NPTEL

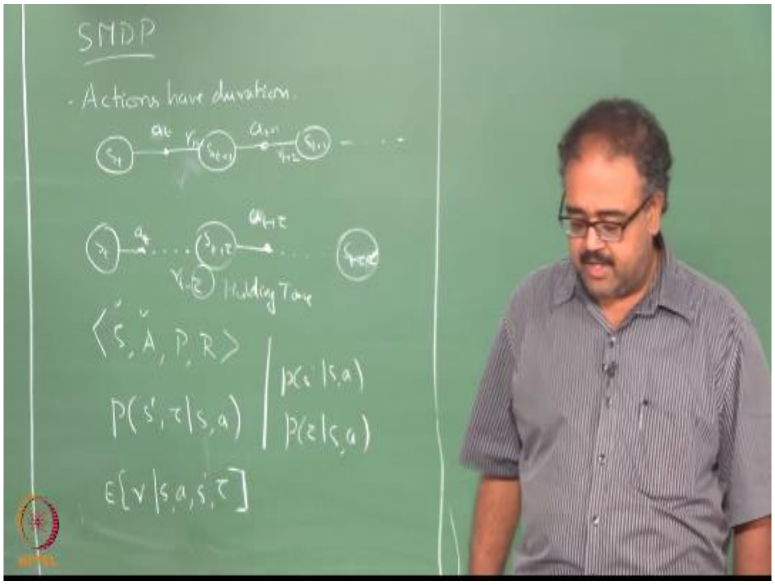
NPTEL ONLINE COURSE

REINFORCEMENT LEARNING

Semi-Markov Decision Processes

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

(Refer Slide Time: 00:15)


SMDP
· Actions have duration.
s_t a_t r_{t+1} s_{t+1} a_{t+1} r_{t+2} s_{t+2}
s_t a_t $s_{t+\tau}$ $a_{t+\tau}$ $s_{t+\tau'}$
$r_{t+\tau}$ Holding Time
$\langle S, A, P, R \rangle$
$P(s', \tau | s, a)$
$p(s' | s, a)$
$P(\tau | s, a)$
$E[r | s, a, s', \tau]$

So I said recursive optimality is good if you have a mechanism by which you can treat the solutions that you have as action. So that your horizon you are learning horizon becomes smaller right I do not really have to look for many, many, many actions to come before I make a, before I can learn something new right.

Otherwise I said it will not be a very useful thing, so one of the foundational structures that we use for doing that is called a Semi-Markov Decision process. So what is a Semi-Markov decision process or a Semi-Markov decision process, no that is just non Markov which is just came Markov, it depends on to two previous states is second order Markov.

So in Semi-Markov decision processes right, so a Markov process what do you have so I have a state ST I take an action as AT then I go to ST+1 I get RT+1 right. So and this happens again and again, so next I take an action right it keeps going after I have to see RT+2 so you can see my font from that way right okay.

So in Semi-Markov decision process what happens is I am in ST I take AT, then some time elapses I would find myself in the next state at time T+τ so a τ is some kind of a random variable that is sample based on ST and AT right I also get some reward 3+ τ. Now I take another action it is not T+2τ okay so T+τ + some other time τ' we are bother about the time right.

So because we use gamma power sorry, so I will use gamma power T RT+1 right gamma power one basically I will use gamma into RT+1 or no, no gamma into RT+2 gamma squared into RT+3 and so on so forth. Now what I will do is I will use gamma power τ into RT+τ+ gamma power τ' τ+τ' into whatever the next R and so on so forth.

And the next time I do the same actions right the the time taken might be different. So now I am not only interested in what the next state I am going to land up in is how long will it take me to get to the next state because that is how much I will discount the reward by, how do you get to sub problem.

Now maybe that is the best way to solve this problem it takes you five time units to solve one sub problem right. So I am not even talking about any sub problems here by the way you are getting ahead of yourself okay, this is just a Semi-Markov addition process there are no sub problem here, there are only states and actions okay, there are also problems here and they are saying actions take different amounts of time to complete.

So optimality now would depend not only on where the action takes me but how long it take it is to take to get there, how long it takes to get there right. What is it, explain the τ, τ is the time taken for the action to complete it is a random variable, so every time I take an action it may take a differing amount of times to complete.

Dependent on τ, in reality it would in many cases it would, but we kind of wave our hands here and decoupled the τ and the S' distributions right. So typically we end up decoupling the τ and S' distribution so it makes it little easier, but typically it should be a joint distribution right. So historically SMDP models used to do that right, but then Tom Dietrich introduced a variant of the SMDP model into the literature.

Specifically meant for modeling hierarchies right where he actually defines the distribution as joint distribution over τ and S' right. So historically, so people used to talk about this as holding time and keep saying historically people who use SMDP reward and other things will kill me because that is how they use it even today right.

So it is called holding time right, so holding time essentially the idea is that at time T, I decide A is the action I am going to take right, but I do not apply the action A till T+τ. It could be a variety of reasons I mean I could actually be sensing the market at some point of time and by the time I place my order it might be 15 seconds later or 20 seconds later or 30 seconds later that could be variety of things that come in.

These could be many reasons why you want to do that, I mean you could, that should be gap between the sensory input coming to you that end you taking in action right. So it is Semi-Markov in the sense that for the duration of the action I do not know how the system is evolving right. So to determine when the next state is going to occur I really need to know how much time has elapsed since the last state was seen right.

So some amount of history dependence is there, so that is why it is called Semi-Markov right. So I know that τ seconds have to elapse before the next state happens right, but how much of the τ seconds is left right. So that is why it becomes Semi-Markov not completely non Markov because once this is happened the probability of ST+τ it depends only on ST and AT right.

But during this transition when this will actually show up it depends on how much time has elapsed okay. So it is only for this brief periods you have the time dependence okay. In cases one is after taking action almost time does it take and for the next state to agree yeah, and how much time we wait before taking the action.

That is semantics right, so some people refer to it as holding time, holding time is when you actually, when do you apply the action right other one is you can think of it as transition time. So I apply the action immediately how long does it take for the next state to actually happen that is like a transition time right, both mean the same thing.

So for all people its transition time makes more sense right, but in the other the holding time the explanation was given only because of the semantics of things right. So what really happens if I already applied the action what happens to the system after the action is applied. So instead of asking that question you say okay I will stay in this state for τ seconds and then I will apply the action immediately I will get the next state.

So there you kind of side step the question of what happens after you apply the action and before the next state turns up. But we do not want to sidestep the question, I apply the action go to root 1 and I want to know what happens after that right. So I will pay attention to this, that is why the transition time definition is more acceptable for RL folks right.

So we have some more time so the SMDP is defined as the S, A, P, R, see the thing is these are explanations these are models that you have right, so this is at the end of it mathematically what is happening is there is a state that occurs at time T, there is a state that appears at time T+τ okay that is it.

So what happens in between people come up with different ways of interpreting that see one way that says that okay I actually apply the action at time T itself and it takes a system some time to evolve and settle down into ST+τ. In what way, say if we are applying we are assuming that action will be updated as soon as I sense and the state does not change before I apply the action that is one way we have.

Another ways the state changes before I apply, but as soon as I know generally we assume the state space for time τ that is why it is holding time so come to a state you hold the state for τ seconds and then I apply the action I go to the new state. So as soon as I apply action I, you go to the new state.

The other one is, but in some cases we will have both kind of delays in. It does not matter see the point about SMDP is I really do not tell you what happens in those τ seconds right and you can interpret it different ways but mathematically you end up with the same thing, see the quantities I am using for modeling as I will write it down now you will see what I am going to say that eventually end up with the same thing right.

In fact the holding time interpretation says the following I see ST I pick AT immediately, but I wait for τ seconds before I apply it right. So the decision is made before the waiting time starts, before the holding time starts. So that the holding time is still a function of the decision okay. So I have a SAPR so you know S we know A so we know all of this from the standard thing.

So P is defined as the following P I get, I define P(s, τ) given is S', τ given S, A. So earlier or probability distribution was S given S' given S, A now the S', τ given S, A right. And the reward becomes expect value of R given SA, S' and τ so why does τ figure in, I mean because it will actually end up getting some kind of discounting on it.

So it could be accumulating some rewards along the way, it could be just the waiting time so for every second you wait, you accrue reward at a rate of minus one per second or something right so you could give some kind of a rate of reward right. So it could be some –R is the rate of reward and then if I am waiting for τ seconds and basically I do an integral over that to get you the total sum of the reward right the rate is –R right.

So I could get some things like that right so the the reward that I get could depend on the the transition time or the holding time and so on so forth. So earlier people used to write this as two different functions, so they, right notice that this is not a perfect decomposition of that I am assuming τ is independent of S’ if I actually had used the product rule to decompose it τ should have been dependent on S’ or S’ should have been dependent not one way or the other right.

So but now I am just saying this, so this is the classical way let me not put it in earlier is the classical way of writing this SMDP transition dynamics as into two components one on the τ, one on the S’ yeah, but this is more meaningful for the RL domain but it is more meaningful for RL domain because depends on what is the problem that where I am going to land up in right the time that I am going to take.

For example, if I am going to go out through A or go through B right the time taken will be different for executing it start from the same state if I am choosing to go through A it will have some time if I am choosing to go through B it will have a different time. So depending on which S’ I am going to go for the τ will be different right.

So those things should matter right, so I cannot do a separate distribution like this I have to do a joint distribution over S’ and τ okay. Any questions on SMDPs how is that clear right we can write SMDP value functions, we can write SMDP value in equations hope all of you can write that, you can write SMDP value in equations, well depends if you can write SMDP value functions first right.

Write the value function and then write the recursive version of the value function right, and then then convert that into a bellman equation how would I decide on the what gamma, gamma is well that is the same problem with all discounted about formulations right, gamma has to come from the domain expert yeah the expectation has to be over with also.

So that becomes they are not very hard is it a little tricky to write this thing, but is not too hard right so I am giving as an exercise, I am going to leave all of you to go and look up bellman optimality equation or rather the bellman equation for SMDP can you put that recent advances in hierarchical RL paper bought one and had even right.

So there were lot of hierarchical RL frameworks that have been proposed in the literature and surprisingly you would get more information about these frameworks by reading a survey paper that was written a little later after all this hierarchy or the architectures are proposed then reading the original papers themselves right.

So there are two reasons for it, so kind of the authors of the survey paper had a little bit more perspective so they could see all the things together in a unifying framework and they could present it in a better fashion. And of course, I am very biased and therefore the one of the authors of the survey paper is a fantastic writer and therefore he wrote a better paper than the original writer stem cells right.

So my advisor so of course I am biased so read that paper so the recent advances in hierarchical RL and so he describes SMDP and SMDP value functions and everything very, very beautifully in the paper right. And then I am going to talk about a few other things that we can do first thing I want to talk about is something called SMDP Q-learning right.

So what do you think is SMDP Q-learning that plus minus one we need a -1 there yeah, in the gamma alright τ time steps if τ is one I do gamma power gamma right. So one then it will be τ only of τ is one I do gamma so that is basically SMDP Q-learning it looks very simple radiance it is just that the rewards are discounted.

So the tricky thing here is I have hidden some stuff into this RT+τ right, so what is RT+τ right. So it is the expected reward that I am going to get over the τ time steps it is not expected once I still sample right I just taken a sample over τ time steps, so it could be variety of things right so normally that in the classical SMDP framework we do not tell you how this τ + RT+ τ comes about.

So you have some mechanism by which you can generate it right it could be some $E^{R\tau}$ integral 0 to τ okay E^{R} something RT integral 0 to τ Dτ right. So that could be one way of defining this so that is a rate function, so R is the rate function and I accumulate that function for τ time steps right and that that gives you a reward right.

But what we are more interested in is to actually think of RT+τ as okay, so this RT+τ is a visual RT+τ right, so the reward the one-step reward that you get τ time steps after you took the action right. So maybe this makes it clearer, so I am not written something that is recursive so maybe there you go. So R/T+τ is essentially the return I have accumulated for the τ same steps after I have chosen action AT.

So in some sense if you think about this to world example, so I will start here let us say and then I say do B when I go to root 2 right, then it is basically the reward I get for till that point right and the value of go to root 2 here will have get updated by the value of the state here right. So I will take the Q here and do the max over A right and for the RT+R/T+τ I will use all the rewards I get till that point that will you take both those quantities and then update the Q of go to root 2 here makes sense okay.

So this is called SMDP Q-learning I am sorry, it depends on the reward function, the reward function is the usual -1/ time step reward function then you are minimizing time or if the reward function is also one when you reach the goal and gamma discount along the way, then also you will be minimizing time except that for this action RT+τ will be R/T+τ will always be zero right because you get no reward.

But then the slowly your reward will start propagating back okay. So notice that SMDP Q-learning does not actually look into the structure of AT right, it just assumes that this reward somehow comes when I take action AT right. So it assumes that AT has been learnt already so what is AT here in this example go to root 2 AT was go to root 2 right.

So it is not a simple action this is what you mean by the temporal abstraction so when you say go to root 2 it actually takes some 10 time steps to go to root 2, if I had said going to root 2 from here it was just taken two types if to go to root 2, so it would be different right. And there is some kind of stochasticity in the world right let us say with probability 0.9 action will succeed right otherwise it will fail.

Then it will take different amount of times even from the same state if I say go to root 2 it will take different amounts of time to complete. So that is basically so we will stop here with SMDP Q-learning. So from next class I will actually look into the different hierarchical architectures that are available and then we will see how we can modify SMDP Q-learning to work with different architectures.

Whatever you want, SMDP Q-learning defined it like this, because it is defined on the discounted return. So eventually at the end of the day you will still want to solve for the discounted return right, because you are trying to optimize the discounted return eventually right. So if you are trying to optimize average return then you can define your work functions differently.

IIT Madras Production

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

www.nptel.ac.in