SEI Cyber Talk (Episode 2)

Learning by Observing via Inverse Reinforcement Learning
by Ritwik Gupta and Eric Heim

Ritwik Gupta: Hey everyone. I'm Ritwik Gupta. I work here at the Emerging Technology Center.

Eric Heim: And I'm Eric Heim, also at the Emerging Technology Center.

Ritwik Gupta: And we're here to talk to you today about inverse reinforcement learning. Eric, I heard you've been working on some really, really cool stuff that basically lets agents learn from their environment. Can you tell me a little bit more about that?

Eric Heim: Yeah. So inverse reinforcement learning is a particular formalization of a problem known as imitation learning. Imitation learning is the task of learning from observed behavior. So imagine you wanted to learn how to drive a car. One of the ways to do that is to actually watch somebody actually drive a car. So you could actually watch-- have someone drive a car, see which turns they make to certain destinations. You could see things like the phenomenon like they don't run into poles, they decide to stay in these lanes, and certain things.

Now, inverse reinforcement learning sort of differentiates itself from the broader scope of imitation learning in that it models it in a very specific way. So specifically-- so imitation learning could be-- it could just be what's known as behavioral cloning. So the idea is that you could just take-- anytime the human or thing you're observing takes an action, you take the same exact action and you're just modeling exactly the behavior. Well, the thing of IRL is that they try to model the reward structure behind the decisions they make. So you can imagine when a human drives up to an intersection, they decide-- they have some intrinsic reward associated with actions in that particular state they're in, which is the intersection. So they would say, "Well, a more rewarding action than running into a pole is making the correct right turn to my destination." And this formalism is really useful for a couple reasons. One, it allows you to generalize tasks. So if I'm just copying a behavior of an agent, or some observations, and I'm using that for an agent to do the task, if I'm put in a situation which I haven't directly seen before, it may not know what to do. But if I learn sort of the intrinsic reward in the relationship between the actions and states that the human found themselves, or where the observation behavior found themselves in, then I can generalize to other states and actions. So I can do things like, "Hey, I'm in a similar situation than I was before, and because I know that, I know my reward structure loosely and make decisions that way."

Ritwik Gupta: Gotcha. So I guess in my world a little bit-- let's say in an ideal world, if I just watch Roger Federer play tennis for like 1800 matches, are you saying that I-- assuming I was using inverse reinforcement learning-- I could learn what things Roger Federer prioritizes when the environment is in a certain state and the certain actions that he can take in that specific state.

Eric Heim: Yeah, absolutely. And so I think that there's a lot of interesting questions there. I think what you could learn, and specifically in the formalisms that IRL provides-- IRL being inverse reinforcement learning-- you could learn that-- given the way you model the problem, if you say, "If Roger Federer is in this position on the court, the ball is coming at him at this angle, it's at this position," you could say, "He prefers going to his right and doing a forehand as opposed to going to his left and doing a backhand," because potentially the ball is going this way, right? And so this is useful for a couple reasons, actually. One of them is-- so if I wanted to have a tennis-playing robot, I could then use the formalisms to have them mimic the same actions. Again, the idea of IRL is that it could generalize to a lot of different states and actions. So for instance, if we just saw him do forehands and a couple backhands, maybe then for certain situations where he needs to do a backhand that it hasn't seen before, this portability of the reward that we learned implication could be useful.

Now, the other way that-- so this is what's known as inverse optimal control. I want to do the optimal actions for this control task of this robot. Now, I think that the other reasons-- this is sort of less reflected in the literature for inverse reinforcement learning-- is that there's a lot of this-- there's something to be said for understanding the model that experts have for certain behaviors. So if I wanted to say, "What are the interesting decisions that Roger Federer made as opposed to someone else?", what you can do is compare the rewards they assign to certain actions in certain states to potentially other choices they could have made, and this is useful for learning how experts perform tasks and then teaching people potentially. So if someone who's a novice in tennis goes, "Man, I'm in this situation. What would Roger Federer do?", that could be useful for that. In addition, things like-- so certain phenomena in biology that are not well understood-- if we wanted to get a lot of observation of them happening, then we can say, "We can probe this reward function." Say things like, "Hey, what are the most important actions or states that led to this behavior?" So we can observe those.

Ritwik Gupta: This sounds really interesting, because it seems to me-- and correct me if I'm wrong-- that basically by just watching something happen, or just observing something happen, we can kind of understand why they're happening by making no assumption in the world itself besides, "These are the possible states and these are the possible actions you can take in that space."

Eric Heim: Yeah. And the thing that's sort of beneficial about inverse reinforcement learning as opposed to another reason is that it fits into this idea of-- so inverse reinforcement learning is actually an extension-- or not an extension-- maybe uses the same formalisms as reinforcement learning. So I think that there's a lot of benefit in using that. So while it is sort of this-- well, I would say imitation learning is very general. Reinforcement learning poses a particular structure that's useful for practical learning tasks.

Ritwik Gupta: So actually let's dig into that a little bit. We've heard a lot of stuff about inverse reinforcement learning. We've also heard a lot of stuff about imitation learning from you. What's the difference? Why is there a distinction between this general class of imitation learning, which I guess is what inverse reinforcement learning attempts to do, but what's the difference between the general class and specific class of algorithms?

Eric Heim: Sure. So I think it's important to talk about that, for inverse reinforcement learning, these sort of formalisms that it assumes, it's useful to talk about reinforcement learning, or sometimes known as forward reinforcement learning, and the idea there is that you want to take these particular formalisms of states and actions of what an agent can find them in-- so this could be a human agent or an intelligent agent or a software agent or a robot or anything-- and learn an optimal policy.

Ritwik Gupta: Hold up. What does a policy mean?

Eric Heim: So a policy is a mapping of state and actions to probabilities. Or, conversely, you can think of it as, "If I'm in this state, what is the action?" I can score an action.

Ritwik Gupta: Right. So basically if I'm at a red light, what's the chances I'm going to go forward or turn left, right?

Eric Heim: Right. And so in the driving example, since you brought it up, the states that the agent can find themselves in is positions on the road, or potentially, if you want to model it in such a way, the speed they're going, and the position of the lanes, maybe even the type of road they're on or the type of intersection they're approaching, the signs, that sort of thing-- the same sensory input you would get if you were driving a car. "This is how I observe the world." Those are the states.

The actions are the valid actions that can take me from one state to the next. So if I'm in an intersection, I choose the action Turn Right, if that's the granularity in which you want to define actions. You could then say, "It takes me over to this road to the right," and that's the particular formalism that they use. Now, the formalism goes-- it varies between different people how they formalize it, but there's also the idea of, "If I take a right," that's a deterministic-- what they note as a Markov decision process-- that means I know that I'll go exactly on this road, but if you assume the dynamics are not fixed or deterministic, you could assume that the state and action doesn't completely capture what the next state will be. So things like a car could potentially cut you off or something, and you'd have to go around or do something like that.

And the reason why this is useful-- and I sort of went off on a little bit of a tangent-- is that the forward reinforcement learning problem is learning this policy from observed states and actions. So you have an agent drive around, and after doing this and receiving some reward back that,

"Ooh, it was good that I took a right because I didn't hit anybody, and eventually I went to my goal," the inverse problem is, instead of having the agent do stuff in the environment-- observing states and actions and getting a reward back-- the inverse problem is learning the reward from watching an expert take some states and actions in the space. So these trajectories of an agent, of an expert driving, is-- there's no observed reward in those cases, but you could learn their intrinsic reward that they guided them to those policies.

Ritwik Gupta: That's actually really interesting, because what that tells me is that this intrinsic reward, it's dynamic and it can change depending on what's going on in the environment.

Eric Heim: Yeah.

Ritwik Gupta: So it actually-- continuing the driving example-- sounds like perfect for self-driving, right?

Eric Heim: Yeah. So there's a lot of issues with the practical applications of these methods. One of them is-- so there's a lot of assumptions in how you model the world, right? So given all the sensors you have on a car potentially for self-driving, there's a lot of things that aren't captured in those sensors potentially, and this is sort of the dynamics that's not represented in state. So I think there's a lot of engineering involved in defining a proper state space, action space, and assumptions about transitions and stuff like that. The other issue is if you want to do forward reinforcement learning, what ends up happening is you that you need-- for really big state and action spaces, meaning that you can be in a lot of different states and take a lot of different actions, you have to have a lot of-- you have to let the agent take a lot of actions and a lot of states to actually learn an actual policy. Inverse reinforcement learning allows you to alleviate some of that burden by watching an expert. But again, it still runs into the same problem of you often need a lot of observations to learn-- how to learn a reward and thus a policy that abstracts to a lot of different states and actions. So yes, I would say it's useful for it, and IRL solves a particular problem with forward reinforcement learning in that it's not entirely exploring the state and action state to learn a policy, you can actually learn from observed behavior, but there's also issues in computational complexity of these problems. These are typically-- and relatively speaking to a lot of other machine learning methods-- computationally expensive algorithms. So yes, while I think it can be applied to things like self-driving cars, I think there's a lot of technical and research hurdles, both on the engineering and sort of the basic science side that sort of need to be overcome before I think it's ready to put it out on the street or whatever.

Ritwik Gupta: What would you say the biggest limitation is? Are you saying that the science isn't there mainly, or is it that-- I know both of these issues are concerns, but is it mainly the computational cost right now, or is it mainly the lack of basic science?

Eric Heim: So I think that there's essentially two really pressing issues within the inverse reinforcement research right now. One of them is on the computational side. So reinforcement learning, generally speaking, is a very computationally expensive problem. Inverse reinforcement learning essentially iteratively solve-- and this is the current techniques out there-- iteratively solve for a reward and then solve the forward reinforcement learning problem. So you're solving multiple reinforcement learning problems each time you want to learn a reward for the actions that were observed.

So this is incredibly computationally expensive. So if there's techniques that can alleviate the burden on the reinforcement learning side so that step is easier, or if we find ways so we don't have to solve for forward reinforcement learning every time we want to update a reward, that would be very useful.

The other issue is largely on sort of the data side. So if I only see an expert drive on a particular route, how do I abstract that idea to all of Pittsburgh, say?

Ritwik Gupta: Gotcha. That seems complicated. Or San Francisco, right?

Eric Heim: Sure, right. And so these different environments. How do we handle this? So there's a lot of different ways I think people are looking at this. There's, "How I inform the inverse reinforcement learning problem with prior knowledge about the world?" Typically you may have not seen some of the roads in San Francisco, but you still know how to navigate between them because you have some innate knowledge or prior knowledge about how roads work.

Ritwik Gupta: Hopefully.

Eric Heim: Yeah. Otherwise I'll be driving if we go on trips together, yeah. So I think that-- so how do I get this sort of what some people call common sense about driving into these things and injecting things like prior information about the world into these techniques, so this allows the generalized ability to increase for these things.

Ritwik Gupta: We talked about a lot of things, and I know you mentioned a lot of words, like Markov decision processes, forward reinforcement learning, states, policies, etcetera. If someone wanted to go learn more about this-- and I want to go learn more about this-- what should they look up?

Eric Heim: So I think there's-- one of the great things about the way the machine learning has evolved is that there's tons of resources out on the internet for these things. But if you want to start at a little more classical sources, like in the research for reinforced learning, there's a really great textbook from Sutton and Barto. The title itself is escaping me right now, but if you google

reinforcement learning, Sutton and Barto, you'll probably get there. That's a really good, thorough description of reinforced learning. And inverse reinforcement learning, there's certainly blogs out there that'll get your foot in the door, but also sort of the more fundamental works are based off of-- let me try to remember. So there's Bayesian inverse reinforcement learning, is a good resource. That's a really classic paper for inverse reinforcement learning. There's max causal entropy inverse reinforcement learning. That's another paper. There's tons of work since those. Those are sort of the classic basic methods for a lot of the research that's built off of them. But I think starting there and sort of working your way up through the literature I think is a good resource. But of course there's tons of informative blogs and sort of more introductory material, but the research is sort of-- you can follow the paper lineage I think forward and backward from those two works.

Ritwik Gupta: Fascinating. Do you have any introductory blogs that you yourself have put out?

Eric Heim: Not at the moment. I'm currently assembling a practitioner's guide with hopeful release, but if we are able to do that, I would gladly put that out for people to read.

Ritwik Gupta: And all I really have to say is Roger Federer, watch out. Inverse reinforcement learning, Ritwik Gupta, world number one coming soon.

For more information on inverse reinforcement learning, please hit us up at info@sei.cmu.edu, or just check the links in the description below. We're more than happy to add anything that you guys might have questions about. Hope to hear from you soon.

Related Resources

For Reinforcement Learning:
https://blog.insightdatascience.com/reinforcement-learning-from-scratch-819b65f074d8
http://www.cs.toronto.edu/~zemel/documents/411/rltutorial.pdf

For Inverse Reinforcement Learning:
https://thinkingwires.com/posts/2018-02-13-irl-tutorial-1.html