NPTEL

NPTEL ONLINE COURSE

REINFORCEMENT LEARNING

Learning with Option

Prof. Balaraman Ravindran

Department of Computer Science and Engineering

Indian Institute of Technology Madras

So how do you learn with this how do you learn that now that you have to know how to define options can I solve problems then by using these options and this is their SMDP Q learning comes in, you all of your remember assembly SMDP Q learning, right so I am going to update I mean the normal, if you take an action that part is clear I mean there is a normal Q learning, right in a particular state suppose I take a, option instead.

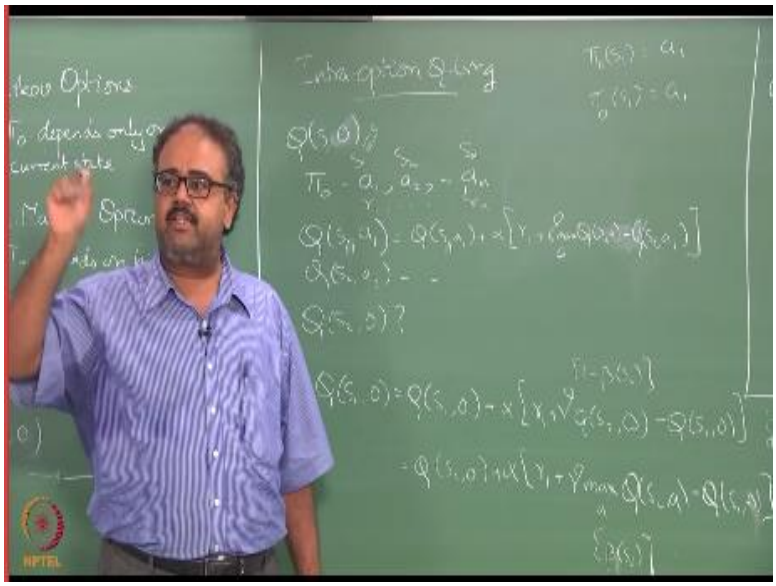
So what we do, I treat this as a derivative action, right in the SMDP Q-learning what happened we assume that actions have duration right so here in hierarchical RL, I start off with an MDP, right I have defined a set of options on top of set MDP, right so when I use these options on that MDP I am going to treat these as derivative actions, so the MDP plus options gives me SMDP, good question you could it will come to that in a minute right.

But the simplest version is you just do so you take option O in state s then you update the value of that option in the statu s, won't that we are trying to learn the eternal policy by zero no massage has come to it hold on okay, right do we need to write down the SMDP Q learning update again, you guys all know it from last class right great so the only tricky thing here is the reward and the reward for the action is going to be the the sum of the the return that you get, from when the option started till when the option ended, right.

So that is basically so last time we had it started from RT and ended at RT plus δ and then we gave a proper operate discounts and everything so all of that will come into play here because, because the option is the derivative action, right so that is essentially why we introduced the

SMDP framework in the first place so that we could handle all of these things, great so SMDP Q learning is fine, right but then we want to do something better right.

(Refer Slide Time: 03: 26)



So we want to do, they call intra option Q-learning, so we actually want to take advantage of the fact that the option consists of, primitive actions, right and we want to go and, use that as well right so what would we do in that case, is I'll update Q (s,o) in the normal way, right let us assume that when I start π_0, right so I do actions, these are the actions I see okay this is this is because of the definition of π_0 on the sequence of states s1,s2,s3 that I am seeing, right

So I see let's say that I see s_1,s_2,s_n and after I do a_n and s_n I terminate, okay let us assume that this is the sequence of things I have seen okay, so what I can do is I can actually update Q (s,a_1), right so how will update $Q(s,a_1)$ this one what goes in here max, okay max over a $Q(s_2,a_1)$ cannot use a_2 in that expression because, why because it is Q learning why wouldn't I want to use a_2 in an expression the because, this is where we come to your point I am following policy π_0 here right.

So this is not necessarily reflective of the value of taking a_1 in state s_1, right if I had you stats a_2 hear it is the value of taking a_1 in s_1 assuming I am going to following π_0 I not thereafter and then I will after π_0 terminate I will do whatever I want, right so if I had used yet to there that is what I

will be learning does not make sense normally what do you expect in Q learning or salsa is that this expression, is the reward you get for following, policy π thereafter right not following optimal policy that after whatever the following policy π thereafter is what you expect in salsa.

So if I reduce the a_2 here I am not following policy π there after I am actually following $\pi_{0,}$ it could very well be that π_0 is the policy that I would have picked there in a_2 also in s_2 also but I do not know that for sure, because I never had a choice of picking an action in s_2 to a pick in action to in s_1 I was constrained to follow that action in s_2 also since I didn't have the choice of picking the action in s_2 too I really shouldn't use a_2 there and claim that it is salsa okay.

So that is why in all this kind of interruption, kind of learning mechanisms you have to use Q-learning you have to be off policy, okay so you said anything else I can do other than update so likewise I can do it for a 2 and so on right I can update Q there anything else I can do can update any other value functions are we maintaining two kinds of patterns or the maintaining there are primitive a transfer for every state and there are options well it's like an MDP that has like 10 different actions that you can take right when forward from the values and six were options.

So I'm maintaining a Q function for for primitive actions and this exaction six options also it is a single Q function and it is for solving the original MDP problem essentially I'm optimizing the reward function given that MDP right no I'm not solving sub problems are being solved I've been encapsulated and given to me as an option policy, I am not changing π_0 anywhere yeah, because I'm allowed to take the primitive actions also, right when they come to the state s_1 next time I can either choose to take whoa o or again choose to take a_1 or a_2 or a_3 are whatever wait I am allowed to take primitive action, change $\pi_{0.}$

So I'll talk about that when I come to max Q now in the options framework itself the acid was originally proposed they never had, a option of changing the option policy, right so they just let you they just assume that the option policy was frozen, and then they said okay here go learn right they never did this multi-level learning it's possible to think of mechanisms for doing that as well right then the max cube framework made it explicit, right.

So there are ways of learning the policy in the options framework also it is nothing very deep or anything you just have to be very careful about how you do the book keeping so you have to make sure that the appropriate or what function is fed into the appropriate value function so I will explain that more when we talk about Max cube, which is one of the other architectures freedom, anything else that we can update, about that option gets the option option is ended, can I update this on the option ended, you need a value for your fee oh my desk this is $Q(s_{2,}o)$ can I write an expression like this for $Q(s_{2,}o)$ my question, when we do have an option of taking all these two part of the zero great anything else such has to be Marco right.

So what $Q(s_2,o)$ be if all starts from s_2 runs what will be the value, but who I started in x_1 right if it is now if it is semi Markov option then the policy would be different because it started in s_1 when a call o here, right so I shouldn't be using this trajectory for updating the value of s_2,o right but if o happens to be a Marko option it does not matter whether I start in s_1 whether a start in s_2 or a start in s_3 I can start in s_2 and I can update the value of, Q $s_{2,}o$ right it because it is likely to give me the same sequence of actions I mean it is a valid sequence of actions right but the only thing is the SMDP reward that I am using should start start subbing from R_2 or not $_{R1.}$

So I should leave R_1 I should start something from R_2, great so assume I am with Marko options you saw something more that I can do, assuming I am doing Marco options is a more efficient way of updating the value of Q(s,o), just do like the next state that it goes to directly use the you get transmitted electrocuting over there all right if you have an max over next of all the high so that you can take that this is fine thus it sound right way did we take the max over all options that is ah tell me why I'm asking you people you see right away policy is it is off position and all this is that isn't that match and that same function because the policy itself is a manner you don't know if $\pi 0$ the max right.

So we don’t know π_0 just make defines and a policy that has been created for me right, I don't know if I do Max so that I get π_0 or not all the other hand I am constrained to take actions according to π_0 right where in the Marko frozen no I am chosen O in s1 right so in s2 I have to take actions according to π_o right so I have to take actions according to π_o I am constrained that

way except when when they can end it is to accept when I can end a test, if I ended less to then I have to do a max right.

So now I do not have to keep a running sum of all the returns and I can get by doing this kind of bootstrapping, so I am writing it separately so that it is clear but you can actually do this combination of this 2 into a single update by using your β, right so with probability 1-β this is what you will do and with probability β this is what you will do, so if you are terminating β of s_2 so if you are terminating in s_2 right so then β is to will be one right.

So this will not happen so you will multiply this by 1-β here so if you are terminating in s_2 then this will not happen if you are not terminating in s_2 then this will happen basically for β will be 0 and this will happen so essentially I can multiply this play, and multiply this by and then I can write everything combined into a single function, make sense I remember so we have done all kinds of complex things, so we the easiest thing of course makes your life a lot easier to implement is SMDP Q learning but then you can do interruption Q-learning can do a whole bunch of things right in fact you can do even something better what else can you do what else you think you can do now that you know how to do this, so follows right.

So my π_0 of s_1 is a 1 and my $\pi_{0'}$ of s_1 is also a_1 something's these are a_1 not not the first action but three the the first index action not the first action according to time but the first action according to some index right they essentially what I am saying is both of them give me the same action for s_1, right if I mean state s_1 both π_1 $\pi_{0'}$ give me the same action now can you think of how I can update given that I have taken action option oh I am executing option oh can I update Qs' listen what is that a generalization if you take go then we can update and no function approximation here I can do the same thing exactly so I can do the same thing right.

So I am just assuming that you can multiply those things right I just write the first expression around this is amazing thing right so as far as the update equations are concerned I am only looking at s1 r1, right I do not care whether that s1 a1 R1 basically that is a combination I am looking at I do not care whether the a_1 came from executing π_0 or π_0^-, so I can still do this abrasion.

So if you do a single option execution I can actually update the value function for all the options which are consistent with that option that I am executing right, now let us ask me let us ask another question I o Q has to come over I already said right I can you can do the Q is this is a good struggle the same kind of bootstrapping is exactly where the whole of policy ratio updating thing got introduced into the oral literature, not not in chapter five right.

So when so during a break up and rich certain were working on this interruption Q-learning then they came up with this inside this they even if the options I mean the first said that if the options are consistent agree with each other then I can use the same experience, and update the values of different options and then they came up with this inside hey even if they do not agree exactly with each other as long as they are consistent in the important sampling sense, and I can use the important sampling ratio and update the values right.

So that is the inside at the camp and then they went back and say why did I have to be in the options face, right I just need two different policies the behavior policy and the estimation policy this is what happens either your behavior policy is and estimation policies is π_o not over Prime sorry π_o is your estimation policy it behavior policy is π_o go then they said okay this is a great thing we see we can do more generalization we can learn so we can actually maintain this huge library of options like that differ from one another with in some small respects but I can learn about all of those options, well I am executing only one option.

So that is the inside that led to this whole loft policy set up that we looked at the Monte Carlo chapter, and then they went back and then they started talking about off policy learning and in fact this is where the on policy of policy the dichotomy that we talked about, actually God came into the literature from this interruption key, learning areas and then we went and back fitted everything we started calling, Q-learning ass off policy sources on policy and all of those but they started only in the late 90s so cool.

So any questions on this so that we can implement very efficient option learning algorithms right was it you have to come up with options to us that's a great question, yeah we have to come up

with the options first, still a very active open area of research, so I don't know if time permits maybe next class or something or maybe in the class after the exam I will talk about attempts people are made for option discovery, and so on so forth right not too many out there but there are some interesting items that have been done right.

So we will talk about that in the maybe in the last class right now I want to move on talk about the other hierarchical frameworks so so far so good nice options are good yeah and I should say that because of the simplicity of the entire option framework that options are the most popularly used, hierarchical aural things so that's why I spend so much time I went so slowly on this one thing which you should point out there are a few things which are little dissatisfying right.

So for example the policy π_o is it is not entirely clear where π_0 is defined you know. so if you think about it we have the initiation set these are states where policy can start right but once it starts here can the policy go everywhere, not really right it depends on the termination said if the dynamics of the problem is such that the termination sets cuts you off from the initiation set right so those states you cannot you cannot actually reach, right.

So there is no there is no clear specification of you know admissible states for the option these are the states that this option will visit and things like these are all left to be inferred from other things that have been defined, right so what else is missing is things like whatever I was talking to you about here right so when I am actually executing option whoa when I come to a state I will behave differently even if it is a Marko option right and behave differently, if I as opposed to when I am executing option ō, right but for an outside observer it look like either it look like I am behaving in an on Marco fashion or I am behaving in stochastic fashion right I come there I do I do action a1 sometimes I do action a2 sometimes it's the same state s I do action a1 sometimes I am doing action a2 sometimes but I might be executing a deterministic policy it depends on what option I am executing when they come to a1 right.

The fact that the option is really part of the option you are executing is really part of the internal state is never made explicit in the options framework there is no notion of internal state or anything they are all left to you know kind of the implementers interpretation of options ok so

the lot of things which the options framework really does not specify in some sense that is the power of the option swimmer you can go fill in whatever details you want for it but is also in some sense dissatisfying about the framework in that it lets you do a lot of this work yourself.

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resources Development

Government of India

www.nptel.ac.in