NPTEL

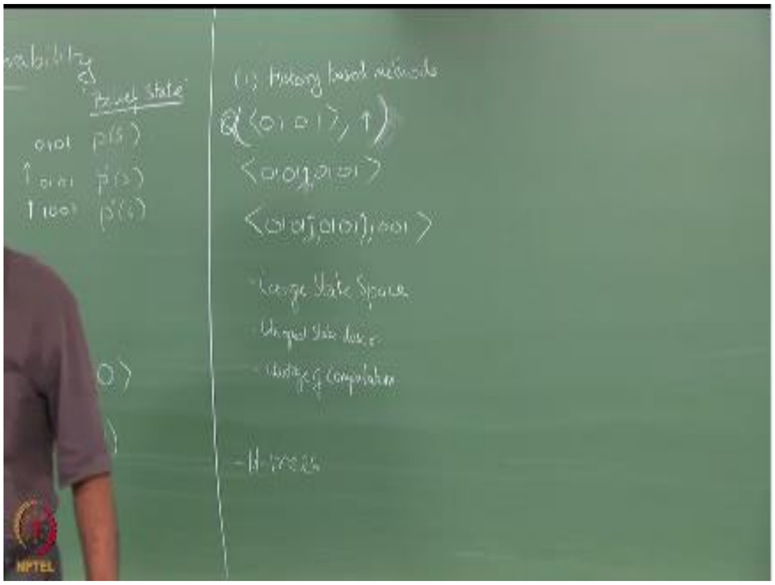
NPTEL ONLINE COURSE

REINFORCEMENT LEARNING

Solving POMDP

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

(Refer Slide Time: 00:16)



So I will talk about different ways of thinking about solving this problem right, the first one is actually something that does not use belief states I pretty straightforward pretty very nice way of solving this is basically so I am going to call the mystery based methods, but almost all the partial observable solution methods use history in some fashion or whether but I call these history based methods because they he use history very directly so what do they do and then keep a track do what with it.

No, I told you read this first method does not use belief how else can you use the observations directly. We talked about it we talked about in the options case when I said we talked about semi Markov options what did I say look at the history since the option started or something like that exactly the same thing you look at history forever right, so your state space will start looking really funny your state space to look like the first state is 0101, okay.

So the next state is next state is for this right so the most recent state will be the last entry here, right so and then you go back in history you remember the whole history like this, right so this is my state. So if I am learning a q function I learned q of, good point if you want it in this action as well, so the history can contain just the sequence observations you may or it could also have the action if you want you can put the action itself that is a good point so you could watch possibly do this, nice so you can also have the history as we have been action as part of the history.

What do what is the, who you gather out of actions having actions what do you get. So the thing is so 0101 and then 1001 and I keep going up right, so that is one thing if it go up i have 0101 when if I go down I have 0101 that is a completely different interpretation if I did two ups and I still get 0101 it is very high probability that 0101 is a noisy measurement say go up I get 0101 and then I go down I get 0101 that is probably the right thing it probably went from here to here and then it came back here.

But if it gone from here to here to here and I still get the same 0101 there is something wrong right, I could have done either this measurement was wrong or this measurement was wrong or could be something even inverse my sensors could be stuck at 0101 hits I could have been anywhere in there anywhere in the world, okay so knowing the actions helps in the same equate those kinds of things.

So going I just keep doing up down up, down up down and I keep seeing 010101 that is fine, right if I keep doing up, up, up, up, up and I keep saying 0101 0 and there is something wrong it is good to have that means that is a good question what do you think why you would not repeat I will need to talk about episodic MVPs and things like that right, so next time you start an episode we will start from the beginning right.

It is like keeping track up a game from when you started playing the game, but you have a point I mean why do you want to repeat but so people I see how this is this can be useful right what are the problems with this state space is very large right, so large state space yeah that is why it is large state space here so step here, so this episode is large it with the length becomes large anything else. If you are thinking of doing parameterized representations and things like that your states have different enough links not only the state space large.

But each description is of unequal links right, anything else my competition on stage that have quite low. I will come to that in a minute that is the wastage of computations but states of low probability I mean that is the problem we have with MDP is also right, I mean if you do not visit them often enough you are not going to waste computation resources on them, right. Sorry you could maybe I am not sure.

I mean given the way the field is moving outside you would want to use an LS steam and not a HMM but you could think of sequence models for doing this history modeling that is exactly what I am suggesting to you guys to use talk to him later, right. So one thing that you want to think about so when you think this will work is it a good idea will it always work. What is it that we require of a state in MTP think of the math hold that you say given what is enough we have a specific definition for a Markova in this way what is the definition.

I see all was missing a point I see an important point so they said it nominate you as a TA is it, missing an important point probability of St+1 given St, At is equal to exactly, so sufficient enough will determine the next state all of that is fine, but only to the extent of what you could have done with the entire history that is all I am asking for right I do not want the current state to have more information than I could have had way remembering the entire history that is exactly what we are doing here.

If you can remember the entire history you are essentially in an MTP right, so you cannot ask for anything more is it fine, is it correct I am just remembering history of the part is the history of the observations right, so here that is a very nice question about the broken vision system in the book I do not think we did we are say anything think about that now, right. I remember the history from the beginning and the history is 01010101010 is it enough, I mean I shall be anywhere else in the world, right.

So I do not get the state information but say because there is a significant level of noise introduced in the observations, correct so assuming that remembering history is good enough so it is assuming that I know where I was to start out with and remembering history is good enough then this should be completely sufficient but sometimes it is not, okay. Suppose my observations is sits that, right I set up a kth order Markov system what is a kth order Markov system it depends only on the k last state, right.

The normally what we talked about Marco is first order, right so kth order of the Markov is it depends on the K previous stage, right. So if I, if my observations is sits that I have a Kth order Markov system so for the last k observations k can be however large you want can be arbitrarily large right, but if I fulfilled that exists a k such that if you remember everything it is Markov then all of this will work okay, that need not necessarily be the case as I just told you in the like in the broken vision system right, but then there is a question that I asked you is a broken vision system Markov or not, right. In the textbook and the answer was yes, because there is no underlying state space, right.

So we just assume that that is all the status right so we did not assume that there was an underlying space observations was the state, right and therefore I can predict the next observation very clearly right, it is going to be 0101the correct, in the broken vision system in the broken vision system I can always predict what the next observation will be it will be a completely black screen, right.

I do not need history because it is broken I know that it will not change therefore it is Markov what is somewhere yeah, that is a different issue that is a different there is a, there is not a, its starts out being broken yeah, yeah but we did not put that into the problem statement Edwin it is broken its shape broken. What has to be in model MDP. Yeah look so they are not there, they are not mentioned means is not there if you making, you always know that rule, right.

If you are making any assumptions about the problem write the assumptions clearly and then write the answers, so if you are not making any assumptions then but anyway so getting back to this so if your problem is k^{th} order Markova where history is good, right having history is actually useful thing to have right. So but we still have the problem of unequal state distributed straight line, right how can we fix that, justice to the state representation in unequal.

Many ways of fixing it for example, if you use LSK you will actually get a single I mean fixed length representation and all that, but one simple way to do it is to fix your history we are going to say that okay, k^{th} order Markov I am going to look at the last case states and then if I have only seen three states so far to fill it with them, that should be like a no-op kind of a symbol, right that is a maybe 1111 basically I am surrounded all sides by obstacles that can never happen right.

So I can just put 1111 as my symbol right, where I can move right and also point, since this can still report this shade yeah, sure, great. Then pick another thing say -1,-1,-1,-1 or whatever, right so pink some sentinel kind of thing that we know for sure it is not a state, right so then what would happen is suppose my history is three right, this guy will be reported as 0101, -1,-1,-1,-1, -1 and this scale we report our 01 010101, -1,-1,-1,1-1 right, this is full state after this everything will be fine.

So the x there will be a small fraction of the history a small fraction in the beginning which will have this weird -1s but after that everything will be fine okay, that is good. And so we also saw the last start space problems I mean large state space problem right, by using this and what about where is the wastage of computation coming I am are talking about wastage of computation it still is there . That is one part of it anything else.

For example let us go back to the noise free scenario, if I have a trajectory of 10 states right, and let us say state number 10 is where I am now and state number 9 my observation was 1001, so do i need to remember 1 to 8 right, because 1001 is absolutely localizing right, so I know I said I see 1001 I know I do not need the history before that so as soon as I am able to localize myself perfectly the history before that becomes 10 to 11 right.
So localizing yourself perfectly just one example that could be other cases where some places you need a history of length 3 some places you need a history of length 2, some places you need a history of length 7 and so on and so forth. It might not be uniformly k order Markova throughout right, but once you decide that it is going to be kth order Markova then you are stuck with keeping a k window history around and since you are treating this gay window history as your state you are unnecessarily relearning the same thing over and over again so that might be the first five states might be irrelevant.

But every time I visit the different set of states in the beginning I will treat it as a different state a different, a different observations state, right so this thing is wastage of computation that is what I meant, right. So if you can fix it that will be great right, there is also another problem with using kth order Markov why, what is the problem determining the k so you might actually get the k wrong in which case either you will be doing wasteful computation.

Because I have a two larger k or you will be missing out on important information because you chose a too small a k okay, so these are all the problems with these history based methods but history based methods are very easy to use, right you do not have to worry about too many complicated things for example you do not have to worry about a belief state, right we already saw updating the belief state is a very complex operation, right.

So we needed to worry about how would you do this updation things like that we will talk about that depending on how far I make it today, right so we will talk about that and okay, so it is just a lot easier to implement intuitively easy to understand and if your MDP happens to have a small k this is kth order model with a small k right, these are pretty good that is all about it can go ahead and do that.

So there is one method which I am not going to go into in detail to encourage you to read no, I will not ask questions on this in the exam it is called the U trees, right the U trees are proposed by Andrew McCullum when was this long time ago okay, I forget 92 maybe yeah, so what he does in U trees it is U tree stands for utile trees so it basically he only makes distinctions in the history, so he treats all the states has just one thing to begin with, right so just a single state your problem does not have multiple states it just as a single states.

And then as you make more observations and you get more rewards he only makes distinctions okay, that allows you to make better predictions of the rewards you are going to get, okay. So I will not worry about everything else I only worry about the first bit I see okay, so if the first bit is 0 or 1 should I make a distinction based on that right, so I split it into two states in one state the first bit is 0, in other state the first bit is one, right so does it make sense does it give me better prediction if it does okay, great.

Now should I look at the second bit now or should I look at the first bit one step ago alright, so there are multiple things I can I can look at more bits in the current state itself what I can start looking at history, right okay I'm mixing up two things I'm mixing up a little later work so what Andrew did was the following do I look at one step down, okay so what is the first stage so last eight was 1001, right I don't need to look further back in history, I have done because any further back in history a look I am not able to make any distinction between the predictions, because as soon as I know that the last date was 1001, I know exactly what will be the outcome of each of those actions.

So I do not need any further history, but then I look back and see that the last eight was 1010 and now I need to look further one more step back, right so at look at 1010 that look one more step back if it is 1001 I am done, right if I look one more step back and it is also 1010 and then maybe I have to look back one more step, right or look for work so I had looked back one more step right.

So like that so we'll have to look at what the history is see what is it that you observe and based on that you have to look further and further back in the history, right so this is essentially what he what he did so he didn't say that I always you say history of length 3 are always you selection of length 4 what length history I use depends on what are the actual elements that go into the history, I didn't tell you that you trees was data efficient on memory efficient it’s not it's in fact it requires a horrendously large amount of memory, infect yew trees had it was originally proposed by Miquelon, is also very wasteful of data, right.

He wanted to give some guarantees about, statistical independence and so on so forth so what he essentially did was whenever he makes a split right he kind, of throws away all the data, and then he starts all over again okay so that's pretty wasteful use of data right, so it's not I’m not using the same trajectory to determine multiple splits that gives you independence in the tests right but also wasteful of data so we need to come up with other mechanism for the uteruses amazingly powerful but he actually has solved, nontrivial problems which the other methods for partial observability would struggle with in this because given the kind of state spaces he's looking at right.

But he does to do a very good job with that so that’s basically on history based methods so the second class of approaches, which I would call this is the QMDP like approach I can't think of appropriate term for it but QMDP is a prototypical example of such kinds of models is that you assume that the MDP is available too yeah you assume that you know the MDP okay since you know the MDP what you do is go ahead and solve it, you solve the MDP get a policy then what’s the big deal, you assume that you know MDP is known.

So you have this form DP model right I assume that may SAPR is known, I'll solve the pond you can solve do value iteration polish iteration whatever you want solve it I get my policy π, know what haven, exactly so execution becomes a problem, right so I can solve execution is still a problem so can you think of ways are fixing that, yes so I have the form DP I have to have the form DP I have let us say I have the form DP.

So I solve the MDP component of it now when they have to execute what do I do, because I learnt the policy on the stage how can I do it all of this ok, even if I want to do it all observations, oh shit so we are done I says we have the end is a form DP that take the MDP component of the

form DP I solve it, so I have a policy defend on the states, right in the history based methods we are assuming you only have these observations as states, right the QMDP as we are assuming you are given up form DP you how to solve it right.
So we solve it we find the policy because they stake the MDP component out and they find the policy now, how do I execute it lets say found the Q function that's why it's a QMD people have somehow iron-shod saw it on Q-learning or Iran value iteration on the Q function Iran policy traction with Q function whatever it is I have the Q* for the MDP exactly but I have 0 believe somebody said believe exactly.

So what do you do with the brief policy is a like step to action probability not huh she always seemed to p/s of this priority of that whether you believe you big Bell of s good policy of a and then we use that probability distribution yeah QMDP is a Q function, use the Q function so essentially what you do is you is the some kind of score for right and then action that you pick would be so you are in the current belief state that is all or you can think of this as so the current belief state,

So what is the value of that for taking action A the current belief state can you believes hit the valley of taking action is essentially given by the sum over all states, beliefs take belief of being in the state into QSA and then when you pick an action you basically do this, way so now I have solved the q and converted that into a value function over believe states come actions so we look at the other popular method of solving partially observable MDP, right.

Unfortunately all the methods are out there except, history based methods assume that you know the form DP right or you take a Beijing approach to solving this where you start off with some distribution over all parameters of the form DP and then you try to solve using that but in whatever it is you have some kind of models, that you are using right.

So given that you are anyway having a model anyway the assumption you make is that you know the model way of solving for form DP thesis a very easy way of solving it we just solve MDP first and do this what is the problem with this method though, you have to maintain believe over all states for all belief state methods, so that's the only non belief state thing everything else we have we will talk about will be made any states if you clown we need the MDP rule assess what I’m saying right.

If you don't have MDP don’t know anything about it that's your only option is over the states and then were doing so when you are executing only one policy yeah but you're getting somewhere close, huh following the very functional I have never knew valuation new policy take whatever you have our MDP you know how to solve it, yeah that is there for any belief state methods, since we solve the problem assuming that you would have access to the state at the point.

And now you are only doing it heuristic for me converting that into an execution policy, the policy that we execute might not be optimal for the community if you solve the form DP directly then you can find something that is optimal for that form DP, right given now we are the solution you have discovered is assuming that you would know these state at all points right. So given the uncertainty in the states, that might have been a better action to pick in terms of the total reward that you get made if I had for example right here is a very dumb way of thinking about it is actually not too dumb, later as you will see.

I can take my form DP, right I have this way of computing the belief right so we will come to that we but we you have a gist of how to compute the belief now I can define an MDP Prather, my states or the belief states right, on the transitions between the belief states, is given by the computation that I did for finding the belief escapes rightly that that function will give me the transition probabilities right so given that thisis the initial belief state and this is the observation that you made right.

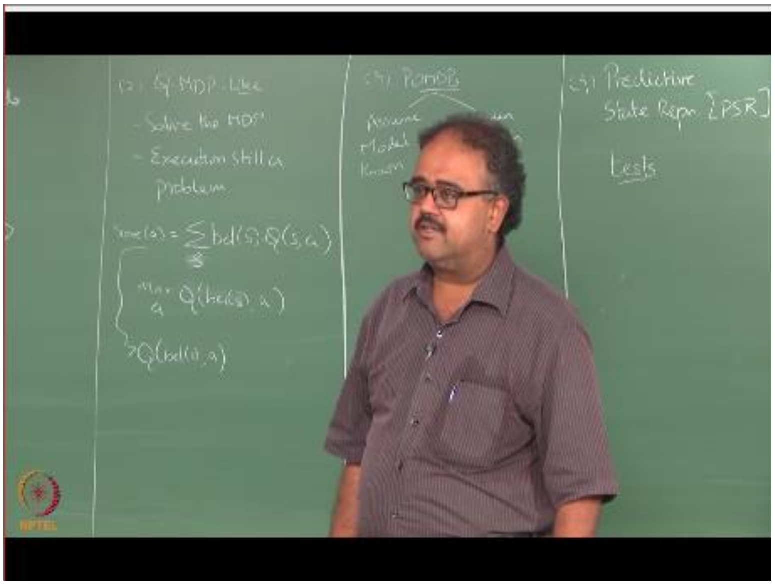
So given this action you took or something what you in the next belif tale in fact that will become actually almost deterministic because it says it’s actually a computation that will determine what the next state is so I will have a belief state i will do some computation will get to the next belief state and now it's an MDP state. I know the previous state right and then they know the previous belief state my current action will determine what my next will escape really so current if I know the previous beliefs safe come on the observation, that if I know the current action I’ll know what the next up next belief state, will be right.

So now it becomes an MDP I can solve it and what’s the problem is solving that MDP I have very high-dimensional, continuous state space, that if you original stay originally I had like a million states now my state to sell million dimensional simple x hey so it's like a crazy state space, and I have to learn value functions on top of that, right so it becomes a little faithful so what we will see in the I'm going to defer it to the next class I don't have actually energy to teach the whole of form DP today.

So what we will see in the next class is how to solve the belief state MDP in in a clever fashion, by making use of certain kinds of regularities, that will be available in the value function of belief state MDP and degrees the very fact that the belief stats are simplex right and the belief the is a simple and then the value functions have to be related in the certain way.

So we use those relationships and try to come up with the efficient way of solving form DP directly where you are essentially learning in action from history, and running a policy from history to actions, so that's essentially what you will do, when we solve from DP so such solutions where you are actually cognizant, of the entire history of what has happened right and then making choices and knowing of knowing that there is a certain uncertainty in where you are and being cognizant of that and making your action choices certainly yields better policies then something and Huck like this, okay what is nice and what the ad hoc thing easy, right it's much easier to implement and, turns out that in practice it actually yields good policies, they too bad.

(Refer Slide Time: 34: 32)



So people are tried in most real problems that you encounter, right so things are not so pathological that this will perform very badly, many real problem domains at the encounter QMDP is a seem to give you good, in a solution so people actually like to use this only problem is you can’t do anything, right so the third class of methods are directly solving directly solving from DP, so it requires setting up a lot of machinery for that so we'll come to that later but there are two, classes of algorithms here.

So one assume that assume the model other one is a other one is like a Bayesian approach where you do not assume the model right but you assume some kind of distribution over all possible models, right and then as you gather more data you try to reduce that distribution also but at every point of time you can actually solve a solver form DP even though you do not know the true model right you can solve some assumed model and you start to generate some kind of experience, and use that experience to refine your probability distribution over the models, okay.

So it would assume the model and solve it assume that you know the model and solve it or you can take a base in the price of course they can take a non Bayesian approach also when the model is not known right you can just do some kind of a frequents approach just count the number of times some things that happened and so on so, forth but it’s more fun so the last class of algorithms which I will not cover a lot right but there is a video for you guys to look at can you ask them to link the PSR videos something called predictive state representations the kind of turned the whole idea of history based methods on its head so what did they do they said why am I bothered about history.

So what do I need the history for, so they can make better predictions about the future right so instead of worrying about history, can I just look at how well can you predict the future right so what are the things that we need to know about the future right so that I can predict everything that can happen in the future, so the basic assumption here is if you make certain assumptions about the dynamics of the world right I can do the following I can say that if I knew what is the probability, of let us say if I go forward two steps I will get a wall from wherever I am if you go forward two steps I will get a wall if I go to the left five steps I wall, get a wall if I go to the right two steps and go up one step I wall get a wall.

So if I know all of these like what is the probability of these four things happening then I know where I am in the world, kind of make sense read instead of trying to know about the history if i can make predictions about where I will end up at if I if I do some small policy fragment right and I will make, certain observations along they right so essentially if I say after three steps and hit a wall that essentially means one step no wall two-step no wall three-step wall right.

So it is a sequence of action observations that you decide that you have to decide based on the problem dynamics, so they have mechanisms for coming up with the tests themselves right so these things are called tests so what is the probability of North clear North clear north wall okay that is one thing right there is one sequence so action North, observation clear action North observation clear action North observation wall that is one sequence likewise action west observation wall that is one sequence then action East no wall action is now all action North one wall right that is another sequence, these kinds of observation action sequences are called tests, what they did was they said that predictive state representation consists of a set of tests right and the probability is that those tests will be true, right.

So if I have this collection of tests and their probabilities that will determine what a state is for me is why it's called predictive state representation, right so what is a nice thing about it is they do not assume that there is an underlying MDP like the form DP does right they don't ever want to look at the MDP it's like saying that okay i do not need all the history sequences, I will need only some of the histories and that is enough to represent the world for me right.

So in predictive state representations we do not assume that there is actually an underlying state space we only have these tests, and we say this tests are adequate for me to make any decisions I want in the one that might be a true underlying state we do not know, right but that's a very power everything about PSR, if you build PSRs on top of form DP then you can how that they are as powerful as, the direct form DP solution methods the guild the same solutions and so on so forth,. but the nice thing is PSR can go way beyond the regular form DP setting because they do not have to have any notion of what the underlying state spaces they can have they do not need a state space see the video.

So this was proposed by sather shing and his group and the video is by sather shing so getting it from the horse's mouth, so something that came down here for the reinforcement learning workshop, so I asked you to do a session on PS arts he did so that makes it that means that I don't have to ever teach PS abs again that's a cool idea, unfortunately it is hard to learn the PSR wait that's it so it's the bass lessons easy just look back so many time steps and you just remember whatever you wanted to learn but learning the PSR turns out to be a little tricky you can learn linear PSR right.

So what I mean by linear PSR linear PSR or a set of tests, from which I can recover the probability of anything else being true, I suppose I said I gave you three examples right so north, north, north and then east and then west, west north or whatever right I gave you three examples suppose using these three things I can give you the probability of west clear west clear west wall, by using a linear combination of these probabilities then they are called linear PSR right and linear PSR will necessarily be large, there will be if there is an underlying form DP if there is an underlying MDP on which this PSR is defined, the number of linear PSR linear tests, that you need right will be the same as the number of states, it is test that you need will be the same as the number of states, in a linear PSR.

But if you are able to define nonlinear relations that can maybe products and divide one by the other and those kinds of things right you can even get a representation which is smaller than the number of states, right but then finding non-linear PAR is incredibly hard we only have like existence she'll proofs but actually finding these things are very hard, but so that this is whole literature on predictive state representation, is very interesting.

If that is interest next semester if people are around we could think of looking at PSRs X there is interest but I am more personal, inclined to look at more of the deep RL stuff but we will see so we but I am not going to cover this in more detail so if you want we should look at the lectures so next class what is left is I am going to talk about form DP right so we will just do that in more detail and that will be the end of it.

IIT Madras Production

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

www.nptel.ac.in