NPTEL

NPTEL ONLINE COURSE

REINFORCEMENT LEARNING

Eligibility Trace Control

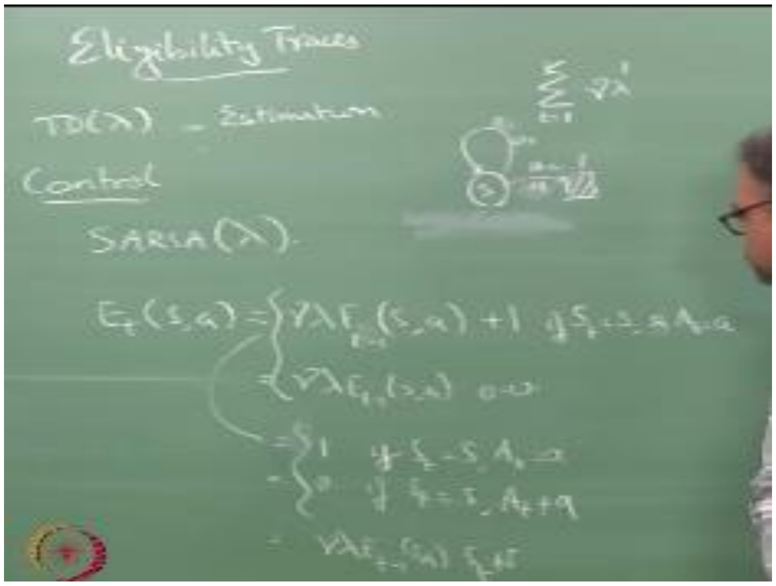
Prof. Balaraman Ravindran

Department of Computer Science and Engineering

Indian Institute of Technology Madras

So we are looking at eligibility traces.

(Refer Slide Time: 00:16)



Eligibility Traces

TD(λ) – Estimation

Control

SARSA(λ)

$$E_t(s,a) = \begin{cases} \gamma\lambda E_{t-1}(s,a)+1 & \text{if } S_t = s, A_t = a \\ \gamma\lambda E_{t-1}(s,a) & \text{o.w.} \end{cases}$$

$$= \begin{cases} 1 & \text{if } S_t = s, A_t = a \\ 0 & \text{if } S_t = s, A_t \neq a \\ \gamma\lambda E_{t-1}(s,a) & S_t \neq s \end{cases}$$

$\sum_{t=1} \gamma\lambda^t$

Actually look at TD λ so what is the next step to go naturally would be to look at control right after looking at TD λ which is essentially a estimation method so we have to look at control so the easy has to think about in terms of control is version of SARSA that will use an eligibility trace right so what does it mean to have eligibility trace in SARSA, so the backward w is kind of easy to see where we are going with right so backward view in fact is almost like TD λ.

So essentially what I do now is introduce a eligibility trace for so I increase eligibility trays for state come action just like you went from various state values you go to action values for q learning and SARSA so you go to state comma action plus you have an eligibility trace like this right and how do I update right exactly what we did that so this is kind of your you're accumulating trace.

So if you want replacing trace what you okay, so there is one other thing you have to be careful about when you are talking about eligibility traces on state action pairs right suppose I am in some state yes right this is action a1 this is action a2 I do not need you in this is action A to that is action near one this is a zero river right and then I have a policy right now that keeps taking action a one many times and then finally takes action a two and gets a reward of +1 that if I am using accumulating traces so what will happen is every time I go around this loop I will keep adding one right.

The first time I go I will have one right the next time I go I will have $1 + \gamma \lambda$ the next time I go I will have $1 + \gamma \lambda$ this into $\gamma \lambda$ wreck so I will have $\gamma \lambda + (\gamma \lambda)^2 +1$ okay I will keep going there at some I will basically my ribs my trace for this will be some number of times $\gamma \lambda^t$ right t running from 1 to sometime some k right depending on the exact values of $\gamma \lambda$ this number can be much larger than 1 right I have $\gamma \lambda + (\gamma \lambda)^2$ so for it is decaying right but you think about every time I am upping it by one rate so.

So it will be a number larger than one could be a number larger than one and what will be the trace of a two and I finally take a2 it will be one right and which action you think is going to get the higher fraction of this plus 1 a 1 will get updated more than e2 right so what will happen next time I come to say this and move them again I will be more I am more likely to take a 1 right again I will do this a few times and then I will take a2 again a1 will get rewarded yes even if I use replacing trace what will happen that k will get $\gamma \lambda$ okay it could still be a large number right.

So the rate at which my a2 will actually come to beat my a 1 will be very very slow right but replace interest will eventually fix your problem okay so accumulating traces are bad typically if you are doing control right it was not that much of an issue when we are talking about the estimation because I was not changing my actions based on this right but here I will be changing my actions based on the estimates I am making so the replacing Machiavelli traces are bad replacing traces are better right.

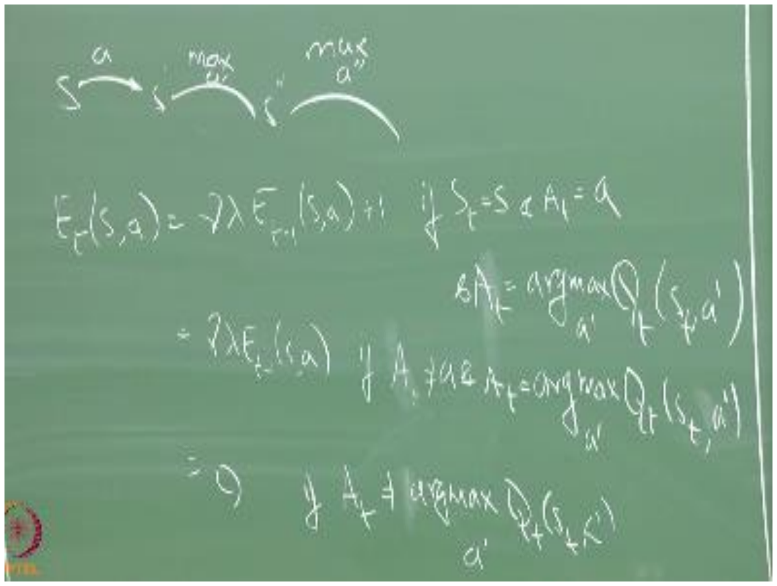
And you can do a slightly more aggressive version of replacing stresses what do I do in that case whenever a revisit a state and I take an action if it is for all the other actions that I could have taken from that state I make the trace 0 so I come to a state right I take an action from there so for all the other actions I could have taken from the state right I set the place to zero that means if there is a specific reward I get by taking this action that will not go back to all the other actions I could have taken from the state right.

So to see what will happen suppose I come S for the first time right and there from here I take a2 I have not taken a1 so what will be the trace for a1 0 I am taking only a2 right so whatever I get here it will be a change the value of a2 only it will not change the value of a1 right, so I want that effect regardless of how many times have visited state s before if I visit yes right and then I take a one right then I come back to s after a while and then I take a2 and I get some reward right so it is not likely that I would have gotten this reward by taking a1right.

Because the different completely different path I am choosing there so I do not want this reward to go update the value for a1 right so what you can do is this is one way of doing this is accumulating trace this is replacing trace with three conditions this is a special case of replacing traces so you could do the regular replacing trace also wherein you have decayed the value for all the other actions right this is one aggressive case of replacing traces so wherever you come back to a state for that action you set the trays to 1 for other actions you set the trace to 0.

So what would be the forward view for SARSA λ I am ignoring this part so what would be the equivalent we saw the forward be for TD λ what will be the equivalent for would be for SARSA λ, so unless the same thing you do not think too much about entering so it is 1 - λ times λ wait for the first the first return one step return right and then λ^2 wait for this a two-step return λ^3 wait for the three-step return and so on so forth do not worry do not do not do not think too much about it so that is basically it right.

(Refer Slide Time: 09:57)



$S \xrightarrow{a} s' \overset{\max a'}{\frown} s'' \overset{\max a''}{\frown}$
$E_t(s,a) = \gamma\lambda E_{t-1}(s,a) + 1$ if $S_t = s$ & $A_t = a$
& $A_t = \arg\max_{a'} Q_t(S_t, a')$
$= \gamma\lambda E_{t-1}(s,a)$ if $A_t \neq a$ & $A_t = \arg\max_{a'} Q_t(S_t, a')$
$= 0$ if $A_t \neq \arg\max_{a'} Q_t(S_t, a')$

And once you have this et so what you do your Δ Q t Sa will be some α times δ T x right where δT = Rt what okay and I do not even have to write this thing stone when you can generalize directly from TD λ almost everywhere I wrote v of s I have written q of s, a this I will just done the backward view the forward view we can likewise very easily write down the forward view for SARSA λ.

Anyway good any questions SARSA λ okay it is over the next control algorithm we should look at Q λ yeah it turns out that there are many variants of Q λ in the literature right so if you read the book you will get picks Q λ naïve Q λ and then also get something called Watkins Q λ Watkins Q λ that is one we will look at very quickly because that is one it is kind of you know justifiable even though in practice it does not work well right the other things are just ad hoc stuff you know both banks Q λ and nave Q λ which will be discussed in the first edition of the book.

They are pretty ad hoc right I mean it is not at all clear what they are converging to and there are equivalent lee many ways in which you can define a queue learning version for eligibility traces but what Kansas Q λ seems to be the most sensible way of doing things right even though it is when empirically it can be very bad okay so if you think about what Kansas Q λ what would be the generalization or just if you think about q-learning itself what would be the generalization to using eligibility traces if you think of what we are doing in q-learning we are taking the immediate reward for taking an action.

And then assuming you are going to behave optimally thereafter that is why I use the max and take the q so if I am going to use a two-step return in stuff one-step return what should be the two-step return for q-learning I should be behaving optimally subsequently right it cannot be a arbitrary action like in SARSA I never had that issue right so I could just take whatever action I did I could just use it is not an issue right but in q-learning it has to be the optimal next action right.

So we basically have this right so for this δ a is fine but I am competing the return I have to make sure that I am actually taking the optimal action and sampling the rewards right how do I ensure that I have to behave greedily see I can't do epsilon greedy or anything like that if I do epsilon greedy then there is no guarantee that whatever sample I am drawing is going to be from the optimal policy when I mean there is no guarantee in the beginning when your value function is all messed up.

But even as learning progresses if I am exploring right the return I am generating will not be from the optimal policy right so as long as am being greedy and generating my trajectories I can use the entire trajectory for q-learning right so if you think about what I really need is okay I start from yes I take some A that path is fine where I go to some s prime then what I nearly need to do is take max a’ I will go to some s double prime and what I really need to do here is take max over a double Prime and keep doing this right.

So if you think about it this is essentially the greedy policy all right I am just executing the greedy policy if I execute the greedy policy then I can use how many other steps I have gone I can use that entire return in my updation you start first we get the same type traction always forgive us we're not exploring you know we would not follow the same trajectory always but we have other arguments about why we have to explore like we have looked at it all the wine and just trying to build up why we need to be greedy fest but we all know we have to explore.

So we have to figure out how to address that which we will come to the acceleration portal imminent right but we would understand why I need the greedy trajectory because that is what the assumption of q-learning is right I am behaving greedily so like she pointed out and like we know from the Bandit times right we need to explore if you do not explore we will get stuck in something that is suboptimal and I cannot start regardless of where I can start with an arbitrary initialization of the Q values right.

But if I do not explore enough right I am going to stuck in since we get stuck in suboptimal policies is I need to do exploration so how do I balance these two tensions right yeah then the chain gets broken right because I want to find the return yeah that is because we’re only doing one step right if I am going to take the rib disease the thing with λ is that I am taking some kind of weighted average of one step return to step return three separate and forced of return and so on so forth right.

So one way are thinking about doing this is to saying that I am going to keep track of the optimal action trajectory from every state I have visited imagine the amount of memory that you would need and I start from yes I take the optimal action go to explain okay but then s prime I explore right but then if I want to use the two-step trajectory for updating s I will need to know what the optimal action from s prime is that I can do what about three separately from is so what will I need I need to know the state I would have gone to from s prime like I am the reaction I will have to take from that state.

And I remember all of that to get the three-step trajectory for updating at s assume it is were correct as soon as the explorer at x prime all I can do that is use the greedy action from s prime that is the longest trajectory that I can consider let update the value based on that right and then 03 eligibility trace I cannot do anything in New York to that it does it this because as long as I do not explore if I do not explore for ten steps I can use from one step return to the 10 step return we are breaking the Liz ability trace that is the main problem with what means Q λ.

So the eligibility trees will not run till the end of the trajectory so it will be only for short periods of time so that is a trade-off that you are giving that the thread of that you are taking it's some form of λreturn it is if you try to write the forward view of Watkins Q λ it turns out to be very messy expression because whenever you take an exploration only that till that point you can do the averaging right so now what we say is that we do an averaging over all possible and step returns rate and then all the waiters are sent to the last written that is what we are saying so now what happens is if I can do only up to some say for step return after the into exploration.

So for that particular state I will be doing an average over only the four returns right the state that came after that I will be doing the average over only three returns because any explanation all the states that came before me I will make it zero right and then the state just before that I am doing the update for just one right the state just before you do the exploration for example an S prime if I take an accelerate reaction when s prime will get the regular q-learning update know λ business there and I reset all the traces and start all over again.

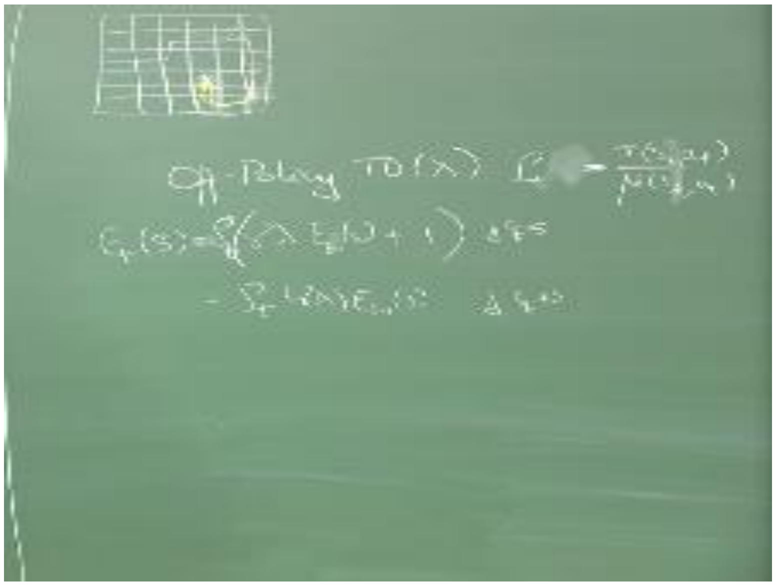
So that is the tricky part with what consists you λ then the forward view seem incredibly hard to write so I am not even going to attempt to do that and not worry I will not give it as homework as well maybe we'll save it for an exam right so this is my same δ that we have in q-learning the δ does not change this update rule does not change right so this is exactly the same update rule that you have for SARSA what changes is this so if it is a greedy action if the current action is the greedy actions are not a sorry if the current action is the greedy action right and s and a happens to be a current state in action I will increment it by one right.

So this should be st sorry so if the current action is the greedy action but that is not the current action okay is not the current action then a decayed eligibility trace of a I do not add a one if the current action is not the greedy action a zero out all these abilities regardless of what state what action everything is it fine now should I do something else what is external this is the exploring but I am going to anywhere about action selection right so when you are doing the action selection you do it in the ϵ greedy fashion say whenever you actually pick an action that is not the greedy action that 1-ϵ times you make this update to the eligibility trace.

If you have picked the greedy action then you do the regular accumulating trace update here so likewise you can write one version for replacing traces as well right or if you want to do Dutch traces you can write one version for that stresses as well that version here and the Dutch trail solution here so you could do all of that makes sense so the main problem with Watkins is Q λ is all the traces will be very short unless you decrease your exploration by a significant amount and if you say I will explore with say epsilon off point one or something then 10% of the times you are going to take accelerate reactions right.

So things can get very either the traces will get chopped up a little and you lose the advantage of doing eligibility traces okay, and so here is a here is a picture that they have maybe I should draw this on the board drawers over easy enough okay.

(Refer Slide Time: 25:00)



So that is the final state I am going to get to okay and that garden there is a reward let us say some reward say plus 10 everywhere else it's zero okay and I have say λ 0.9 and   λ 0.9 okay so I am doing some random exploratory trajectory I go something like this I reached there and I reach the goal and you get the reward okay if I had been using regular q-learning or SARSA if I had been using regular q-learning or SARSA which are the states that will get their values changed after one trajectory like this online q-learning so this K alone right this state alone the value of this action will go up right.

Value of that action will go so what happens the TD λ or Q SARSA λ in this case all the state so but unfortunately no SARSA λ probably it will stop somewhere there this decays to such a small value after that I do not update it right so unfortunately your you are getting wrong updates in a right action due to do here is to probably go here but the point to notice with SARSA λ with one trajectory I update I know something or the other for so many states and actions.

Well when it if I done just plain SARSA or plain q-learning it would have been just for one step and that is one of the reasons we expect this to converge faster just to give you a very rough intuition as to why you expect us to converge faster this is this intuition comes from the backward view right because I am updating in so many states good so there are a couple of other things I am debating whether I should talk about other should make you guys read about it so what does q-learning do for us.

We keep calling q-learning as an off policy learning method right but it really does not fit into the structure of off policy algorithms that we saw in the Monte Carlo chapter right so we said we had one behavior policy and one estimation policy and we are trying to estimate the policy by behaving according to estimate the value of that policy behave according to another policy and so on so forth here we took a lot of shortcuts because the estimation policy was the optimal policy so we try to take all kinds of shortcuts right first of all we do not know what the optimal policy is right.

In the typical of policy learning we talked about we have a we have a π and a μ and then we did all of those things right now we do not know what the π is we know what the π is because mu is what we are behaving according to but pie is the optimal policy I do not know what π is so really q learning does not fall into the off policy mechanisms that we spoke about so you could think of having an off policy version of TD λ right.

So I could have a π which I am trying to evaluate but am a trajectory could be coming to me from some μ right so how do I how do I handle that so showing the equivalence of a SARSA and backward and forward view of SARSA λ you still have to assume that you are doing this offline not online so the end of the trajectory only you are changing the values okay that becomes very tricky you are never ever be running shorts are offline that because you want the values to be changing as you are executing so that you start learning right so you do not want to be executing random trajectories and then.

So it becomes a little tricky so there are more recent works where you can have an online version which allows you to which gives you equivalence between the online forward view and an online backward be anyway that is an aside so let us go back to policy TD λ so what do you think we should do so we basically have this right, so that is the alright so that is the update equation I the other one is γ λ ET -1 so I have to somehow figure out how I life how am I going to insert my importance weight into this and I have to figure out how I am going to insert any importance weight into this how do you think I will insert my importance weight into this.

So let me define something that may can make my life looking simpler I am going to define ρ t of s as right whatever then you see π s of whatever action I took at that time oh yeah that is good point so I do not need to so it is a ρ t is the importance weight at time T it is for that particular choice right so at that time I made this choice right, I made a choice between ah some action at time is T is state s t I picked an action right what is the probability of me picking that if you remember our expression for the importance weight rate.

We wrote it as a product of π STAT summing over all T right so that is essentially that that item let us say entry I want so ρ T will be π STAT even it by μ STAT that is at one term from the importance weight if you if you actually have notes you can flip back and look at the important sampling thing that we talked to go right so I basically multiply this way ρ t the question is what do I x ρ t just the γ λ et – 1 or should I multiply the plus one also right.