NPTEL

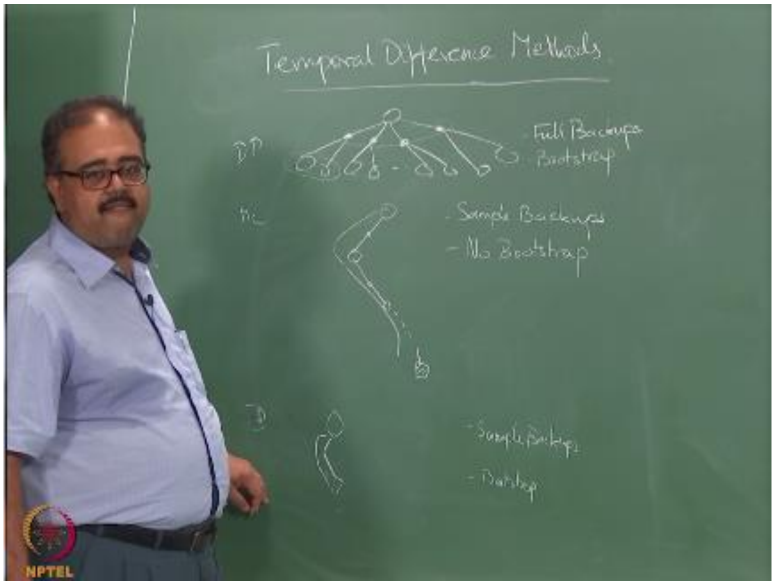
NPTEL ONLINE COURSE

REINFORCEMENT LEARNING

TD(0)

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

(Refer Slide Time: 00:18)



Okay so now we come to what I think it is in some sense these the soul of RL okay temporal difference methods right I think I think TD is very important semantic we can argue that RL is all about learning from samples learning to control dynamic compute complex system by just sampling from the system and soon so forth in which case like the Monte Carlo methods we talked about are also reinforcement learning methods right but for me what is really cool about

RL is the whole idea of temporal difference okay to what do temporal difference methods are many ways we shake and motivate this did any of you hear me in cinema securities class last lecture you did what is event count variance you are securities less.

Yeah there are you are notice you in the class you keep too quiet or maybe we did not show up that day you did okay anyway so there are only processor but he do not count now you are not taking the class for cadet are you know even car maybe so I motivated in a very different way temporal difference is I motivated in a very different way from a completely neuroscience or not even Euro sign is more like a behavioral psychology point of view okay so now I will do it purely from a computational point of view right.

So let us start up so what did we do when we did the dynamic programming right so when we are thinking about what we did with dynamic programming you start with some state s right then you look at all possible outcomes if you look at all possible outcomes from that state right so what is so this is all possible let us say these are actions right so all possible actions I can take and all possible outcomes from those actions right so what does this correspond to make my π SA right.

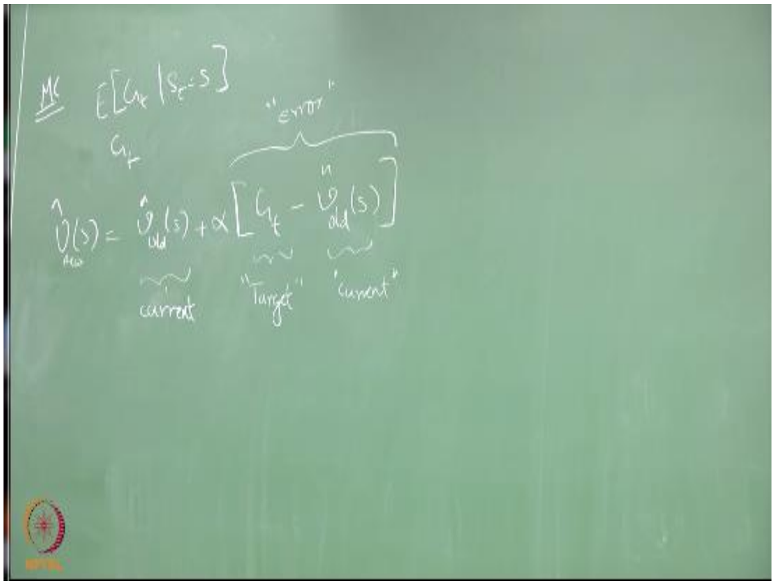
So sum over A π SA so that is what this corresponds to and what does what does this thing's correspond to this month sum over S’ PS’ is given s, right so this is essentially what I did when a dynamic programming right so every time I did a backup I took the entire possible outcome right and it took looked at each one of the state's here I use the values of the phase and did that back up right I did a backup so this kind of backups are called full backups right and it is also distinguished by the fact that I am using the values here right this is called boot strapping I did a full backup and it did that using bootstrapping.

Okay so next we looked at Monte Carlo method so what we do in 10 Carlo methods I started in the state took some action went to another state took some other action went to another state I did this till I reach the terminal state then I use the rewards along the entire trajectory and updated the value of that state right so what did I do here these are called sample backups because I did not take the full possible set of outcomes I just sampled an outcome okay it is sampled an outcome and I do not use any bootstrapping so I do not use the value of this state update the value of this state right of course.

In the three such thing we are talking about we did use bootstrapping but now we are not using any bootstrapping the plain vanilla things you do not use any no bootstrapping correct so what do you think I can do next bootstrap and sample yeah so I can do sample backups with bootstrap now will be extreme that is what is one sample and I will bootstrap just like we did here one sample but you find all the way to the end we did no bootstrap so here what I will do I will start from some state I will take an action I look at the next outcome and use the value of the state to update the value of this state it in the full backup I do all of this and update there is no simulation also involved right I am just using all possible outcomes well here there is a simulation involved here there is a simulation involved but here is only a one-step simulation I just used the value of this state to back up.

So such methods are called temporal difference methods why are they called temporal difference yes the value at T + 1 to get a estimate of the value at T right so essentially you look at the prediction you make at time T + 1you said for changing the prediction you made at time T right so look at the difference over time so it is called temporal difference you look at the difference in predictions over time we call this call this a temporal difference production okay hey that is a very pictorial motivation for why you want to do something like that okay let us do a little bit more formal thing right.

(Refer Slide Time: 07:37)



MC $E[G_t | S_t = s]$

G_t

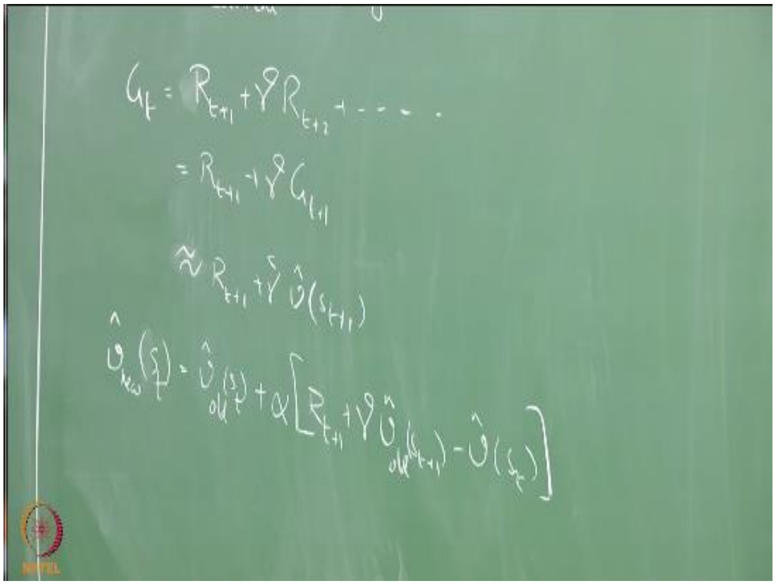
$$\hat{v}_{new}(s) = \underbrace{\hat{v}_{old}(s)}_{\text{current}} + \alpha \overbrace{\Big[\underbrace{G_t}_{\text{"Target"}} - \underbrace{\hat{v}_{old}(s)}_{\text{'current"}}\Big]}^{\text{"error"}}$$

So if you think about what we did with Monte Carlo methods is that I was interested in looking at that right but at every point of time I had some sample GT okay I did not have the expected value I just had some value that they had sample as GT right so if I want to write this as a stochastic averaging method right so what will I do I have so v hat is okay let me not put that little confuse people a little bit so v hat let me not put little bit new so I want to form a new estimate of v hat what will I do I will take v hat old + some α times right this will give me we had ν so this is the classical stochastic averaging rule let me look at from very beginning you have been see stochastic averaging rule I told you that almost everything that we will see later will get converted to a stochastic averaging rule right.

So this is stochastic averaging version of what Monte Carlo evaluation right this is Monte Carlo policy evaluation this is stochastic averaging version of it right so you can think of this as old value okay that is target-current okay so the target – the current prediction that added to your old prediction gives me the new prediction right and this is some form of a error term right okay so I can rule I can make this me at this is this is what we say sophistic averaging rule right so in the Bandit case then we showed you this α should be 1 / N and all can keep a constant α on all of those things we talked about it so this all of you are on board with this expression.

So this is essentially the incremental version of simple Monte Carlo simply Monte Carlo estimator right without all the other complications I spoke about today no importance sampling the thing have a policy file and evaluating the police file target is this the expected value of GT right the target does not expected value of GT but I do not know the expectation I do not know the expected value of GT so all I have is a sample right one sample of this quantity of whose expectation I am taking right so I plug in that sample here instead of the target yeah no I am talking about Monte Carlo here I am not talking about this why I say that is why I kept saying this is the incremental version of simple MC okay everybody on board so this is an MC not really everybody onboard with that.

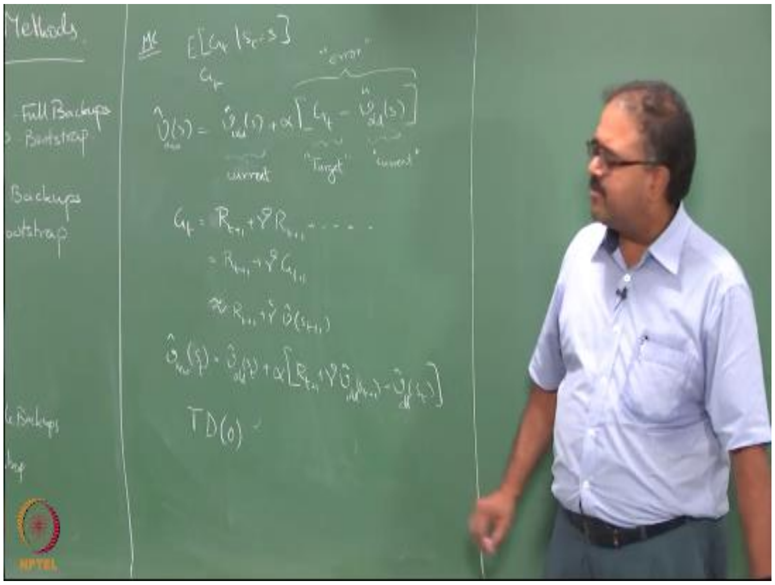
(Refer Slide Time: 11:15)



$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots$$
$$= R_{t+1} + \gamma G_{t+1}$$
$$\approx R_{t+1} + \hat{\gamma} \hat{v}(s_{t+1})$$
$$\hat{v}_{new}(s_t) = \hat{v}_{old}(s_t) + \alpha\left[R_{t+1} + \gamma \hat{v}_{old}(s_{t+1}) - \hat{v}(s_t)\right]$$

Yes no maybe so now let us go look at this target fellow closely so what is the target it is GT right so what is GT going to look like right so we will do the same trick that we did earlier okay if I know the true value actually I do not know the true value I am going to write this as now can you plug this into this if I want to I mean right so now I can say that.

(Refer Slide Time: 13:34)


Methods
Full Backups
Bootstrap
Backups
ootstrap
e Backups
trap
MC
$E[G_t | S_t = s]$
G_t
"error"
$\hat{v}_{new}(s) = \hat{v}_{old}(s) + \alpha[G_t - \hat{v}_{old}(s)]$
current
"Target"
"current"
$G_t = R_{t+1} + \gamma R_{t+2} + \cdots$
$= R_{t+1} + \gamma G_{t+1}$
$\approx R_{t+1} + \gamma \hat{v}(s_{t+1})$
$\hat{v}_{new}(s_t) = \hat{v}_{old}(s_t) + \alpha[R_{t+1} + \gamma \hat{v}_{old}(s_{t+1}) - \hat{v}_{old}(s_t)]$
TD(0)
NPTEL

Let say everybody on board with that yes no maybe so notice when new is where the old is and where the T is and where the T + 1 this is important okay so remember it right so I do not what you would tell me later that you are using value at time T to form an estimated time T +1 very careful you are forming an estimated time T + 1 so that is why this is the indicated ν initially I wrote VT and I erased it wrote ν because I wanted to make the distinction okay so the value function estimate is ν estimate but I am making it for the state I encountered at time T okay.

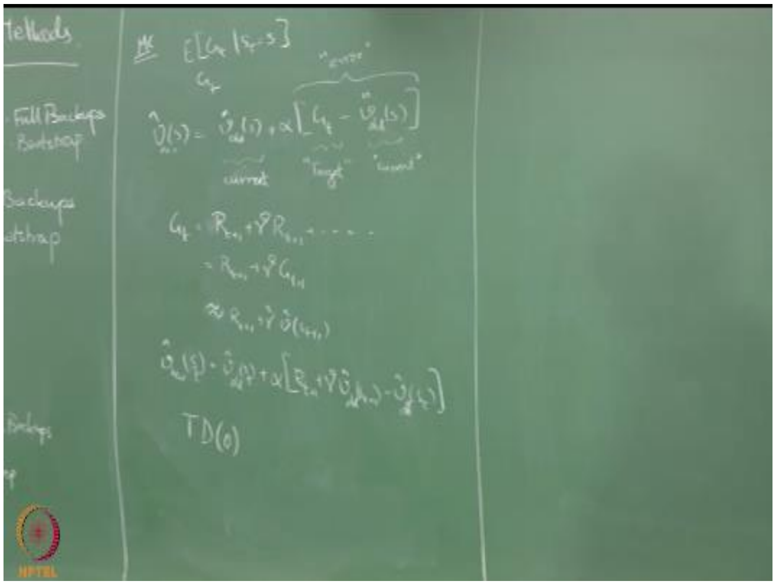
And I am using the old value I had for the state I encountered at time T+ 1 right so this derivation clear right so what we did here very simple lesson see the same trick that we use for deriving the bellman updates right so the same trick I used here except that I plugged it into instead of plugging it into the original return equation I plugged it into the Monte Carlo update alright I use the same trick let me get into the Monte Carlo update I get this expression okay so if you think about it this is exactly what I wrote here in the picture right.

So I am going to update the value of this state I will use the reward I get plus the value at the state okay so a value I mean a policy evaluation algorithm that uses this kind of an update for finding the value function okay is called a TD(0) algorithm right so you should be able to write a whole algorithm around this and be able to implement it and all that right so you start off with some arbitrary guess for the value function you have a fixed policy π you start off with some guess for the value function then you start generating trajectories right as you go along the trajectory you keep updating the value function using this update.

So we come to the end of an episode you go back reset the state but do not do you reset the value function okay reset the states and continue okay so that is essentially what is going to happen you could have been all kinds of tricks you could use you could have an optimistic initial value right you could do all kinds of different kinds of explorations on top of the basic policy right if you are going to use exploration on top of the basic policy what it should be careful about my god this is a chapter yeah this is chapter one question should you should you not if you know I am talking about valuation here I am not talking about evaluation here so I am not talking about control right.

In evaluation you should not I am trying to evaluate a fixed policy if you update over an expository action you are evaluating a different policy right lower excavate reactions you should not make any updates so that is basically TD(0) right so we have done one algorithm this is the basic policy evaluation algorithm right so people understand the concept of TD what we are doing here and looking at things that is one step away and we are trying to update things okay so let us see that is what when you are look when you're doing then the depth limited rollouts right in some sense you are doing something like this right.

(Refer Slide Time: 18:08)


Methods
Full Backups
Bootstrap
Backups
otstrap
Backups
MC
$E[G_t \mid S_t = s]$
G_t
"error"
$\hat{v}_{new}(s) = \hat{v}_{old}(s) + \alpha [G_t - \hat{v}_{old}(s)]$
current
"Target"
"current"
$G_t = R_{t+1} + \gamma R_{t+2} + \cdots$
$= R_{t+1} + \gamma G_{t+1}$
$\approx R_{t+1} + \gamma \hat{v}(s_{t+1})$
$\hat{v}_{new}(s_t) = \hat{v}_{old}(s_t) + \alpha [R_{t+1} + \gamma \hat{v}_{old}(s_{t+1}) - \hat{v}_{old}(s_t)]$
TD(0)
NPTEL

So when we are thinking about death limited rollouts is that right I doing depth limited rollers that is what you are doing I am counting the reward until I reach a certain point in time after that I am adding the value function so that is exactly what you are doing there right so you can think of this as just one step prologue not even so it is like it is take one step and immediately use the value function is one we are thinking about but I am doing only one sample and all those are there done that multiple times but here I am doing just one sample will come back and explore this question a little bit later right in the next chapter will explore this question where we will explain why the 0 is there.

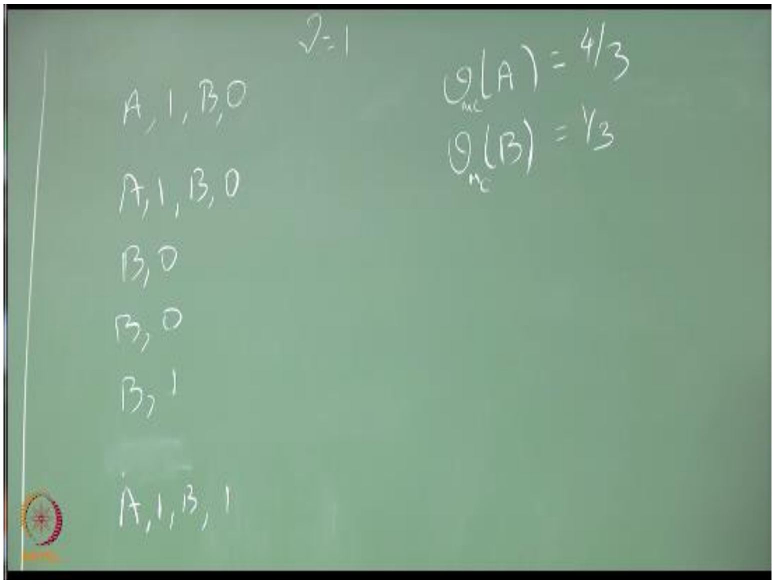
In is a family of algorithms called TD λ algorithms where we will actually look at this kind of having multiple step rewards and then having the value function case we will talk about that next time yeah well you did not ask me that question when we are doing Monte Carlo no do I am asked I am asked you to think about what we did in the Monte Carlo case right so we started in some state s right we generated an entire trajectory and we use that update this value so I'm instead of generating meant a trajectory and they substituting this GT with a biased sample but I am still measuring the return starting from s st okay.

Because I am not measuring the return from st + 1 tray tax could update the values st +1 but not using this expression I will not use st +2 here no how no why am I getting a estimate of we had a st +1 from V hat of st see v had of st +1 if I do not have any loops in my state space we have a v+1 will not even have v st in this expression only way we had a V st will enter into the expression for V out of st +1 if you come back to st right but I see the you know it is a time index random variable so we will never come back to st Kalman filter you're trying to predict what the next state is going to be right.

So here I am not doing that prediction right so if you want you can write a kalman filter like equation here to predict what the next state would be what I am trying to predict here what I am trying to learn here is the reward starting from the state I am not trying to predict the next state it is entirely possible for you to write the expression for st + 1 in terms of st but not V hat st + 1 in terms of Vhat st does that make sense no we talk we can talk off line right yeah they stopped I actually had some other public and talk about it later anyway so that is TD(0) right so any other questions on TD(0) okay good.

So I am going to talk a little bit about some time to make it give you a clearer as to what the difference between Monte Carlo and TD methods would give you right so we start off with a very simple data set right.

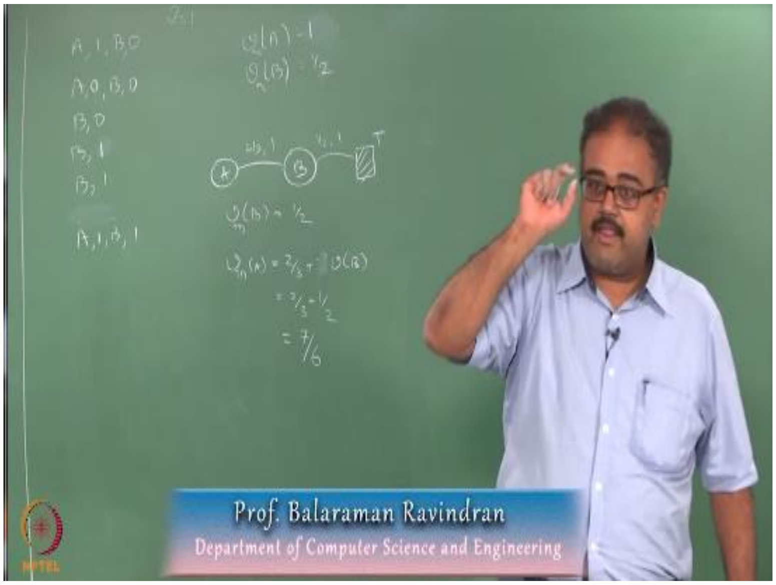
(Refer Slide Time: 22:23)



So I am going to give you as a terminal state after that right so is the sequence of states and rewards that you have observed let us assume the policy is very dumb policy okay because I am doing evaluation right the policy is fixed I do not have to worry about the actions so whatever is the policy right I just observe the sequence of states and rewards words right so what is the value of A what is the value of B if I use Monte Carlo where if you say let us say I use first to set Monte Carlo let us keep things simple I use first visit Monte Carlo so what is the value of A what is the value of B or let me make things even simpler.

Let us have no first visit every visit business so what is the value of A the last one is not critical for our example so well let us keep calm acquire 4/3 right what about B so here okay I will make your phases let us not have 7 case okay so this is essentially a Monte Carlo estimate they make a mistake here.

(Refer Slide Time: 25:13)



Hey sorry guys I made a mistake in the number I just cooked up some numbers suggest you realize now both Monte Carlo and TD will give you the same answers so I need to give you a set of numbers where Monte Carlo entity will give you different answers okay it is Monte Carlo kicked we have A is 1 + 2 / 3 so 1, B is 3/ 6 this 1/2 okay so what will be the value is estimated by TD so essentially what you have to do is you have to keep updating this rule repeatedly till you converge I am not that cruel okay what you can show which we lock through or what we can show is that TD has this amazing property that you are applying that rule incrementally but it computes the value function by implicitly constructing an a Monte Carlo process right.

So if you think about it every time I will go from A to B right so whenever I update the value of A I will use the value of B so in some sense of implicitly encoding that transition into my updation so it essentially it looked like something like this so A go To B and B will go to a terminal series we can see the right this is with probability 1 all of these transitions will happen right what is the reward forgoing from A to B okay the reward is 2/3 the probability is 1 right like what is expected of all right I am just removing so what is the work from going from B to the terminal state right.

So now the value function that the TD update will converge to is the value function of a and B in this MDP or in this Monte Carlo right so what is the value of B in this Monte Carlo shape that would not change okay here is the important interesting what is the value of a in this Markov shape it is the immediate reward + γ times value of B at γ is 1 7/ 6when people accept this right so what is the value of A is the immediate reward plus the value of the next state right so that is the bellman equation so the immediate reward is 2/3 the value of the next state turns out to be half in this case which we already computed is 7/6 okay.

So we came to a different answer than what recovery right Monte Carlo gave me one and half TD is giving me 7/6 and ½ which is the right answer so you see why TD tells me it is 7/6 see whenever A occurred before B right B is been very unrewarding right only one out of the three tens be gave me 1 but other times B occurred it has been very rewarding twice it gave me a 1 and once it gave me 0 okay so what TD methods assume implicitly is that the whole system is mark of so regardless of how I came to B when I go out from B I am going to get that expected reward of half right but it so happened.

That the few samples I drew from A turned out to be a bad sample for B right when I do the Monte Carlo estimates I am getting a lower value for A because of that I do not take into account the structure right the structure is from A I go to B and then I go to the terminal state and what TD methods assume is okay from be if you go to the terminal state this is the reward you are going to get and therefore from A if you go to B then you will get whatever reward will get from going from B to the terminal say it makes that very strong Markov an assumption right because it solves for this Markov chain okay.

So now let me go back and ask you which is the right answer thank you here finally somebody said it depends on the depends on the problem and it turns out that what Monte Carlo methods are doing is essentially trying to minimize the prediction error right between this particular set of samples that you have drawn the returns computer according to these samples alone right going back to the definition of written okay it is actually minimizing the squared error between the value function you get and the return that you get from this right is just trying to minimize the squared error from this fixed set of samples that you get right.

So what do TD methods do on the other hand is they form what are called a certainty equivalent model okay so people in control theory probably understand this better so there is something called the certainty equivalent model which essentially is the model best explains the data that you have seen so far under certain assumptions okay so under certain assumptions you can assume that this model is equivalent to the system that generated the data that you have seen so far and in this case we are making a very strong Markova an assumption.

Then what TD methods try to do is try to make the prediction according to the certainty equivalent model I trying to minimize the error between the value function in the certainty equivalent model and the predictions you are making here right so they are optimizing very different objective functions given an infinite amount of data drawn from a truly Markova an system it both will converge to the same answer but given a finite amount of data right this will converge to the minimize of the squared error this will converge to whatever is the Markov model you can form with that limited data okay.

This need not be the right model this need not be the model that gave rise to the data right given the limited data this is the best you can do that is all you can see we are saying here given this data what is the model you can form it will converge to that right whether this is the right answer or that is the right answer given a limited amount of data depends on what kind of application you are looking at all the other hand since the Monte Carlo methods make a less of an assumption about Markov witness if you suspect that your system is not very markup okay then do not use theory methods you are better off probably better of you see Monte Carlo kind of methods okay.

But remember I told you that if you use the notion of a value function already or making an assumption about Mark of Venus right as you should be using values of history's right you should not be using values of states if you are already using values of states you have making a Markova assumption but that is okay right in some sense provided the is not estimates you are making since then right you take care of the Markova awareness non- Markova I think still it is not correct but at least we will be more robust to violations of the mark of assumptions and as you can see very clear from here right just going by the evidence of the trajectories I have given you should not think this is a marker system right so whenever A happens B seems to have a depressed outcome right display the evidence of the data here you should not think it is a Markova system but TD will just assume it is a Markova system and go ahead and do the updates okay so you see the difference between what TD does and Monte Carlo does okay we will stop here we will worry about TD control the next class I am still converts to the thus the least square estimate of the returns predicted by the data.

IIT Madras Production

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

www.nptel.ac.in