NPTEL

NPTEL ONLINE COURSE

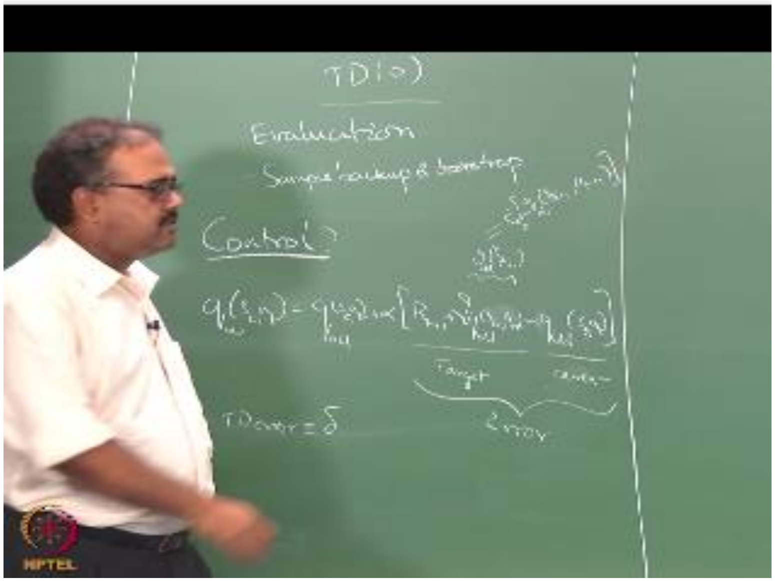
REINFORCEMENT LEARNING

TD(0) Control

Prof. Balaraman Ravindran
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Okay, so we started looking at the TD learning right, we look at TD(0) yes so what does TD(0) do for us okay.

(Refer Slide Time: 00:35)



Yeah but it does essentially right, he does evaluation right it just gives you given a policy π and finds out what Vπ is right and use a sample backup and boot strap how do we do control is the q function instead of the v function that is it, what we do is fix a policy π and use the q function and learn the qπ for the policy and then max of that and then you get a new policy and then you go ahead with it is it okay.

So that is one way of doing control with TD(0) right so in fact I can do that v if I have V but finding a greedy policy is hard so what i can do is i can take the q function right, so can you we want to write down how this is going to look like what goes in here what was in there I think so okay, so let us because I put RT+1 we have to add the T everywhere sorry, okay so first day I said as I put ST+1 AT+1 okay, so if you stop and think about it what should really go here is right so if you remember this is the that is the current estimate, right that is a target so this whole thing gives you a error right.

So did I mention TD error in the last class right, so this expression that you get in the TD(0) update that expression is called the TD error right, what does the TD error RT+γ V is Vo this T+1-Vo ST 1 that quantity is called the TD error so I said it is some error term so whatever error term you use in the TD update is called the TD error and the error term you would use in the Monte Carlo update is called the Monte Carlo error bla, bla whatever is one but TD error is the more important things okay, so sometimes we so the TD RF is special enough to get its own symbol right, δ so typically when I say TD error δ without any other quantification it is the error on the v function okay, just the more of a historic note so wherever I use δ I will define what δ is but typically delta by itself typically stands for the error on the v function okay, good.

So ideally I want this why because I am using this as my target and what is my target it is expected return shafting from this point right, that is the target since I do not have the expected return I am using a sample return which is what one Monte Carlo gave us as I said instead of using a sample return I will use bootstrapping to correct that written so I will use the immediate reward plus γ times that gives me the correct editor it is ideally I should be using the return here but why am I using the Q function here yeah, so why am I using that just the action i took a loan right ideally what should i should have used that it's expectation over my policy.

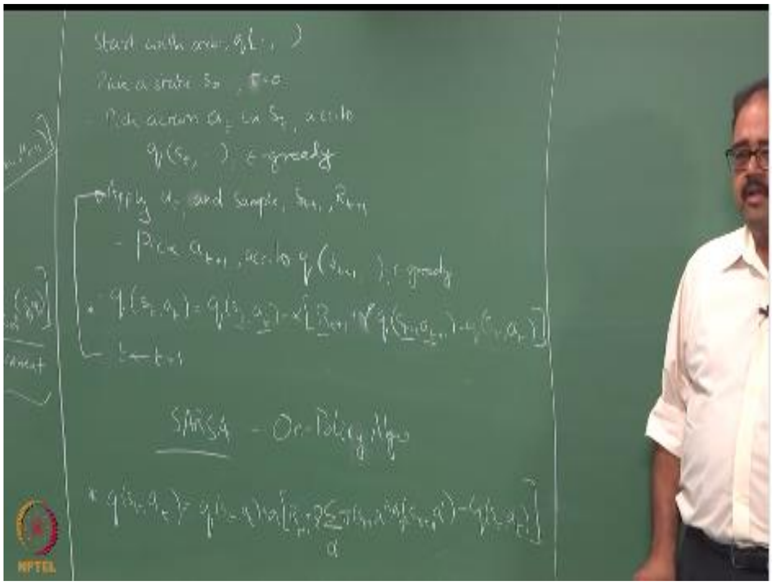
People see this is essentially equal to right whatever is my current policy that is expect so expected value of Q old comma stat plus 1 where AT plus 1 is taken according to the current policy right so that is V but instead of completing this expectation right I'm just taking one sample from that expectation, that's basically what I'm doing so if i apply this rule often enough i will eventually compute this expectation by former cure, π you fire under that single policy we are taking this salmon if you are not updating the policy does not matter, if you want to put the π here but since I am fixed a policy I do not I do not worry about it you should ask me this question envy it makes more sense, but here there is another reason why I am avoiding the π.

I will tell you in a minute right so we if you give one to use people do not use Vπ as π,is fine right, so this is how we will estimate q right so you understand why we are putting in this here okay we are putting this here as a estimate for this expectation right as an unbiased estimate for this expectation everyone with me okay good, so what do I do now I won a few iterations of this learn the Q function be greedy with respect to that Q function and I can update my policy and then run it again okay that is one way of doing control okay.

There is another way of doing control well this is exactly the same way whatever tell you except that instead of running it for several iterations with the same policy π you take one step with that policy π update the Q function and the next time you have to choose an action be greedy with respect to the current Q function, if you remember we talked about this in generalized policy iteration right you can just take one step and then next time you can be greedy with respect to that is that one update you've done right, so do not fix a policy in this sense in this case in so instead of having a fixed policy every time you have to pick an action we just look at the Q function and pick the greedy action so this is like running extreme generalized policy iteration, right.

So for every step you just take one step with that policy for the next time step you be greedy with respect to the value function update, that you have done right you just keep alternating between being greedy and being and updating the value function you just keep going back and forth for one step update the value function next time you be greedy right, so putting all of this together you can actually get a algorithm like this so you start off with arbitrary Q function okay then right, so pickle action 80 to perform in state according to the current q value right so why'd I have put according to the current Q value, you could just pick a greedy action but we know from our bandit experience being greedy always is a bad thing so you need to do some kind of acceleration.

(Refer Slide Time: 11:11)


Start with some q(·,·)
Pick a state s_0, t=0
Pick action a_t in s_t acc to
q(s_t, ·) ε-greedy
Apply a_t and sample s_{t+1}, R_{t+1}
Pick a_{t+1} acc to q(s_{t+1}, ·) ε-greedy
$q(s_t,a_t) = q(s_t,a_t) + \alpha[R_{t+1} + \gamma q(s_{t+1},a_{t+1}) - q(s_t,a_t)]$
$t \leftarrow t+1$
SARSA – On-Policy Algo
$q(s_t,a_t) = q(s_t,a_t) + \alpha[R_{t+1} + \gamma \sum_a \pi(s_{t+1},a) q(s_{t+1},a) - q(s_t,a_t)]$
NPTEL

So you be greedy a lot okay explore right so you can either do this with some kind of a epsilon greedy kind of an approach right or you can do softmax or you can do anything whatever, whatever takes your fancy come up with a good exploration strategy right so people do Bayesian exploration and there are many different ways optimal ways in which to do this exploration, right. Now apply 80 and your sample St+1 Rt+1T right, you could sample this if you have a sample model you could just run this through simulation or you could just perform it in the real world and get the St+1 Rt+1, right.

Now comes the important thing pick 80+1 according to the q function okay, the current q function, now what you do and keep doing this until your episode terminates right, if once episode terminates then you go back to kind is not and keep doing this again and again until you are happy with your q function, okay right fine you do not agree. So there is no explicit evaluation stage there is no explicit gratification stage right, so the evaluation is done here right and the gratification is done here right, here, okay.

There are some small caveats here so if you want this algorithm to converge to something right what do you think you should do and then anything else decrease alpha also over time right, so decreased alpha over time decreased epsilon over time so that eventually you will stop updating things but the eventualities should happen a long, long, long time into the future right, so you should explore a lot, right so that you are sampled every state action often enough and at alpha should not decay so quickly that even while you are sampling if alpha goes to zero then there is no point in sampling right.

So the alpha also should not decay till you have sample enough okay, so both should decade but should be decaying at a slower alpha should decay at a slower rate than epsilon, okay is it good okay. So this is probably the second most popular reinforcement learning algorithm that is used for solving a lot of problems so this is called anyone SARSA.

So what is SARSA okay, sounds like a Mexican dish or a dance so it is a dumbest possible explanation for the name okay, state action reward next state next action so you need the next action also before you can perform these updates therefore they called it SARSA right, the name stuck I mean and we have reached certain to blame for it I mean he came up with the name not the guys who originally proposed this algorithm okay, so they are in come up with this names so the guys are Rummery and Nurujan.

So in fact couple of years ago Nurujan was going to come and spend like a semester here on a sabbatical from England but then last minute he changed his mind so you never made it the guy who proposed SARSA who going to come and spend a semester here but he did not come anyway, so this is SARSA this is an on policy algorithm why, because I am following this epsilon greedy policy but I am also updating the value function corresponding to the epsilon greedy policy, right.

Because I amusing the action that was actually taken, right so if you go back here this will be the expected value according to π which is whatever π I am following, right so only for that is this a sampler correct right, there for this is on policy algorithm, okay. In fact if we stop and think about it for a second in fact I can complete this expectation without knowing anything about the world. No, the expectation is only with respect to i and the quantity and taking the expectation of is qπ, right I know both π because that is as policy for according to which I am taking actions, right, and I know the cube because that is a quantity I am estimating anyway so I might as well compute this expectation explicitly I do not have to take a sample from further expectation here in fact I can do the exact computation so if this line I can replace with something like this.

That is why I said it is computing the epsilon greedy value, since you are updating for the exploration steps as well it is actually computing the policy correspondent value function corresponding to the epsilon greedy version if epsilon GD policy and that is why I also call it on policy right, so you could do something like this right. So in this case there are a couple of interesting things here I do not have to pick an action A priori right, so here we first pick the action for 80+1right.

And then we did the updating and the way I have written the loop you notice that I am forcing you to take this action here again the same action you took I am only saying apply 80 okay, I am not allowing it to pick again right, because if you pick again then you are not on policy anymore because I use 80+1 to update here if I allow you to pick another action you are not on policy anymore because my value function has changed right.

I will be picking a different action so this is slightly subtle things that right, but here since I am taking the expectation it does not matter I can go back and take a new action and I can do this but this also makes this may be kind of policy is it somewhere in the gray region between on policy enough policy because I am not actually evaluating the action I took, right because at this point I am not trying to even build the π right.

Let us just have a think about it a little bit, right so at no point I am I having an exclusive representation of a π anywhere in this SARSA algorithm there is no policy here and it is implicitly defined by the q functions so when the q function changes the policy is automatically changing right, but here I am explicitly talking about the π so I have to compute this π, right and then I have to throw all of this in and then I can become greedy with like epsilon greedy with respect to the new q function I compute therefore the π also will change every step right. So apparently this has a name this is called expected SARSA, okay fact I never came across this until the second edition of the book and I had to go back and see if there is actually a paper on this but rich also cooks up things for the book, right and all the off policy Monte Carlo stuff that was there it never used to exist before they wrote the first edition of the book he just cooked it up for the book okay, so like that he might have cooked up this expected SARSA for the book I am not sure but he just likes to add things.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in