STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Engineering

Indian Institute of Science, Bangalore

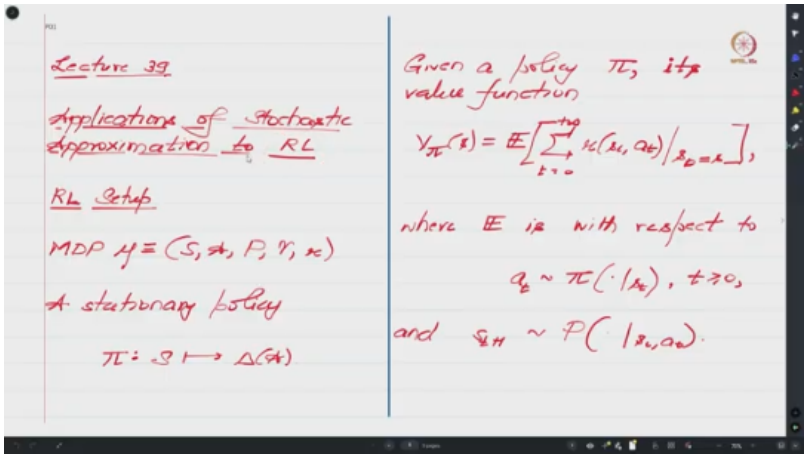
Week 11

Lecture 39

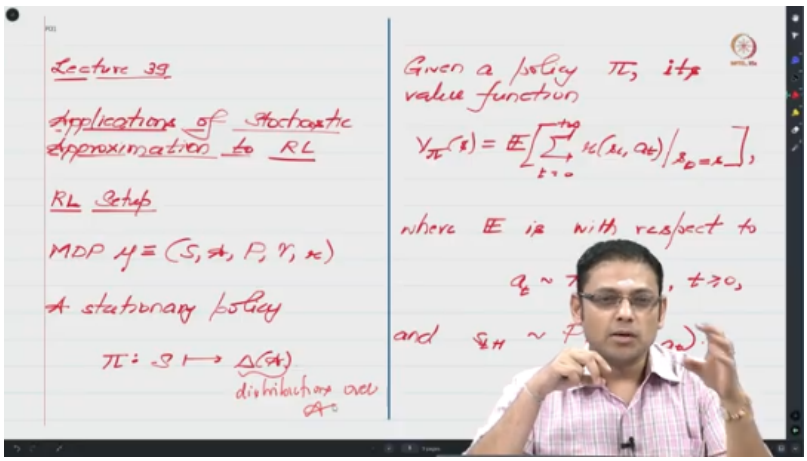
Introduction to Reinforcement Learning and Value Function Approximation

Hello and Namaste everyone. Welcome to lecture 39 of this NPTEL course on Stochastic Approximation. Over the past 10 weeks, we have been looking at various forms of Stochastic Approximation, including Single Time Scale Stochastic Approximation, Two Time Scale Stochastic Approximation, and also Stochastic Recursive Inclusion, and we have at a very broad level discussed their convergence and convergence rate analysis techniques. In the next couple of lectures, we are going to look at applications of stochastic approximation to the reinforcement learning context. In particular, we will consider different problems of reinforcement learning and see how we can design algorithms, in particular stochastic approximation algorithms, for solving those problems, and then we will see how we can also use the theory that we have developed for convergence analysis of such algorithms.

With that in place, let us begin the formal discussion. So, let us recall that we want to do reinforcement learning, and in the first week, we sort of gave an overview of what the reinforcement learning problem is. So, in reinforcement learning, we have a Markov decision process. A Markov decision process is defined by five things, which includes the state space, the action space, the transition probability, the discount factor gamma, and the reward function r. And we said, you know, in reinforcement learning, we want to find, in some sense, the optimal policy, and a typical example of such a policy is any function which takes as input a state from the state space and spits out a distribution over the action space.



So, this notation over here is the space of distribution space. Over the action space A. So, for example, you know, you could say with some probability pick this action and with some probability pick the other action. So, that would be an example of policy, and stationary over here means that this strategy does not change with time, right, and the goal in reinforcement learning is to find the best among the stationary policies. Right. And one can show that that actually outperforms even non-stationary policies.

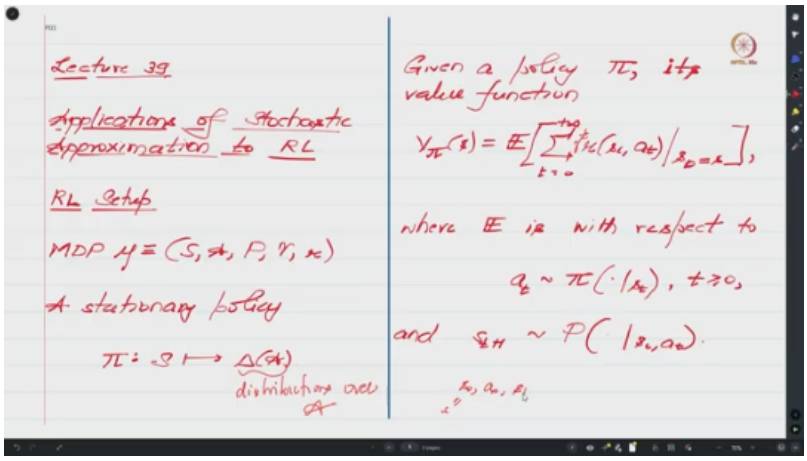


Right. But that's a discussion for the reinforcement learning course. Now, one of the problems in reinforcement learning in particular, it's one of the fundamental problems, is to evaluate how good your strategy is. Right? And the way we quantify the strength of a policy is via what is known as the value function.

And the value function of a policy pi is a vector whose size equals the cardinality of the state space. In particular, the s-th coordinate of this value function is given by the

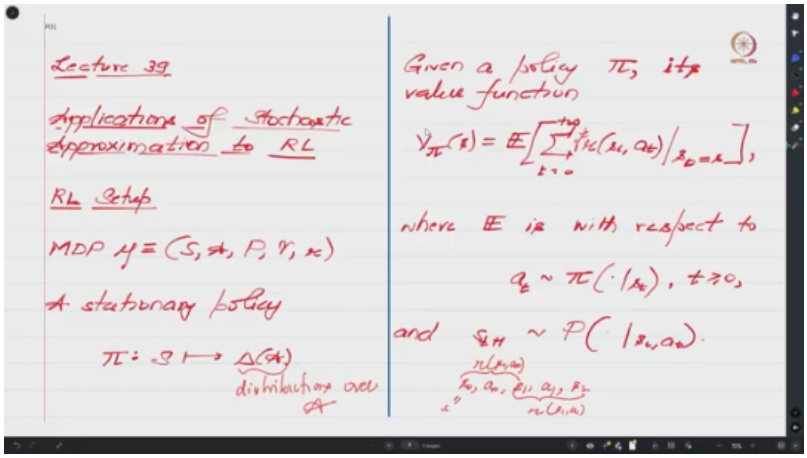
expected value of this quantity over here. So, I forgot to put T over here, gamma to the power T, right. So, you can, you know, imagine that you start at the state

S which is also S0. So, S0 equals S right, and then you take an action A0 which is sampled according to your policy right, and then you move to your state S1. This state S1 is governed or dictated by this transition function right, and in this transition, you sort of get a reward which is a function of S0 A0, and then again you pick action A1 again according to your underlying policy pi right, and then you get hold of the next state and again you get some reward which is a function of S1 A1. Now, this reward could also depend on the next state, but to keep things simple we are presuming that it only depends on the current state and the next state.

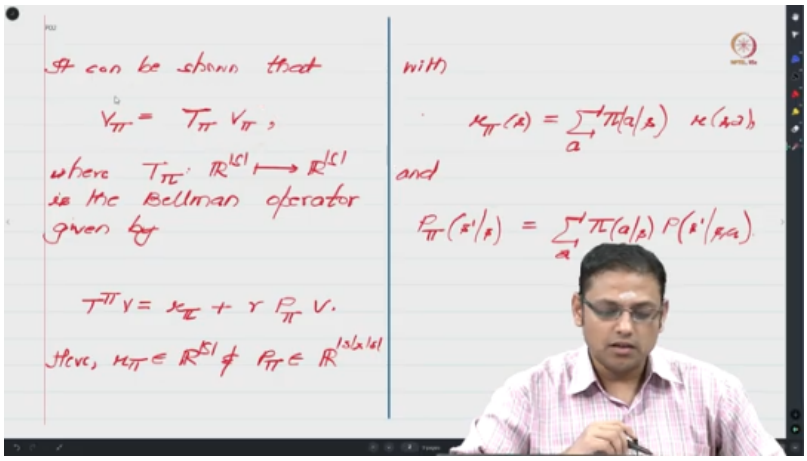


So you see that in this way we will get a sequence of rewards. And then what you do is you look at their cumulative sum, right? And we sort of care more about current rewards than future rewards. And this gamma factor sort of dictates how much we care about, you know, the future reward at, I mean, the reward that we obtain at time instance t. How much do we care about it at time instance 0? So that is why we multiply it to the power gamma to the power t. And then we sum all this up.

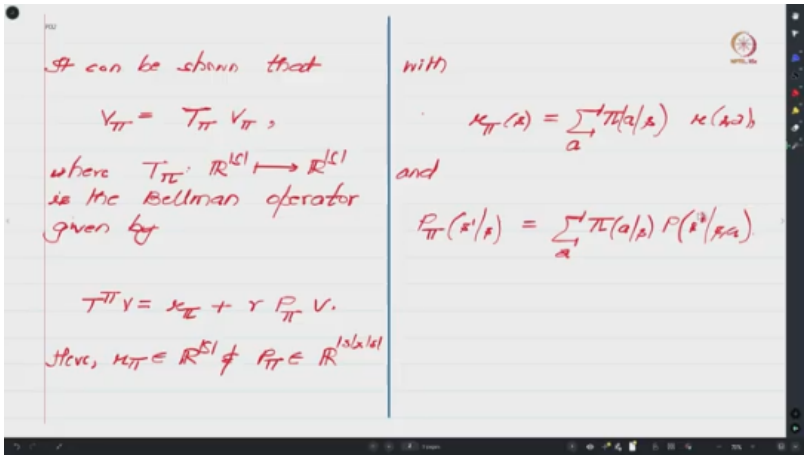
And because these actions and states are random, right? We take their expectation and whatever this value is, we denote it as V pi of S. So, this is known as the value function of a policy pi. And then there is a problem of policy evaluation, which means that given a policy pi, can you tell me how good it is? And the way to say how good it is, is by coming up with the value of V pi. Is this okay?

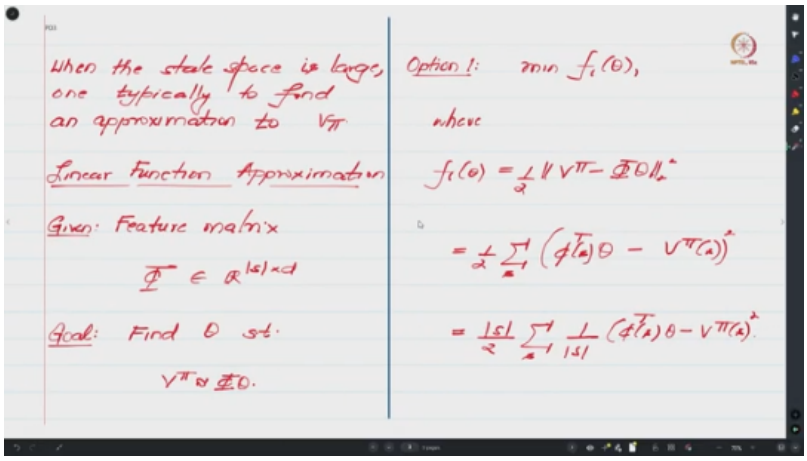


And in the previous class, I had mentioned this, which is that, you know, this V pi actually can be shown to be a fixed point of operator T pi, right, which is known as the Bellman operator and it is defined in the following way. You give any vector V as input to T pi, then T pi of V is r pi plus gamma times P pi V. where r pi is a vector whose size equals the cardinality of the state space and p pi is a matrix of dimension cardinality s times cardinality of s and the s-th coordinate of r pi is defined in the following way. It is the average of the reward that you will get at a particular state s where the averaging is according to your policy pi. right and P pi S prime given S is basically if you start at S what is the probability that you will reach at S prime and the way you compute this is you start at S you ask what is the probability that you will pick action A according to the current policy pi right and then once you pick an action A you sort of you know check with the MDP what is the probability under the transition kernel you know that



Starting from state S and taking action A, you reach state S prime, and then you find, you know, all possible ways of starting from S and going to S prime by picking different actions, and this is what is P pi of S prime given S. Is this okay? Now, one of the, you know, real-world problems of reinforcement learning is to, you know, evaluate how good a policy is, but the challenge is You know that often the state space may be very, very large. Sometimes the state space could be continuous, in which case the state space will actually be infinite. In such a situation, you know, the goal is, you know, instead of trying to find V pi, can we find an approximation to V pi itself?



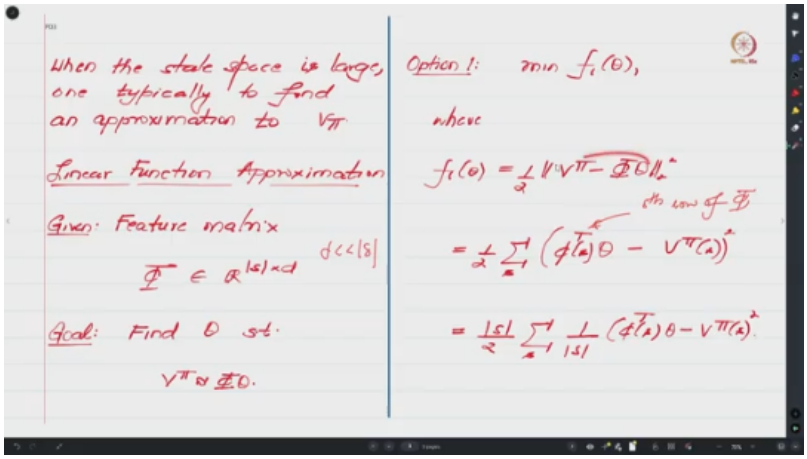


And one of the simplest ways to come up with such an approximation is what is known as the linear function approximation. Here, what is the goal? Well, we have been given some matrix V. So, this matrix V is known, and this matrix V is presumed to be of dimension cardinality S times D, and this D is typically assumed to be significantly less

than your matrix. cardinality of the state space, okay, and the goal then is to find a theta such that V pi is approximately phi theta, right? So we cannot find V pi, which is a vector in a, you know, cardinality s dimensional space, and we are in the problem instance where cardinality of s is very, very large, and So what we do is we try to instant find a theta, and as you can see that the number of columns of phi is d. So theta will be d dimensional.

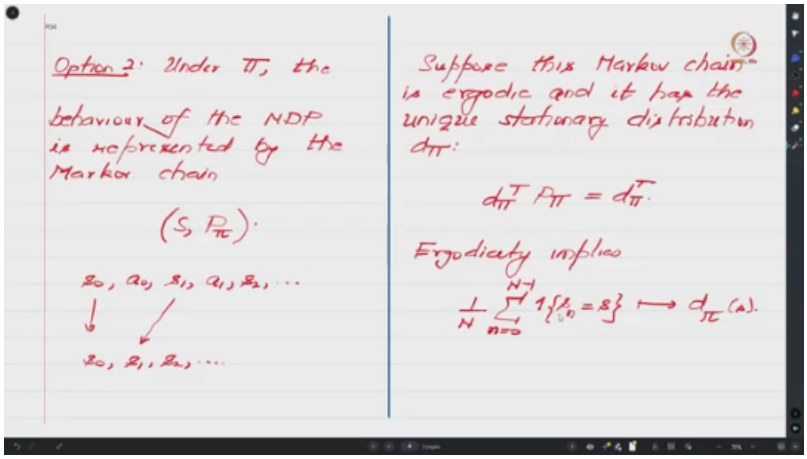
So in some sense, we are trying to reduce the dimension of the problem and asking, you know, in this reduced space can we find a good proxy to V pi. Right. And, you know, to solve such problems, one can look at various optimization problems. The first among this would be, you know, to somehow try to find the theta which minimizes the Euclidean distance between V pi and phi theta. So, if we define, you know, f1 of theta to be this objective function, right, then the goal here would be to find the theta which minimizes this distance.

And, you know, since this is the Euclidean norm, if I look at the s-th coordinate of v pi, which is v pi of s, and if I look at the s-th coordinate of phi theta, it will be phi s transpose theta, where this phi s transpose is basically the s-th row. So, this is like the row vector. So, if you take phi s transpose theta, this will be the s-th coordinate of phi theta. So, you can take the difference, square it up, and sum it over different little s's. So, this will be the Euclidean distance between V pi and phi theta, and we could try to minimize it.



And one can see that this expression can be, you know, sort of multiplied and divided by the cardinality, and one can then imagine this to be the uniform distribution; that is, we

pick a state S at random, right, and then we look at this distance, and then we sort of look at the expected distance. So, one can view this expression in the following way. But then one can ask, why are we only interested in looking at the uniform distribution? Also, does the uniform distribution sort of make sense here? So, to sort of answer that question, we will try to understand, if we look at the states under this policy pi, what would be the distribution of these states?



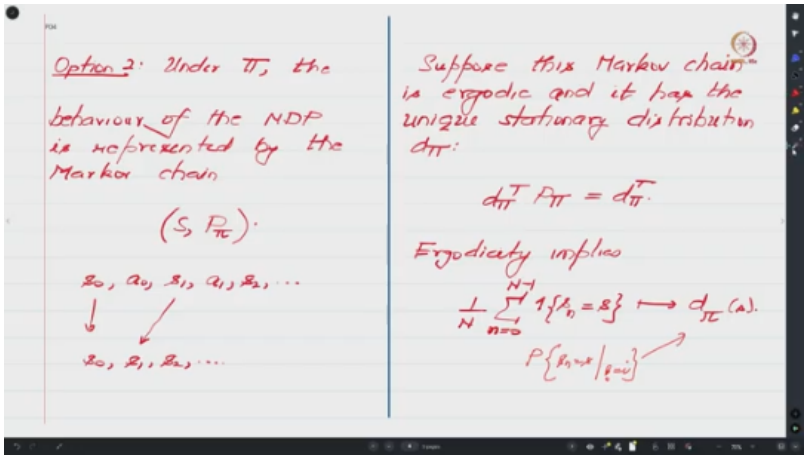
So, towards that, one can see that if you act according to your policy pi, the behavior of the MDP can be represented by the Markov chain S, P pi. So, let me elaborate. See, in your MDP, you start at a state S0, then you pick an action A0 according to this policy pi, and then you end up with this state S1, which is dictated by your underlying transition kernel. Then again, you pick your action A1 according to your policy pi, then you end up in S2, and so on and so forth. Now, since your A0, A1, and so on are always picked by your policy pi, if you ignore these actions and only focus on

The resultant states, then one can show that the resultant states actually form a Markov chain, right? The sequence of states can be imagined to come from a Markov chain, right? And a Markov chain is dictated by two things: the state space and the transition probability matrix. One can show that the probability of going from any state S to any state S prime is governed by this P pi, right. Now, suppose this Markov chain is ergodic and since it is ergodic, it will have a unique stationary distribution. A Markov chain being ergodic means that it is irreducible and aperiodic. So, you can look it up online what

these terms mean, right? So, if it is ergodic, then it will have a unique stationary distribution d pi, which means that if you take

the transpose of your column vector d pi right and multiply it with p pi then you will end up with d pi transpose right. So, this is like the eigenvector, the left eigenvector with eigenvalue 1 right. One can show that, you know, because of this ergodicity, if you fix a state S and you take the, you know, sample of this quantity, right? I mean the average of this quantity, I should not say sample, the average of this quantity. So, what is this quantity?

At time n, you ask whether the observed state is S or not. And then you take, you know, this indicator is 1 if the observed state at time instance n is S and 0 otherwise. And if you sum these indicators and take their average, one can show that because of ergodicity, this distance actually goes to d pi of S. One can in fact also show that, you know, the probability... that S N equals S if you start from any state S I, right, this also actually converges to d pi of S, right. So, one can see that if you start from any arbitrary state I and allow this Markov chain to evolve, then one can see that, you know, as N becomes larger and larger, right, the probability with which you see state S is precisely dictated by d pi of S.



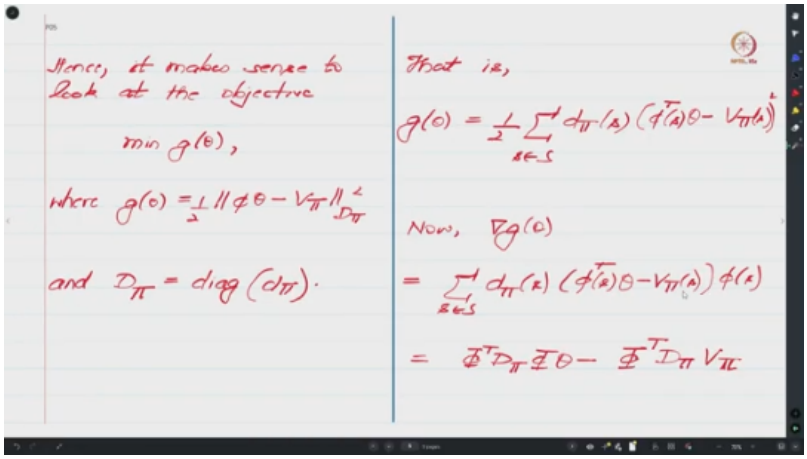
So, in this sense, it, you know, sort of seems reasonable that instead of looking at the Euclidean distance between phi theta and V pi, why do we not look at, you know, the distance between phi theta and V pi, which is induced by this matrix d pi, right, where d

pi is the diagonal matrix whose diagonal entries include phi. Where recall little d pi is the transition, sorry, stationary distribution of the Markov chain, you know, s comma p pi.

$$d_{\pi}^{T} P_{\pi} = d_{\pi}^{T}$$

$$\frac{1}{N} \sum_{n=0}^{N-1} 1\{s_n = s\} \mapsto d_{\pi}(s)$$

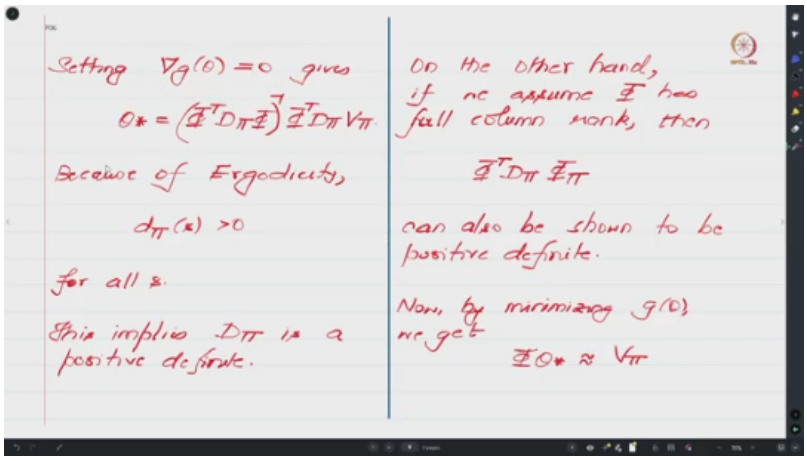
So this Markov chain is also referred to as the Markov chain induced by our policy pi. Then one can see that if you expand this expression over here, one can see that this has the form some little s in capital S. d pi of s phi s theta transpose phi s transpose theta minus v pi of s square, but the difference between this and what we had seen in this f of theta expression is that, you know, this expectation was computed with respect to the uniform measure on the state space. Here, instead, we are presuming that, you know, the expectation is with respect to the stationary distribution, so this sort of makes more sense.



Because as you allow your Markov chain to evolve, the fraction of times that you will be sitting on state S more or less looks like this. So, this sort of tells you how much weight we should give to a particular state S. So, if this has a larger value, then we would prefer that the error be smaller on that state and so on and so forth. So, you know, we are going to now try to approximate the value function V pi using this objective function and let us see if we can do that or not, right? So, first let us see what is the gradient of this expression, right? In particular, let us see what is the theta value that minimizes this

objective function and since this objective function is quadratic, right, and one can show that this is actually, under some reasonable assumptions, this is actually a convex function, a strictly convex function, and because it is a strictly convex function if we set the gradient equal to 0, one can find the value of theta where this quantity is minimized. So, towards that, let us take the gradient. So, if you take the gradient, this half and this two will cancel off and so you see that I have nothing over here. So, this d pi of s comes down as it is and you know I will first take the derivative of this. So, this two vanishes and in fact it cancels with this two.

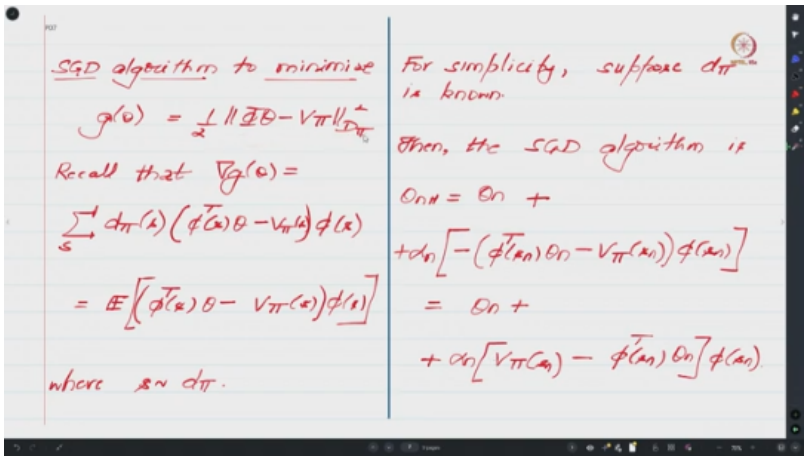
So, I have written it as it is, and whatever is the expression that multiplies with theta, I will take its transpose and write it in this fashion, and one can see that this expression is d-dimensional in nature. And, you know, in compact form, one can see that this can be written as phi transpose d pi phi theta minus phi transpose d pi v pi, right. So, these quantities lead to this, and this along with these quantities, okay, leads to something like this. This is, you know, some simple linear algebra, and you can check it on your own. And now if you set the gradient equal to 0, one can see that theta star equals phi transpose d phi phi inverse phi transpose d pi v phi.



And now whenever you see an inverse, one of the first things you have to check is whether this inverse exists or not. So let us do a quick check. Because your Markov chain is ergodic, one can separately show that this d pi of s is strictly positive for every s, and hence one can show that this capital d pi matrix is positive definite. Recall that this capital d pi matrix is a diagonal matrix whose diagonal entries consist of d pi, little d pi.

And since this is strictly positive, one can conclude that this capital d pi is actually a positive definite matrix.

And on the other hand, if you presume that this phi matrix has a full column rank, then one can show that this phi transpose d pi phi, which is the matrix that is sitting over here, this matrix is also positive definite, and hence this inverse actually exists. And one can see that if we minimize g theta, then we will end up with this theta star, and hence our candidate approximation for V pi would be phi theta star. So, if we manage to somehow solve this problem, the hope is that we can find phi theta star and that would be our approximation for V pi. And now let us see if we can indeed minimize this objective function or not. Is this okay?



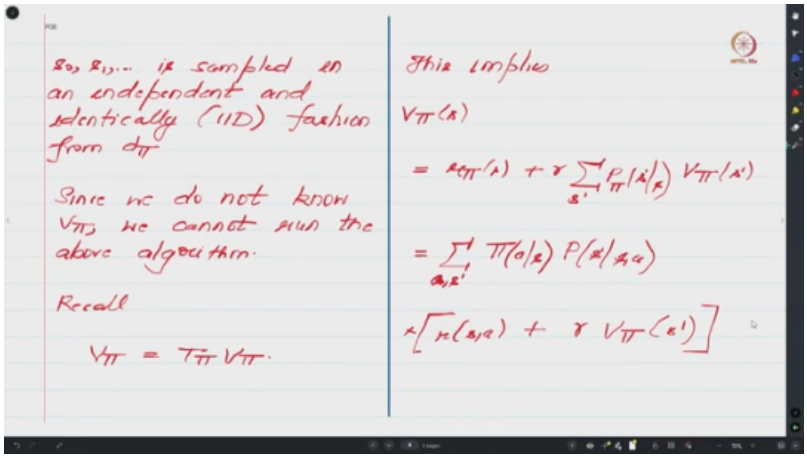
In particular, can we design a stochastic approximation algorithm to minimize this objective function? Right, so to begin with, let us see if we can come up with, you know, a stochastic gradient descent algorithm itself to minimize this quantity. So towards that, recall that, you know, if this was your objective, the gradient has this expression, right? And you can think of this as a distribution, and, you know, one can then imagine this summation to be an expression. Expectation, and the expectation is of this quantity where this state S is actually sampled from this distribution d pi. So this is an expectation, and one can see if you have an expectation, we can use a sample of this quantity as a stochastic noisy estimate of the gradient at theta, and then one can use this fact to come up with a stochastic gradient descent algorithm. And for simplicity, what we will presume

is that we know this stationary distribution d pi. Later on, I will tell you how to relax this assumption.

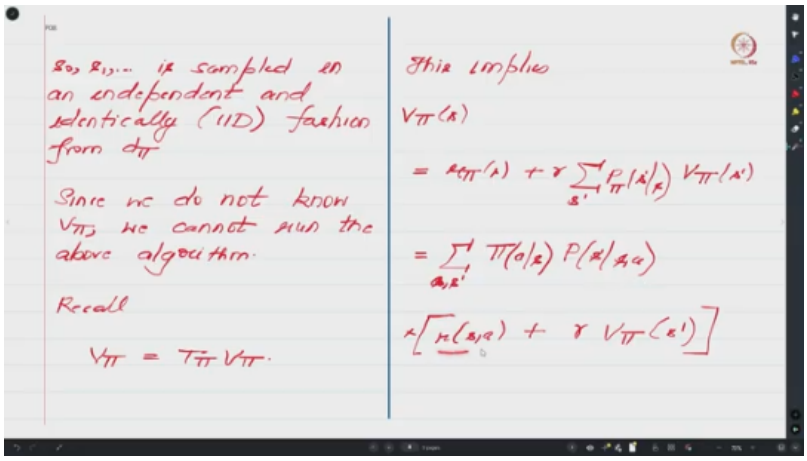
But at a high level, as I told you, because of this ergodic assumption, if you allow the Markov chain to evolve for some time and then... take a state from, you know, like take the last state after you allow this Markov chain to evolve for some time, whatever is the last state you get, one can see that that state can be presumed to come from this distribution d pi, right? And then one can write the SGD algorithm in the following way. So observe that theta n plus 1 is theta n plus some step size times theta. You know, the negative of a sample that is sitting inside the expectation, right? So, why the negative?

Because we want to do gradient descent, and, you know, why do we not have an expectation here as it is here? Because we want to work with samples. So we take one sample of this quantity; by that, I mean you take this Sn and presume that it is sampled from d pi, and whatever this expression is, you take this expression multiplied with a negative sign and use that to update your equation. theta n, right? And one can rewrite it slightly in the following way; this negative and this negative sign becomes plus.

Hence, this will become V pi of Sn minus phi Sn transpose theta n times phi of Sn that we have over here. And as I said, we will, you know, to run this algorithm, we will presume that this S0, S1, dot, dot, dot is sampled in an independent and identically distributed fashion from this distribution d pi. And as I told you, although we do not know d pi, there are ways using which we can sample from this distribution d pi. Now the main challenge in running the previous algorithm is that we do not know V pi. Since we do not know V pi, we cannot run the above algorithm.

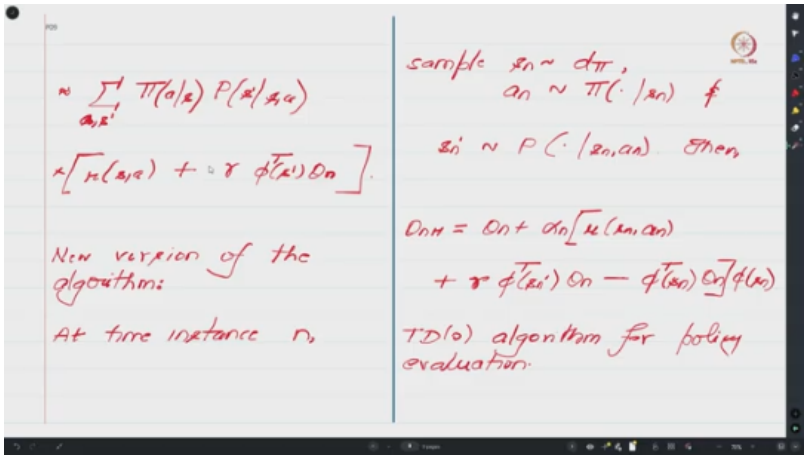


So now let us see what we can do to come up with a variant of this algorithm that can be implemented in practice. So towards that, recall this Bellman equation which said that V pi equals T pi V pi. In particular, if you look at the S-th coordinate of the left-hand side it is V pi of S, and if you look at the S-th coordinate of the right-hand side one can show that it is R pi of S plus gamma times sum over S prime P pi of S prime given S V pi of S prime. Now, recalling the definition of r pi and p pi, one can see that this right-hand side can be written as sum over a comma s prime pi of a given s p of s prime given s a times r s a plus gamma times v pi of s prime. So the purpose of writing it this way is that this quantity can be imagined to be an expectation of a quantity that has a form that looks like this.



So this quantity is an expectation of a quantity which looks like this, where the expectation is taken with respect to this product distribution. Now one may ask, okay, we

do not know V pi right, and this right-hand side also contains V pi, so how does this, you know, expression help? But notice that while we do not know V pi here, we can approximate it with what we know right now. In particular, at time instance n right, we know theta n and phi theta n. Recall is, you know, can be imagined to be the estimate of V pi at time instance n right, and hence one can imagine that this expectation is approximately equal to this expectation. So of course, whether this approximation is good or not will depend on how good your phi is and how good your n is. So, that is why this is, you know, in some sense an approximation, and later on we will see how good of an approximation this is.

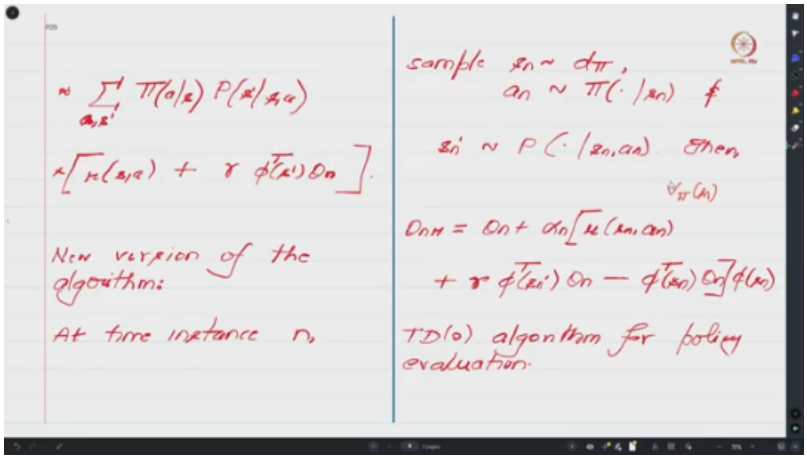


But nevertheless, one can think of V pi of S having an expression like this, right? At time instance n, we are replacing V pi of S prime, but this quantity and hence this thing can be viewed as an approximation to V pi of S prime. I mean, in particular, this thing can be viewed as an approximation to V pi of S prime and this whole thing can be imagined to be an approximation to V pi of S. Now, based on this, we can now come up with a new variant of the algorithm. At time instance n, what we will do is we will presume that this Sn. That is some state at time instance n is somehow obtained from the stationary distribution d pi.

Then we sample an action a n which is supposed to come from this policy and you sort of sample it presuming that the current state is s n and the next state s n prime is presumed to come from the stationary distribution condition on the current state being s n and the current action being a n. So wherever we had v pi of Sn, what we have done is we have

replaced this quantity with a quantity that we have over here. So, you can see that here I have replaced it with R Sn An plus gamma times phi Sn prime theta n. So, you can see that this quantity has replaced V pi of Sn. So, why am I able to replace it?

Because V pi of Sn, if you keep it on the left hand side, then this quantity is approximately V pi of Sn. In particular, the expectation of this quantity is V pi of Sn. And keeping that in mind, I sort of replace it with one of these samples. Now, this is the algorithm that is often used in practice to evaluate how good your policy is. Now, notice that in this update rule, no knowledge of Vpi is required.



Instead, you can take these samples S n, A n and S n prime and use those samples, compute this immediate reward, compute this inner product, compute this inner product. Take, you know, these sums and differences and hence you will be able to compute the square bracket multiplied by phi S N, right? And then you ask, as N becomes larger and larger, can we say something about the limiting behavior of theta N? And because of this approximation, one can show that, I mean, I will discuss this in the next class, that this update rule is no longer stochastic gradient descent. Instead, you know, one needs to view it as a stochastic approximation algorithm.

So in the next class, we will see how to analyze such an algorithm. Until then, goodbye and namaste. Thank you.