D pi to the power half, one can show that this expression is basically the norm of this vector x, but which is the one that is induced by this matrix D pi. So recall that x transpose D pi x is basically norm of x D pi the whole square.

So, this expression that we have can be written as D raised to half pi x the whole square. So, the square of D raised to half pi x is basically norm x square D pi, and since I only this expression without the square, I have written it as norm x D pi. And in the same way, one can see that this expression can be written as the norm of P pi x under, I mean, under the norm that is induced by D pi, right? So, this is the quantity that we have here, right?



This is the right-hand side. And the left-hand side, as I told you, is basically norm x square where the norm is induced by this matrix D pi, so we want to show that this quantity is bigger than this, and this quantity we have written in this fashion, and this quantity right is less than this quantity, and hence to show this is bigger than this it suffices to show that this is bigger than this; however, you know you already have this. So, we can cancel off one term like this, and hence in order to, you know, derive this inequality, it suffices to show that the norm of P pi x under this, you know, in the metric induced by D pi, is upper bounded by the norm of x under this, you know, norm induced by this matrix D pi. So in order to show this, what we will do is we will look at the square of this quantity.



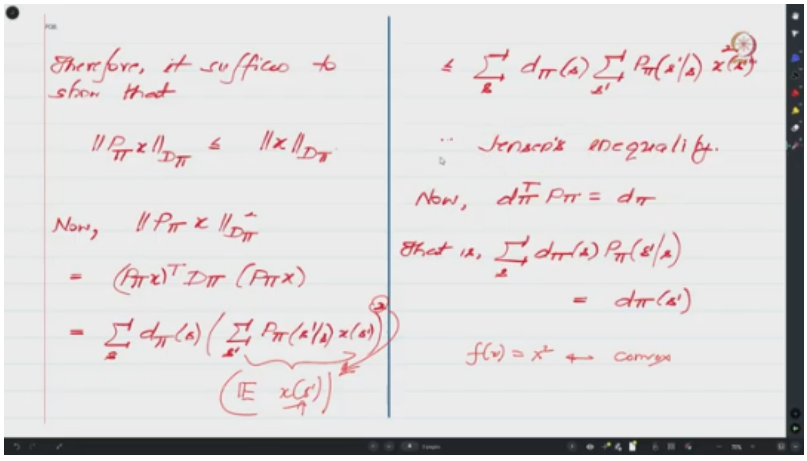
The square of this quantity is basically the norm of p pi x under this d pi induced metric square. So this expression I can write it as p pi x transpose. And since this is the metric that is induced by d pi, I will have d pi here and again this vector over here which is p pi x. And one can see that this expression actually has a form that is given over here. Because of the d pi, you will have d pi s and because you have, you know, the transpose and the same vector over here, one can see that you will end up with the, you know, The s-th term over here and the s-th term over here is basically you take the sum over s prime p pi of s prime given s times x of s prime.

Right. And you look at the square of this expression over here. Right. So one can see that this square is actually this thing. Now the next step that we do is a very interesting step and a very powerful step.

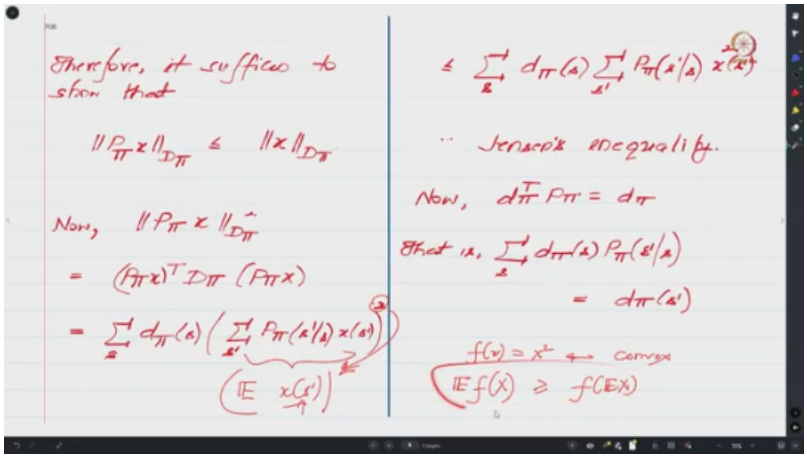
So observe that we have sum over s prime p pi s prime given s of x x of s prime. Now one can view this as an expectation. Your x of s prime where the, you know, quantity that is random is s prime and the distribution from which your s prime is coming is this distribution that you have over here, right? And in particular you can think of this as an expectation of excess, sorry I should say not this thing so it's like an expectation and then you have a square over here. Is this okay? So this square is what I have written here and this expectation I have written here, right? And now because the square function, so if I write f of x equals x square, now this function is convex in nature.

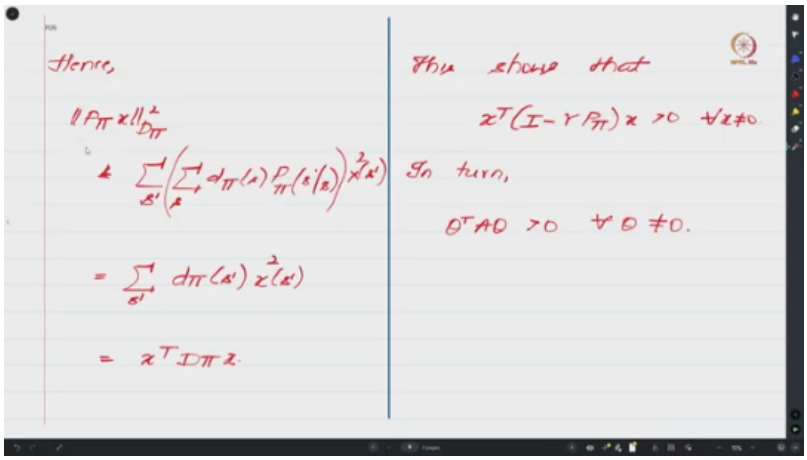
Now, because it is convex, one can invoke what is called Jensen's inequality to now conclude that the square of the function of the expectation is upper bounded by the

expectation of the function. So this is where I use Jensen's inequality. And hence one can conclude that this sum over here is upper bounded by this sum, where notice that this quantity that we have, that is P pi of S prime given S, has now come outside the bracket and this expectation is outside the bracket; that is, we first square it and then we invoke this expectation. In other words, what I am using is



So, the expected value of f of x is greater than f of the expected value of x whenever your function f is convex. So, if you use this property, one can see that this expression over here is upper bounded by this expression. You know, recall the basic fact about d pi: so d pi transpose p pi should equal d pi transpose because your d pi is actually a stationary distribution. So what this implies is that if you take any—so this is a row vector—so if you take any particular coordinate of this row vector, let us say we look at the s prime-th coordinate, the s prime-th coordinate is given by summation over s d pi s p pi of s prime given s. So this sum must equal d pi s prime. So, keeping this in mind, if you interchange the order of these two summations, one can see that this expression is upper bounded by first you take the sum over s prime, then you take the sum over s, and we would end up with some quantity like this.





Now, this quantity is actually d pi of s prime from the fact that your d pi is a stationary distribution, right? And hence, this whole expression will basically be this. And now we can write this expression as x transpose d subscript pi times x, right? And this is the inequality that we wanted to show. In particular, this is, you know, x.

d pi the whole square and we have managed to show that this quantity over here is upper bounded by this quantity and if you sort of look at the chain of arguments one can see that we have eventually managed to show that x transpose I minus gamma P pi x is greater than 0 and that in turn implies that theta transpose A theta is greater than 0 for all theta not equal to 0 which in words implies that your A matrix is a positive definite. In the next class, we will see what this condition implies for the solution trajectories of your limiting ODE, right? And from there on, we will see what can we say about the asymptotic behavior of the TD0 algorithm and then we will see what kind of, you know,

approximation does the TD0 algorithm manage to find with regards to your original value function that is V pi.

So, I hope you will join in the next class to, you know, understand this mystery. Until then, goodbye and namaste.