INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

REINFORCEMENT LEARNING

LSPI and Fitted Q

with

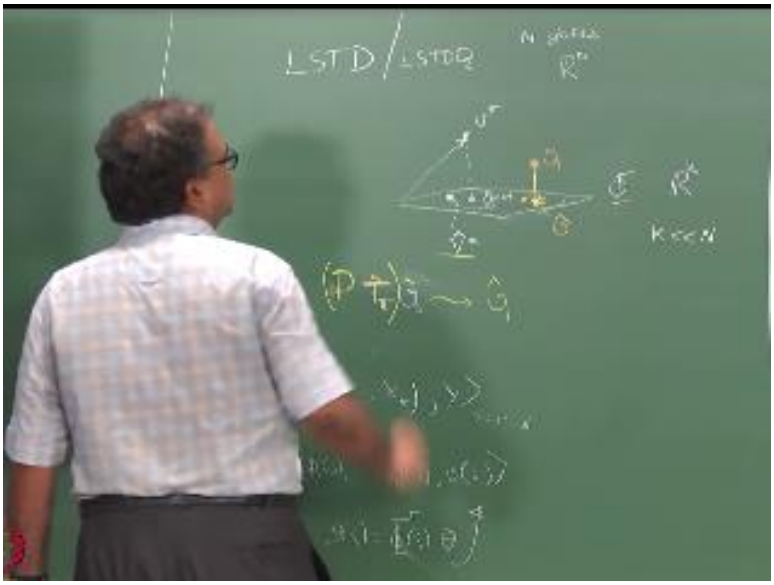
Prof.Balaraman Ravindran

Department of Computer Science and Engineering

Indian Initiation of technology, Madras

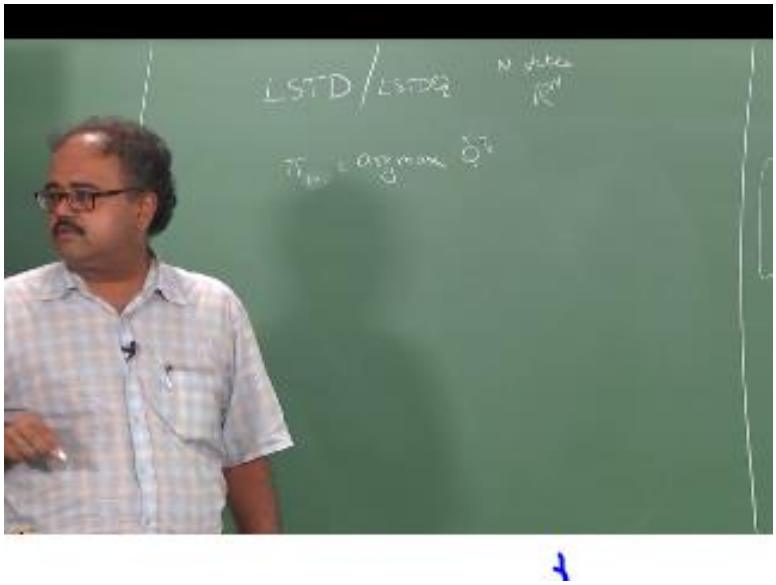
So what is next well I have found something for the policy by then what should i do control right .

(Refer Slide Time:00.28)



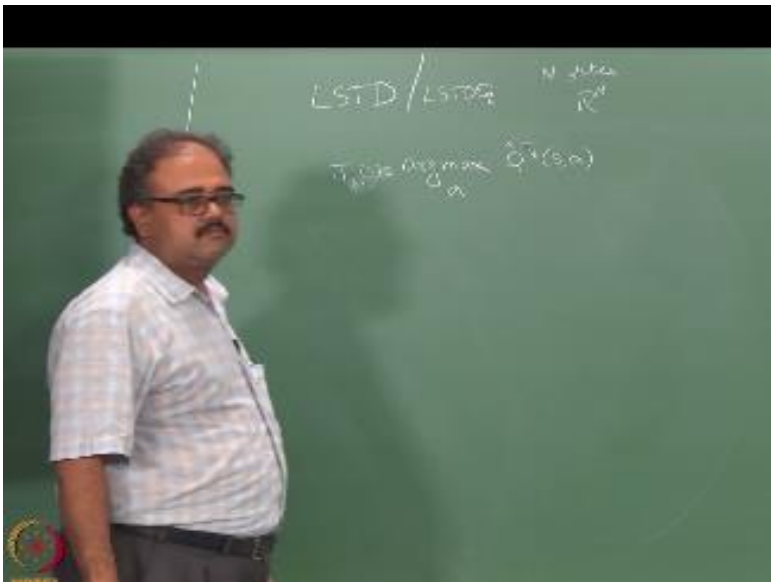
Come on somebody say something before they say that use the key values you already go do you sell STD you and then look on the next one okay . Take a policy solve it find the Q function be greedy with respect to that hit a new policy evaluate that so and so what's the tricky part

(Refer Slide Time:1.09)

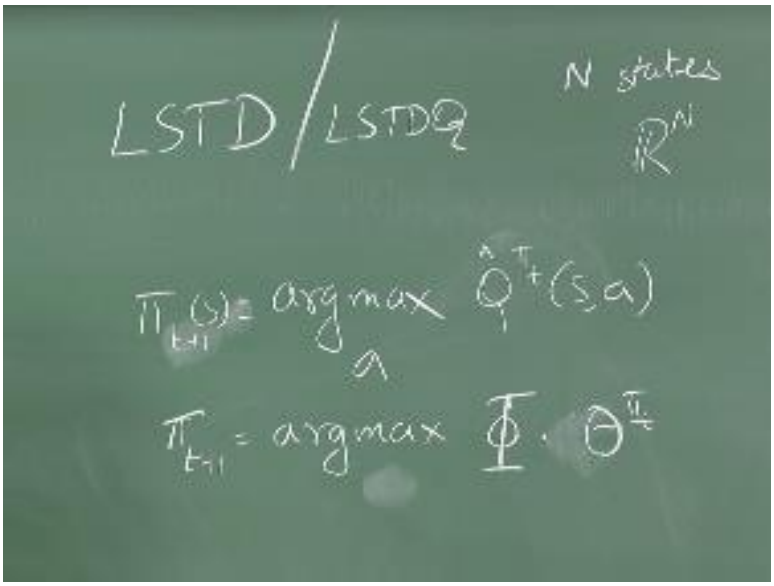


Right so this is what we are saying so for every statement where the arc max is done state-wise right if this is little confusing then

(Refer Slide Time:1.45)



(Refer SlideTime:2.18)



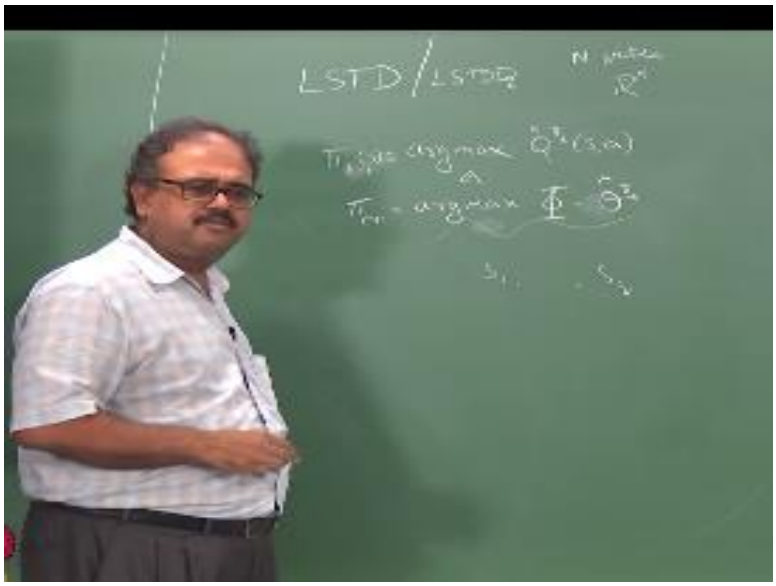
So now what is this and this little easier for me right or more appropriately theta hat free tea okay now what are the problems this is where we have gotten three guys know what are the problems here. The arguments is over 1 Artemis is about one action our map is all one max Anderson always will we do policy creation they always look at deterministic well it all depends on how my states and actions are represented right so I am looking at the maximization problem if it may dimension is simple it is fine but my fee SE a parameterization some how translates that into a continuous value right or if my action itself is a continuous value hood action.

Right how will I do this that if my state is a continuous valued state what do I mean by saying I am going to find the max one very state right state might be cutting is valued but my VS might be something smaller write something more compact right so if VSA is happens to be really spanning RK right so what does it even mean to say I am running a maximum maximization over that representation right.

So we have to come up with some clever way of doing this right there are many, many ways in which people have people are set this up as a linear program with love set this up as other ways of solving this right there are clever tricks Lucy people come up with and the many, many ways in which you can solve this right so one thing which I particularly liked okay this is from the old technique from like 2003 or something so what people do is this somehow solve this for a few states this all this maximization problem for a few states.

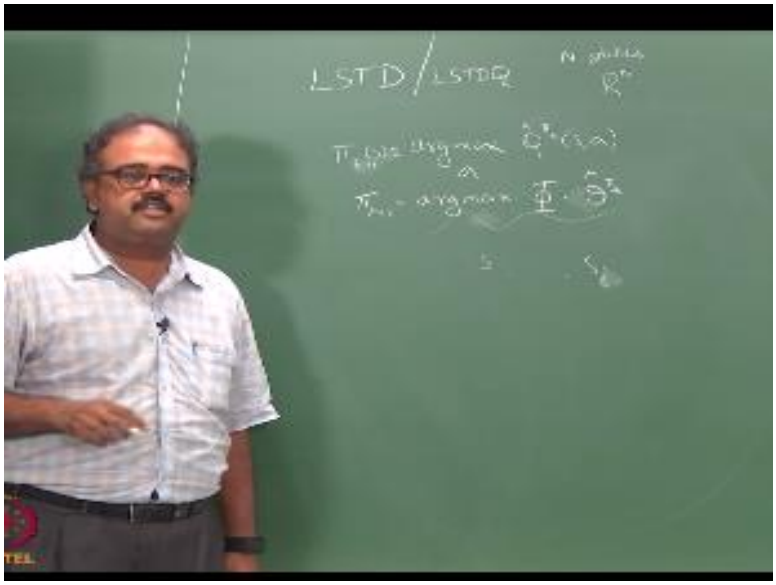
And then what they do is they get a estimate of the hat by solving a classification problem so what will I do is I will solve this for some number of states right some, some , some number

(Refer Slide Time:4.46)



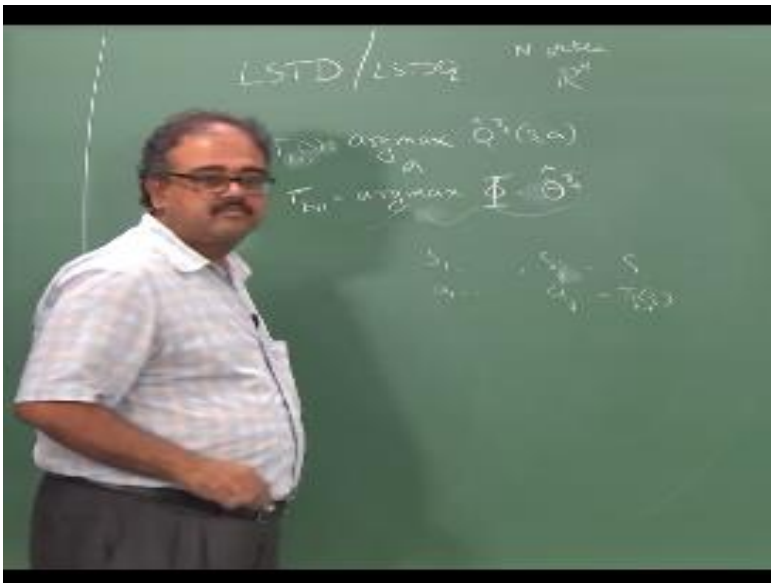
SSS so S 1 to S Q right choose a bad, bad thing to use here I use of all, all letters what letter haven't I used SX

(Refer Slide Time:5.15)



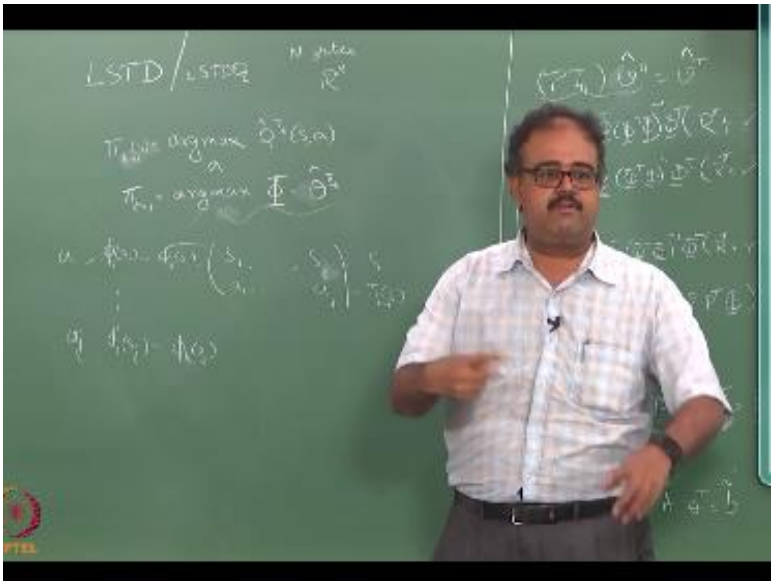
Yeah so S 1 to S J right so I am solved it for some problems this one test and what do I have I have corresponding actions A1 to AJ what this is this is yes this is like JI JI T +1

(Refer Slide Time:5.29)



I pick some arbitrary specific points in state space it can be a continuous state that it can be continuous actions with continuous action I have some way of maximizing this so I just pick some points in space and I solve this so now what I do is give this as a training data not this by.

(Refer Slide Time:5.53)



This I mean I will give see one now first one to come I give this as a training data to a classifier I didn't have it solve the problem learn a classification problem and why a classifier I mean because A 1 to AJ are some fixed set if it is a finite action case right if it's a finite action case I will give it to a classifier if it is a continuous actions case I will give to a regress or that will give me some value that will generalize across all the unseen streams was it representing the policy using approximator right now we do not have an explicit representation for the policy right we are just using the cube function as a representation for the policy.

We are saying if you want to recover the policy you just be greedy with respect to the Q function right but this since this is a hard problem to solve right i am trying to approximate it by saying okay instead of solving this exactly this maximization execs explicitly and getting a Л representation I am going to represent Л using some kind of a function approximated.

And now I still have a way of generating the samples if I want right. Yes no people see that see once I have a policy I can start from state yes go to this policy pick out whatever is matchimg according to this policy perform that if the next state in their stash what I am doing here so this is the problem that I really want to solve right suppose my S is a continuous state space right then it becomes hard for me to solve this and get a closed form expression or I'll have to solve this maximization every time.

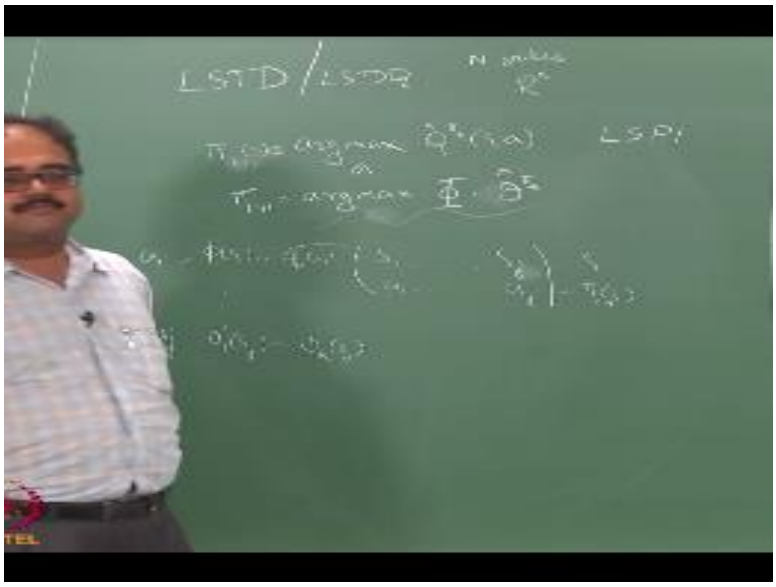
I come to this state S I'll have to solve a maximization problem and find what my action is and I do not want to do that I want to say precomputed and keep it as an explicit policy so that I do not have to do the maximization every time I have to pick an action it right now we just do the maximization every time you pick an action because it's easy to compute because we have a look up table is easy to compute but if I have to solve a linear program every time I have to find the action I do not want to do that I want to pre-compute this right and I do not want to solve it for every state what I do it I what I do is I sample say J States I solve the maximization problem on those J says.

I find the best action for those J states so I have A1 to AJ right now what I do is I create a training data that says ok the input is C1 S1 to PKS and then output is A 1 so like this I form a training data right where the, the said output is the action that came out of the maximization on the input to that is the encoding of the state some encoding of the state right so now this will earn some function F right.

So next time I come to a state I passed the state to the function F okay and it is going to spit out a action and I will take that action so the F is my representation for the policy JI it is another, another approximation so I'm just maintaining an approximate representation for the policy so that is one way of handling this right of course it has is one thing the fee need not be the same as this fee it is a completely different function approximated right so I can use a different fee if I want right.

First of all if I am doing Q functions this fee will be a function of S,A right and that is a function of states so I have to use a different, different set of features anyway and this is one trick for doing this right and yeah so what do you think this, this kind of an approach is called policy iteration

(Refer Slide Time:10.08)



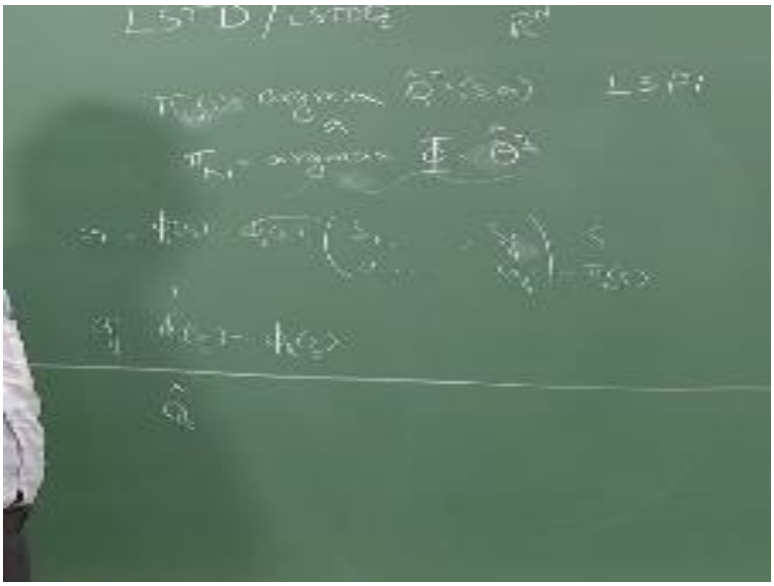
Policy iteration least-squares policy great events policy iteration noise least-squares policy duration right so can we put up the LSPI JML our paper also online so I took I took the discussion for LSTD & LSPI from the same LSPI paper so if you read the originals LS TD paper is a little hard to follow what they are doing because it was like a seven page paper or eight page paper a short paper and there is much, much more expanded discussion here and they used to see uniform notation for both LSTD & LSPI so it's easy to follow .

And then they have a whole bunch of experiments there are small, small things that you have to be careful about which they describe very nicely in the paper okay. So that is one thing the next thing which I wanted to talk about was something called fitted Q iteration it is very similar in spirit like very similar inspirit so what do you do there is you do you do Q learning right so how do you do Q-learning you start off with a with a fixed set of samples you start off with a fixed set of samples then you generate your targets.

Right so how are the targets generated like so like we talked about here right so I am going to generate a V JI as my target right now but that VJI is not the target so what will be my target and Q-learning R plus max over next I so I generate those targets right so what do I do is with the same data right I train a function approximated to approximate that target okay sounds very similar .

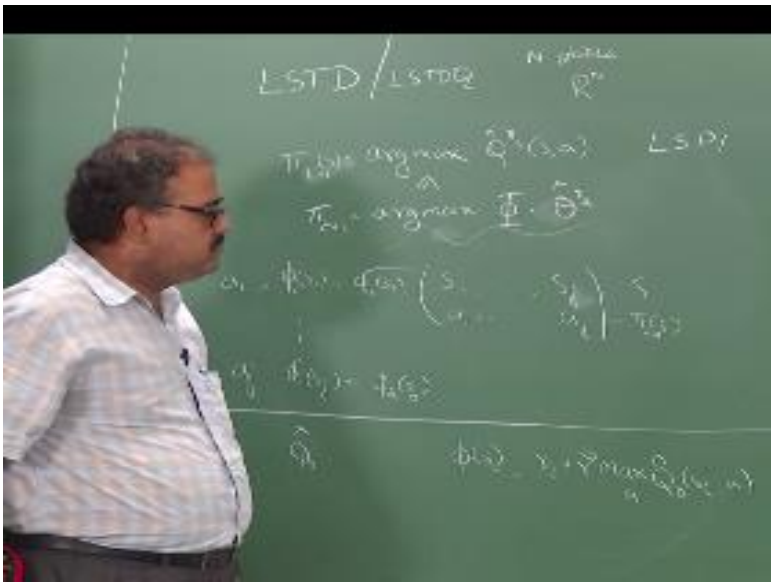
I train a function approximate, approximate the target then what do I do so one thing here is I did not put any linear constraints or anything here that they can take any function approximator right that will approximate this Q function then I will take the approximated Q function and create a new training data why because we are training data dependent on the Q functions output right I am seeing too many blank so I have some Q had to begin with right.

(Refer Slide Time:12.37)



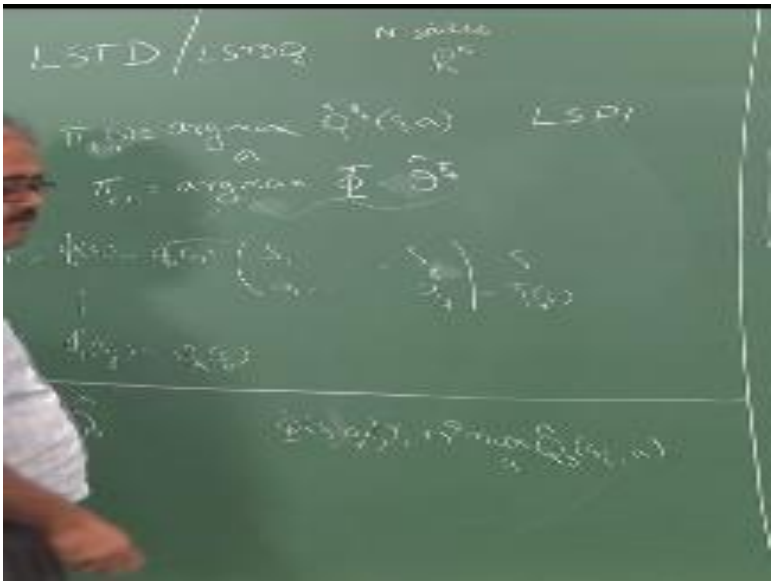
So my training data will be

(Refer Slide Time:12.45)



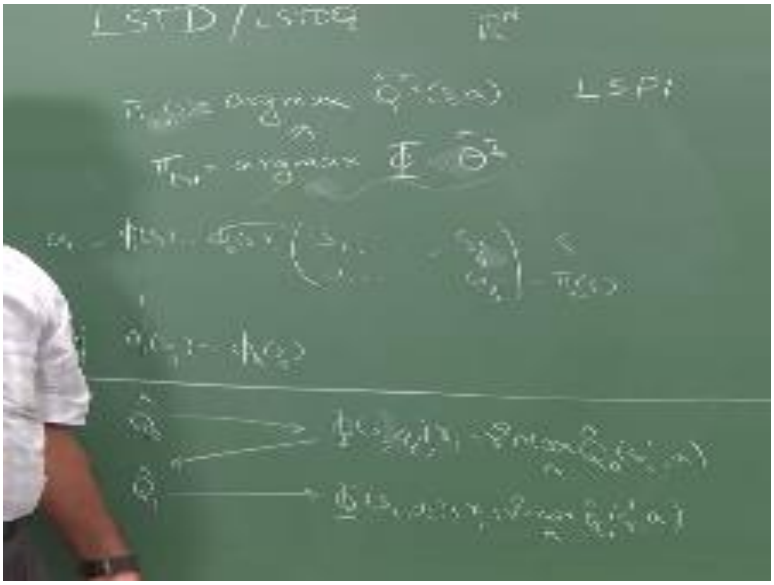
So PSI remember that PSI is a vector right so that is a the encoding for the state right or encoding for the state.

(Refer Slide Time:13.21)



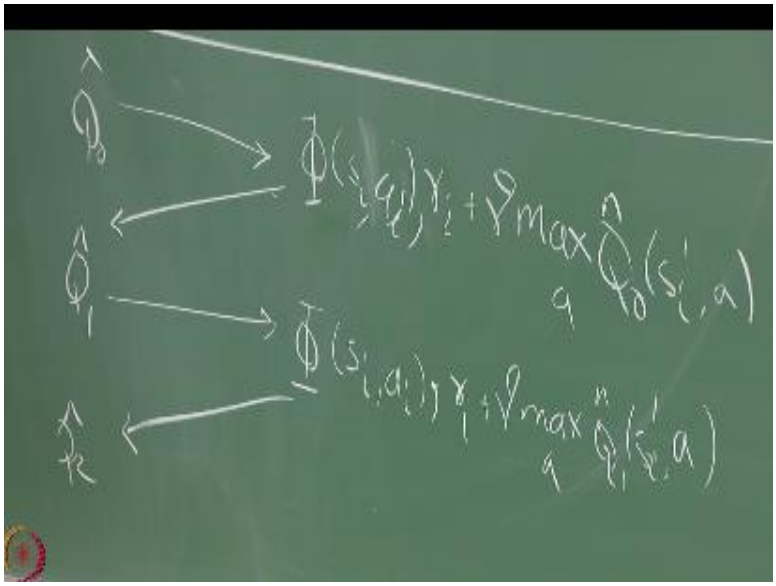
Action pair okay so I give that as the input and the output is RI plus γ max over a Q naught hat SI prime comma a right so like this I will do this for all I so the data is of this form okay the data is of this form I form the training data of this form I feed it to my function approximator so whatever is my function approximator right so my function approximator will take this as the input come here I create this is going to give me back q 1 hat right so I'm going to take q 1 hat.

(Refer Slide Time:14.09)



I create that okay it makes sense and solve that.

(Refer Slide Time:14.44)



give me Q2 hat right and I keep going so what is what are the things are to be careful about here well whether I can do this maximization properly that means I should hide all actions from here right for me to be able to do this maximization so I need to have generated sufficiently random

samples that if I have not generated sufficiently random sufficiently insufficient number of samples right then this is going to become a little tricky right.

So this are some caveats with the filtration so how is it generating that are I from that cube so my data is of this form way SIAR is SI dash is the drain data I have this is the trajectory that I've written from these trajectories I am generating that for my classify but that form are regarsar right so my regression problem has to be something like this right so this is what where we saw erased it unfortunately but that is where we started off with for the whole thing right and from that I said we are going to do a least squares fit and then we assumed linear parameterization and then we derived all of this right so here we are not saying we're not rush eating or.

So this quiz you can you could do this quiz with a linear function approximation but what we are trying to do is the iterative path here right so I am trying to converge to the optimal Q function without going through the policy iteration cycle this is more like getting inspired by q-learning kind of an approach this is called fitted q iteration so it is iterating over the Q function and I am fitting a function to its source called fitted q iteration and I don't know so I might actually comeback to fitted duration and tell you a little bit about some of the nice properties and caveats on fitted duration because it turns out that if you have a fixed set of samples right and you have really no way of generating more samples right so empirically fitted q iteration seems to do pretty well that is one part of it the second thing there are a lot of things on in the fitted q iteration setting that people are now reusing in deep q-learning made in the deep learning all this ring a bell Atari alpha ago kind of stuff right a lot of the ideas that were developed in this fitted duration sitting is being used so I would really like come back and cover that a little bit maybe

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resource Development

Government of India

www.nptel.ac.in