LECTURE 2

Markov Decision Processes, Part I

Recap

Last lecture, we discussed the Agent-Environment Framework, as well as introduced the Reinforcement Learning Problem.

To review, an agent is what is taking actions within an environment, which responds to the agent's actions by providing a reward (or no reward). The goal of RL is to tell an agent the optimal actions to take that maximize reward.

But let's focus on the concept of environment. While the environment may seem abstract by nature, we're now going to talk about how to formally describe an environment for reinforcement learning. This is achieved through Markov Decision Processes. For now, we are going to assume that the environment is fully observable, meaning we know everything about the environment such as all the states the environment could be in. Later on in the course, we'll see how we deal with scenarios where we do not have all the details of the environment.

Markov

Markov Property. The future relies only on the current state.
The central idea is that we have some state, S, that explains the current state of the environment. The state has all the relevant information about the past. So as we move forward, if we are keeping track of the current state, we do not have to keep track of the past anymore.

What this means is that the current state, S(t) describes all that is needed from previous states S(1) up until itself. the probability of going to the next state is only determined by our current state

Example: Walking on two feet. As I'm walking, I don't need to know whether I was standing on my left foot or right foot 20 seconds ago. All that matters is that at the very last step, I moved my left foot, so I should move my right foot next.

Why do we care about this? Because it allows us to move between states efficiently, and not have to look very far back into the past for information.

NOTE: There is no concept of behavior yet: the environment is moving you. The probabilities determine where you go.

Transitions

In an environment, we move between different states. You are more likely to visit certain states over others depending on where you currently are. For example, if you are on Facebook during my lecture, you are much more likely to continue on Facebook (its very addicting) than to listen to me talk.

So for the current state S, there is a probability associated with every other state S', which signifies the chance that you will end up at new state S' given youare currently in state S. So for the Facebook example, let's say we have two states you can go to from being on Facebook: paying attention or continuing on Facebook. The probability of continuing on Facebook could be .9, while the probability of paying attention again is .1.

We arrange these probabilities in a matrix, called a transition matrix. Each row tells us the probabilities of going to every other state from state 1, for example. Some states may have a 0 probability, meaning you can't go to that state from your current state.

Markov Chain

Markov Chain. Now that we have our transition matrix we can describe a Markov Chain: a sequence of states with the Markov Property

There are two pieces to a Markov Chain: a finite set of states and the transition probability matrix.

Example (Student Markov Chain): There are three classes you have to sit into in order to pass the class. But there are distractions that may prevent you from going to class and listening. The numbers near the arrows signify probabilities, which would be included in the probability transition matrix.

Episodes. Episodes are possible sequences of states through a Markov Chain.

Markov Reward Process

Now, lets throw rewards into our Markov Chain, which now makes it a Markov Reward Process. So now, we also have a value judgment telling us how good is it to be in a particular state.

So before with the Markov Chain, we only had the states and probability transition matrix. With a Markov Reward Process, these two components are the same, but we also add in a reward function that gives a reward for being in a particular state, as well as a discount factor, gamma, which I will touch on in a bit. The reward function depends on the state you are currently in. This reward is immediate. And as we've

discussed since the first lecture, we care about maximizing cumulative reward, the sum of all the rewards by the end of our episode.

Student Example:
So for our student example, although we may get negative immediate rewards for being in class (maybe you think it's boring and would rather be asleep), in the end you can end up with a +10 reward which has a much larger cumulative reward, making up for the negative immediate rewards of being in class.

So we've been talking a lot about the concept of maximizing cumulative reward, but how exactly do we compute the cumulative reward? Through addition!

Return. Return is the cumulative discounted reward from time step t.

I haven't touched on discount yet, but I will in the next section. The takeaway from here is that we are simply adding the rewards from the time steps after the current time. So Rt+1 is the reward you get at time step t+1.

Discounts. A real number from 0 to 1. It is a parameter we use in the value function to tell us how much I care now about rewards I'll get in the future.

We're introducing a preference on when our award is coming: Do we prefer recent rewards or do we not care? The closer gamma is to 0, the more we prefer recent or immediate rewards. (ex: money given to me now is sometimes more important than money I get in the future, maybe because of interest). If gamma were to actually be 0, we would ONLY care about immediate rewards in our return. Gamma value closer to 1 means you want to include rewards that are longterm.

Why Discount?

There are many reasons to discount:

- There is more uncertainty in the future (we usually do not have a perfect model of the environment, and due to the uncertainty of the future, we can weigh recent rewards more heavily) But if you really trust your model in assessing the future, let's say you think you'll be getting a pot of gold sometime far in the future, you can discount closer to 1.
- It's more mathematically convenient: avoids infinite returns in cyclic Markov processes. Makes sure you don't end up having infinity as a reward

So now we understand the concept of return and why use discounting in our return.

When we think of reward, we think of an immediate signal we get from taking an action or being in a state. And return is the cumulative reward after a certain time.

But how can we evaluate how good it is to be in a state, in the long term? This is what the value function tells us:

The value function gives us the long-term value of state s. The value function tells us, if we are in state s, what is the total reward we will receive onwards? So how does that relate to return? Return has no concept of state in our definition before: its just the cumulative reward after a certain time. We can define the value function as the expected return (aka expected cumulative reward) starting at state s.

If I drop you into a MRP at some state, lets say Class 2, what is the expected return from that state onwards until you reach the end or terminal state? That is what the value function tells us.

The expectation is there because the environment is stochastic.