STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Engineering

Indian Institute of Science, Bangalore

Week 12

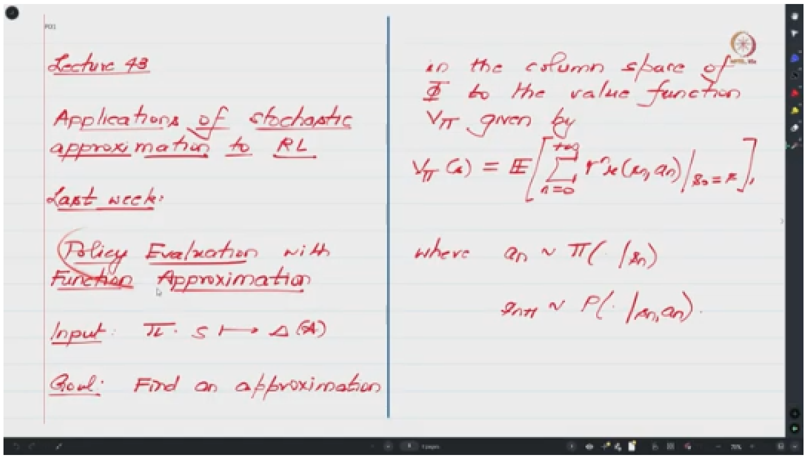
Lecture 43

Best Policy Algorithm for Q-Value Functions: A Stochastic Approximation Formulation

Hello and Namaste everyone. Welcome to lecture 43 of this NPTEL course on Stochastic Approximation. In the last week and the next couple of lectures, we are looking at applications of Stochastic Approximation to the reinforcement learning context. In the previous week, we focused on this sub-problem within reinforcement learning called policy evaluation, wherein the goal was given a policy, can you quantitatively say how good is this policy? And we looked at this problem in the context of what is known as function approximation, which is useful when the state space is very, very large.

And then we designed the TD0 algorithm and came up with an analysis of the asymptotic behavior of this algorithm using the stochastic approximation theory that we had studied over this course. This week, what we will do is we will go beyond policy evaluation and try to come up with an algorithm using stochastic approximation principles for identifying the optimal policy. So, in the previous week, we said given a policy how good is the policy, and we sort of evaluated the value function. This week, our goal would be can we use similar ideas as in our previous week to, in fact, identify the best policy itself. So, let us do a formal study of this goal right.

So, as I said last week, right, we looked at the problem of policy evaluation where given a policy pi, we wanted to, you know, get an estimate of V pi; in particular, we looked at this context of function approximation. So, wherein the goal is can we find the value of V pi or can we get an approximation to V pi in the column space of some feature matrix? I mean, I should say this is not fully the function approximation; I mean, this is the special case of function approximation called linear function approximation. So, in practice, people usually work with non-linear function approximation by making use of neural networks and so on. But whatever ideas we are studying now, similar ideas will be applicable in the context of, you know, neural networks as well.



Lecture 48

Applications of stochastic approximation to RL

Last week:

Policy Evaluation with Function Approximation

Input: $\pi : S \longmapsto \Delta(A)$

Goal: Find an approximation in the column space of Φ to the value function V_π given by

$$V_\pi(s) = \mathbb{E}\left[\sum_{n=0}^{+\infty} \gamma^n r(s_n, a_n) \Big/ s_0 = s\right],$$

where $a_n \sim \pi(\cdot / s_n)$

$s_{n+1} \sim P(\cdot / s_n, a_n)$.

So, this was the problem that we studied in the previous class, and recall that the definition of the value function of a policy pi was that it is a vector of size equal to the cardinality of the state space, and the s-th coordinate of this vector equals the expectation of this infinite discounted sum of intermediate rewards where the state S_n is sampled from the transition probability, and this action a_n comes from your policy pi.

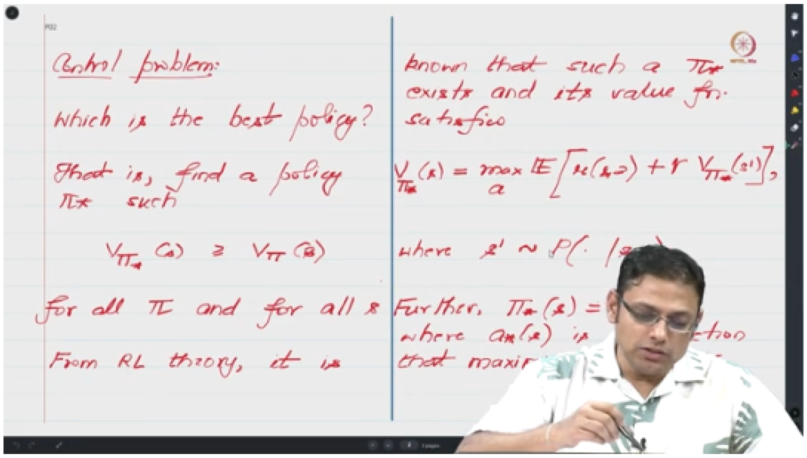
$$\Pi: S \mapsto \Delta(A)$$

$$V_{\Pi}(s) = E\left[s_0 = s\right]$$

$$a_n = \Pi\left(s_n\right)$$

$$s_{n+1} \sim p(\ |s_n, a_n)$$

So this was our understanding. So today, what we want to do is we want to look at this control problem. In other words, we want to ask, okay, I know how good a given policy is, but I would like to know which is the best policy.



Control problem:

Which is the best policy?

That is, find a policy π_* such

$$V_{\pi_*}(s) \geq V_{\pi}(s)$$

for all π and for all s

From RL theory, it is known that such a π_* exists and its value fn. satisfies

$$V_{\pi_*}(s) = \max_{a} \mathbb{E}\left[r(s,a) + \gamma V_{\pi_*}(s')\right],$$

where $s' \sim P(\cdot \mid s$

Further, $\pi_*(s) =$

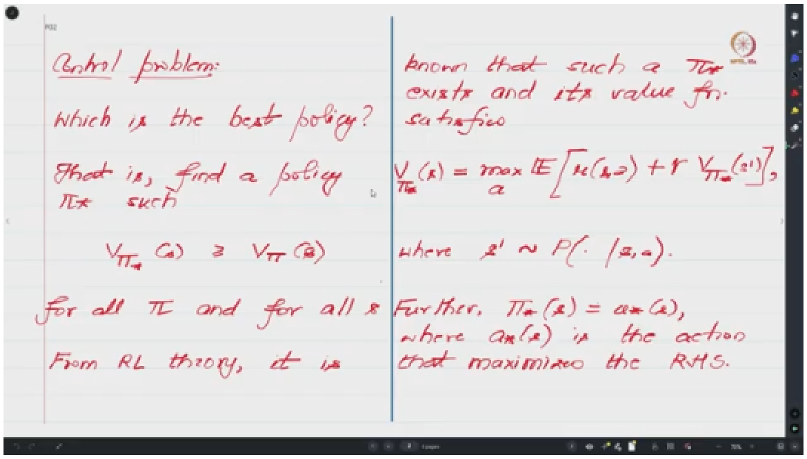
where $a_*(s)$ is ction

that maxi

Can we somehow figure it out? In other words, we want to find a policy, let us denote it as pi star, such that v pi star, that is the value function of pi star, is better than the value function of any other policy at every state s. Recall that this s over here denotes the initial state. It says if I start at s, and you know this quantity over here, the interpretation is that if you start at s, and if you always act according to the policy pi, what is the expected discounted sum of, or expected sum of discounted rewards, that we will get starting from state s and acting, you know, as per policy pi. This is the quantity over here.

So we want to find the policy pi star which has this property—that is, you would get the highest returns compared to any other policy, and you know that statement holds true from whichever state you start from. Right? So, from the theory of reinforcement learning—which I am not going to cover in this course—right, so you can look up the standard textbooks on reinforcement learning to understand the statement that I am going to make, which is that, from the theory of reinforcement learning, it is first of all known that such a policy pi star exists and its value function, that is the value function of your policy pi star, satisfies a relation of the following form. So, of course, if you remember, I had mentioned some Bellman equation that V pi satisfies. So pi star will also satisfy that same relation where pi is replaced by pi star, but pi star is also special in that it satisfies one more type of Bellman equation, which is known as the Bellman optimality equation, and that equation is stated over here.

So what is this relation? It says that if you look at the s-th coordinate of your V pi star, then this equals the max of this quantity over A. So let us understand what is written over here. So on the right hand side, if you see, you have the immediate reward plus gamma times the returns of from state S prime onwards. So the way to interpret this is you start at S, take an action A, you will get this reward, and from there you will sort of randomly reach to some other state. So from that state onwards, you act according to pi star, right, and you know then you would get this expected return. So if you take the expectation of this quantity where S prime is the random variable in this expression and if you take the max over A, so notice that A here is not, at least in this expression, it is not chosen to be pi star.



Control problem:

Which is the best policy?

That is, find a policy π_* such

$$V_{\pi_*}(s) \geq V_{\pi}(s)$$

for all π and for all s

From RL theory, it is known that such a π_* exists and its value fn. satisfies

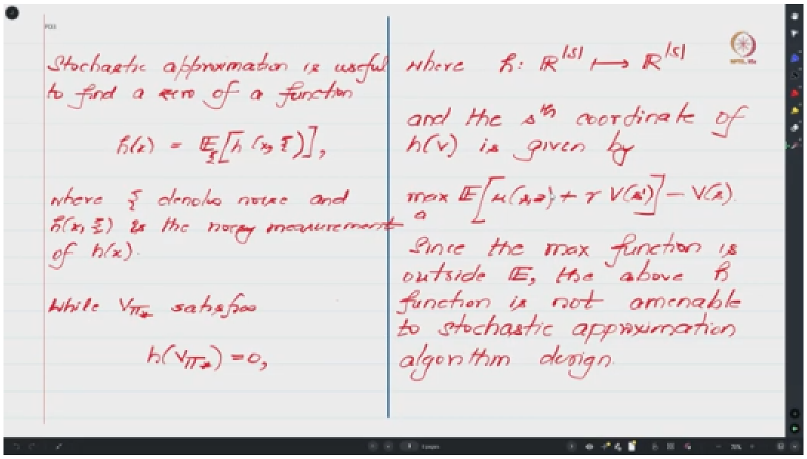
$$V_{\pi_*}(s) = \max_{a} \mathbb{E}\left[r(s,a) + \gamma V_{\pi_*}(s')\right],$$

where $s' \sim P(\cdot \mid s,a)$.

Further, $\pi_*(s) = a_*(s)$, where $a_*(s)$ is the action that maximizes the RHS.

So you pick an A and then look at the expected value and take the max over A. So your V pi star of S, you know, will be equal to the max of this quantity. So in some sense, you know, V pi star satisfies this equation and it somehow tells you that V pi star gives you the maximum possible return. And this is known as the Bellman optimality equation. And as I said, the expectation here is with respect to this random variable S prime, and S prime is chosen according to this distribution.

And from the theory of reinforcement learning and based on this equation, one can then figure out that the optimal policy pi star is actually suggests at state S to take the action which maximizes this right hand side. So if you see, you know, there is a max over A. So whichever action maximizes the right hand side, your pi star S will suggest taking that action itself. Right, so this is the, you know, interesting properties of this optimal policy, and our goal for today is to see if we can somehow exploit these two properties and figure out pi star. However, if you notice in this relation, the max and the expectation have, you know, the max appears before the expectation. Right now, in stochastic approximation, typically, right, we want to, I mean, the goal of stochastic approximation or stochastic approximation is useful to find the zero of a function, okay, which is in some sense of this form, right.



Stochastic approximation is useful to find a zero of a function

$$\bar{h}(x) = \mathbb{E}_{\xi}\left[h(x, \xi)\right],$$

where ξ denotes noise and $\bar{h}(x, \xi)$ is the noisy measurement of $h(x)$.

While V_{π_*} satisfies

$$h(V_{\pi_*}) = 0,$$

where $\bar{h}: \mathbb{R}^{|S|} \longmapsto \mathbb{R}^{|S|}$

and the s^{th} coordinate of $h(V)$ is given by

$$\max_{a} \mathbb{E}\left[r(s,a) + \gamma V(s')\right] - V(s).$$

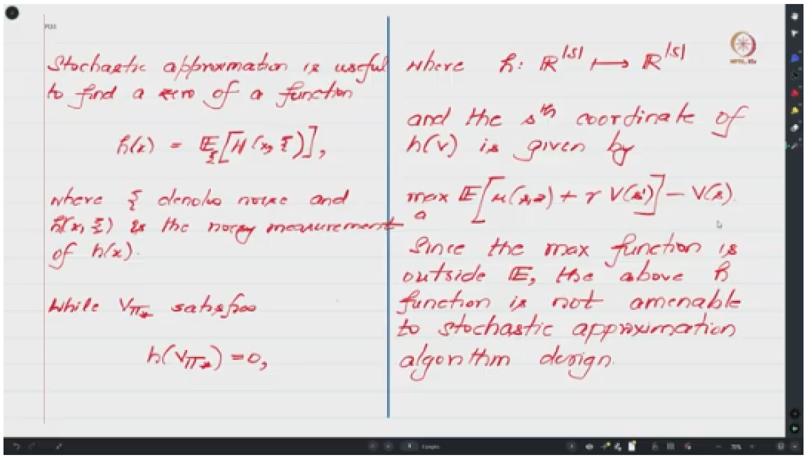
Since the max function is outside $\mathbb{E}$, the above $\bar{h}$ function is not amenable to stochastic approximation algorithm design.

So, let me elaborate what we have over here, and maybe I should use a different notation here. So, let us say this is, you know, some capital H over here. Is this okay? So let us say little h of x is the expected value of capital H of x, xi, right? And the expectation is with respect to xi.

So the idea here is h of x is the expectation of this quantity where xi denotes noise, right? And capital H of x, xi is the noisy measurement of h of x. So suppose you wanted to find the zero of little h of x and h of x has this property, then you can design stochastic approximation algorithms for solving such kinds of questions. However, in the previous slide, you observed that the max and the expectation sort of appear in the wrong order, so if you sort of try to find, you know, define H in the following way: so let's say H is a function which goes from this space to this space, right? And you know, so H of V will be an element of size equal to the cardinality of the state space, and similarly H of V will be a vector whose size equals the cardinality of state space, then if you look at the s-th coordinate of this vector

Right? One can, based on the previous equation, define it in, you know, H of V's S-th coordinate in this way. Right? And one can then see that your V pi star will indeed be the zero of this function. Right?

However, this function and this function, if you see here, there is a max sitting outside the expectation. So, because the max function is sitting outside the expectation. Right? Finding the zero of this function in some sense is not natural in the stochastic approximation context. Right?



Stochastic approximation is useful to find a zero of a function

$$h(x) = \mathbb{E}_{\xi}[H(x, \xi)],$$

where ξ denotes noise and $H(x, \xi)$ is the noisy measurement of $h(x)$.

While V_{π_*} satisfies

$$h(V_{\pi_*}) = 0,$$

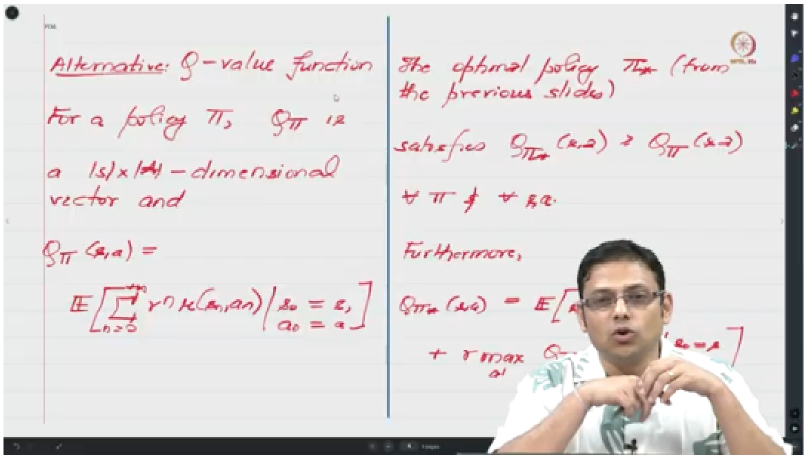
where $h: \mathbb{R}^{|S|} \longmapsto \mathbb{R}^{|S|}$

and the s^{th} coordinate of $h(V)$ is given by

$$\max_{a} \mathbb{E}\left[r(s,a) + \gamma V(s')\right] - V(s).$$

Since the max function is outside $\mathbb{E}$, the above h function is not amenable to stochastic approximation algorithm design.

So, hence the question that we would like to ask is: can we somehow come up with a relation where the expectation and max are in the right order? That is, you have the max sitting inside the expectation. If that was the case, we could have designed a stochastic approximation algorithm for finding the zero of that kind of function. So, towards that, the alternative is to make use of what is called a Q-value function. So, as I said, for a policy pi, V pi measures how good it is.



Alternative: Q-value function

For a policy π, Q_π is a $|S| \times |A|$-dimensional vector and

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{n \geq 0}^{\infty} \gamma^n r(s_n, a_n) \,\middle|\, s_0 = s, a_0 = a\right]$$

The optimal policy π_* (from the previous slides)

satisfies $Q_{\pi_*}(s,a) \geq Q_\pi(s,a)$

$\forall \pi$ & $\forall s, a$.

Furthermore,

$Q_{\pi_*}(s,a) = \mathbb{E}[$

$+ \gamma \max_{a'} Q$ $s_0 = s]$

Now, Q pi is another quantity that similarly measures how good it is. But the difference between V pi and Q pi is the following. V pi is a vector that is of size equal to the cardinality of the state space; however, Q pi is a vector whose size equals the cardinality of the state-action space product. Right?

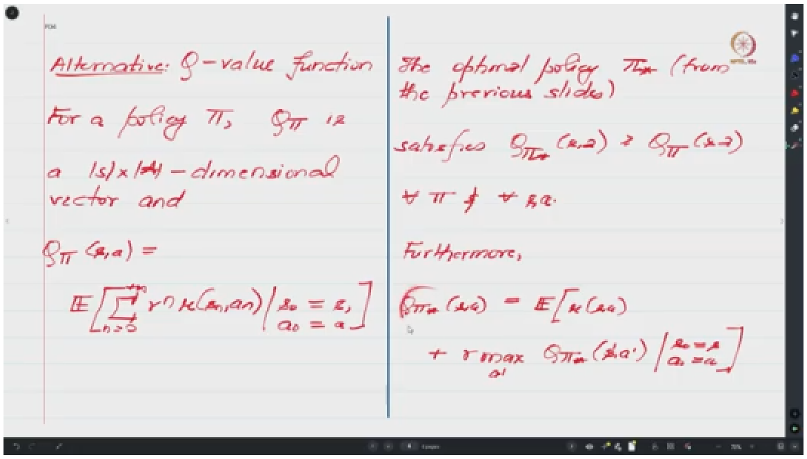
So Q pi actually sits in a bigger dimensional space. And what we will show is that A policy that is optimal in the V value function sense is also optimal in the Q value function sense, and the Q value also satisfies a Bellman optimality relation like your V pi star, and this Bellman optimality relation for Q pi star actually has the desired form; that is, the expectation appears first and the max appears later. So let us discuss this more formally. So, for a policy pi, the Q pi, right, is the value function, and it is known as the state-action value function or the Q value function associated with policy pi.

So, this is, as I said, of size equal to the product of the state and action space cardinalities. And if you look at the SA-th coordinate of Q pi, then it has the following definition. Now, if you notice, this definition and the definition that we had for V pi share a lot of similarities; in particular, this sum over here is very similar to what we had in the definition of V pi, right? And similarly, we had an expectation here. The difference is in what is there in the conditioning.

So, if you look at the definition of V pi, you only have conditioning with respect to S0 equals S. Whereas in the definition of Q pi, you can observe that in addition to requiring that S0 be S, you also require that A0 be A. So the way to interpret this definition is that suppose you start at state S. Forcefully pick an action A. So, this action A need not come from this policy pi, right? So, you pick a state and forcefully pick an action A, right? And from the next point onwards, right?

Or from the next state onwards, you act according to policy pi. That is, you pick an action that is recommended by your policy pi. And whatever will be the expected value of the infinite sum of discounted rewards that is referred to as Q pi SA. So I hope you are able to see the distinction between V pi and Q pi and one can show that the policy pi star that is optimal in the V pi sense

is also optimal in the Q pi sense and in fact that same optimal policy satisfies the relation of this form that is Q pi star SA is greater than or equal to Q pi SA for all pi and for all SA. Furthermore, one can show that Q pi star satisfies the Bellman optimality relation which is that Q pi star SA equals the expected value of this quantity on the right hand side and I should perhaps be careful here. Here I should not put this conditioning. It is just the expectation here and the random variable here is S prime. So, this SA and this SA are the same.



Alternative: Q-value function

For a policy π, Q_π is a $|S| \times |A|$-dimensional vector and

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{n \geq 0}^{\infty} \gamma^n r(s_n, a_n) \,\middle|\, \begin{array}{l} s_0 = s, \\ a_0 = a \end{array}\right]$$

The optimal policy π_* (from the previous slide)

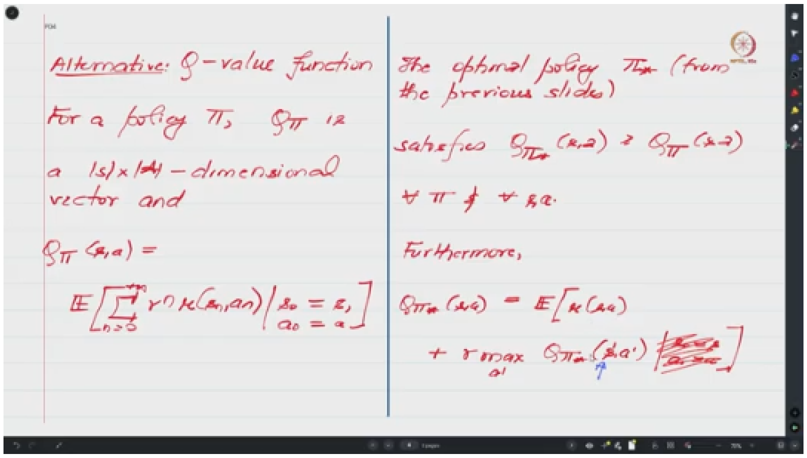
satisfies $Q_{\pi_*}(s,a) \geq Q_\pi(s,a)$

$\forall \pi$ & $\forall s,a$.

Furthermore,

$$Q_{\pi_*}(s,a) = E\left[r(s,a) + \gamma \max_{a'} Q_{\pi_*}(s',a') \,\middle|\, \begin{array}{l} s_0 = s \\ a_0 = a \end{array}\right]$$

So, the random variable that is there in this expression is S prime. So, the reason I say this is nice because you can see that the expectation and the max appear in the right order. By right here, I mean in the form that is amenable to stochastic approximation algorithm design. So, this relation can also be formally written in the following way that is Q pi star that is the Q value of the optimal policy is a fixed point of the Bellman optimality operator T which is given in the following way that is T you know takes as input SA dimensional vector and spits out another SA dimensional vector right and the SAth coordinate so if you give Q as input TQ will be a capital S cross A dimensional vector and if you look at the little SAth coordinate of TQ



Alternative: Q-value function

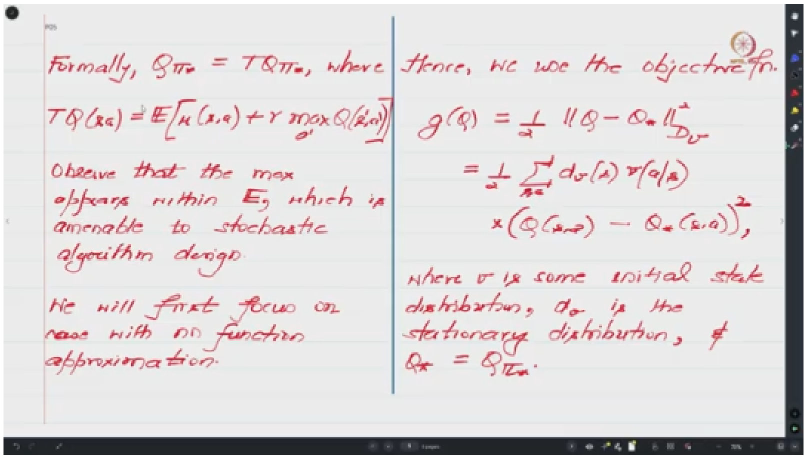
For a policy π, Q_π is a $|S|\times|A|$-dimensional vector and

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{n\geq 0}^{\infty} \gamma^n r(s_n,a_n) \,\Big|\, \begin{matrix} s_0 = s, \\ a_0 = a \end{matrix}\right]$$

The optimal policy π_* (from the previous slide) satisfies $Q_{\pi_*}(s,a) \geq Q_\pi(s,a)$ $\forall \pi$ & $\forall s,a$.

Furthermore,

$$Q_{\pi_*}(s,a) = \mathbb{E}\Big[r(s,a) + \gamma \max_{a'} Q_{\pi_*}(s',a')\Big]$$



Formally, $Q_{\pi_*} = TQ_{\pi_*}$, where

$$TQ(s,a) = \mathbb{E}\Big[r(s,a) + \gamma \max_{a'} Q(s',a')\Big]$$

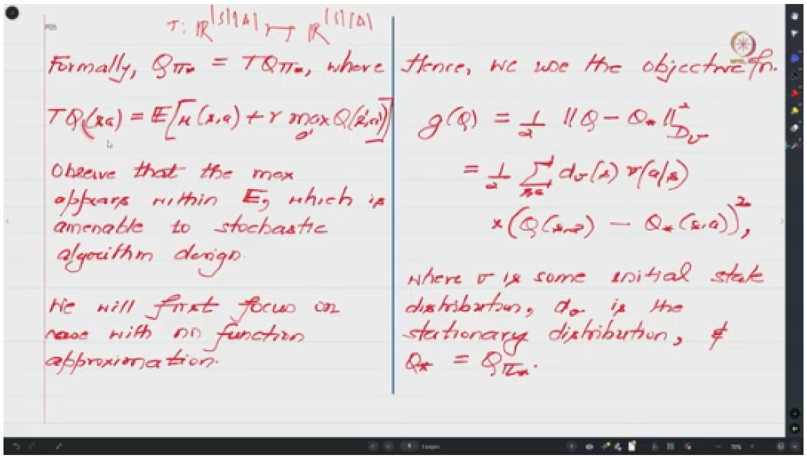
Observe that the max appears within $\mathbb{E}$, which is amenable to stochastic algorithm design.

We will first focus on case with no function approximation.

Hence, we use the objective fn.

$$g(Q) = \frac{1}{2}\|Q - Q_*\|^2_{D_\nu} = \frac{1}{2}\sum_{s,a} d_\nu(s)\,\pi(a|s) \times \big(Q(s,a) - Q_*(s,a)\big)^2,$$

where ν is some initial state distribution, d_ν is the stationary distribution, & $Q_* = Q_{\pi_*}$.



$T: \mathbb{R}^{|S||A|} \mapsto \mathbb{R}^{|S||A|}$

Formally, $Q_{\pi_*} = TQ_{\pi_*}$, where

$$TQ(s,a) = \mathbb{E}\Big[r(s,a) + \gamma \max_{a'} Q(s',a')\Big]$$

Observe that the max appears within $\mathbb{E}$, which is amenable to stochastic algorithm design.

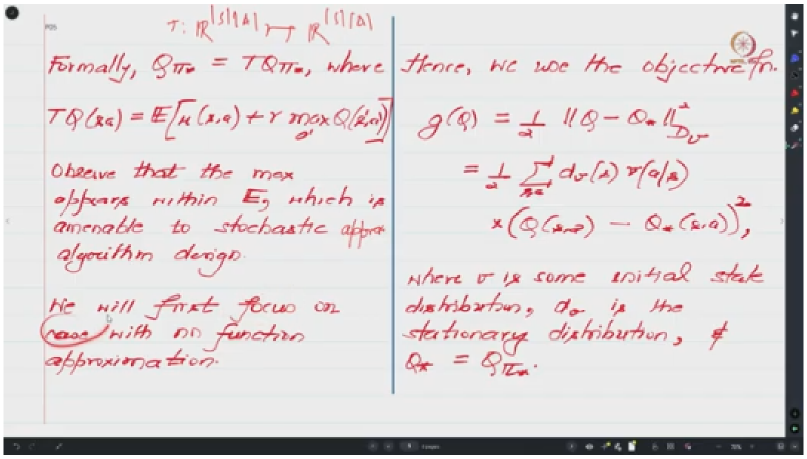
We will first focus on case with no function approximation.

Hence, we use the objective fn.

$$g(Q) = \frac{1}{2}\|Q - Q_*\|^2_{D_\nu} = \frac{1}{2}\sum_{s,a} d_\nu(s)\,\pi(a|s) \times \big(Q(s,a) - Q_*(s,a)\big)^2,$$

where ν is some initial state distribution, d_ν is the stationary distribution, & $Q_* = Q_{\pi_*}$.

It will be given in the following way. And as I said here, the max and expectation are in the right order, which means that we can design a stochastic approximation algorithm to find this Q pi star. So let us begin our formal analysis of this. And to keep things simple, what we are going to do is we are going to focus on the case with no function approximation, right? When we were doing policy evaluation, we directly went ahead with the function approximation case.



$T: \mathbb{R}^{|S||A|} \mapsto \mathbb{R}^{|S||A|}$

Formally, $Q_{\pi_*} = TQ_{\pi_*}$, where

$TQ(s,a) = \mathbb{E}[r(s,a) + \gamma \max_{a'} Q(s',a')]$

Observe that the max appears within $\mathbb{E}$, which is amenable to stochastic approx algorithm design.

We will first focus on case with no function approximation.

Hence, we use the objective fn.

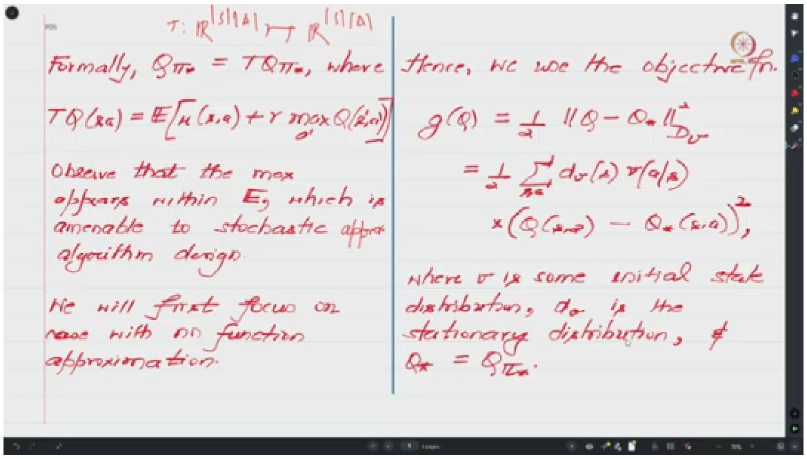
$$g(Q) = \frac{1}{2} \|Q - Q_*\|^2_{D_\nu}$$

$$= \frac{1}{2} \sum_{s,a} d_\nu(s) \pi(a|s) \times \left(Q(s,a) - Q_*(s,a)\right)^2,$$

where ν is some initial state distribution, d_ν is the stationary distribution, & $Q_* = Q_{\pi_*}$.

However, in this control problem, we will first focus on the no function approximation case, and in the subsequent lectures, we will actually see what happens with function approximation and so on, right? So, for the next few lectures, we will focus on the no function approximation case. So let us follow the same principles that we had gone over, used last time to design the algorithm. So let us first come up with an objective function. Let us say g of q is the distance between q and q star.

Right. And this time what we will do is we will take the norm that is, you know, influenced by d mu. Right. Where mu is some initial state distribution and little d mu is the stationary distribution associated with the Markov chain induced by mu. Right.



$T: \mathbb{R}^{|S||A|} \mapsto \mathbb{R}^{|S||A|}$

Formally, $Q_{\pi_*} = TQ_{\pi_*}$, where

$TQ(s,a) = \mathbb{E}[r(s,a) + \gamma \max_{a'} Q(s',a')]$

Observe that the max appears within $\mathbb{E}$, which is amenable to stochastic approx algorithm design.

We will first focus on case with no function approximation.

Hence, we use the objective fn.

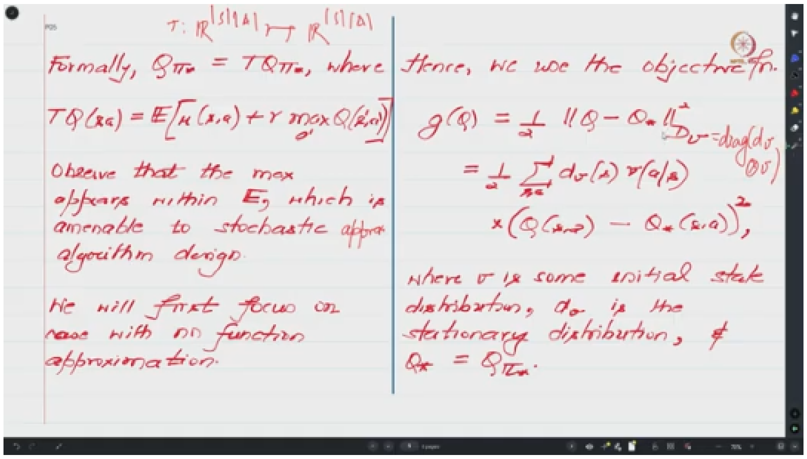
$$g(Q) = \frac{1}{2} \|Q - Q_*\|^2_{D_\nu}$$

$$= \frac{1}{2} \sum_{s,a} d_\nu(s) \pi(a|s) \times \left(Q(s,a) - Q_*(s,a)\right)^2,$$

where ν is some initial state distribution, d_ν is the stationary distribution, & $Q_* = Q_{\pi_*}$.

And this capital d mu is basically equal to diag of mu. d mu. Is this okay? And I should perhaps say it is a diag of d mu times or this times mu. I will explain what I mean by this.

So your d mu is actually a diagonal matrix of size equal to the cardinality of the state space and the action space. And if you look at the s ath coordinate of this matrix then that will be d nu of s times nu of a given s okay so this is the s ath coordinate so let me just denote it by d nu of s comma a equals this right so this is what this matrix over here is so now You know, you may ask why we are having this d mu over here, you know, and why do we not have something perhaps that was similar to what we had in the policy evaluation case. So, first let us recall what we had in the policy evaluation case.



$T: \mathbb{R}^{|S||A|} \mapsto \mathbb{R}^{|S||A|}$

Formally, $Q_{\pi_*} = TQ_{\pi_*}$, where

$TQ(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q(s',a')\right]$

Observe that the max appears within E, which is amenable to stochastic approx algorithm design.

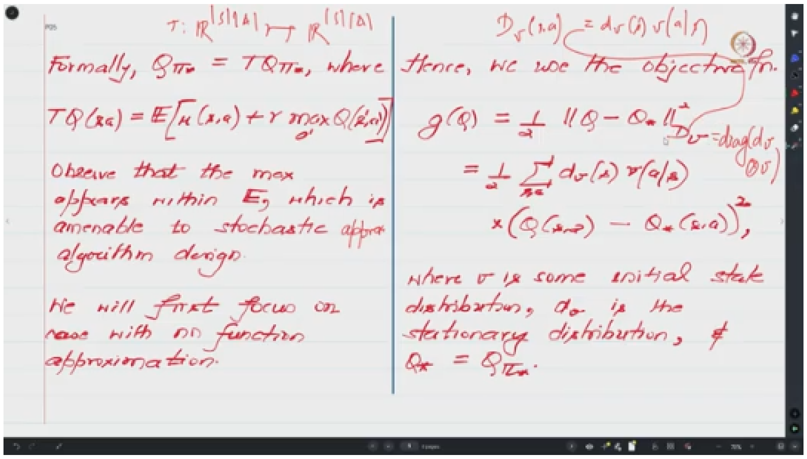
We will first focus on case with no function approximation.

Hence, we use the objective fn.

$$g(Q) = \frac{1}{2} \|Q - Q_*\|^2_{D_v} \quad = \text{diag}(d_v \otimes v)$$

$$= \frac{1}{2} \sum_{s,a} d_v(s)\, v(a|s) \times \left(Q(s,a) - Q_*(s,a)\right)^2,$$

where v is some initial state distribution, d_v is the stationary distribution, & $Q_* = Q_{\pi_*}$.



$D_v(s,a) = d_v(s)\, v(a|s)$

$T: \mathbb{R}^{|S||A|} \mapsto \mathbb{R}^{|S||A|}$

Formally, $Q_{\pi_*} = TQ_{\pi_*}$, where

$TQ(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q(s',a')\right]$

Observe that the max appears within E, which is amenable to stochastic approx algorithm design.

We will first focus on case with no function approximation.

Hence, we use the objective fn.

$$g(Q) = \frac{1}{2} \|Q - Q_*\|^2_{D_v} \quad = \text{diag}(d_v \otimes v)$$

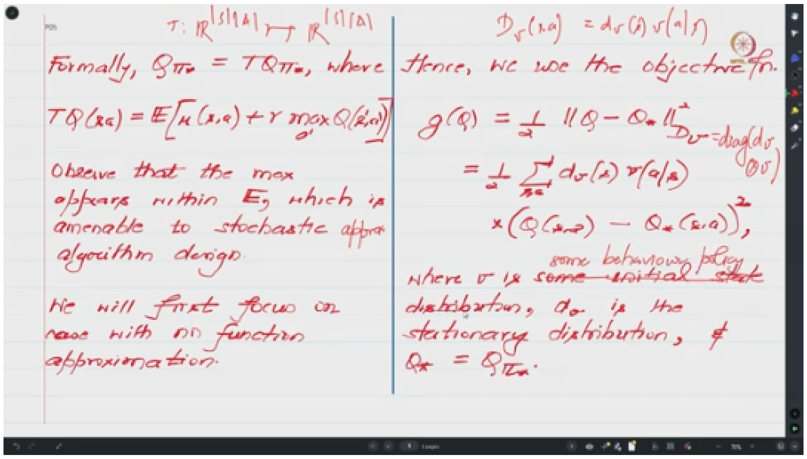
$$= \frac{1}{2} \sum_{s,a} d_v(s)\, v(a|s) \times \left(Q(s,a) - Q_*(s,a)\right)^2,$$

where v is some initial state distribution, d_v is the stationary distribution, & $Q_* = Q_{\pi_*}$.

So, in the policy evaluation case, given a policy pi, we wanted to evaluate v pi and when we define this, you know, norm over here, we had d pi sitting over here. Now, in the case of control we do not know the optimal policy. I mean in policy evaluation we say look here is a policy or here is a strategy tell me how good it is. So, in that problem the policy Pi is known. However, in the control problem Pi star is unknown.

If we knew Pi star, then we could have, you know, just evaluated its value. You know, the goal is, like, without knowing pi star, can you somehow find q pi star? That is the kind of question that we are trying to answer. So here, you know, we sort of take an arbitrary, you know, so I should be careful here. I think I made a mistake.

So here, mu is not the initial state distribution. Sorry about that. Here, mu is some behavior policy. So mu is some behavior policy. It is not the initial state distribution.



$T: \mathbb{R}^{|S||A|} \mapsto \mathbb{R}^{|S||A|}$

Formally, $Q_{\pi_*} = TQ_{\pi_*}$, where

$TQ(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q(s',a')\right]$

Observe that the max appears within E, which is amenable to stochastic appr algorithm design.

We will first focus on case with no function approximation.

$D_\nu(s,a) = d_\nu(s)\,\nu(a|s)$

Hence, we use the objective fn.

$$g(Q) = \frac{1}{2}\|Q - Q_*\|^2_{D_\nu = diag(d_\nu \otimes \nu)}$$
$$= \frac{1}{2}\sum_{s,a} d_\nu(s)\,\nu(a|s) \times \left(Q(s,a) - Q_*(s,a)\right)^2,$$

where ν is some ~~initial state~~ behaviour policy distribution, d_ν is the stationary distribution, & $Q_* = Q_{\pi_*}$.

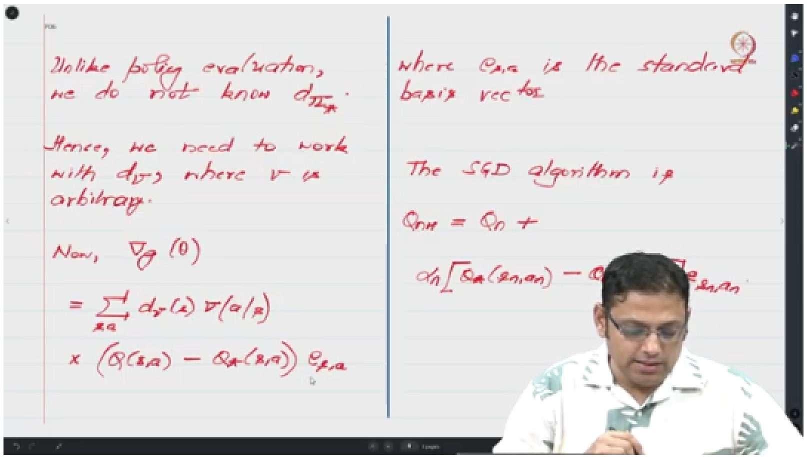
Sorry about that. Okay. Nu is some behavior policy. So what we will do is we will interact with the environment using nu. Right.

And the way to imagine is you start at some state S. Okay. Whose distribution is sampled from the stationary distribution of the Markov chain induced by this behavior policy nu. Right. And nu is some policy that we know. Right.

It is like you know you want to find the optimal way to play chess. But you know you don't know that optimal policy so what you do is you start with some arbitrary policy and play chess right repeatedly and figure out you know what happens if you do this what happens if you do that and so on and so forth so this nu is that behavior policy right so this is the policy with which you interact and this d nu that you have over here is the diagonal matrix whose s-a-th coordinate is d nu of s times nu of a given s and this d nu is the stationary distribution of the Markov chain induced by this behavior policy nu. Is this okay?

And as I said, this differs from what we had in the policy evaluation case. There, if we wanted to evaluate, you know, v pi, we had d pi here. But since we do not know pi star, we are now working with an arbitrary d nu, right? And we will see how this is good enough. So now this is the objective function and Q star over here is basically a shorthand for Q pi star.

So Q pi star was the value function of your optimal policy. We will just denote it by Q star. So, now, as I said, we do not know d pi star, and hence we work with d mu. Now, whatever objective function we have designed, one can see that its gradient has the following relation, right? So, grad g of theta equals d mu of s times mu of a given s, and this is the gradient.



Unlike policy evaluation, we do not know d_{π^*}.

Hence, we need to work with d_ν, where ν is arbitrary.

Now, $\nabla g(\theta)$

$= \sum_{s,a} d_\nu(s)\, \nu(a|s)$

$\times \left(Q(s,a) - Q_*(s,a)\right) e_{s,a}$

where $e_{s,a}$ is the standard basis vector

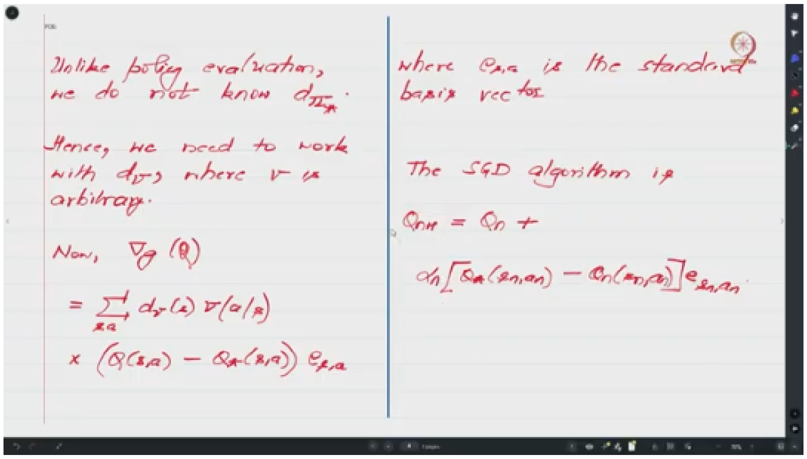
The SGD algorithm is

$Q_{n+1} = Q_n +$

$\alpha_n \big[Q_*(s_n,a_n) - Q \quad e_{s_n,a_n}$

So, I should be careful. This is not grad g of theta, but rather grad g of q. So if you take the gradient of G with respect to Q, then this is the expression that you get, where this E_SA is actually the standard basis vector in the SA-dimensional space. By standard basis vector, I mean that it is a vector whose SA-th coordinate is 1 and all other entries are 0. It is that vector over here.

So this vector you put in here, and you can see that the g function that I had defined on the previous page, its derivative with respect to q actually satisfies the relation that is mentioned over here. So once we have a gradient of the following form, one can look at the stochastic gradient descent algorithm. So that would have the update rule that is given over here. So you have q_n plus 1 equals q_n plus some step size times the negative of the gradient that we have over here.



Unlike policy evaluation, we do not know d_{π_*}.

Hence, we need to work with d_{ν}, where ν is arbitrary.

Now, $\nabla g(Q)$

$$= \sum_{s,a} d_{\nu}(s)\, \nu(a/s) \times \left(Q(s,a) - Q_*(s,a)\right) e_{s,a}$$

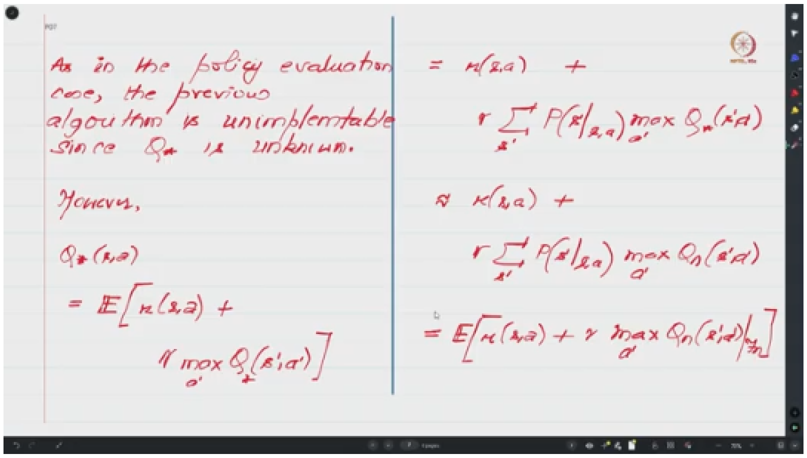
where $e_{s,a}$ is the standard basis vectors

The SGD algorithm is

$$Q_{n+1} = Q_n + \alpha_n \left[Q_*(s_n,a_n) - Q_n(s_n,a_n)\right] e_{s_n,a_n}$$

So here we somehow pick some state action S_n A_n and look at the sample. So you can think of this as an expectation where the expectation, I mean the random variables are S A, and the expectation is taken with respect to this distribution. So you ignore these quantities and whatever we have over here, you sort of use that as a sample. So the way to interpret this is you somehow sample S_n A_n using this distribution right, and then you run this update rule. So this algorithm can be viewed as the SGD algorithm for minimizing the objective function G that I had specified on the previous slide, right? So of course the problem with this algorithm that I have written is that I don't know the value of Q star. So of course, how to sample S_n A_n is another problem, but the value of Q star is not known; that is the major problem. And we will worry about how to sample S_n A_n according to this distribution soon

or later on, but right now let's see how to get rid of this Q star. So towards that, we adopt the same principle. as in the policy evaluation case. So, Q star is unknown and hence the previous algorithm is unimplementable. However, Q star satisfies this relation, that is, Q star is, you know, T Q star, which means that the S-A-th coordinate of Q star is the expectation of this, where, you know, the random variable in this whole expression is S prime, and S prime, okay, is sampled according to your transition equation.



As in the policy evaluation case, the previous algouthm is unimplementable since Q_* is unknown.

Howevr,

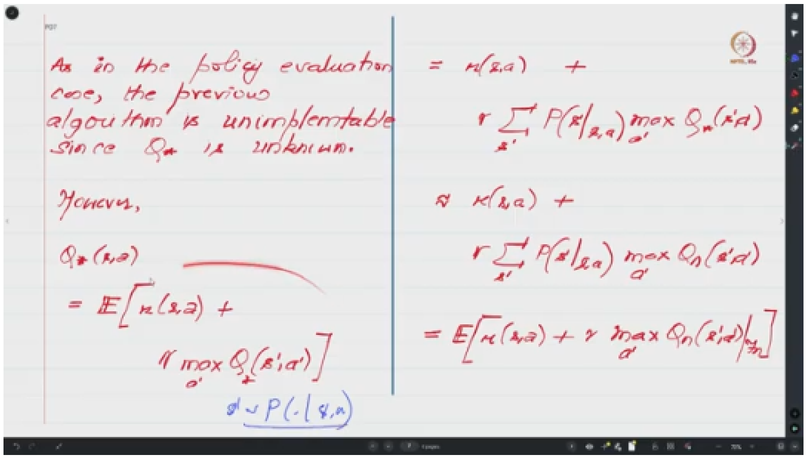
$$Q_*(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q_*(s',a')\right]$$

$$= r(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q_*(s',a')$$

$$\approx r(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q_n(s',a')$$

$$= \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q_n(s',a')\Big|_{\mathcal{F}_n}\right]$$

probabilities. This is how it is sampled. Hence, this expression can be written as shown here. So R S A comes out as it is, and this gamma also comes out as it is, and whatever we have over here, we first multiply it by the probability of choosing this S prime, and this is what you have over here. So, we do not know Q star.



As in the policy evaluation case, the previous algouthm is unimplementable since Q_* is unknown.

Howevr,

$$Q_*(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q_*(s',a')\right]$$

$$= r(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q_*(s',a')$$

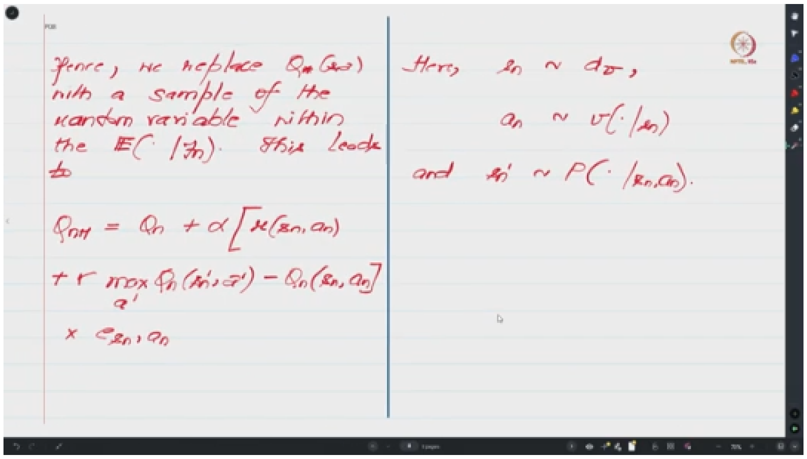
$$\approx r(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q_n(s',a')$$

$$= \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q_n(s',a')\Big|_{\mathcal{F}_n}\right]$$

$s' \sim P(\cdot|s,a)$

However, we know that Q star satisfies this equation. Furthermore, at time instance n, we have an estimate of Q star. So, wherever you have Q star, we replace it with our estimate that gives this Qn. So, as I have said, this equality now becomes an approximate equality. And instead of having this quantity, we can write it as Rsa plus gamma times this quantity.

Sum and wherever you had Q star we have now replaced it with Qn and now the question is you know is this choice good enough or not right so we will see that but whatever we have here can now be written in the following way that is this quantity is the expectation of this quantity conditioned on fn so fn means you know Qn which means that the only thing that is random over here is s prime so that you get back this expression that I have written above Is this okay? So, now with this in place, notice that this expression has no Q star, right? So, wherever there was Q star of S A, we are going to take an expression like this and plug it in its place and that would end up giving us this algorithm, right? So, wherever we had Q star of S N A N, we have now replaced it with a sample whose conditional expectation is approximately equal to Q star.



Hence, we replace $Q_*(s,a)$ with a sample of the random variable within the $\mathbb{E}(\cdot/\mathcal{F}_n)$. This leads to

$$Q_{n+1} = Q_n + \alpha\Big[r(s_n,a_n) + \gamma \max_{a'} Q_n(s_n',a') - Q_n(s_n,a_n)\Big] \times e_{s_n,a_n}$$

Here, $s_n \sim d_\nu$,

$a_n \sim \nu(\cdot/s_n)$

and $s_n' \sim P(\cdot/s_n,a_n)$.

So, I say approximate with you know some caveats here. So of course when Qn is very far away from Q star it would not be a very good approximation but nevertheless it is some approximation and the idea is can we somehow make use of this. The advantage of this algorithm is that notice that there is no Q star and hence it becomes implementable, right? So at least it is implementable and here I would like to now talk about, you know, what are these SN, ANs, SN primes and so on.

So to begin with, we will presume that somehow for a given behavior policy, we know its stationary distribution or the stationary distribution associated with the Markov chain induced by nu. So we can either presume that or the alternative, as I mentioned in the previous week, is to allow this Markov chain to evolve for some time and then pick a state that you get at the end. That will roughly be a way to sample this. Of course, in practice we can get rid of such assumptions. but you know it would be beyond this course, so we are sort of looking at a slightly idealized algorithm that somehow we have knowledge of this d mu and we can sample from this and as I said a loose way of being able to sample from this distribution would be to allow the Markov chain where you act according to mu you allow it to run for some time right and then whatever state you get at the end you take that as Sn right and

get the state at SN plus 1, you do the same thing. You start the Markov chain at some arbitrary state, allow it to run for some time, and whatever state you get at the end, you treat that as SN plus 1 and so on and so forth. So this is how you sample SN. This AN is basically sampled from your distribution or the behavior policy mu. So whatever is your current state, you ask according to your behavior policy how you should pick your action and that would be your AN and SN prime

is supposed to be sampled from your transition probability, right? So, even though you do not know P, you know, in reinforcement learning, we presume that you can interact with the environment and say that, look, here is the current state. If I take this action, what is the next state that I can see? So, this is your SN prime, right? And this SN, AN, SN prime, if you have, you can implement this algorithm.

And again, I would like to emphasize that this algorithm requires no knowledge of Q star; hence, it becomes implementable. And now the question is: can such an algorithm be useful to find Q star? And this would be the focus of our discussions over the next couple of lectures: whether such an algorithm can be used to find Q star. And I would like to highlight that this algorithm is very popular in the RL literature, and it is known by the name of Q-learning. It is known as Q-learning because this algorithm is designed to find Q star, which is the Q-value function of the optimal policy.

So, I hope you will join me in the next class and see how we can analyze this algorithm using our stochastic approximation principles. Until then, thank you and Namaste.