NPTEL

NPTEL ONLINE COURSES

RENIFORCMERNT LEARNING

LSTD AND LSTDQ

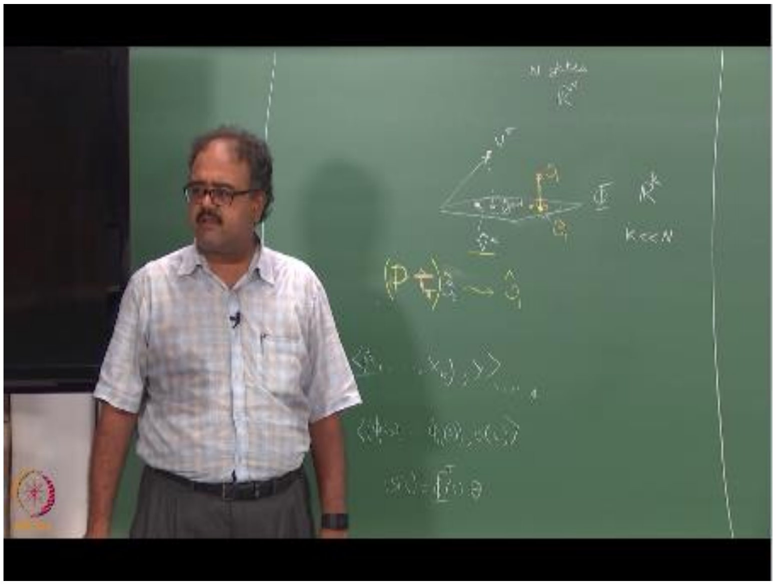
Prof. Balaraman Ravindran

Department of Computer Science and Engineering

Indian Institute of Technology Madras

so what exactly does linear function approximation do, right so ideally what you would want it to do is the following right so, so I have some abstract space in which I have some vπ that is being represented, okay so I have an abstract space in which vπ is represented and there is some plane right then some space which is which kind of expand by my feature vectors Φ, right there is some space that is spanned by my feature vectors Φ right but there is my real value function vπ sitting in some other space this is clear, right.

(Refer Slide Time: 01:11)



So what pace would be π residing suppose I have n states right, so vπ resides in a R^n state space right so it is an R^n state, space where each quadrate corresponds to the value of the respective state, right so the core first coordinate will be the value of the first state the second coordinate will be the value of the second set and so on so forth will be a point in RN, right and my Φ might be something that spans a much smaller state space that my Φ could be something like R^k where, ideally I would like that to be the k because I do not want to be representing an n-dimensional stats in which case I can represent the tabular form right I can difference in the actual value function right.

So I do not want this k should be far far less than it, so in this case I am assuming that n is 3 and k is 2 just so, that I can draw any other higher dimensions it will be hard for me to draw, right so ideally what would you want to happen, right let us say that, right so that is ideally that is what I would want my we had π to be right that is a close as possible I can get to given that I have to stay in this plane, right people must have done some kind of linear regression curve fitting in things like the projections is essentially what you are doing is the closest projection to the plane, right.

So this is essentially what we are trying to look for right and what exactly is this in terms of, regression fits that you are getting, least square solution to the fit, right so the least square solution to the regression problem that you have, where your targets are drawn from vπ, right will essentially be the we had π, right it turns out that there are some results that show that if you are actually going to be updating vπ we had π incremental right.

So essentially I have this vector ϴ which I will be updating incrementally under certain conditions on Φ and so on so forth, right your v̂ will not converge to this so we will call this v had*, right so I am not going to go into the proof of that thing if you're interested can give you pointers, good whenever talk about pointers for one just walked in you can put the citric listen van Roy paper on the convergence, of TD width value function approximation, right.

Now we can give a link to the paper on I think it used to link that paper any way from the previous editions but since the previous edition simple of disappeared you can add the link, to the paper right so we had star would be the least squares projection when it turns out you will not get it normally so what you will end, up doing whatever you can end up showing is you will converge to some we had right which is within a certain distance of we had star right that distance to which you converge is bounded right.

So that so it is within abounded distance of v* you will converge okay so this is basically what you can show, so I so let us not call that star I am going to call that v had optimal way that is an actual optimal thing I want and star is the one that I will get at, convergence right so maybe hat star will be very close to we had optimal but except in some rare cases it will not be v had optimal okay, so why you think this is happening ideally the best solution because anyway we remember you are minimizing the least squares error right we are minimizing least squares right that is a way we wrote the error function right.

So ideally the minimizer should be we had opted right why are we not converging to we had talked some to later noise what kind of noise non-stationary so that is the problem right not just when Evan is a sample error it's essentially the target keeps changing over time right I pointed this out last class also, write the target that you are updating towards keeps changing over time because you are using the v̂ itself, as your target that keeps changing over time so that introduces a little complication in the dynamics.

So you cannot guarantee that you will converge to v̂ opted, that if you knew the true vπ values and if you knew the true vπ values you will converge to, v̂ opted if it since you do not know the true vπ values and you are essentially boot striping you end up converging sub something v̂ stared okay, nice thing is at least that given some mild conditions on Φ, the distance between we had often we had storage bounded, okay.

So there is a nice thing is what makes what mild conditions they had to be orthogonal the Φs have to be orthogonal, and then you have a linear you know not wrestle orthogonal yeah linear linear independence is fine thereto be linearly independent and you have to sample on policy, right so on policy sampling is very important, if you are linear function approximation with the dynamic programming style updates can be shown to diverge, so dynamic programming dress uniform sweeps right it is not on policies every state is of every state is updated equally often, right so that is certainly not on policy and we can show that that will actually diverge right.

So think of what is possibly happening in the in the update, set up right so I start off with some, suppose I suppose I want to do the updating on my let us let's say you and even on this case right I have a fixed π right so I what do I start off with, I start off with some guess, for we say I'd say in start off with some v^0 right so I have it start off with some representation for v^0 right so that will be somewhere in this space, then what do I do some kind of an update operation right so this is I am just using my, value an operator to summarize it but if it's Q learning or if it is TD learning for everything I can write corresponding operator right.

So the TD learning operator would essentially be some kind of a sampled version of this valve an operator right so essentially I apply some kind of an approximation of the bellman operator to my v and what will I do I will get a v1, right but my v1 will typically be not here, so start off with self me not right there is no approximation here right I start off with a V^0 I apply tπ on it right whether it is a one round of TD update that or anything else I apply something on it, right then what will essentially happen is this will go to v1 p 1 right so we not really apply πe on it t π on it will go to v1 and what do I do at that point I project it back because that is all I can represent right.

So that is the best I can do in some sense what I am essentially doing is projecting it back till it hits the space so that is my V1 hat, so I apply, see normally the projection matrix protect projection operator is indicated by T or row right so I am using p here even though p we are using for transition probability I'm also using p for the projection right so I am do a projection so in some sense, that is a joint operation I use to go from V0 hat to V1 hat right so in this case there is no approximation really because I started there but I can do this so this takes me to you one hat right and I keep doing this again and again until I converge to do I do it that way if you think of the update that I am doing and I just change the Ɵ directly right the very thing that he does is I find the gradient of the performance in the direction we'll find the tilt angle Ɵ in the opposite direction of the gradient right.

So that's essentially what I just to be reading this look at the approximation error I can they go in the opposite direction so why did they compute the new v1 I am not I am not explicitly computing we weren't right I basically computed v1 hat anyway conceptually what we are doing is this kind of it well you can think of it as a composition that I applied t followed by p, I mean if the word joint is confusing you okay it is a composition operator, it so this is essentially what is happening in terms of what we are trying to do in function approximation right.

So one thing which people started thinking about is, hey we are trying to solve the regression problem right can we use can you set it up as a proper regression problem, and so what do I mean why I set it up as a proper irrigation problem how do you solve how do you normally solve linear regression problems you are given some data set we are given some x_1 to x^2 input dimensions and then you are given some y right.

so like that you are given many many such data vectors, and then you form a matrix out of it so we form a what is what is the matrix called what is the matrix called, data matrix yeah he should know a different input the setting up of nothing spells only happened no no so that's good right off her shoes instant domes here no way before one has only down to prison today I am going to pick on you unless all the, design matrix right so the data if you get is also called as I am pretty sure many of you have heard the terms in design matrix it essentially means that that data set it's, all the excess.

So the X that you ate by comparing all the except of the same the normal regression problem will require us to have something like, shall have an input vector and a Y and then I will have this index over right if I want to do something like this for, value function approximation what should I do I should have some vector is basically we have right.

So this is what I should have right this should be the data that we should Φd to the repressor like this I should have some n samples makes sense right the only problem here is maybe of ass I don't know maybe FS so what we do for V of s and I am going to assume that my V of S comes from right smoky transpose s into ϴ right you could if you want solving this semantically where you want to TV then you can set it up as a multi-core no problem and then you could just take the return starting from that state and then give that as the sample on the so we’ve set up a perfectly valid but only right why is it uninteresting, because what you will converge to we talked about this even the I had this question in the exam also right.

So what have what will you converge to if you use Monte Carlo a fixed sample Monte Carlo is the least squares minimizer of the prediction of the return right but what you will converge to if you use TD for a fixed sample is the certainty equivalent system a same with the Markov model that is a more interesting quantity to estimate than just the least squares estimator, this quiz anyway right so we can assume that vs is given by this right and for my current setting of Θ right I could just plug this inhere right.

So whatever is may I make initial guess for Θ then I make one big data set like this then what I do I choose my favorite least square solver right and get a new estimate, for Θ right so then I keep doing this again and again right so now if you think about it my input is in the form of some Φs right and the output is in the form of some Φ times Θ right now I can actually try to solve this in some kind of a what are all the bigger data point here we are talking about all the states we've come to the data generation part later, of the plate almost surely you will probably not have samples or all the steps but then for some stage you might have oceans ok so now let's let's look at it so what I really want is suppose I have this this ,whole operation that is happening right my P times Pπ of V of Vπ hat Vπ at right this is my fixed point that this is this is essentially what I want to achieve right I want to get to my fixed point solution.

So if I get to my fixed point what will I have so what will this look like now fixed by π right so this will become my pπ if you remember the matrix pπ that we had earlier so that essentially the transition reduced to what happens when I take my action corresponding, to the state, so instead of PESA s̓ that it will be PS π of s separate right so that is this matrix oh I have reduced later I can remove the action column from there so this is essentially PS s̓ right where the action is fixed to be π of s we looked at this matrix when we saw the convergence and the of the bellman operator right now what is remaining I really have to look at the projection business, right.

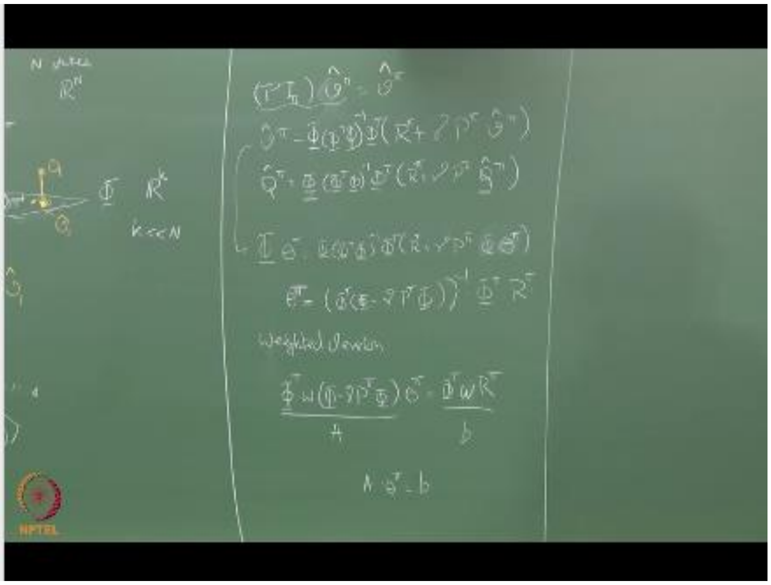
So I've written the t π operator here, I now I have to look at what the projection is great look like watching the projection look like what is the X what is X here, that's the projection you remember the linear regression solution okay where Φ is written for all these states, right Φ of s this one row corresponding to state when I don't write any arguments to Φ it is entire matrix so each row corresponds to one state, I have written the entire bellman equation there for me, so it turns out that with this with a slight change because I'll use it later I just write it down here I'll use it later I can also write this form where this one will be taken appropriately for s' right.

So whatever is π of s' so q of q π of s', s' okay, that will be the q π here and one other changes Φ will be Φ of sa my Φ will be defined on state action pairs, later remember we talked about how we do you go from v π to q π I gave you several methods that one of them was to have VSA at instead of vs, I am assuming people remember anything we did in the last class right, sorry I'm assuming people remember everything we did in the last class is right and so that is the change here right, so Φ will be Φ SI right, and this is this is this is different and this will have to be careful about okay.

So otherwise you can write something very similar to what we did for v okay, so now what do we do, now that we have this so I can write this as that is the entire vector okay so this is a Φ matrix, at Φ Θ so each row will get multiplied Θ so that will give me the whole vector V, this is v π solid yeah so perils of on the fly notation translation whenever I make whenever I put some symbol there which I have not seen before please stop and ask me it might be something which I forgot to translate anyway right.

So I have replaced my V̂ with my Φ Θ by we're here no so this is free transpose yes because of doing one is a vector thing these are this is a matrix Φ right so when it take this will essentially give me a vector so this is Φ Θ this is matrix multiplication not inner product of two vectors, makes sense so where the Φ each row in the Φ is a state right when it take this Θ is going to go like this it will multiply the whole state and it will give me, the value for that state okay, right now, you want me to do the algebra or you guys want to do it and then tell me what Θ π would be, I can take it to that side do some simplification blah blah blah, okay the people are convince good.

(Refer Slide Time: 24:07)



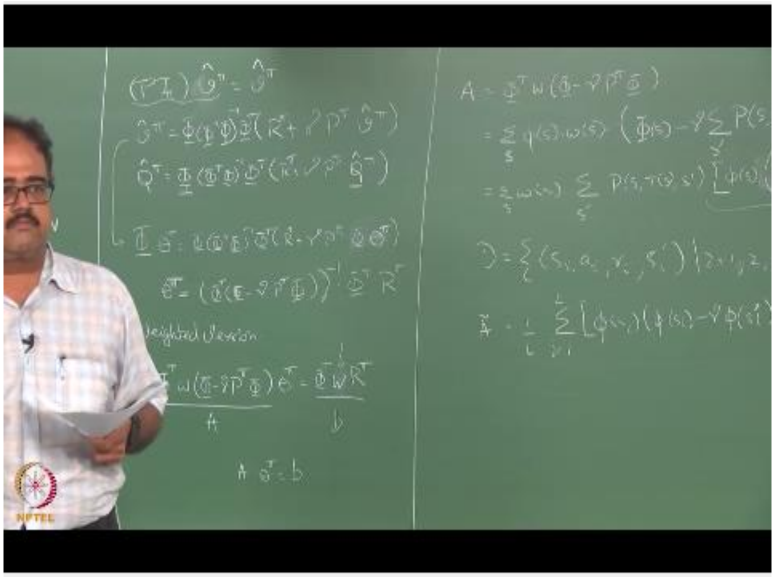
So now I am going to do as little bit of the thing, so one of the so this is this is this gives you the least square solution here sorry Oh yes I mean so this is the least square solution right so what is what is striking about this least square solution is that it treats everything as an equal contributor towards the error, every state has an equal contributor towards error right every sample as an equal contributor towards error but then we know that that is not always see the right thing to do, right well we have a small enough expression cover right.

So we really want to approximate those states that are more important for us right and then leave out those states that are not that important, right so what you do essentially is instead of looking at minimizing a least squares fit you minimize a weighted least squares fit so then what would happen is in your solution you will have a weight cropping up, at different points, right so what we will do is we will rewrite this and while I do the rewriting I'll also introduce the weight, right I am not going to redo the whole, thing again so I am going to write a weighted version, transposes we're w some kind of a weight function over all states, right that tells you how much you want to weigh statement what is the typical weight function that you want to use, the steady-state probability distribution rights steady state probability distribution it could be anything else, whatever takes your fancy right.

So you could use that weighted the probability distribution so this is essentially what we get so I am going to make it little simple I am going to call that A I'm going to call this B, so I'm getting A×B, so the rear ID is not just for fancy okay, so this is going to allow us to do little look at each one of those quantities their little closely, and see what they are actually doing so A and B, okay but now we are not doing any incremental updates are integral I am giving you a bunch of samples and I'm solving for it, yeah yeah I want to approximate the steady state distribution hey and if I do the sampling on policy are the problem it is whatever physical, yeah.

So that is a good point so all I am talking about now is getting the least squares fit for a given set of samples, given a fixed set of samples I am trying to get a least squares fit right and I can get that least squares fit regardless of what the w is I will get a least squares fit given a set of samples, right I have no guarantee about how good that fit will be from the viewpoint of my true v̂ opted and if you use any arbitrary thing it so I with in terms of how close I am to be had talked I have no no guarantees if I use any arbitrary w right.

(Refer Slide Time: 40:24)



So we know that if it is on policy, right then we will get that v̂* given infinite number of samples I mean of course so as it is all asymptotic convergence guarantees right the more samples you are given the closer you will be to be v̂ right but you'll ever get to v̂ opted right and that is guaranteed if your w is going to be your own policy distribution right but if I am looking at any fixed set of samples and I am talking about just getting some estimate that fits that sample, I can choose whatever w I want, I said you would want it to be the steady state distribution but even if you choose something else.

It is fine why would you want to choose something else could it be because you do not know the steady state distribution right you do not know the Surrey she ate distribution one way of approximating it is just look at the relative frequencies in the samples that you have drawn already, right that could be very very far away from this steady state distribution, infect we end up using that distribution only because you do not know what the steady state distribution is so you end up using the distribution that you infect from the samples that you have drawn, people with me or people lost know okay see I said that even if w should be the steady state distribution probability said all of you agreed to that, and that is the ideal situation that we want right.

But we do not know what the steady state distribution is I mean you would know that until you either you solve the Marko chain set up by the, policy and then you know what it is but for solving that you need the transition probabilities, so that is not going to happen so you do not know what the two steady state distributions are, right how do you approximate the steady state distributions by counting it from the sample that you have right, right when you have count from any fixed set of samples right

The distribution that we infer can be something very very far away from the true w that you want right the weight set you get by inferring from a fixed set of data could be very far from the W that you want so the data is very very large write as T tends to infinity doing an incremental update will approximate the w, but we are not talking about T tending to infinity we are talking about a fixed set of samples, here so given a fixed set of samples the W that you get can be very far away from the true w so you just live with it okay.

So let us go and look write out a right, now this is essentially my pπ that transfers comes from, this whole thing gets modified because I went formal matrix notation to a individual entry notation right so essentially if you look at what is a products that are being computed here, this will work out correctly okay so I flipped the free thing do not think I have flipped the transpose that is because I went from the matrix to component waste notation right.

So this Φs times fierce is the product that you have here, right and it will sum over all s’ so this will drop out okay so essentially you are getting your Φs Φs product from here, right the cells like I have added an additional I unnecessary Φsa prime to that product so that I can take it inside, the inside the summation, okay does it make sense right so I have for this this terms I've added an unnecessary Φ s’ term to it.

So that I can take it inside the inner summation right so that is essentially what I have done the rest of it is the same, right nothing else no no fancy tricks here at the reason we did this is so that we can write this as some kind of a sampled thing this kind of an expectation so if you think about it what they say this is from some kind of a weighted sum of this quantity, right so what I can do now I can keep sampling transitions from s to s’ according to this distribution right compute this quantity difference between s and s’ way I compute this quantity again and again by sampling according to this distribution, right and then take a average weighted by how much importance I want to give to each of the states, and that gives me my, a right.

So we got around one of the biggest problems nobody actually asked me how can is what can I say I have solved the linear regression problem here, when I actually have the Φn in here, right I have the Φn in here and I have R^n in here right I cannot say that I have solved the linear regression problem from samples, right because I am assuming that I know the model and if know the model I might as well solve the model directly right why am I doing this the whole idea of doing all of this is I do not know the model, right.

So I need to somehow I figured out what my pπ is and our πe is instead of figuring out P&R directly right so we can have actually way of estimating the e matrix directly, right so what I can say is given the set of samples, right of the form my samples are going to be of the form I am going to get last samples of this form remember, so this is the current state action reward and the next state okay that is that is the form of samples I am going to get ok.

So I now normal notation it will be st at rt+1 st+1 right, the I index just denotes this is sample ok nothing to do with the time indexing, is it make sense this is this I is nothing to do with our time index it just denotes is IH sample so I am going to get states like this now what I can do is I can from an estimate for Ã this is my Ã, okay so a is fine what about B, sorry I’m not a hard present ship yeah system number right so the whole vector is the Φ is fine right.

So what about B I need to B also something like this we can write is almost most, clear was it good question yeah yeah so this will get I am kind of wrapping the W and the Into the frequency right so yeah we are making a very strong assumption that everything is fine, right so that the number of times yes occurs here right is exactly the number of times you want to got into the w right then it is fine, so I can do something ok let me to B first before I move on to anything else, right

So that is what Rπ is, if people numbers Rπ rates Rπ is essentially summation over all s’ Φs is s’ ×rs s’, so which is essentially the expected reward that you are going to get and when we wrote it as an expectation I am just writing it as a simple things, and I can do the same trick that I did last time, I took the Φ inside, right so just like last time I have taken the Φ inside so what I can do now is same thing that we did earlier for A so I can write BΘ as, sorry ,right I can do that so I can also write the action version of this right.

So yeah here there is no transpose, so like I saying great the Φ transpose here lose the transpose when I rite aid component wise, so I can write the action version of it also like this right remember we wrote the same thing earlier, here so I can run through this simplify the whole thing right and I can get the same thing here right except I am leaving out this part I want you to fill it in tell me, about I use there’s a tricky part I might not have this in my data in s’I might not have done, see a si’we have to be careful about whenever we use this things right.

So you make sure that you have done is that's the on policy part of it right now likewise I can go eat beef tilde here this is a relatively straightforward in fact that what I wrote first, so this kind of an approach where we use an estimated Ã and B̃, right and solve for a value function π is called

LSTD least squares TD, right so when you operate with the v function it is LSTD when you operate with the Q function it is so it's called, LSTDQ.

So you have LSTD & LSTDQ and you also have an LSTD lambda, where you do this for TD lambda as well, ok their things become little tricky, so here in some sense my ace repository of my model right so in the LSTD lambda, the γlambda terms will appear here also, here the γ term appears, right we need two LSTD lambda you will get a gamma lambda term than your a matrix right.

So so that is the trickier part right so that’s essentially what will happen in the difference between LSTD and LSTDQ lambda is very convenient, because it looks like it is a very small change in fact right and said something else that you note here so we're adding my solution forΘ I am solving for Θ right and in case you forgot with all this mechanization we are doing we’re actually solving for Θ π right.

So when you solve it for Θ π have here this say they have be that same then I am going to do now Ã B̃ and what is one thing that you notice if I have to Ã B̃ it should be invertible let us assume that if we solve if my Φ are linearly independent it will be invertible, right I can show little linear algebra I can show that if the Φs are linearly independent, okay this will be linearly invertible and then, what else may 1/l will go away right.

So the one by I do not have to carry it around I can just throw it away because it will get cancelled out on the A and B, right one way I will go and then what are you left with just a summation of terms of this side right, so what does this tell you that as I keep getting more transitions, I can just keep adding to the A matrix that is very very easy to update my a matrix incremental, right I can just keep doing this iterations right and then as I am sampling more states I just keep adding to the I matrix so I can get a much larger I can keep improving my estimate of the value function right as I keep getting more transitions so at any point of time I can just take my current state of the A matrix and solve it for v π or q πgreat so this is called LSTD or LST DQ.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resources Development
Government of India
www.nptel.ac.in