NPTEL

NPTEL ONLINE COURSE

REINFORCEMENT LEARNING

Policy Gradient Approach

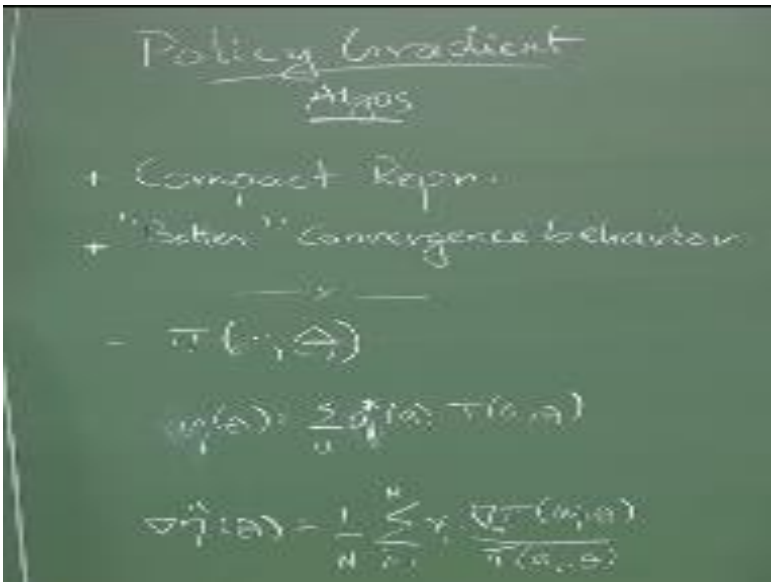
Prof. Balaraman Ravindran
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Sort of the advantages you do not try to find a value function good why so why, why is it a good idea not to try to find the value too far back a whole history if you do not remember it so it could turn out that the value function is a lot more complex in terms of the—the distribution of the value over the state space than this the policy so the classical example I give for this is the inventory control problem.

So the inventory control problem essentially the policies that you will be considering or what are called threshold policies that if the inventory is below X you buy if the inventory is above X you do not buy right so that is essentially what it will reduce to the policies will reduce to that but if you look at the value function so depending on some kind of a cross product of what are all the things that are available you will get some very complex function depending on how the rewards are work and so on so forth right.

So from are presentation point of view it is lot easier to represent policies in some cases than it is for representing value functions right especially when we go to the parameter setting right you can possibly represent policies with a lot fewer parameters then you can represent value functions possibly so that is one advantage right so more compact.

(Refer Slide Time: 01:47)



Right so the second advantage what do you think a second advantage again when we move to the parameter x forms right so value functions value function approximation methods rights are not guaranteed to converge any more right I mean they work in tactically so we have DQ n and other things but you really do not have much of a intuition as to how they will converge except in the case of linear function linear parameterization right and it is not one hundred percent clear how things would work but in the case of policy gradient algorithms.

So you have some kind of guarantees that they will at least converge to local Optima right and so that seems to give us a lot more leverage right in sense the sense that you can say though things will be stable you know if I keep running this for sometime it will converge to something right in the case of value function methods I do not know it might even divergent start by it is not oscillating so I have no guarantees as to the behavior of the methods under value function approximation.

So I am working with really large problems it might be better off working with policy approximations stand with value function approximations okay having said that I should also point out that almost all the large-scale successful applications of reinforcement learning right

have involved value function approximations okay so in theory with it does not guarantee to work but in practice it does seem to work of course you can always come up with examples where things did not work also in practice right so you can always come up with very very simple examples.

Like people talk about X or killing neural networks right so you can always come up with very simple counter examples where you can show that value function approximation is really not going to converge right there for people are always a little very about doing neural networks and other things until deep mind came along and now everybody does neural networks so if you now try to write a reinforcement learning paper without deep networks in it immediately your chances of getting accepted goes down by 20% right.

So that is how the times have changed of course except if you write banded papers okay so banded papers do not need deep networks in it right so the second thing is I am going to put this in codes better convergence behavior okay so what do value functions measure would not do policy gradient methods do so you actually have some kind of a report better within codes because it is I mean in practice value function based methods also seem to work with the function approximation but policy gradient you can actually give some kind of a guarantees about the business local Optima.

So what are the problems with policy gradient approaches we will come to that in a bit okay so after I describe what we are doing then it will ask you some questions and then you will know what the problems are okay so essentially you are going to save a policy π depends on some set of parameters θ so I am always going to write this s right so π parameter s by some σ right and then you have some definition of the performance of the system and then you are going to look at the gradient of the performance with respect to θ.

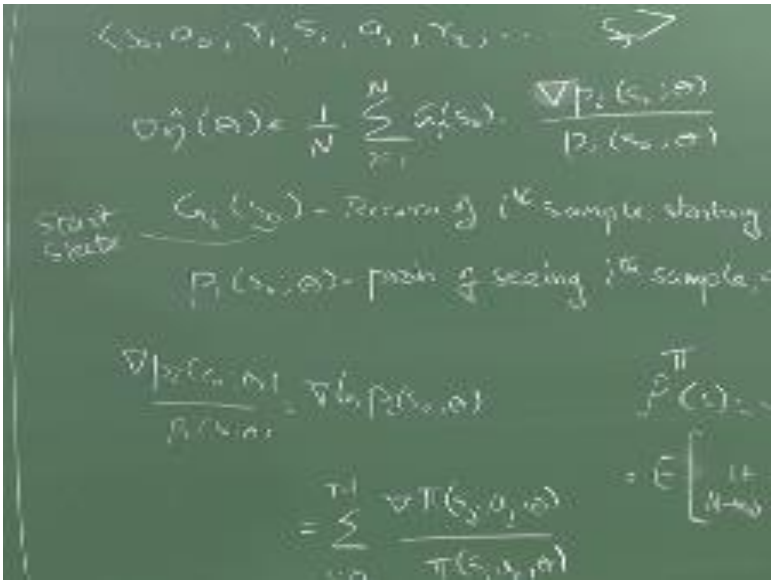
And then move in the direction of the gradient not direction opposite to the gradient right so far we have moved in the direction opposite to the gradient because you are minimizing error now we are maximizing performance so we move in the direction of the gradient right and then now

that is gradient descent we all know that it will eventually reach a global or local Optima under certain conditions right.

You have to make take small enough step sizes and there are small conditions on the step sizes and then your gradient estimate should be unbiased or at least biased favorably and so on so forthright so if you remember so we essentially talked about the right this is from the reinforce lecture right so we kept that as the performance measure right so which of course we could not estimate directly because we did not know Q* right so we did not know Q* so we could not estimate this directly so what did we do we did a whole bunch of sleight of hand right.

So we divided this by we took the gradients evaluated by π blah blah blah and finally we came up with a estimate for the gradient that looked like this right so where a is the action you took at the IH sample and r is the reward you got when you took the action EA ok so people remember this is for the bandit this is this is remaining refreshing see bandits are so far in the past rate I thought people would have forgotten all of this so I am just putting it down great. Now I want to do this for the full RL problem undo this for the full RL problem and what is the most straightforward way to extend this sorry yeah okay fine so my policy becomes s, a.

(Refer Slide Time: 09:08)



So my policy now becomes okay what are my samples my samples in this case where A I, R i right so my sample said I say I so what will be my samples in this case I say to think a little bit yeah I step the sample or something else a sample there are multiple ways of doing it right I am going to give you the simplest form of doing it let us go back to the basics right so here the Q function was the average of the rewards right what do u equivalent for writing it in the full our else case details Q function is average of the returns.

So if I am going to take the average of the returns so what should be the sample I am collecting on trajectory is not transitions you should make trajectories yeah so I should be collecting trajectories right so my samples will be of the form okay, so the time being I am going to assume that it ends wait I will be collecting trajectories like this so now what I can do I forgot what notation we use for this forgot what notation we use for this but this is GI which is the return on the IH sample starting from s not right.

And this is the probability of seeing the H sample starting from s not right, so what is this probably PA we starting I am starting from a snore the trajectory right that is not a not our one the whole probability of seeing that is the probability of the sample rate is C so what did we do here we said I take action Ai and then I am going to get our a right so the so this probability of taking action AI is here right then I get Ri right here.

Similarly the probability of taking this trajectory is here and if I take this trajectory then I get DA right given this trajectory GI is probability of g is one because the rewards are all fixed in the trajectory given this trajectory probability of GA is one but what is the likelihood that I will actually take this trajectory under policy that is why is the θ is here right under this policy π what is a probability of me taking this trajectory all right it is more tech off policy of policy learning when we talked about it is very similar to the off policy learning.

Except that there are one policy at the top one policy at the bottom the transition for already is not dependent later they are going to cancel but if you do a little bit of sleight of hand I can actually convert the product it is a product right it is a product if you think about it is actually a

product term you can convert the product term into some term it is actually the derivative of the log right so this thing do not try to write P I first and simplify it take this right.

So now what is P a P I is the probability of taking a knot in s times the probability of going to s 1 times the probability of picking a 1 in s, s 1times the probability of going to blah blah blah all of those things then times the probability of seeing our one given you have picked this all of this is actually a very long product right so that product can be written as so it is lon of the product rights with some of the launch so I can write it as the sum of the launch right so what will be the terms I sum over will be exactly the ones that I want to cancel out right so essentially I will say okay some of lon going from s not to given is not a not going to s1 okay.

Times the probability of going from s1 a1 to s2 so all of that if you take as one term they are all canceled because the derivative will be 0 and so what will be left out with the probability of taking action here not in its not as the probability of taking action a1 in s1 and so on so forth so that will be those will be the only terms that will be left all right so this essentially reduces to I am not going to work it out we can do that right.

You would not take a minute and convince yourself so this is the case I mean so right out π as a product right take the lawn convert that into sums right and see which of those will where the gradient will vanish and then whatever is the left out terms are exactly the ones that I have written that right so even though I do not know p i-- because I do not know the transition probabilities I can write the ratio or this is called the likelihood ratio if you remember so I can write the likelihood ratio as just the terms in π okay.

So here one small thing which I have kind of any questions on this so once I estimate the gradient then I can essentially update my parameters along the direction of the gradient right I can do an online of data can do incremental update whatever I want to do I can do we have various variants of whether I am doing online incremental info on surface but that is one thing which I have done a small slate here it explicitly talk about it if you look at the performance measure η right.

So performance measurements is a function of γ alone it does not depend on state s or anything like that right so essentially here there is no notion of a value function associated with the state but I still need a notion of a value that summarizes the utility of a policy right and what I had done here no, what have I done to give a single value for the policy read I mean if you remember the value function case for starting in every state I had a different value right but now I am saying they will use a single value for the entire policy so how did I get that single value for the entire quality what have done here?

The s not does not have an eye on it I am assuming there is a unique start state right I am assuming there is a unique start state and I am using the returns from that unique start States as the evaluation for the policy and this is fine right if you think about board games rights always we will start at the same board game a lot of other control things you will always start at some particular given a certain amount of resources and we always start in a particular start configuration and so on.

So in many cases this is actually an acceptable thing to do right using a unique start state what happens if you have in stuff a single start state but I have a set of states over which you can start you can take an imaginary start state and the uniform probability you can transition to one of those states without any action and from there you run a trajectory so then you look at the value of this imaginary so we spoke about this earlier also right so with so I am basically using this s not as my unique as a unique start state right.

But more commonly there is another assumption that we make for reducing it to a single value function single value for a policy so what is the assumption we make no no no no yeah I mean as with all Monte Carlo methods that you need to assume that the trajectory ends here right if it is trajectory running on forever you cannot use this method as it is you will need to modify it somehow I will talk about that in a bit but that is not the end state is not the is show here says something else which people do.

Which is exactly where the average reward RL field fit tin right so if you think about it what does average reward reinforcement learning tell you but we already looked at the average about

formulation right so 1 / n as n 10 infinity R1 2 right so R_1 R_1 R_2 R_3 + R_N tight, so essentially you take the average over n steps and then as n tends to infinity your computer right so if you think about it because n tends to infinity any initial variations that you would see in the value for starting instate 1 versus starting in state 2 will all get wiped out at any initial variations will get wiped out.

Because you are summing up a infinite number of infinite series of numbers so as intense infinity the values for starting from any state we can average value for a policy starting from any state will converge to the same quantity so in average rewards right so you essentially you have only something called ρπ, where there is no like we value function V we have so in average reward you have value of a policy denoted by ρπ regardless of what is the state right so ρπ of S will declare as being equal to the constant ρπ for all s and that is defined as.

Of course there are some conditions for all of these average values to exist and be well defined and so on so forth so we are assuming that your mdp satisfies all those conditions then all you need to do is for a given policy estimate ρπ and use that where instead of G is not instead of G is not you would use ρπ initial variance by doing this which by doing what the average reward will essentially as lower variance thing right but the point is well obviously you are not running it will n tends to infinity right so it is only a sample right.

So you is a sample estimate so the ρπ that I use there will not be the same it is suppose to be the same but obviously it will not be the same just like we know that g is not will not be the same right the ρπ is the expected value of this right so I should put the expectation around this right is it clear so we so I am not going to get into the 90^0 of this if I if I have a lecture I will go do this I tyπcally do average RL but almost always I find that 90% of the class does not get it okay.

So one lecture is two short a time for me to explain all the nuances of average reward RL right so I end up teaching the class but people not actually getting anything from it so if you remember that that is enough for the purposes of this lecture and maybe for active critic also later right so any questions people understand what Monte Carlo policy gradient is averaged about it turns out that there are some certain convergence whistles that are easier to prove if you are using average reward RL one thing.

Second it removes completely removes the definition the need for a γ right so γ is actually a weird thing right for some problems γ is naturally defined that in such cases it is okay to use γ but in many cases what happens is you try to artificially define what the γ should be γ becomes a meta parameter that you tuned to get the result that you want right so that can lead to all kinds of complications right and it also turns out that there is more recent work which I also personally have not looked up yet is showing that function approximation is better behaved with average about value functions rather than discounted value functions in fact there are cases where you can argue that for discounted value functions the target for function approximation is not even well defined right which is not the case for average reward value function.

So there are many advantages some of them are theoretical many of them are theoretical right but some of them are so practical in the sense that it is better it is a certainly a more robust model than using discounts because when you change the discount factor even slightly sometimes the values can change significantly right so that said these are the advantages the main disadvantage is usually average in what RL is numerically very hard to get it to work.

When you implement averaged reward reinforcement learning and then you try to see so get it to converge and things like that it cancels a major pain right so numerically averaged reward are always getting hard to get to work, but ironically they had that better theoretical properties but in this case well you need something that is well defined like this it is a single value for the entire policy in case you do not have a s not and also turns out it is easier to show results for policy gradient when you have averaged rewards okay.

So there are a whole bunch of the caveats and other things which I am kind of seeing under the carpet so for example size limit n tends to infinity right so what if all your trajectories are only finite length as we are assuming here right what is the definition of average reward if n can never tend to infinity the simplest answer to it is well you always assume that n tends to infinity right you always assume that you have a recurrent class of states right so which if you get into those class of states with the probability 1 you will stay in those class of states right.

So this is some subset of your state space so that with probability 1 you will always stay within the set of states right if it is a single absorbing state if you have a single recurrent state right not a recurrent class but a single say that has absorbs everything then at least you assume that you do an infinite number of transitions into that right you do not stop the trajectories and your average in what becomes meaningless because it just becomes a reward of that one action right so you are assuming that you have a non-trivial recurrent class of states.

Which is these are the set of states that are of interest so under this policy you will keep visiting these set of states again and again and you will not stop right so only then the average or what becomes more interesting right and so this is called okay, let me put this so and man I do not want to get in to more complications yeah let us assume this is I will just stop here right I do not want to talk about other things if I have time I will do all of that I will do one more detailed lecture on average reward.

And so any other questions on this? How Monte Carlo policy grading will work all of you can go and implement one know I can ask you the policy gradient on the puddle world problem are we asking you to concentrate not it you fear for your life rolling statements already states it as hard it's already thought is it is very hard functional but from the one we gave him one is lasted which is the function approximation for Q's the polar parameterization and then we asked them for bonus Marcel for a better parameterization way no I remember asking that see they do not want you to get bonus there was a good suggests better parameterization part of the question.

Anyway so I used to ask the policy gradient version of this equation also all right so drawn policy gradient on the puddle world but let us not do that for you guys okay fine so there are a few caveats here so what do you think would be the biggest problem with this right somebody already mentioned this I think you mentioned this what will be the biggest problem with running Monte Carlo gradient estimates like this the trajectories are very long right you get you will have very high variance.

So every time you run the trajectory I mean you are drawing so many samples for the reward right if the rewards unless the rewards are all deterministic and the transitions are deterministic

so every time you draw a sample it is going to be different right so there will be significant amount of variation in the kind of samples at your drawing right and therefore the variance will be very high and this is one of the reasons why policy gradient methods typically tend to fail because the variance is so high just to just start to run these things forever right.
Unless you do something clever to reduce the variance right, so what do you think you can do to reduce the variance cannot do that right you are essentially having some parameterization θ then you change the teeter by something essentially you will have to say that for every parameterization think I am going to have a parade of mystic policy you have to come up with a parameterization like that that will severely limits the glass of things are possible truncate predictable exactly so that is what you have.

I will keep at it the problem is if in trajectories are very long so next up short all wake up shrunk it right you just say that okay I will only do a short term estimate of my return right I will γ is going to decay anyway right if I am using average reward why is it okay to truncate it see if I have done enough circles around the states right I would have a good enough estimate of the average of all right I do not have to keep doing this infinitely long for me to get an estimate.

Suppose I go five times around the loop right I know what is average in what for the loop is right so train of roughly so I can do that race I can I seen stop saying that I will run till time end of time which might not happen if I am assuming a recurrent class right it might not end at all I need to have some way of stopping the trajectories so what typically we do is we say that okay, here is a unique state I will start the trajectory from the state whenever I reenter the state I will stop the trajectory.

So that is all I take a bit of the trajectory I keep doing this so this will allow me to reduce the variance so discounting is one thing right so I can because of discounting my trajectory will get truncated the other one is to say that I will designate a unique recurrent state whenever I enter the recurrent State L will stop the trajectory even if even after doing that if you get long trajectories what you can do you can designate multiple states at which you will stop the trajectories essentially the idea is to get shorter and shorter trajectories okay.

Yeah so sorry we had a question answer coming from this average written but the variance was coming actually from me from the whole sampling process here yeah so there is another method for reducing variance right so where is the variance coming from because I am sampling G's right so what are G's the returns right so do I know of a way of getting an estimate for G that has a low variance close value functions value functions are the expected values of these gradients rate this is G's right if I know value functions right then I can low variance estimate for the G's because every time I do not have to draw the sample I can look at the average of the samples I have drawn so far and soon so forth right so if I can plug in the value function instead of G at some function derived from the value function instead of G's right I can reduce the variance of the updates I am making variants of the estimates of the gradient right.

So that leads us to what are called as actor critic methods so the actor is the π representation the θ gives me the actor right the critic is a guy who evaluates this actions right he is against it now is good bad thing this is the critic the critic is the guy who I will plug in here so the critic is used to evaluate the actor right and the critic comes from any of my favorite value function estimation techniques right I will typically use a parameterization for the Critic as well right.

I will use some function approximation for the critic and that function approximation will feed into the actor function approximation I will estimate the act based on that so these are called actor critic algorithms so I will look at actor critic algorithms later possibly the next class but basically I have told you what is the idea behind actor critic now only you know answers that will have to think about on how do I actually write this in terms of a value function.

Right how do I take out the return and replace it with the value function so that is the only thing that we need to think about as well as there are some technical conditions on the relationship between the parameterization of the actor on the critic right, I am going to use some function approximate for the actor some function approximate for the critic right and they cannot just be arbitrary function approximation there has to be some kind of a connection between them so these are the two things that you really need to think about so we worry about this in the next class but right now you stop here**.**

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resource Development

Government of India

www.nptel.ac.in