STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Automation

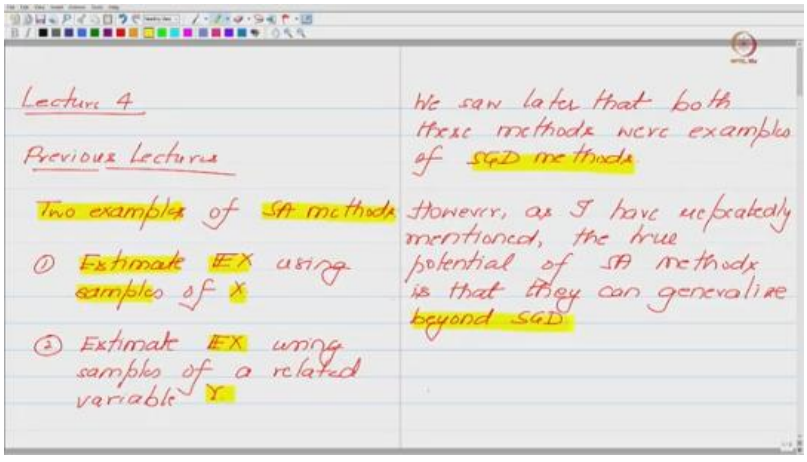
Indian Institute of Science

Lecture 4

An Introduction to Reinforcement Learning

Hello and Namaste, everyone. Welcome to Lecture 4. So, let us do a quick recap of what we have covered so far. In the previous three lectures, we have looked at an introduction to stochastic approximation. I gave you this general abstract form of how a stochastic approximation algorithm looks.

I gave you two examples of stochastic approximation methods. In the first example, we looked at estimating the expected value of X using independent and identically distributed samples of the random vector X itself. In the second example, the problem broadly was looking at estimating the expected value of X using samples of a related random variable Y. We later saw that both these methods were examples of SGD methods. However, as I have repeatedly mentioned, the true potential of stochastic approximation methods is that they can generalize beyond SGD.

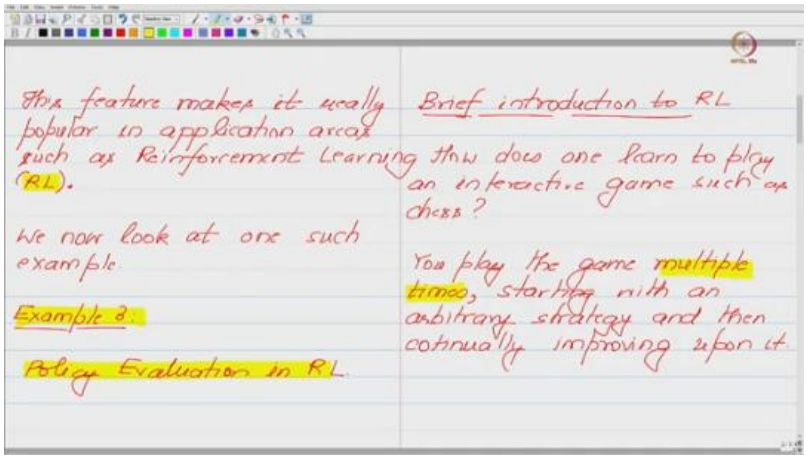


This feature makes it really popular in application areas such as reinforcement learning. Often, I will abbreviate reinforcement learning as RL. So, in this lecture and the next, we will look at an example of a stochastic approximation method which is not an SGD. That

is, it is not a stochastic gradient descent. At a very broad level, this example involves what is called policy evaluation in reinforcement learning.

So that the students and the audience can get a better understanding of this problem. I am now going to give a very brief introduction to reinforcement learning. So, what is the philosophy of reinforcement learning? Well, it looks at how one learns to play interactive games such as chess. So, how do you do that?

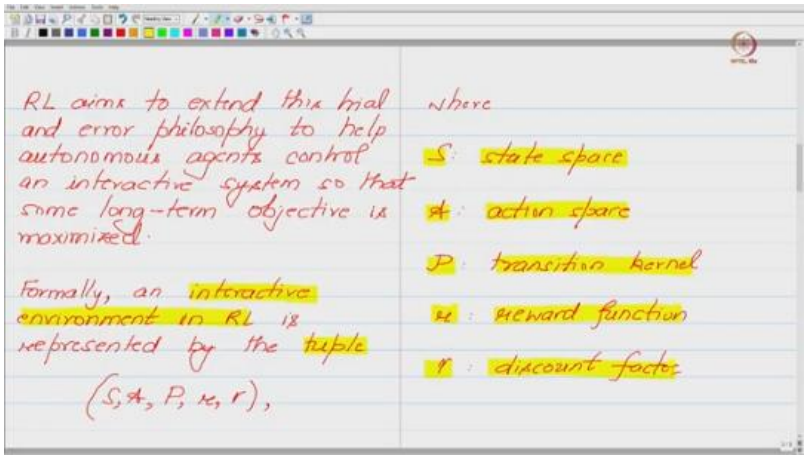
Well, you play the game multiple times. Starting with an arbitrary strategy and then continually improving upon it. I think there is a typo in the word 'continually'. It should be continually improving. So, RL aims to extend this trial-and-error philosophy to help autonomous agents control an interactive system so that some long-term objective is maximized.



Let me elaborate what I mean by that. Formally, this interactive environment in reinforcement learning is represented by a tuple made up of five things. It is S, A, P, R, and gamma. So, what do these five things represent? Well, S is the state space, which is the different possibilities for the environment.

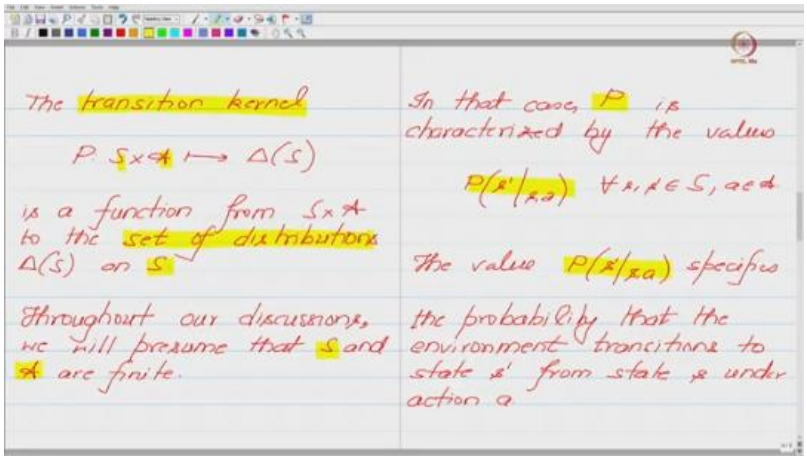
For example, in the context of chess, the configuration of pieces will determine the state of the game. A represents the action space. This is the set of actions available to the autonomous agents. P is referred to as the transition kernel. I will describe what P is in the next slide.

R is the reward function, and gamma is the discount factor. Again, I will explain what R and gamma mean in the subsequent slides. So, what is this transition kernel? The transition kernel P is actually a function. For a function, I need to specify the domain and the range.



The domain of this function P is the Cartesian product S, which is the state space, cross A, which is the action space, and the range of this function P is delta of S. Delta of S is the set of distributions on the state space S. Now, throughout our discussions, we will want to explain things in a simplified fashion. For that reason, we will assume throughout that S, the state space, and A, the action space, are finite. In that case, this transition kernel P is characterized by the set of values P(S' | S, A), where S and S' come from the state space S, and A, which is the action, comes from the action space A.

The value P(S' | S, A) specifies the probability of the environment transitioning to a state S' from a state S under action A. So, you can think of playing the game of chess against a computer. So, the current state, which is S, is the current configuration of pieces; the action is the move that you make. Then, the computer looks at the move you have made and makes a different move, right? The move you made is A over here, right?



After the computer makes its move, the configuration of pieces will change, and the new configuration of pieces is S prime. Now, here, P(S' | S, A) specifies this random transition. When you are playing against a computer, you can imagine that you are in a state S, you play an action A, and the computer changes this configuration to S' in some random fashion. This is what is captured by this function P or the set of values P(S' | S, A). That captures the definition of this quantity P.

Next, we move on to the reward function. The reward function is denoted by this function R.

$$P: S \times A \longmapsto \Delta(s)$$

$$P\left(s'|_{s,a}\right) \forall\, s, s' \in S, a \in A$$

$$P\left(s'|_{s,a}\right)$$

$$\mathcal{U}: S \times A \mapsto \mathbb{R}$$

The domain of this function is again the Cartesian product between the state space and the action space, and the range of this function is R, which is the set of real numbers. Now, this function, R, specifies the immediate reward that the agent gets when the environment is in state S and an action A is taken. I will elaborate a bit more on this soon.

And finally, the discount factor gamma is a scalar value that lies between the interval 0 and 1. So, it is closed at 0 and open at 1.

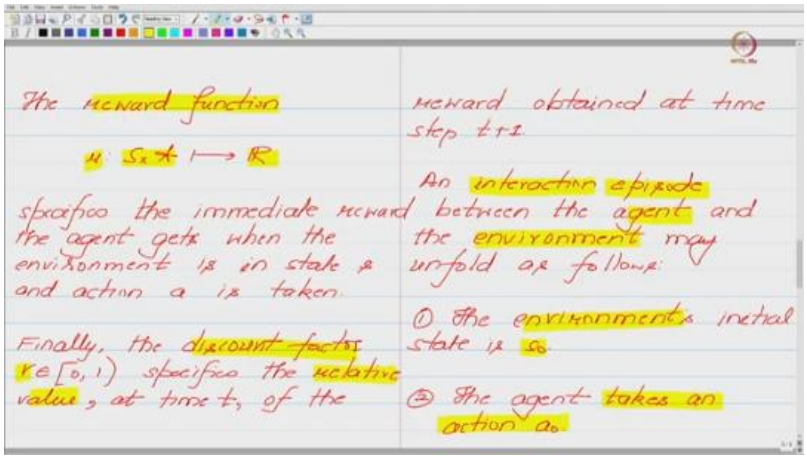
$$\gamma \in [0,1)$$

So, we will not allow gamma to take the value 1. Now, this discount factor gamma specifies the relative value at time t of the reward you obtain at time step t plus 1. So, the intuition here is as follows.

Imagine someone promised to give you 1 rupee today. It will have some value for you today. So, let us try to understand the notion of a discount factor. The idea is the following. Suppose someone promised to give you one rupee today.

It will have some value. Now, imagine someone promised to give you one rupee after one year. Now, what value will this one rupee that you will get after one year have today? What value will this one rupee that you will get after one year have for you today? You can already see it will have different values, and this difference, in some sense, is captured by this discount factor. So, with this understanding of these five different things, let us see how an interaction episode between the agent and the environment may unfold.

The environment's initial state could be S0, and then the agent takes an action A0. So, the environment is in state S0. So, in the context of chess, you can imagine some initial configuration of pieces, and then you, as an agent, take an action A0. Now, as soon as you take action A0 or the agent takes an action A0, the agent gets an immediate reward specified by $r(s_0, a_0)$. So, you can see that this reward is a function of S0, which is the initial state, and A0, which is the action that you took.

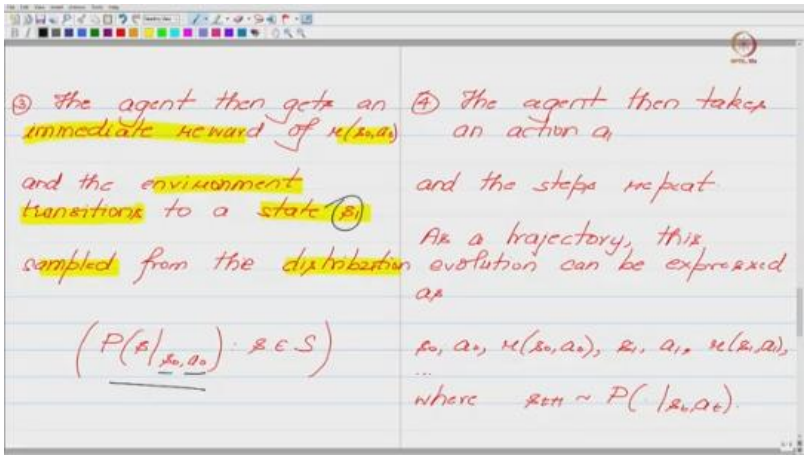


And after the agent gets this reward, the environment transitions to a new state, S1. And this new state, S1, is sampled from the distribution which is specified by P of S given S0 comma A0, where S lies in this state space S. So, let us try to understand what I mean over here. So, notice that you started at S0, A0, right? So, that is present over here.

$$\left(P\left(s|_{s_0,a_0}\right): s \in S\right)$$

Now, what is the next state? So, first, we need to understand that this next state is actually a random state, right? So, which means there is some probability of moving to, say, the first state, some probability of moving to the second state, and so on and so forth. So, which quantity describes these probabilities? Well, it is specified by this P, right?

And since we start at S0, A0, that is fixed. The next state is a random variable, right? And P of S given S0, A0 specifies the probability that, starting from S0, A0, you move to a specific state S. So, from this distribution, you sample a state, and whatever state you get is represented by S1, right? Now, later on, the agent sees the state S1 and takes the action A1, right?



And from this point on, the sequence of steps repeats. So, what do I mean by that? You are at S1. You took the action A1. So, you will now get a reward, which is a function of S1 and A1, and because you have taken the action A1, the environment will now transition to a new state S2, sampled from the distribution P of S given S1 comma A1, right?

$$\left(P\left(s|_{s_1,a_1}\right): s \in S\right)$$

So, as a trajectory, this interaction can be viewed as an evolution in the following way. That is, you start at state S0, the agent takes an action A0, then you get an immediate reward R of S0 comma A0, and the environment transitions to some random state S1. At this point in time, the agent takes an action A1 and gets a reward of R of S1 comma A1, where S_{t+1} for any t comes from the distribution P of dot given S_t comma A_t. So, the dot here represents a variable, or alternatively, you can think of this as coming from the distribution P of S given S_t comma A_t, where S comes from the state space S, capital S. So, once you are at S_t comma A_t, the next state is sampled from this distribution.

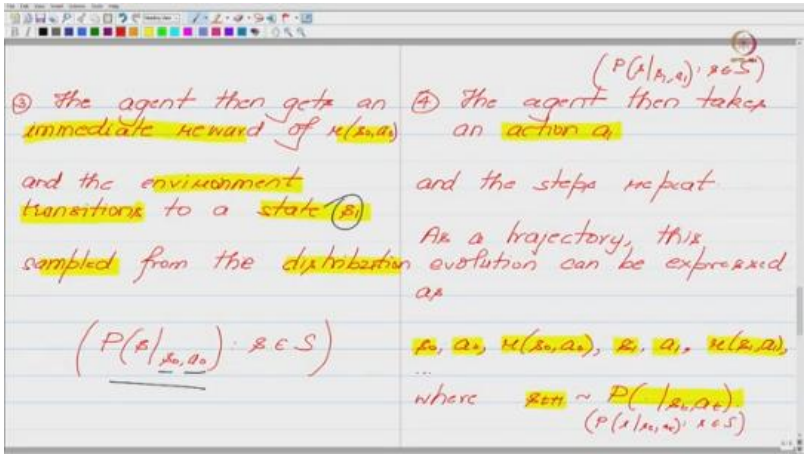
Then you move to S_{t+1}, then you pick action A_{t+1}. Once you have S_{t+1} and A_{t+1}, again, the environment first gives you a reward and then makes a transition to S_{t+2}, and so on and so forth.

$$s_0, a_0, r(s_0, a_0), s_1, a_1, r(s_1, a_1)$$

$$s_{t+1} \sim P(\ |_{s_t, a_t})$$

$$\left(P(s|_{s_t, a_t}) : s \in S\right)$$

The goal of the reinforcement learning agent then is to find the sequence of actions A0, A1, A2, A3, and so on, such that this expectation is maximized. So, let us look at this expectation. First of all this expectation involves an infinite sum, right?



So the index of this infinite sum is t which denotes time and it starts from zero and goes all the way up till infinity, right? And rst comma at specifies the reward that you get at time

step t, right? so rst comma 80 is the reward that you get at time step t And this gamma to the power t is multiplying it. This denotes the value that this reward that you obtain at time step t, you have at time step 0. So, you have some relative quantification of how good that reward, which you will get at time step t, holds for you at time step 0.

That is why you multiply by gamma to the power t, and one can see that since gamma is strictly less than 1, as t increases, the value of gamma to the power t decreases. So, which means that the reward that you get far away into the future has less importance to you today compared to the reward that you will get in some nearby time instance.

$$\mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t \; r(s_t, a_t) | s_0 = s\right]$$

So, you look at this infinite sum. So, this infinite sum, because the states are randomly evolving, will be a random variable. And hence, we look at its expectation, which makes sense, and which will now be a real number.

And this expectation has some conditioning which specifies from which state you start. So, let us go over it one more time. You start at this state S, right, and then you take this action a0, then you move to s1, right? So at that point in time, you will get this reward r(s0, a0). Right, then you move to state s1, right, and then you take this action a1. So at that point in time, you will get this reward r(s1, a1). Then you move to state s2—by 'you,' I mean the environment moves to state s2—and you, as an agent, take this action a2.

And you will get this reward r(s2, a2). What value this reward that you will get at time step 2, you know, relatively holds for you at time step 0, right?

$$r(s_0, a_0) + \gamma \, r(s_1, a_1) + \; \gamma^2 \, r(s_2, a_2) + \cdots$$

So, this is gamma squared r(s2, a2), and so on. Is this okay? So, you can see that this infinite sum is described in this fashion, right? And this sum is random because the states that you observe are random, right?

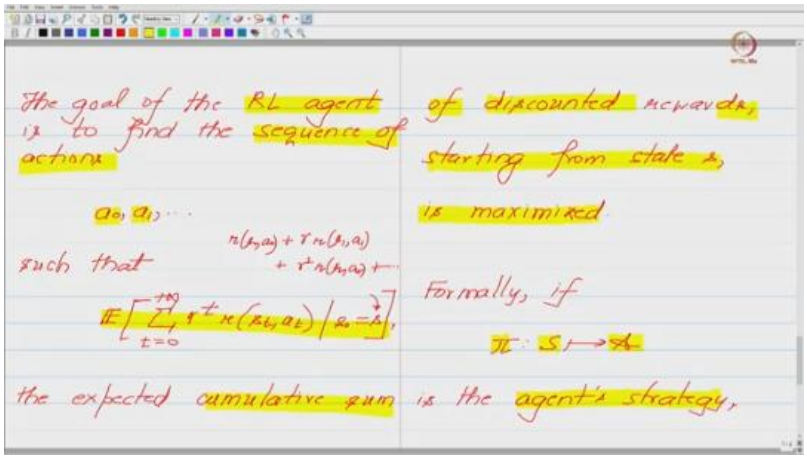
And the goal of the RL agent is to find the sequence of actions so that the expected value of this cumulative sum of discounted rewards, starting from the state S, is maximized. So,

this is the goal of the RL agent. So, we can state this more formally. Suppose you have a policy or a strategy pi. So, I am introducing this formal description for the first time.

So, here the agent took these actions A0, A1, and so on, but how does the agent decide which action to take? That is described by the agent's strategy, and we will formally denote this by the symbol pi. Now, what is pi? Pi is a function, right? Its domain is the state space, and the range is the action space.

$$\pi: S \mapsto A$$

So, an agent's strategy can be viewed in the following way: you know the environment is in a state s, and pi of s specifies the action that the agent prefers to take. Similarly, if the environment is in state s prime then pi of s prime specifies the action that the agent plans to take and so on and so forth. So, now given agents strategy or policy, so I will interchangeably use this word strategy and policy. Now given agent strategy, its quality is measured via vector which we denote by V superscript pi. So, once in a while I will also denote it by V subscript pi.



So, this vector is a vector of size, I should maybe emphasize that it is the vector of size cardinality of the state space S. So, if you have two possible states, then V pi will be a vector of size 2, it will be a vector in R2. If similarly, if the cardinality of the state space is 100, 1 billion, 1 trillion, then V pi will be a vector of corresponding dimension. So given strategy or policy of an agent which we denote by pi, its quality is quantified by a vector which we denote by v pi and we call this vector as the value function. Now, what is the s-

th coordinate of this vector V pi? Well, the s-th coordinate of this vector V pi is given by this expectation.

$$V^{\pi} \in \mathbb{R}^{|S|}$$

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t}\ r(s_t, \pi(s_t)|s_0 = s)\right]$$

So, let us go over this expectation and see how this differs from what we saw before. You see this 's' over here. This 's' prescribes the starting state. So, the s-th coordinate of V pi means that you start from state S and thereafter pick your actions as prescribed by your policy pi. So, what do I mean by that?

So, you start from S. Now, since you are at state S, pi specifies an action pi of S. Now, since this action is specified, you will get a reward R of S comma pi of S, and then you will make a transition to the state S1. By 'you' again, I mean the environment makes a transition to the state S1. Now at S1, the agent again decides to take the action pi of S1. Recall that pi is a function from the state space to the action space.

So, pi of S, pi of S1, and so on and so forth—they are actions. So, as soon as the agent takes this action, it will get the reward R of S1 comma pi of S1, right? And so on and so forth, right? So, again, we can look at the sum of rewards. So, maybe I will write it over here. So, we will look at the sum of rewards R of S0 comma pi of S0, right?

Plus gamma times r of s1 comma pi of s1 plus gamma squared r of s2 pi of s2 and so on and so forth.

$$s, \pi(s), r\big(s, \pi(s)\big), s_1, \pi(s_1), r\big(s_1, \pi(s_1)\big), r, \ldots$$

$$r\big(s_0, \pi(s_0)\big) + \gamma r\big(s_1, \pi(s_1)\big) + \gamma^2 r\big(s_2, \pi(s_2)\big) + \cdots$$

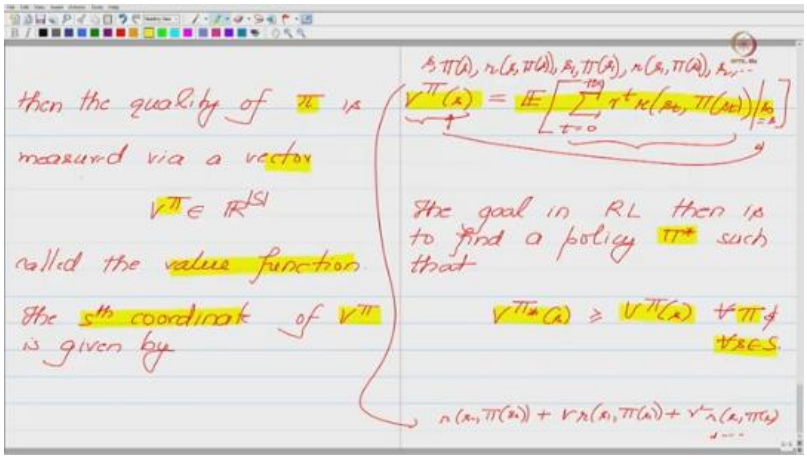
So, again we can see that you have a cumulative sum of discounted rewards, right? Again, this sum is random because the evolution of the states is random, and it is random because, you know, the environment evolves randomly, which is as prescribed by this transition

function P. So, this infinite sum is random, and we are interested in looking at the expected value of this. So, what is V pi of S?

Well, it is the expected value of this cumulative sum of discounted rewards. The difference between this infinite sum and what we saw before was that previously it was some arbitrary sequence of actions. However, in this case, the sequence of actions is as prescribed by this policy pi. So, that is why this left-hand side is V pi of S. Now, what is the goal of reinforcement learning?

$$V^{\pi_*}(s) \geq V^{\pi}(s)\ \forall \pi \notin \forall s \in S$$

Well, the goal in reinforcement learning is to find a policy pi star such that V pi star of S is greater than or equal to V pi of S for all policies pi and all states little s, right? So, in words, the goal of reinforcement learning can be said to be to find a strategy or a policy pi star. So, what does a strategy or a policy specify? It specifies which actions to take at which states. So, you can imagine that, you know, when you are playing chess, you look at a configuration of pieces and say, okay, I will take this action.



You look at a different configuration of pieces, which corresponds to a different state of the environment, and you say, 'Okay, I will take this action now.' So, you come up with a strategy which says, at this configuration of pieces, you will take this action; at a different configuration of pieces, you will take this action, and so on and so forth. And you can see that this mapping from the configuration of pieces, which specifies the state space and the action space, corresponds to your strategy. And the goal in reinforcement learning is to

find a strategy or a policy, pi-star, which gives you the largest returns. By returns, I mean the expected sum of discounted cumulative rewards.

So, you can think of playing this game forever and looking at the rewards you get, and you would like to get the maximum sum of these rewards. And in that way, pi-star is the policy such that V-pi-star of S is bigger than V-pi of S. And notice that we require that pi-star be better than pi for any starting state S. So, with this, we come to the end of the lecture. Now, I will summarize what we have done in this class.

So, as I said in the previous lectures, we looked at examples of stochastic approximation, but they were SGD methods in disguise. In this lecture and the next, we want to look at an example of stochastic approximation which is not an SGD method. Towards that, I gave you a basic introduction to reinforcement learning and tried to explain to you what the goal of reinforcement learning is. In particular, I told you what the value function of a strategy or a policy pi is. And I told you the goal in reinforcement learning is to find the strategy which has, in some sense, the largest value function.

And one of the intermediate goals in reinforcement learning is what is called policy evaluation, which basically refers to the problem of finding V pi given pi. And we will discuss an algorithm called the temporal difference learning algorithm in the next lecture to estimate this V pi vector. And you will see that the temporal difference learning algorithm that we will discuss cannot be expressed as an SGD, and hence in that sense, that is an example of a stochastic approximation algorithm that is not an SGD. With this, let me stop lecture 4. Thank you and Namaste.