SDS PODCAST EPISODE 773: DEEP REINFORCEMENT LEARNING FOR MAXIMIZING PROFITS, WITH PROF. BARRETT THOMAS

Jon Krohn: 00:00:00 This is episode number 773 with Dr. Barrett Thomas, Research Professor at the University of Iowa. Today's episode is brought to you by Ready Tensor, where innovation meets reproducibility.

00:00:14 Welcome to the Super Data Science Podcast, the most listened-to podcast in the data science industry. Each week we bring you inspiring people and ideas to help you build a successful career in data science. I'm your host, Jon Krohn. Thanks for joining me today. And now, let's make the complex simple.

00:00:45 Welcome back to the Super Data Science Podcast. Today, you're in for a treat with the eloquent and deeply knowledgeable Professor Barrett Thomas. Barrett is Research Professor in Business Analytics and Senior Associate Dean at the University of Iowa's College of Business. As will soon be unsurprising to you, when you hear how well he communicates complex concepts, he's won multiple teaching awards amongst other academic prizes. He holds a PhD in industrial and operations engineering from the University of Michigan. Today's episode is a technical one that will appeal primarily to hands-on practitioners like data scientists, software developers, and machine learning engineers. In the episode, Barrett details what Markov Decision Processes are and how they relate to deep reinforcement learning, how operations research leverages neural networks to maximize business profits and minimize business costs, how same-day delivery has been made possible by machine learning, and how aerial drones and autonomous vehicles will revolutionize supply chains and transportation. All right, you ready for this fascinating episode? Let's go.

00:01:47 Nice. Well, welcome to the University of Iowa where we have an amazing, if you're watching the YouTube version of this, the University of Iowa is super generous with

recording equipment. I'm sure the audio even sounds spectacular, but I can see the video in real time here. And wow, this is some great production value. The reason why I'm at the University of Iowa is because I'm interviewing Professor Barrett Thomas. So welcome to the show.

Barrett Thomas: 00:02:14 Well, thank you for having me. I have had a chance to follow your show and you do great work, so it's exciting to be a guest.

Jon Krohn: 00:02:22 Nice, thank you. Well, let's dig right into your research. So you sit at the intersection of delivery logistics, and machine learning. Do you want to tell us a bit about why that's interesting, and I also know you have a lot of association with the business analytics program, you have an operations research background, so maybe you can tie those different fields together?

Barrett Thomas: 00:02:42 Yeah, sure. So my first job out of college, I was an intern at Schneider Logistics, which was a subsidiary of Schneider Trucking. So if you've seen-

Jon Krohn: 00:02:56 Orange trucks.

Barrett Thomas: 00:02:57 The orange trucks on the road, that's Schneider, and they're based in my hometown of Green Bay, Wisconsin. So I ended up there in an internship, so found my way into the trucking business. One of the things that we were doing at that time, it was the dawn of third-party logistics. So we were working with a lot of different clients, helping them in many cases move different components to manufacturing facilities. One of the things that happens in that context is that you don't necessarily fill up a full truckload. And so, we were looking at ways of improving the cost by along the way, stopping somewhere else and putting more in the truck to get better, essentially utilization.

00:03:57 And so, that led me into a career in logistics and particularly vehicle routing and eventually to a PhD program in operations research where we were exploring these problems as part of a Sloan Industry Foundation Program at that time. So I come to these as an operations researcher, as somebody trying to optimize. And so, first and foremost, that's the lens that I take to these problems and only came to the machine learning side much later. So a lot of the early work that I did in these applications, we were trying to certainly solve problems, essentially find policies for the decision making, but we were incredibly limited at that time. The class of problems that I was working on are called sequential decision problems. And so, essentially we're going to make a decision now. We'll incur a cost, we'll receive some sort of reward, and that will have though an impact on the future. But at the same time, of course, the future is unknown.

00:05:23 So we're making this decision and we're going to make another decision in an unknown future, but by making this current decision will affect perhaps the decisions that are available to us in the future. And so, you model those as Markov Decision Processes. A Markov Decision Process or an MDP is well known to have the curses of dimensionality. The traditional method of solving these, at least a finite horizon problem is to use what's called backward dynamic programming. But what that means I need to be able to do is I need to essentially enumerate the different paths in this problem. And so, these curses of dimensionality come into play and the one that's most famous is what we call the state space, the curse of state space dimensionality, is that I can't possibly, in most cases, write out every possible future state that I could find myself in based on every possible decision that I could make along the way.

00:06:37 And so, we really, particularly in the late '90s, that reduced a lot of, if we wanted to solve it exactly, that

reduced us to very toy problems, and in logistics that meant maybe I could take a small road network and with some very limited stochastic outcomes and we could solve these small problems. So what we spent a lot of our time doing at that time was trying to understand the mathematical structure of the optimal solution. This has been incredibly powerful in things like inventory control. The structure didn't, it wasn't as evident. Unfortunately, when you get into the vehicle routing problems, this dimension of time and space that you have in routing often made that difficult. And so, from there, we sort of turned to a lot of heuristic decision-making. The easiest thing to do is simply to ignore the future when you make your decision right now. We call that a myopic decision-making. It's easy to show example after example where that puts you into really bad positions in terms of the future decision-making.

00:08:10 And so, the goal in a lot of the work then that I was doing was how do we solve something using a heuristic that has at least some accounting for the future? And that became a big part of my research. Then, though, we have the reemergence of machine learning, particularly the advent of deep learning. And you realize now that we can approximate the future through the neural net as that approximation architecture and we wipe away the problem of the state space. And so, that for the last, I don't know, 10 years, that has been the, I mean, just an amazing advance I think for the kind of work that I do.

00:09:09 The challenge though is that there are still other curses of dimensionality. When you're dealing with a routing problem, that can be the combinatorics of the solutions. So some of you may be familiar with the traveling salesperson problem, where I have, let's say, a set of cities and I want to build a path that visits all those cities and maybe technically I want to return to the starting point. If you think about that as the number of cities is growing,

the number of different ways in which I can order that is exploding. Of course, there has been an incredible amount of work on this. Many of you may know Bill Cook and his Concorde Solver and they can solve just incredibly large-

Jon Krohn: 00:10:05 Tell us about that. I don't know about that problem.

Barrett Thomas: 00:10:07 So this goes back to, I don't even know how long, the Königsberg bridge problem and I think has similarities to this and traveling salesperson. And so, people have been working on various math programming methods to solve this traveling salesperson problem, at least since Danzig and post-war.

Jon Krohn: 00:10:39 Yeah, I suspect that the traveling salesperson problem, which that's a really great way to call it, I'm so used to it being called a more gender-specific kind of.

Barrett Thomas: 00:10:48 And it traditionally would've been, and I wanted to be a little more gender-neutral.

Jon Krohn: 00:10:52 Yeah, absolutely. It's one of those ones where that's something that I try to be really mindful about, but that is one, the traveling salesperson problem is one place that I had not stamped-

Barrett Thomas: 00:11:01 Or we can just call it the TSP and avoid that issue altogether.

Jon Krohn: 00:11:06 And with that problem, probably most of our listeners are aware of the problem, but we can quickly rehash it here, which is that it's that if you have a number of cities and you want to find the optimal route between those cities so that you spend, I guess, as little time and money cost like your classic-

Barrett Thomas: 00:11:27 Right. Some measure of cost, whether it's distance or time, you want to minimize that cost in terms of the order in which you visit those cities.

Jon Krohn: 00:11:36 And when there's three cities, it's computationally tractable, but then it very quickly-

Barrett Thomas: 00:11:40 [inaudible 00:11:41].

Jon Krohn: 00:11:41 It's even I think with 10 cities or something, it starts to become crazy, the number of possible paths between them. And so, I know this is one of the places where I think there's been completely different kinds of computational approaches like genetic algorithms where, and I don't mean a genetic machine learning algorithm, I mean literally using biological material.

Barrett Thomas: 00:12:06 I mean, the idea is that each solution is and of itself like a genome. And so, you sort of mix and match solutions to get new solutions. The idea would be that I'm going to cut one good solution that I found with and then add a piece of another good solution. And so, essentially I'm reproducing and hopefully improving the fitness of my populations of solutions, and we continue that reproduction to hopefully find the best solution.

Jon Krohn: 00:12:46 And so, you're describing a genetic machine learning algorithm there.

Barrett Thomas: 00:12:52 I mean, that's what's called a genetic algorithm.

Jon Krohn: 00:12:53 But what I'm talking about is actually using like a liquid and having actual biological genetic material. And so, I read about this years ago, but it's one of those out there. It's because it's so computationally complex to try to model the traveling salesman perfectly with our computing methods that we use today predominantly,

and there might be quantum approaches to solving traveling salesman problem out there.

Barrett Thomas: 00:13:21 There could be, yeah.

Jon Krohn: 00:13:22 But I remember reading years ago about using genetic information floating in solution, and I guess it must be similar to what you just described there, genetic material with genetic machine learning you are using, you're using different strains where you're like, okay, this strain was pretty good. That other strain was pretty good. Let's mate them together and see what the result is. In this case, it's just random, I guess, where genetic, I don't remember all the details, but the point is that it's such a computationally complex problem that people come up with really far out ideas for ways to try to solve it efficiently.

Barrett Thomas: 00:13:59 Right. Whether it's the genetic algorithm, either one of these or ant colony algorithms and all these other things that have sort of emerged around different ways that we can try to solve these more generally combinatorial optimization problems. But the TSP itself has been just this subject of just a tremendous amount of work, and whether it's the heuristic solution methods that we're talking about, or more exact integer programming coupled with various dynamic programming type of approaches. And those have evolved to the point, particularly through this Concorde Solver that is available that can solve tours that have millions of cities in them.

Jon Krohn: 00:14:50 Oh, really?

Barrett Thomas: 00:14:51 And it saw them to exact solutions, which is really-

Jon Krohn: 00:14:55 Is this Concorde like the grape or the jet?

Barrett Thomas: 00:15:00 I think it's spelled like the jet. So I don't know the genesis of that name. I mean, maybe you have to have Bill Cook on your show and you can talk about it. I think he'd be a great interview.

Jon Krohn: 00:15:11 Research projects in machine learning and data science are becoming increasingly complex, and reproducibility is a major concern. Enter Ready Tensor, a groundbreaking platform developed specifically to meet the needs of AI researchers. With Ready Tensor, you gain more than just scalable computing, storage, model and data versioning, and automated experiment tracking, you also get advanced collaboration tools to share your research conveniently and securely with other researchers and the community. See why top AI researchers are joining Ready Tensor, a platform where research innovation meets reproducibility. Discover more at readytensor.ai.

00:15:53 And so, this Concorde Solution, I mean, is there a high level way of describing that or basically that's the point of this is that the Concorde... I can't remember now how I got into this. I interrupted you. You mentioned it briefly.

Barrett Thomas: 00:16:06 Well, I was talking about the fact that the action space of the MDPs when you get into these logistics problems itself is combinatorial because it's essentially, the sub problem's a routing problem. And so, you can't enumerate all these solutions, which becomes the problem.

Jon Krohn: 00:16:25 But the Concorde Solver can help you out?

Barrett Thomas: 00:16:28 Well, see, that becomes the challenge in that when we use the neural net as that approximation architecture, and if everybody's probably familiar with Q-learning. So I have a state of the world, that's everything I know about the world at this time. And so, if you're thinking about a vehicle routing problem, it might be the current location in my vehicles, we at least need to know who the

unserved customers are at that point. Maybe you want to hold on to some sort of route plan that maybe will change, this information can all be useful. So that's our state. And then from that state, we're at this point, we want to make a decision. And the challenge is that that decision in and of itself can result in a routing problem. When you think about that's okay, if I have this neural net, I could pull that out and maybe I've used all linear activation functions, so I now have a linear model. But doing that in a neural net, it could be incredibly large as a math program making this training, then the training becomes almost impossible in and of itself.

00:18:03 And so, some of the things that people have done is instead of neural nets, let's just use some linear approximation of the future. And so, now, that's a little bit easier so I have this single kind of linear equation that goes in my objective and I can solve that, and you can train those relatively quickly. But the truth is that that linear function is obviously, it's a particular functional form, and that isn't necessarily the right form to get a good approximation of the future value of making this of any particular decision, which is why you want to use the neural net. And so, you're at this sort of crossroads there with the one really powerful technology in terms of its ability to approximate, but creating on the other hand then some really significant challenges in how we go about trying to solve the problem of essentially choosing the action in that case.

Jon Krohn: 00:19:12 And so, this sounds like we might be getting into cost function approximation, which maybe we can put a pin in.

Barrett Thomas: 00:19:19 So cost prox and function approximation is a way to get around some of these problems. So maybe I should step back and talk a little bit more about the MDP then to tee that up.

Jon Krohn: 00:19:34 Yeah, let's talk about the MDP because I also, something that's interesting for me, I don't know MDPs very well, but I am familiar with reinforcement learning and deep reinforcement learning in particular. And it sounds like when you're talking about MDPs, with Markov Decision Processes, I'm aware of some of the ideas here. So Markov for example, being that I believe that's the property that all of the information from the most recent time step is all that you need.

Barrett Thomas: 00:20:02 Correct.

Jon Krohn: 00:20:03 So the Markov property, for example, applied to stock markets is that you don't need to know all of this... When you're assuming the Markov property, you don't need to know all the historical stock prices. You say, what were the stock prices yesterday? I'll use those, that one snapshot of data to predict today's.

Barrett Thomas: 00:20:20 Right, yeah. So technically, what it means is I don't need to know the history in order to know what actions are available to me and to know the probability distribution on those transitions in the future and the historical context of MDPs, that's what it would mean.

Jon Krohn: 00:20:40 Nice, nice. Other than that Markov property, something that seems very familiar to me from my understanding of reinforcement learning. So first of all, you apply reinforcement learning to solve sequential decision-making problems.

Barrett Thomas: 00:20:56 Correct.

Jon Krohn: 00:20:56 Which are also using an MDP to solve sequential decision-making problems and similarly with reinforcement learning, we're often talking about estate space. And so, the state being the current situation that you're in.

Barrett Thomas: 00:21:12 The state's the information you need to make that next decision, it helps define what decisions are available to me and also ultimately what the transition would look like if I made that decision.

Jon Krohn: 00:21:29 So you're in some state. Some way in my textbook, Deep Learning Illustrated, I use the example of video games because I feel like that you can kind of imagine a video game being frozen.

Barrett Thomas: 00:21:42 Exactly.

Jon Krohn: 00:21:43 You're controlling a joystick, but let's say you just pause and so there's just a certain state on the screen. Let's say you're playing pong where there's a paddle at the bottom of the screen. And so, you find yourself at some state, you can see where the pong paddle is on the screen, and you learn through experience or an algorithm, a reinforcement learning algorithm can learn through experience the pressing left on the joystick will change the state so that the pong paddle moves left. So you have a state which is represented in the video game example by the pixels on the screen, and then the action you take is the joystick movement. And then, that changes the state on the screen, and you find yourself in a new state. And the state could remain fixed until you move the joystick again or potentially in something like pong, it could also be a ball moving. And so, the state is changing. So in fact, you're actually, by not moving the joystick, you are making the decision. You're taking the action of not moving as that state continues to change.

Barrett Thomas: 00:22:45 Right. Even in pong, well, maybe not in pong, if you had the physics and you could model the physics exactly, you would know what's going to happen. But in the real world, there's exogenous things that are happening to you. Demand is happening, and I don't know what that demand is in the future. So that change from one screen

to the next is random in the real world. And so, that then adds another element into our MDP.

Jon Krohn: 00:23:19 It's random typically around a probability of distribution. If the demand for this car part that needs to be delivered from one location to another is a certain amount on one day or one month, you could have a distribution around how likely it is to be needed the next month, maybe a little bit more, maybe a little bit less. It's unlikely to be a lot more or a lot less.

Barrett Thomas: 00:23:43 Right. And so, I mean, there's a probability distribution, it's just whether or not we know what it is. The most complex problems, it might be really hard to even specify what that is, which is why when we solve these problems, or rather when we're trying to train something like machine learning, we would want to, in general, we turn to a simulation to do that, that because we are not modeling that distribution exactly. We're using the simulation to advance forward the time.

Jon Krohn: 00:24:22 So even when you started your career, you would've been doing simulations.

Barrett Thomas: 00:24:26 So the first work I did know actually, we were assuming a probability distribution existed and one that we knew, and then we would, if we could generally derive structure of the optimal policy without putting restrictions on that distribution, we would generally, we couldn't do anything with that. So then, we would assume a particular distribution and try to again, find structure, optimal structure from that assumption. But again, that's limiting. That's really limiting about what the world looks like. I have to make assumptions about this distribution. I may or may not know what that distribution is. If you end up in a situation where that distribution is a result of many different actions, so take queuing examples, for instance. Those are are very, very difficult to specify.

Jon Krohn: 00:25:23 Queuing examples.

Barrett Thomas: 00:25:25 Queuing is simply the waiting lines, but that could be you at the grocery store or, well, nobody goes to the bank anymore. That used to be the classic example, but nobody does that. So maybe it's at the drive-through. So it happens in terms of information processing, you end up with packets queued, things like that. So there's distributions for say how long you're going to wait in line, but those can be very, very difficult to even specify. So a simulation can help us at least generate the data we need to take the step forward in time.

Jon Krohn: 00:26:07 Nice. And so, the Markov Decision Process, it allows you to simulate in a way, or it's just for evaluating a simulation?

Barrett Thomas: 00:26:15 No. So Markov Decision Process is simply a model of our decision-making. So in my Markov Decision Process, I have my current state, this is what I know about the world, and then that allows me to define the set of actions that are currently available to me. What I want to do, let's say we're maximizing, I want to choose the action that is going to maximize the current reward, the reward that I'm going to receive right now from taking that action. So maybe I'm going to choose to make a sale, so if I make that sale, I'm going to get an immediate reward on that. Somebody's going to pay me.

00:27:00 Now, added to that though is a second term. And that second term is an expectation of the future value that I can earn given what my current state is, as well as the action that I've just taken. So it becomes a conditional kind of expectation on that state in action pair. But since I've just made this sale, well that now means in the future I can't make and sell that exact same item. So that is impacting what that future cost or reward is going to be. And so, if we were going to solve this exactly, then ideally

I need to know that future reward for every single state and action pair. This becomes, that's the challenge of solving these exactly. So now, I have this model of my decision-making process.

Jon Krohn: 00:28:08 And all of that stuff that you just said around the MDP, so it sounds... Oh yeah, now the pieces are coming together for me. When I'm doing a reinforcement learning problem, it is a Markov Decision Process.

Barrett Thomas: 00:28:24 It is a Markov. So the Markov, no, no. Well, I mean the Markov Decision Process is a model of the problem that you are solving with reinforcement learning. So what I'm doing in reinforcement learning is I have a state, and then let's say if you're doing Q-learning, I have a state and I take one of my actions, and that becomes the input into my model, and then that output becomes a value of that state action pair. So that's one decision. Then you step forward in time because I've taken, I have a state and I have an action, and now I'm going to move into some new state. And how I get to that new state is where the simulation comes in, because there's going to be something that happens randomly between the current state and action I've taken and my future state. And it's going, so now it's going to be my current state, the action I've taken plus what we call that exogenous information that then leads me into that future state. And we're simulating that exogenous information.

Jon Krohn: 00:29:41 So we could create a program that simulates the environment-

Barrett Thomas: 00:29:45 Exactly.

Jon Krohn: 00:29:45 That our reinforcement learning agent is then making decisions in. And over time, you can build up through running this simulation many times, you can start to get

a sense of, should I be making this sale now or making other offers first?

Barrett Thomas: 00:30:02 Well, I'm using the reinforcement learning to learn what I should do because I do an entire sequence of decisions, and now I can just do a backward pass. I know the value of each of those decisions. And so now, I can do that backward pass. I can add it all up going backwards. Now I know, oh, here is for that particular trajectory, the actions that I took. This is now a sample of the future value of that particular action in this particular state. And so, now we start running many, many trajectories using our simulation. So now that's the information that we are using to update using the reinforcement learning techniques. Essentially if we're using a neural net update our neural net based on each of those trajectories, and then hopefully at some point in the future we converge.

00:30:59 But I know you've done a lot of work in reinforcement learning in some areas. One of the things that is challenging in the environments that I've worked in is that getting to convergence can be really, really challenging. There's a lot of, and maybe we're just bad at our design of our neural nets, but we find that we get a lot of, I think, jumpiness in the values of our policies that we're returning.

Jon Krohn: 00:31:35 Ready to master some of the most powerful machine learning tools used in business and in industry? Kirill and Hadelin, who have taught millions of students worldwide, bring you their newest course Machine Learning Level Two. Packed with over six hours of content and hands-on exercises, this course will transform you into an expert in the ultra-popular gradient boosting models, XGBoost, LightGBM and the CatBoost. Tackle real-world challenges and gain expertise in ensemble methods, decision trees, and advanced techniques for solving complex regression and classification problems.

Available exclusively at superdatascience.com, this course is your key to advancing your machine learning career. Enroll now at superdatascience.com/level2. That's superdatascience.com/level2.

00:32:18 So converging meaning having just nice and smooth roundly getting to that maximum reward as opposed to hopping all over the place.

Barrett Thomas: 00:32:28 Right, exactly.

Jon Krohn: 00:32:30 And so, I guess another key term here to note is that, so we've been talking about reinforcement learning a lot. We've been talking about neural nets. And when those two things are combined like you've been describing, that is deep reinforcement learning.

Barrett Thomas: 00:32:41 That's deep reinforcement learning, and what you're approximating in that particular case is you're using it to approximate the value of the future. You could of course also use some sort of policy gradient approach and then you're directly learning the distribution on the policy.

Jon Krohn: 00:33:04 A weird thing about, at least my understanding of this deep reinforcement learning term is that it doesn't matter, so typically when we talk about neural networks, it's only a deep learning architecture if it has, say, three hidden layers.

Barrett Thomas: 00:33:17 Multiple layers, right.

Jon Krohn: 00:33:18 But we can call it deep reinforcement learning even if we just have a single hidden layer in our neural network. We don't call it shallow neural network reinforcement learning.

Barrett Thomas: 00:33:28 That's true. Everything is deep reinforcement learning now regardless of the number of layers that we use. Yeah, that's true.

Jon Krohn: 00:33:36 Sorry, I jumped in with that as you were starting to talk about policy a bit more. I feel like policy is a term that maybe we should define. I mean, at the onset of the episode, you talked about policy and it was so, we moved on from that so quickly that it sounds like an insurance policy or something.

Barrett Thomas: 00:33:55 So a policy is a mapping from a state to an action or to a decision. That's all a policy is.

Jon Krohn: 00:34:05 It's like a term that you could use in normal language. You could be like, "When I see a pedestrian on the road, I have a policy of hitting the brake."

Barrett Thomas: 00:34:15 Exactly. No, that would exactly right. It's a mapping from what you've seen. So even going back to the pong example, what you're seeing is that set of pixels, that's our state, that's the input, and now we get an output that's telling us hit the brake.

Jon Krohn: 00:34:32 When I see the pong ball move to the right, I have a policy of moving the pong paddle to the right. Gotcha. Cool. So all right, we've covered a lot of terms. So to quickly recap, actually you could phrase this better than me. Describe how you do this perfectly when you say the connection between reinforcement learning and Markov Decision Process.

Barrett Thomas: 00:34:59 So the Markov Decision Process is a model of the problem that you're seeking to solve. Reinforcement learning is the process by which we take to learn the, well, it's not optimal, but to learn the best possible policy we can for the problem that we've modeled.

Jon Krohn: 00:35:21 Yeah. And then, so within that world, we've got state, which is like pixels in the video game. We've got actions we can take, which is the joystick in the video game. And we have policies which is the mapping of that state to some particular action. And so, this scales up from our small pong example to very complex problems like you have in logistics where a given logistics company like Schneider would have thousands of trucks all over the country, all over the US, maybe probably internationally traveling as well, going to parts factory in Canada and bringing the part down to Michigan. And so, you're trying to optimize across all of this, how can we minimize driver time or minimize fuel expenditure? And so, this framework, this reinforcement learning framework, it scales up to these very complex problems.

Barrett Thomas: 00:36:29 It does. So what we're using reinforcement learning and the neural net for is, let's go back to Q-learning. We want to learn the value of that second term of our Markov Decision Process, that reward that we can earn in the future given this state, given this action.

Jon Krohn: 00:36:51 And that reward is defined. And I guess we're going to talk a bit more about how you can have cost functions be approximated, but typically in the way that I think about cost functions or reward functions, which are really the same thing. If you are maximizing reward like dollars, that is an objective in machine learning that you're trying to maximize. But you could also think about it typically when we think about stochastic gradient descent, which is a very common algorithm for optimizing machine learning models. In that case, we're typically minimizing cost. But in this case, in an operations research sense, that cost can literally be dollars.

Barrett Thomas: 00:37:36 Yes. In the problems we're solving, that could literally be dollars.

Jon Krohn: 00:37:39 And so, you're trying to maximize reward, maximize dollars or minimize cost, minimize dollars spend.

Barrett Thomas: 00:37:44 Right. And so, this is where we can talk about a cost function approximation. So we've been talking about using reinforcement learning and trying to approximate that second term. Well, that can still be very complicated. And certainly before we had something like neural nets, we might've had to have some other heuristic that allowed us to do that. So one of the things we might consider is, you know what, we can just, we're not going to deal with that reward to go or that cost to go. It's too complicated. We're going to try something else. We're going to say, "All right, instead of that, I'm going to just work with a simple penalty term and I'll have my current reward and then I will have some parameterized penalty term." And so, the paper that I think you probably were looking at where we did this, we were exploring how could you support decision-making in this new set of meal delivery services that have grown up, whether it be Grubhub or something like Uber Eats.

00:39:15 We have customers who are going online, they're placing orders from a set of restaurants. You have the platform that then needs to get those out to drivers. So how would you make decisions about what drivers should serve which orders, when, and what goes together? So the complication there is that new orders are always arriving, drivers are moving around to customers in different restaurants, and at the same time, you don't even know when the food might be ready at a particular restaurant. How much visibility do we even have to what their order queue looks like? So it may be longer or shorter times at that restaurant.

Jon Krohn: 00:40:05 It's so crazy complicated. I haven't thought about that.

Barrett Thomas: 00:40:08 And so, there's a lot of different moving pieces. There's a lot of different stochastic elements to that. I mean, just the two alone, what customers are calling or they don't call anymore, they use an app on their phone, when, and then what's happening at the restaurants because there's in-person folks at the restaurants. There might be other delivery platforms that are also visiting that. So there's a lot of randomness that's going on there. And so, we were really interested in how would you build decision support for this? And this did predate the deep learning.

00:40:46 And so, we were looking at other approaches, and at least at that particular time, one of the things that the platforms and certainly the restaurants cared about in getting food to their customer was what we were calling freshness. So the longer it takes to deliver the less fresh that food becomes. So we wanted to, in essence, if you will, maximize freshness. So it's obvious in hindsight, but when you made a decision to assign a meal, even if we just assumed the expected time to prepare that meal, the expected time then to take that meal and deliver it, we could look at that expectation. We could say, "Huh, we're nearing this soft deadline we have for how long we want to go from order to delivery."

00:41:49 And so, as that delivery time pushed out to that deadline, we said, "You know what? Those aren't the best solutions because we don't have then any time that will buffer the randomness that's in the system." So what we want to do is we want to penalize a decision that is going to lead to that delivery happening really close or closer to that deadline. And so, we did something simple. You could just parameterize that as a linear function and penalize the time as you got closer and closer to the deadline. And so that, that's a simple cost function approximation, and you can do different techniques to learn the right parameterization. In this case, it would be what's the slope of our linear penalty term? And that now, that is a

much simpler decision-making tool than obviously putting this into a neural net.

00:42:59 Even with all of these technologies, there are a lot of cases where methods like this can be really valuable. As I am sure you're aware, obviously people want, we talk a lot now as we do machine learning about interpretable models or things like that. Well, this would be a form of an interpretable model because I could describe this to a driver, I could describe it to a restaurateur as I'm trying to sell my platform to them. People can understand this in ways that the neural net's a black box. And so, there are advantages to using tools like a cost function approximation because ultimately you do have to deploy these into the real world and people have to use them and they want to know how it works and they want to be able to trust it. When you can describe that, you can build that trust.

Jon Krohn: 00:44:07 Mathematics forms the core of data science and machine learning. Now with my Mathematical Foundations of Machine Learning course, you can get a firm grasp of that math, particularly the essential linear algebra and calculus. You can get all the lectures for free on my YouTube channel, but if you don't mind paying a typically small amount for the Udemy version, you get everything from YouTube plus fully worked solutions exercises and an official course completion certificate. As countless guests on the show have emphasized to be the best data scientist you can be, you've got to know the underlying math. So check out the links to my Mathematical Foundations and Machine Learning course in the show notes or at jonkrohn.com/udemy. That's jonkrohn.com/udemy.

00:44:52 Gotcha. So it's an explainable linear model where each of the parameters in that model, you can say, if this thing

goes up driver or restaurant, then this is the exact effect it's going to have on cost.

Barrett Thomas: 00:45:11 Yeah, exactly.

Jon Krohn: 00:45:13 So I guess is there potentially a trade-off in that scenario whereby allowing for more explainability, there's potentially less nuance?

Barrett Thomas: 00:45:22 Oh, yeah. I mean, you could just the fact that one, let's say we've chosen that linear penalty term, well who knows if that's even a good functional form of what our penalty should look like. So we've already made that decision. And then, something like a cause function approximation is it's pretty crude. And so, like you said, you lose nuance and really probably more likely you really lose robustness. There's going to be cases where it gives you really, it isn't giving you the best solution. There are going to be particular states where, you know what, the best action right now is for us to actually make a decision that makes it look like this delivery is going to be right up against the deadline. That cost function approximation doesn't know that particular case and you'll have made the wrong decision at that point. You hope that it works in enough of the cases to still give you really good performance. But there's absolutely that possibility that you're going to end up making a bad decision in particular circumstances.

Jon Krohn: 00:46:44 Very cool. All right, changing gears here a bit. Let's talk about drones. So we've been talking about same day delivery and drones come up as something that could be super helpful in having same day delivery. I mean, I guess it's conceivable though, I don't know of this yet, or at least I don't see it in New York, of either a sidewalk drone that just drives itself or a flying drone bringing me a meal or bringing me some small Amazon order or something. Where are we in terms of drone deliveries happening? Are

there places that that is happening regularly, and what's the benefit once we get it to work? How does that complement existing systems?

Barrett Thomas: 00:47:32 Companies have been piloting drones, whether it was Google had a pilot on this, Amazon has been trying to do this. You do see JD.com in China has used drone deliveries. Where I think it's most successful at this point is not so much in rural areas or in cities, but when we're delivering into rural areas and we might send that drone and we're going to a more rural area, it drops off a set of packages in the backyard of somebody who then delivers them to the individuals. But when we're doing research, we also want to look into the future and we want to try to understand what could that future look like. Should I be even in trying to invest in these technologies? Could there be any advantage to doing so?

Jon Krohn: 00:48:30 Yeah, you might discover that actually your cost ballooned. It seemed like a cool idea, but for some reason...

Barrett Thomas: 00:48:35 Yeah, that's exactly right. And so, one of the papers that I have with a co-author from Germany, Marlin Ulmer, and then a former PhD student, Xinwei Chen, we were looking at this question because we wanted to know, wow, if you're Amazon and you're doing your same day delivery in an urban environment, would I ever want to use a drone? Would I just want to stick with trucks or maybe I want to use some combination of the two? And in fact, what we found is that at least at that time, if you considered that the drone technology, it could carry one package. And so, it would take a package, it would go out from the delivery depot, it could deliver it, then it had to come back and pick up another. But if you're using a vehicle, well, a vehicle, I can have multiple packages on, so I can put multiple packages onto it. It can go out and it can do a

delivery route and then return and pick up the next set of packages.

00:49:45 So each of those might have an advantage. The drone moves faster. It's not affected by traffic, but it's only doing one at a time versus the truck that's affected by all those things but I can carry multiple packages. And so, maybe the result wasn't that surprising, but it turns out you do want to use both and you would deliver things that were further from the depot, you would deliver those using the drone. And I would use trucks as we were closer to the depot because I could put more packages on it and I could take advantage of the fact that it wasn't doing those back and forth trips to the depot.

00:50:23 Now, the question is could we ever use those in a city? There are tremendous challenges in a place like New York in the fact that you have really tall buildings. Once you have tall buildings, you have the effect that might have on winds moving through the buildings. Do we really then want drones that maybe these are small drones, but they still weigh something that could have challenges and be knocked out of the sky. So subsequent research is looking at, well, okay, maybe I don't want to have drones delivering those packages, particularly in urban areas like that. But maybe what we want to do is instead of having that truck go back and forth, maybe I want to use a little bit larger drone and resupply the truck closer to where it's going to do its delivery. So maybe it comes out to a loading zone of some kind, an area that is a little safer, and maybe we do that.

00:51:29 So we didn't do that work, but there's been folks starting to look at that, and I think that's a really promising idea to think about. And particularly when you think about many of the dense European cities where bringing in delivery vans can be really difficult, they're already doing things like cargo bikes in many of those cities. Well, those

cargo bikes have a pretty limited capacity anyway, and they're not going, you're going to have to ride the bike back. And yes, they're electric bikes, but ride them back and forth to these depots. Maybe we want to do something in between so that it's just a lot more efficient.

Jon Krohn: 00:52:11 That's cool.

Barrett Thomas: 00:52:12 And we're just getting more productivity.

Jon Krohn: 00:52:14 That's a totally new idea to me. It never occurred to me and makes so much sense. Super cool. In all the years that you've been doing this, obviously we talked about the example of deep learning coming into the picture and then being able to use that to approximate some functions. What else has changed? I mean in terms of programming languages you might use or even the way that you solve problems in general, maybe today that we have more compute, we have more storage, maybe you can approach problems in a very different way than you would've when you started your career.

Barrett Thomas: 00:52:51 So there's that and as well as the problems that we work on. I think that what's really emerged is that last mile delivery, if you go back 25 years, well, what did that mean? That means that maybe it was the dawn of the internet, but you might've still been getting a catalog and calling in, ordering something, it somehow moves through the system and it ended up at a UPS or a FedEx. And so, at the beginning of that day, they could say, "Okay, these are the packages we have to deliver today," and they would deliver that.

00:53:32 And so, in that sense, you had sort of this beginning of the morning problem where I knew everything I was going to know for the day. And so, we talked about Concorde, you had systems like that that were actually pretty effective if you wanted to solve that particular problem.

There were operational challenges to updating your routing every day, but the companies were really good at that. As we fast-forward and you start to get same-day delivery, you get the meal delivery platforms, you get increasing Uber-type technologies. We've really seen an explosion in these new models of transportation that have honestly made this a really exciting time. I mean, some of the listeners may not agree that that's exciting from a research-

Jon Krohn: 00:54:23 I hope some of them did because that's why I picked you to be on the show.

Barrett Thomas: 00:54:25 Well, I appreciate that.

Jon Krohn: 00:54:27 Because it is interesting when somebody says logistics to me, for some reason at a distance, that word doesn't excite me like some other words in technology might. But in fact, it is fascinating and it's such a complex problem, and it impacts probably all of our listeners many times a day.

Barrett Thomas: 00:54:48 It's almost certainly impacting them many times a day. It's related as well to supply chain. We know that particularly during the pandemic, that had an incredible impact on many people's lives as well when we couldn't get goods and-

Jon Krohn: 00:55:05 Couldn't get a couch.

Barrett Thomas: 00:55:05 Yeah, you couldn't get a couch or toilet paper or whatever it was. And so, that supply chain would have, certainly there were manufacturing issues that were going on, but there were also logistical issues like how am I going to move this through the system? Do you have the capacity or not to also move this from one place to another and ultimately to your end consumer? And so, that logistics part does play a role in that. As you mentioned, I mean

particularly in this day and age, I don't know how many times a day the Amazon trucks are coming down my street, but it's significant. And so, we are seeing it every single day. I appreciate that you found something interesting in there.

Jon Krohn: 00:55:52 And so, what do you think is next? So you've given us some exciting ideas, and so I might've already exhausted potentially new exciting things in this space, but drones, whether those are aerial or driven, that's something that's coming, but maybe what else is expected to change about our behavior? You were talking minutes ago about previously ordering from catalogs, companies knowing what their delivery route would be for the day, now things are moving more quickly. Real-time apps, you watch the cyclist come with the Uber Eats and you're hoping that it's hot. And algorithms like you're describing are helping ensure that my meal is getting to me hot. What could happen next? Is it just faster? I mean, what could be more real-time?

Barrett Thomas: 00:56:41 Yeah, I mean, I think it's going to be hard to be more real-time. And you have seen companies that tried to do this and they'll say, "We're going to guarantee a five-minute delivery." I mean, I just can't imagine that the work that I've done, I am pretty sure that that isn't ever going to make sense. Whenever the faster you say you're going to deliver something, that means that that delivery becomes more and more one-off. We like to talk about consolidation opportunities, that is being able to put multiple things onto one delivery vehicle. That leads to economies that you want to take advantage of. So you start doing those one-offs. You're just not going to be able to make money doing that. But the other thing that's happening as we have moved into this delivery economy, whether it's food or goods, is that that means that there's more and more delivery traffic.

00:57:46 And so, I think that that leads to, particularly as the environment becomes more and more urban, that's leading to more and more congestion. You've certainly in New York, been impacted by delivery vehicles not having anywhere to park. And so, this driver has to do something. It isn't true that they want to double park, but I know we've all seen it. It also means that as we increase congestion, we're increasing at least as long as we have combustion vehicles, we're going to increase the emissions.

00:58:23 And so, I think to me, the really interesting next step are the ways that we're going to rethink these delivery models. And drones would've been one example of that. You mentioned delivery robots is another, there's been more than 10 years that companies have been working with delivery lockers or having maybe at a 7-Eleven, you might go and they'll have that, and you go and you pick up your package there. Because that's sort of a one-stop, there's parking there generally and things like that. Two colleagues and I, my colleague here, Ann Campbell, and our former student, Sara Reed, and I have been working on how would we change the delivery model if we had an autonomous vehicle. So one-

Jon Krohn: 00:59:16 I met Ann today. She told me that you were going to quote Tennyson.

Barrett Thomas: 00:59:19 I was going to quote Tennyson. Okay then, now you're putting me on the spot. So Charge of the Light Brigade, I guess.

Jon Krohn: 00:59:28 I completely derailed you, my apologies.

Barrett Thomas: 00:59:34 What's that?

Jon Krohn: 00:59:34 I completely derailed you, my apologies.

Barrett Thomas: 00:59:36 Oh, that's okay. I could have been ready with something for you, but I'm surprised she said that. Anyway, okay. So we've been working on what different things could you do if you had this autonomous vehicle that particularly was working with a delivery person. Probably Ann remembers this as well. We were talking about what would be interesting about this, and we tend to go to lunch and went to this Chinese restaurant that we've had I think a number of good ideas while we've been eating lunch there, and had this realization that an autonomous vehicle could keep moving. And so, what would that mean? Well, that means that I could get off the vehicle, I could take a number of packages that need to be delivered. As that delivery person, I could deliver those around the block and the vehicle could meet me.

Jon Krohn: 01:00:44 Oh, yes.

Barrett Thomas: 01:00:45 And so, what's the advantage of that? Well, one of the advantages is as a driver, I don't have to find a place to park necessarily. So that's saving time. Then, you don't know where you're going to end up parking necessarily because you are subject to the availability of parking. That parking spot might be pretty far away. Now, I have what we might call a deadhead. I have to walk to that first location, and at some point I need to walk back distances that I'd rather not have to walk. And so, what if I could get off the vehicle and then it picks me up at the end of when I've delivered those packages and we go on to the next spot at which we want to deliver?

01:01:30 And so, that was some work that we did while Sara was working on her dissertation. And it turns out, particularly in urban areas, you can save a really large amount of time. And so, what would that mean? Well, that means I need fewer vehicles to do the delivery work. And so, now you have that possibility that we can reduce the vehicle congestion at least associated with some of these

deliveries. But it has other benefits. It could reduce the stress on the delivery person, reducing workplace injury, reducing workplace fatigue. It could help preserve additional parking spaces in urban environments and consumers like that, the stores like that. So there could be a lot of benefits from that.

01:02:24 Of course, there's all kinds of other models. You have models of the Superblock in Barcelona where you're trying to put delivery zones outside of the block. And this gets us to some of the things we were talking about in terms of European cities. Now, Ann, Sara and I are continuing to work on a problem where we're thinking about what's the value of having dedicated commercial parking? How much would a company be willing to pay to have that available? How much should a city be charging in order to make that available? So I think that this problem of the convenience that we have from these services also means that they're creating these other challenges. And so, that question becomes how do we improve those delivery services to then mitigate the challenges that it also creates?

Jon Krohn: 01:03:21 Fantastic. Well, this has been an incredible episode. I knew you would be a great speaker and you've exceeded expectations. Thank you. Really fascinating conversation. Before I let you go, do you have a book recommendation for us?

Barrett Thomas: 01:03:34 So I do have a book recommendation. It's a book by my friend Warren Powell, and he's looking at and also been very interested in this intersection of operations research and machine learning. And so, his book is Reinforcement Learning and Stochastic Optimization: A Unified Approach. If you're interested in Markov Decision Processes and then how that relates to reinforcement learning, I think this is a great book. He talks about cost

function approximation as well, and I think maybe some of the listeners would find that interesting.

Jon Krohn: 01:04:11 Yeah, no doubt. Well, thank you very much. If people want to hear more of your thoughts after this episode, is there any particular, do you tweet or?

Barrett Thomas: 01:04:24 I do have Twitter. I mostly am giving kudos to the Department of Business Analytics and the Tippie College of Business, but people are welcome to connect barrett.w.thomas on, I guess, it's X now.

Jon Krohn: 01:04:39 Yeah, I never say that.

Barrett Thomas: 01:04:41 Yeah. But yeah, please connect or LinkedIn.

Jon Krohn: 01:04:47 Nice. And of course, your Google Scholar profile, lots of fresh papers showing up there regularly. Amazing that you do that on top of all the administrative work I know you have to do today.

Barrett Thomas: 01:04:59 Yeah, but I think that when you get in and you become an academic, that's the thing that you really enjoy most is doing that research work. And so, I try to make sure that I am always doing a little bit.

Jon Krohn: 01:05:15 Nice. Well, thank you also for taking the time to do this today. Absolutely fantastic. And maybe we can catch up with you again in a few years and see how delivery has evolved in that time.

Barrett Thomas: 01:05:24 I'd love to. Thank you very much.

Jon Krohn: 01:05:31 That was spectacular. What a talented communicator Barrett is. In today's episode, he filled us in on Markov Decision Processes, deep reinforcement learning being a blend of reinforcement learning with neural networks, cost function approximation allowing for more explainability but the expense of nuance and robustness,

and how his operations research backs innovations such as large drones, resupplying trucks away from their depot, and autonomous vehicles moving to a pickup location while a delivery person drops off packages on foot. As always, you can get all the show notes, including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Barrett's social media profiles, as well as my own at superdatascience.com/773.

01:06:10 And if you'd like to engage with me in person as opposed to just through social media, I'd love to meet in real life at the Open Data Science Conference East, ODSC East, which will be held in Boston from April 23rd to 25th. I'll be hosting the keynotes and teaching two half-day tutorials. One, we'll introduce deep learning with hands-on demos in PyTorch and TensorFlow. And the other will be on fine-tuning, deploying, and commercializing with open-source large language models featuring the hugging face transformers and PyTorch lightning libraries. It'd be awesome to see you for all of these big events.

01:06:44 All right. Thanks to my colleaguqes at Nebula for supporting me while I create content like this Super Data Science episode for you. And thanks as always to Ivana, Mario, Natalie, Serg, Sylvia, Zara, and Kirill on the Super Data Science team for producing another fascinating episode for us today. For enabling that super team to create this free podcast for you, we are deeply grateful to our sponsors. You can support this show by checking out our sponsor's links, which are in the show notes. And if you yourself are interested in sponsoring an episode, you can get the details on how by making your way to jonkrohn.com/podcast. Otherwise, please share, review, subscribe, and all that good stuff, but most importantly, just keep on tuning in. I'm so grateful to have you listening, and I hope I can continue to make episodes you'd love for years and years to come. Until next time,

keep on rocking it up there, and I'm looking forward to enjoying another round of the Super Data Science Podcast with you very soon.