PROFESSOR: Hi, everyone. We're getting started now. So this week's lecture is really picking up where last week's left off. You may remember we spent the last week talking about cause inference. And I told you how, for last week, we're going to focus on a one-time setting.

Well, as we know, lots of medicine has to do with multiple sequential decisions across time. And that'll be the focus of this whole week's worth of discussions. And as I thought about really what should I teach in this lecture, I realized that the person who knew the most about the topic was in fact a postdoctoral researcher in my lab. Most about this topic in the general area of the medical field.

FREDRIK D. JOHANSSON: Thanks. I'll take it.

AUDIENCE: Global [INAUDIBLE].

FREDRIK D. JOHANSSON: It's very fair.

PROFESSOR: And so I invited him to come to us today and to give this as an invited lecture. And this is Fredrik Johansson. He'll be a professor in Chalmers, in Sweden, starting in September.

FREDRIK D. JOHANSSON: Thank you so much, David. That's very generous. Yeah, so as David mentioned, last time we looked a lot at causal effects. And that's where we will start on this discussion, too.

So I'll just start with this reminder, here-- we essentially introduced four quantities last time, or the last two lectures, as far as I know. We had two potential outcomes, which represented the outcomes that we would see of some treatment choice under the various choices. So, the two different choices-- 1 and 0.

We had a set of covariates, x and a treatment, t. And we were interested in, essentially, what is the effect of this treatment, t, on the outcome, y, given the covariates, x. And the effect that we focused on that time was the conditional average treatment effect, which is exactly the difference between these potential outcomes-- a condition on the features.

So the whole last week was about trying to identify this quantity using various methods. And the question that didn't come up so much-- or one question that didn't come up too much-- is how do we use this quantity? We might be interested in it, just in terms of its absolute magnitude. How large is the effect? But we might also be interested in designing a policy for how to treat our patients based on this quantity.

So today, we will focus on policies. And what I mean by that, specifically, is something that takes into account what we know about a patient and produces a choice or an action as an output. Typically, we'll think of policies as depending on medical history, perhaps which treatments they have received previously, what state is the patient currently in. But we can also base it purely on this number that we produce last time-- the conditional average treatment effect. And one very natural policy is to say, pi of x is equal to the indicator function representing if this CATE is positive.

So if the effect is positive, we treat the patient. If the effect is negative, we don't. And of course, positive will be relative to the usefulness of the outcome being high. But yeah, this is a very natural policy to consider. However, we can also think about much more complicated policies that are not just based on this number-- the quality of the outcome.

We can think about policies that take into account legislation or cost of medication or side effects. We're not going to do that today, but that's something that you can keep in mind as we discuss these things. So David mentioned, we should now move from the one-step setting, where we have a single treatment acting at a single time and we only have to take into account the state of a patient once, basically. And we will move from that to the sequential setting. And my first example of such a setting is sepsis management.

So, sepsis is a complication of an infection, which can have very disastrous consequences. It can lead to organ failure and ultimately death. And it's actually one of the leading causes of deaths in the ICU. So it's of course important that we can manage and treat this condition.

When you start treating sepsis, the primary target-- the first things you should think about fixing-- is the infection itself. If we don't treat the infection, things are going to keep being bad. But even if we figure out the right antibiotic to treat the infection that is the source of the septic shock or the septic inflammation, there are a lot of different conditions that we need to manage. Because the infection itself can lead to fever, breathing difficulties, low blood pressure, high heart rate-- all these kinds of things that are symptoms, but not the cause in themselves. But we still have to manage them somehow so that the patient survives and is comfortable.

So when I say sepsis management, I'm talking about managing such quantities over time-- over a patient's stay in the hospital. So, last time-- again, just to really hammer this in-- we talked about potential outcomes and the choice of a single treatment. So we can think about this in the septic setting as a patient coming in-- or a patient already being in the hospital, presumably-- and is presenting with breathing difficulties. So that means that their blood oxygen will be low because they can't breathe on their own. And we might want to put them on mechanical ventilation so that we can ensure that they get sufficient oxygen.

We can view this as a single choice. Should we put the patient on mechanical ventilation or not? But what we need to take into account here is what will happen after we make that choice. What will be the side effects of this choice going further? Because we want to make sure that the patient is comfortable and in good health throughout their stay.

So today, we will move towards sequential decision making. And in particular, what I alluded to just now is that decisions made in sequence may have the property that choices early on rule out certain choices later. And we'll see an example of that very soon.

And in particular, we'll be interested in coming up with a policy for making decisions repeatedly that optimizes a given outcome-- something that we care about. It could be minimize the risk of death. It could be a reward that says that the vitals of a patients are in the right range. We might want to optimize that. But essentially, think about it now as having this choice of administering a medication or an intervention at any time, t-- and having the best policy for doing so.

OK, I'm going to skip that one. OK, so I mentioned already one potential choice that we might want to make in the management of a septic patient, which is to put them on mechanical ventilation because they can't breathe on their own. A side effect of doing so is that they might suffer discomfort from being intubated. The procedure is not painless, it's not without discomfort.

So something that you might have to do-- putting them on mechanical ventilation-- is to sedate the patient. So this is an action that is informed by the previous action, because if we didn't put the patient on mechanical ventilation, maybe we wouldn't consider them for sedation. When we sedate a patient, we run the risk of lowering their blood pressure. So we might need to manage that, too.

So if their blood pressure gets too low, maybe we need to administer vasopressors, which artificially raise the blood pressure, or fluids or anything else that takes care of this issue. So just think of this as an example of choices cascading, in terms of their consequences, as we roll forward in time.

Ultimately, we will face the end of the patient's stay. And hopefully, we managed the patient in a successful way so that their response or their outcome is a good one. What I'm illustrating here is that, for any one patient in our hospitals or in the health care system, we will only observe one trajectory through these options.

So I will show this type of illustration many times, but I hope that you can realize the scope of the decision space here. Essentially, at any point, we can choose a different action. And usually, the number of decisions that we make in an ICU setting, for example, is much larger than we could ever test in a randomized trial.

Think of all of these different trajectories as being different arms in a randomized controlled trial that you want to compare the effects or the outcomes of. It's infeasible to run such a trial, typically. So one of the big reasons that we are talking about reinforcement learning today and talking about learning policies, rather than causal effects in the setup that we did last week, is because the space of possible action trajectories is so large.

Having said that, we now turn to trying to find, essentially, the policy that picks this orange path here-- that leads to a good outcome. And to reason about such a thing, we need to also reason about what is a good outcome? What is good reward for our agent, as it proceeds through time and makes choices?

Some policies that we produce as machine learners might not be appropriate for a health care setting. We have to somehow restrict ourself to something that's realistic. I won't focus very much on this today. It's something that will come up in the discussion tomorrow, hopefully. And also the notion of evaluating something for use in the health care system will also be talked about tomorrow.

AUDIENCE: Thursday.

FREDRIK D. JOHANSSON: Sorry, Thursday. Next time. OK, so I'll start by just briefly mentioning some success stories. And these are not from the health care setting, as you can guess from the pictures. How many have seen some of these pictures? OK, great-- almost everyone.

Yeah, so these are from various video games-- almost all of them. Well, games anyhow. And these are good examples of when reinforcement learning works, essentially. That's why I use these in this slide here-- because, essentially, it's very hard to argue that the computer or the program that eventually beat Lee Sedol. I think it's in this picture but also, later, Go champions, essentially.

In the AlphaGo picture in the top left, it's hard to argue that they're not doing a good job, because they clearly beat humans here. But one of the things I want you to keep in mind throughout this talk is what is different between these kinds of scenarios? And we'll come back to that later. And what is different to the health care setting, essentially?

So I simply added another example here, that's why I recognize it. So there was recently one that's a little bit closer to my heart, which is AlphaStar. I play StarCraft. I like StarCraft, so it should be on the slide. Anyway, let's move on.

Broadly speaking, these can be summarized in the following picture. What goes into those systems? There's a lot more nuance when it comes to something like Go. But for the purpose of this class, we will summarize them with a slide. So essentially, one of the three quantities that matters for a reinforcement learning is the state of the environment, the state of the game, the state of the patient-- the state of the thing that we want to optimize, essentially.

So in this case, I've chosen Tic-tac-toe here. We have a state which represents the current positions of the circles and crosses. And given that state of the game, my job as a player is to choose one of the possible actions-- one of the free squares to put my cross in. So I'm the blue player here and I can consider these five choices for where to put my next cross. And each of those will lead me to a new state of the game.

If I put my cross over here, that means that I'm now in this box. And I have a new set of actions available to me for the next round, depending on what the red player does. So we have the state, we have the actions, and we have the next state, essentially-- we have a trajectory or a transition of states.

And the last quantity that we need is the notion of a reward. That's very important for reinforcement learning, because that's what's driving the learning itself. We strive to optimize the reward or the outcome of something.

So if we look at the action to the farthest right here, essentially I left myself open to an attack by the red player here, because I didn't put my cross there. Which means that, probably, if the red player is decent, he will put his circle here and I will incur a loss, essentially. So my reward will be negative, if we take positive to be good. And this is something that I can learn from going forward.

Essentially, what I want to avoid is ending up in this state that's shown in the bottom right here. This is the basic idea of reinforcement learning for video games and for anything else. So if we take this board analogy or this example and move to the health care setting, we can think of the state of a patient as the game board or the state of the game. We will always call this St in this talk. The treatments that we prescribe or interventions will be At. And these are like the actions in the game, obviously.

The outcomes of a patient-- could be mortality, could be managing vitals-- will be as the rewards in the game, having lost or won. And then up at the end here, what could possibly go wrong. Well, as I alluded to before, health is not a game in the same sense that a video game is a game. But they share a lot of mathematical structure. So that's why I make the analogy here.

These quantities here-- S, A, and R-- will form something called a decision process. And that's what we'll talk about next. This is the outline for today and Thursday. I won't get to this today, but this is the talks we're considering.

So a decision process is essentially the world that describes the data that we access or the world that we're managing our agent in. Very often, if you've ever seen reinforcement learning taught, you have seen this picture in some form, usually. Sometimes there's a mouse and some cheese and there's other things going on, but you know what I'm talking about. But there are the same basic components.

So there's the concept of an agent-- let's think doctor for now-- that takes actions repeatedly over time. So this t here indicates an index of time and we see that essentially increasing as we spin around this wheel here. We move forward in time. So an agent takes an action and, at any time point, receives a reward for that action. And that would be Rt, as I said before. The environment is responsible for giving that reward.

So for example, if I'm the doctor, I'm the agent, I make an action or an intervention to my patient, the patient will be the environment. And essentially, responses do not respond to my intervention. The state here is the state of the patient, as I mentioned before, for example. But it might also be a state more broadly than the patient, like the settings of the machine that they're attached to or the availability of certain drugs in the hospital or something like that. So we can think a little bit more broadly around the patient, too.

I said partially observed here, in that I might not actually know everything about the patient that's relevant to me. And we will come back a little bit later to that. So there are two different formalizations that are very close to each other, which is when you'd know everything about s and when you don't. We will, for the longest part of this talk, focus on the way I know everything that is relevant about the environment.

OK, to make this all a bit more concrete, I'll return to the picture that I showed you before, but now put it in context of the paper that you read. Was that the compulsory one? The mechanical ventilation? OK, great.

So in this case, they had an interesting reward structure, essentially. The thing that they were trying to optimize was the reward related to the vitals of the patient. But also whether they were kept on mechanical ventilation or not. And the idea of this paper is that you don't want to keep a patient unnecessarily on mechanical ventilation, because it has the side effects that we talked about before.

So at any point in time, essentially, we can think about taking a patient on or off-- and also dealing with the sedatives that are prescribed to them. So in this example, the state that they considered in this application included the demographic information of the patient, which doesn't really change over time. Their physiological measurements, ventilator settings, consciousness level, the dosages of the sedatives they use, which could be an action, I suppose-- and a number of other things. And these are the values that we have to keep track of, moving forward in time. The actions concretely included whether to intubate or extubate the patient, as well as the administer and dosing the sedatives.

So this is, again, an example of a so-called decision process. And essentially, the process is the distribution of these quantities that I've been talking about over time. So we have the states, the actions, and the rewards. They all traverse or they all evolve over time. And the loss of how that happens is the decision process.

I mentioned before that we will be talking about policies today. And typically, there's a distinction between what is called a behavior policy and a target policy-- or there are different words for this. Essentially, the thing that we observe is usually called a behavior policy. By that, I mean if we go to a hospital and watch what's happening there at the moment, that will be the behavior policy. And I will denote that mu.

So that is what we have to learn from, essentially. So decision processes so far are incredibly general. I haven't said anything about what this distribution is like, but the absolutely dominant restriction that people make when they study system processes is to look at Markov decision processes. And these have a specific conditional independent structure that I will illustrate in the next slide-- well, I'll just define it mathematically here.

It says, essentially, that all of the quantities that we care about-- the states. I guess that should say state. Rewards and the actions only depend on the most recent state in action.

If we observe an action taken by a doctor in the hospital, for example-- to make a mark of assumption, we'd say that this doctor did not look at anything that happened earlier in time or any other information than what is in the state variable that we observe at that time. That is the assumption that we make. Yeah?

AUDIENCE: Is that an assumption you can make for a health care? Because in the end, you don't have access to the real state, but only about what's measured about the state in health care.

FREDRIK D. JOHANSSON: It's a very good question. So the nice thing in terms of inferring causal quantities is that we only need the things that were used to make the decision in the first place. So the doctor can only act on such information, too. Unless we don't record everything that the doctor knows-- which is also the case. So that is something that we have to worry about for sure. Another way to lose information, as I mentioned, that is relevant for this is if we look to-- What's the opposite of far?

AUDIENCE: Near.

FREDRIK D. JOHANSSON: Too near back in time, essentially. So we don't look at the entire history of the patient. And when I say St here, it doesn't have to be the instantaneous snapshot of a patient. We can also Include history there. Again, we'll come back to that a little later.

OK, so the Markov assumption essentially looks like this. Or this is how I will illustrate, anyway. We have a sequence of states here that evolve over time. I'm allowing myself to put some dots here, because I don't want to draw forever. But essentially, you could think of this pattern repeating-- where the previous state goes into the next state, the action goes into the next state, and the action and state goes in through the reward.

This is the world that we will live in for this lecture. Something that's not allowed under the mark of assumption is an edge like this, which says that an action at an early time influences an action at a later time. And specifically, it can't do so without passing through a state, for example. It very well can have an influence on At by this trajectory here, but not directly. That that's the Markov assumption in this case.

So you can see that if I were to draw the graph of all the different measurements that we see during a state, essentially there are a lot of errors that I could have had in this picture that I don't have. So it may seem that the Markov assumption is a very strong one, but one way to ensure that the Markov assumption is more likely is to include more things in your state, including summaries of the history, et cetera, that I mentioned before. An even stronger restriction of decision processes is to assume that the states over time are themselves independent.

So this goes by different names-- sometimes under the name contextual bandits. But the bandits part of that itself is not so relevant here. So let's not go into that name too much. But essentially, what we can say is here, the state at a later time point is not influenced directly by the state at a previous time point, nor the action of the previous time point.

So if you remember what you did last week, this looks like basically T repetitions of the very simple graph that we had for estimating potential outcomes. And that is indeed mathematically equivalent. If we assume that this S here represents the state of a patient and all patients are drawn from some sum process, essentially. So that S0, 1, et cetera, up to St are all i.i.d. draws of the same distribution.

Then we have, essentially, a model for t different patients with a single time step or single action, instead of them being dependent in some way. So we can see that by going backwards through my slides, this is essentially what we had last week. And we just have to add more arrows to get to whatever we have this week, which indicates that last week was a special case of this-- just as David said before. It also hints at the reinforcement learning problem being more complicated than the potential outcomes problem. And we'll see more examples of that later.

But, like with causal effect estimation that we did last week, we're interested in the influences of just a few variables, essentially. So last time we studied the effect of a single treatment choice. And in this case, we will study the influence of these various actions that we take along the way. That will be the goal. And it could be either through an immediate effect on the immediate reward or it can be through the impact that an action has on the state trajectory itself.

I told you about the world now that we live in. We have these Ss and As and Rs. And I haven't told you so much about the goal that we're trying to solve or the problem that we're trying to solve. Most RL-- or reinforcement and learning-- is aimed at optimizing the value of a policy or finding a policy that has a good return, a good sum of rewards. There are many names for this, but essentially a policy that does well.

The notion of well that we will be using in this lecture is that of a return. So the return at a time step t, following the policy, pi, that I had before, is the sum of the future rewards that we see if we were to act according to that policy.

So essentially, I stop now. I ask, OK, if I keep on doing the same as I've done through my whole life-- maybe that was a good policy. I don't know. And keep going until the end of time, how well will I do? What is the sum of those rewards that I get, essentially? That's the return.

The value is the expectation of such things. So if I'm not the only person, but there is the whole population of us, the expectation over that population is the value of the policy. So if we take patients as a better analogy than my life, maybe, the expectation is over patients. If we fact on every patient in our population the same way-- according to the same policy, that is-- what is the expected return over those patients?

So as an example, I drew a few trajectories again, because I like drawing. And we can think about three different patients here. They start in different states. And they will have different action trajectories as a result.

So we're treating them with the same policy. Let's call it pi. But because they're in different states, they will have different actions at the same times. So here we take a 0 action, we go down. Here, we take a 0 action, we go down. That's what that means here.

The specifics of this is not so important. But what I want you to pay attention to is that, after each action, we get a reward. And at the end, we can sum those up and that's our return. So each patient has one value for their own trajectory. And the value of the policy is then the average value of such trajectories.

So that is what we're trying to optimize. We have now a notion of good and we want to find a pi such that V pi up there is good. That's the goal. So I think it's time for a bit of an example here. I want you to play along in a second.

You're going to solve this problem. It's not a hard one. So I think you'll manage. I think you'll be fine. But this is now yet another example of a world to be in.

This is the robot in a room. And I've stolen this slide from David, who stole it from Peter Bodik. Yeah, so credits to him. The rules of this world says the following-- if you tell the robot, who is traversing this set of tiles here-- if you tell the robot to go up, there's a chance he doesn't go up, but goes somewhere else. So we have the stochastic transitions, essentially.

If I say up, he goes up with point a probability and somewhere else with uniform probability, say. So 0.8 up and then 0.2-- this is the only possible direction to go in if you start here. So 0.2 in that way. There's a chance you move in the wrong direction is what I'm trying to illustrate here.

There's no chance that they're going in the opposite direction. So if I say right here, it can't go that way. The rewards in this game is plus 1 in the green box up there, minus 1 in the box here. And these are also terminal states. So I haven't told you what that is, but it's essentially a state in which the game ends. So once you get to either plus 1 or minus 1, the game is over.

For each step that the robot takes, it incurs 0.04 negative reward. So that says, essentially, that if you keep going for a long time, your reward would be bad. The value of the policy will be bad. So you want to be efficient.

So basically, you can figure out-- you want to get to the green thing, that's one part of it. But you also want to do it quickly. So what I want you to do now is to essentially figure out what is the best policy, in terms of in which way should the arrows point in each of these different boxes?

Fill in the question mark with an arrow pointing in some direction. We know the different transitions will be stochastic, so you might need to take that into account. But essentially, figure out how do I have a policy that gives me the biggest expected reward? And I'll ask you in a few minutes if one of you is brave enough to put it on the board or something like that.

AUDIENCE: We start the discount over time?

FREDRIK D. JOHANSSON: There's no discount.

AUDIENCE: Can we talk to our neighbor?

FREDRIK D. JOHANSSON: Yes. It's encouraged.

[INTERPOSING VOICES]

FREDRIK D. JOHANSSON: So I had a question. What is the action space? And essentially, the action space is always up, down, left, or right, depending on if there's a wall or not. So you can't go right here, for example.

AUDIENCE: You can't go left either.

FREDRIK D. JOHANSSON: You can't go left, exactly. Good point. So each box at the end, when you're done, should contain an arrow pointing in some direction. All right, I think we'll see if anybody has solved this problem now. Who thinks they have solved it? Great. Would you like to share your solution?

AUDIENCE: Yeah, so I think it's going to go up first.

FREDRIK D. JOHANSSON: I'm going to try and replicate this. Ooh, sorry about that. OK, you're saying up here?

AUDIENCE: Yeah. The basic idea is you want to reduce the chance that you're ever adjacent to the red box. So just do everything you can to stay far from it. Yeah, so attempt to go up and then once you eventually get there, you just have to go right.

FREDRIK D. JOHANSSON: OK. And then?

AUDIENCE: [INAUDIBLE].

FREDRIK D. JOHANSSON: OK. So what about these ones? This is also part of the policy, by the way.

AUDIENCE: I hadn't thought about this.

FREDRIK D. OK.

JOHANSSON:

AUDIENCE: But those, you [INAUDIBLE], right?

FREDRIK D. JOHANSSON: No.

AUDIENCE: Minus 0.04.

FREDRIK D. JOHANSSON: So discount usually means something else. We'll get to that later. But that is a reward for just taking any step. If you move into a space that is not terminal, you incur that negative reward.

AUDIENCE: So if you keep bouncing around for a really long time, you incur a long negative reward.

FREDRIK D. JOHANSSON: If we had this, there's some chance I'd never get out of all this. And very little chance of that working out. But it's a very bad policy, because you keep moving back and forth. All right, we had an arm somewhere. What should I do here?

AUDIENCE: You could take a vote.

FREDRIK D. JOHANSSON: OK. Who thinks right? Really? Who thinks left? OK, interesting. I don't actually remember. Let's see. Go ahead.

AUDIENCE: I was just saying, that's an easy one.

FREDRIK D. JOHANSSON: Yeah, so this is the part that we already determined. If we had deterministic transitions, this would be great, because we don't have to think about the other ones. This is what Peter put on the slide. So I'm going to have to disagree with the vote there, actually. It depends, actually, heavily on the minus 0.04.

So if you increase that by a little bit, you might want to go that way instead. Or if you decrease-- I don't remember. Decrease, exactly. And if you increase it, you might get something else. It might actually be good to terminate.

So those details matter a little bit. But I think you've got the general idea. And especially I like that you commented that you want to stay away from the red one, because if you look at these different paths. You go up there and there-- they have the same number of states, but there's less chance you end up in the red box if you take the upper route. Great.

So we have an example of a policy and we have an example of a decision process. And things are working out so far. But how do we do this? As far as the class goes, this was a blackbox experiment. I don't know anything about how you figured that out.

So reinforcement learning is about that-- reinforcement learning is try and come up with a policy in a rigorous way, hopefully-- ideally. So that would be the next topic here. Up until this point, are there any questions that you've been dying to ask, but haven't?

AUDIENCE: I'm curious how much behavioral biases could play into the first Markov assumption? So for example, if you're a clinician who's been working for 30 years and you're just really used to giving a certain treatment. An action that you gave in the past-- that habit might influence an action in the future. And if that is a worry, how one might think about addressing it.

FREDRIK D. JOHANSSON: Interesting. I guess it depends a little bit on how it manifests, in that if it also influenced your most recent action, maybe you have an observation of that already in some sense. It's a very broad question. What effect will that have? Did you have something specific in mind?

AUDIENCE: I guess I was just wondering if it violated that assumption, that an action of the past influenced an action--

FREDRIK D. JOHANSSON: Interesting. So I guess my response there is that the action didn't really depend on the choice of action before, because the policy remained the same. You could have a bias towards an action without that being dependent on what you gave as action before, if you know what I mean.

Say my probability of giving action one is 1, then it doesn't matter that I give it in the past. My policy is still the same. So, not necessarily. It could have other consequences. We might have reason to come back to that question later. Yup.

AUDIENCE: Just practically, I would think that a doctor would want to be consistent. And so you wouldn't, for example, want to put somebody on a ventilator and then immediately take them off and then immediately put them back on again. So that would be an example where the past action influences what you're going to do.

FREDRIK D. JOHANSSON: Completely, yeah. I think that's a great example. And what you would hope is that the state variable in that case includes some notion of treatment history. That's what your job is then. So that's why state can be somewhat misleading as a term-- at least for me, I'm not American or English-speaking. But yeah, I think of it as too instantaneous sometimes.

So we'll move into reinforcement learning now. And what I had you do on the last slide-- well, I don't know which method you use, but most likely the middle one. There are three very common paradigms for reinforcement learning. And they are essentially divided by what they focus on modeling. Unsurprisingly, model-based RL focused on-- well, it has some sort of model in it, at least.

What you mean by model in this case is a model of the transitions. So what state will I end up in, given the action in the state I'm in at the moment? So model-based RL tries to essentially create a model for the environment or of the environment. There are several examples of model-based RL. One of them is G-computation, which comes out of the statistic literature, if you like. And MDPs are essentially-- that's a Markov decision process, which is essentially trying to estimate the whole distribution that we talked about before.

There are various ups and downsides of this. We won't have time to go into all of these paradigms today. We will actually focus only on value-based RL today. Yeah, you can ask me offline if you are interested in model-based RL.

The rightmost one here is policy-based RL, where you essentially focus only on modeling the policy that was used in the data that you observed. And the policy that you want to essentially arrive at. So you're optimizing a policy and you are estimating a policy that was used in the past. And the middle one focuses on neither of those and focuses on only estimating the return-- that was the G. Or the reward function as a function of your actions and states.

And it's interesting to me that you can pick any of the variables-- A, S, and R-- and model those. And you can arrive at something reasonable in reinforcement learning. This one is particularly interesting, because it doesn't try to understand how do you arrive at a certain return based on the actions in the states? It's just optimize the policy directly. And it has some obvious-- well, not obvious, but it has some downsides, not doing that.

OK, anyway, we're going to focus on value-based RL. And the very dominant instantiation of value-based RL is Q-learning. I'm sure you've heard of it. It is what drove the success stories that I showed before, the goal in the StarCraft and everything. G-estimation is another example of this, which, again, has come from the statistic literature. But we'll focus on Q-learning today.

So Q-learning is an example of dynamic programming, in some sense. That's how it's usually explained. And I just wanted to check-- how many have heard the phrase dynamic programming before? OK, great. So I won't go into details of dynamic programming in general. But the general idea is one of recursion.

In this case, you know something about what is a good terminal state. And then you want to figure out how to get there and how to get to the state before that and the state before that and so on. That is the recursion that we're talking about. The end state that is the best here is fairly obvious-- that is the plus 1 here.

The only way to get there is by stopping here first, because you can't move from here since it's a terminal state. Your only bet is that one. And then we can ask what is the best way to get to 3, 1? How do we get to the state before the best state? Well, we can say that one way is go from here. And one way from here.

And as we got from the audience before, this is a slightly worse way to get there then from there, because here we have a possibility of ending up in minus 1. So then we recurse further and essentially, we end up with something like this that says-- or what I tried to illustrate here is that the green boxes-- I'm sorry for any colorblind members of the audience, because this was a poor choice of mine. Anyway, this bottom side here is mostly red and this is mostly green. And you can follow the green color here, essentially, to get to the best end state.

And what I used here to color this in is this idea of knowing how good a state is, depending on how good the state after that state is. So I knew that plus 1 is a good end state over there. And that led me to recurse backwards, essentially.

So the question, then, is how do we know that that state over there is a good one? When we have it visualized in front of us, it's very easy to see. And it's very easy because we know that plus 1 is a terminal state here. It ends there, so those are the only states we need to consider in this case. But more in general, how do we learn what is the value of a state? That will be the purpose of Q-learning.

If we have an idea of what is a good state, we can always do that recursion that I explained very briefly. You find a state that has the high value and you figure out how to get there. So we're going to have to define now what I mean by value. I've used that word a few different times. I say recall here, but I don't know if I actually had it on a slide before.

So let's just say this is the definition of value that we will be working with. I think I had it on a slide before, actually. This is the expected return. Remember, this G here was the sum of rewards going into the future, starting at time, t. And the value, then, of this state is the expectation of such returns.

Before, I said that the value of an policy was the expectation of returns, period. And the value of a state and the policy is the value of that return starting in a certain state. We can stratify this further if we like and say that the value of a state action pair is the expected return, starting in a certain state and taking an action, a. And after that, following the policy, pi.

This would be the so-called Q value of a state-action pair-- s, a. And this is where Q-learning gets its name. So Q-learning attempts to estimate the Q function-- the expected return starting in a state, s, and taking action, a-- from data.

The Q-learning is also associated with a deterministic policy. So the policy and the Q function go together in this specific way. If we have a Q function, Q, that tries to estimate the value of a policy, pi, the pi itself is the arg max according to that Q. It sounds a little recursive, but hopefully it will be OK.

Maybe it's more obvious if we look here. So Q, I said before, was the value of starting an s, taking action, a, and then following policy, pi. This is defined by the decision process itself.

The best pi, the best policy, is the one that has the highest Q. And this is what we call a Q-star. Well, that is not what we call Q-star, that is what we call little q-star. Q-star, the best estimate of this, is obviously the thing itself.

So if you can find a good function that assigns a value to a state-action pair, the best such function you can get is the one that is equal to little q-star. I hope that wasn't too confusing. I'll show on the next slide why that might be reasonable.

So Q-learning is based on a general idea from dynamic programming, which is the so-called Bellman question. There we go. This is an instantiation of Bellman optimality, which says that the best state-action value function has the property that it is equal to the immediate reward of taking action, a, and state, s, plus this, which is the maximum Q value for the next state.

So we're going to stare at this for a bit, because there's a bit here to digest. Remember, q-star assigns a value to any state action pair. So we have q-star here, we have q-star here. This thing here is supposed to represent the value going forward in time after I've made this choice, action, a, and state, s.

If I have a good idea of how good it is to take action, a, instead of s, it should both incorporate the immediate reward that I get-- that's RT-- and how good that choice was going forward. So think about mechanical ventilation, as I said before. If we put a patient on mechanical ventilation, we have to do a bunch of other things after that. If none of those other things lead to a good outcome, this part will be low. Even if the immediate return is good.

So for the optimal q-star, this quantity holds. We know that-- we can prove that. So the question is how do we find this thing? How do we find q-star? Because q-star is not only the thing that gives you the optimal policy-- it also satisfied this equality.

This is not true for every Q function, but it's true for the optimal one. Questions? If you haven't seen this before, it might be a little tough to digest.

Is the notation clear? Essentially, here you have the state that you are arriving at the next time. A prime is the parameter of this here, or the argument to this. You're taking the best possible q-star value and then state that you arrive at after. Yeah, go ahead.

AUDIENCE: Can you instantiate an example you have on the board?

FREDRIK D. JOHANSSON: Yes. Actually, I might do a full example of Q-learning in a second. Yes, I will. I'll get to that example then.

Yeah, I was debating whether to do that. It might take some time, but it could be useful. So where are we?

So what I showed you before-- the Bellman inequality. We know that this holds for the optimal thing. And if there is a quality that is true at an optimum, one general idea in optimization is this so-called fixed point iteration that you can do to arrive there. And that's essentially what we will do to get to a good Q.

So a nice thing about Q-learning is that if your states and action spaces are small and discrete, you can represent the Q function as a table. So all you have to keep track of is, how good is the certain action in a certain state? Or all actions in all states, rather? So that's what we did here. This is a table.

I've described to you the policy here, but what we'll do next is to describe the value of each action. So you can think of a value of taking the right one, bottom, top, and left, essentially.

Those will be the values that we need to consider. And so what Q-learning can do with discrete states is to essentially start from somewhere, start from some idea of what Q is-- could be random, could be 0.

And then repeat the following fixed-point iteration, where you update your former idea of what Q should be, with its current value plus essentially a mixture of the immediate reward for taking action, At, in that state, and the future reward, as judged by your current estimate of the Q function. So we'll do that now in practice. Yeah.

AUDIENCE: Throughout this, where are we getting the transition probabilities or the behavior of the game?

FREDRIK D. JOHANSSON: So they're not used here, actually. A value-based RL-- I didn't say that explicitly, but they don't rely on knowing the transition probabilities. What you might ask is where do we get the S and the As and the Rs from? And we'll get to that.

How do we estimate these? We'll get to that later. Good question, though. I'm going to throw a very messy slide at you. Here you go. A lot of numbers.

So what I've done now here is a more exhaustive version of what I put on the board. For each little triangle here represents the Q value for the state-action pair. So this triangle is, again, for the action right if you're in this state.

So what I've put on the first slide here is the immediate reward of each action. So we know that any step will cost us minus 0.04. So that's why there's a lot of those here. These white boxes here are not possible actions. Up here, you have a 0.96, because it's 1, which is the immediate reward of going right here, minus 0.04.

These two are minus 1.04 for the same reason-- because you arrive in minus 1. OK, so that's the first step and the second step done. We initialize Qs to be 0. And then we picked these two parameters of the problem, alpha and gamma, to be 1. And then we did the first iteration of Q-learning, where we set the Q to be the old version of Q, which was 0, plus alpha times this thing here.

So Q was 0, that means that this is also 0. So the only thing we need to look at is this thing here. This also is 0, because the Qs for all states were 0, so the only thing we end up with is R. And that's what populated this table here.

Next timestep-- I'm doing Q-learning now in a way where I update all the state-action pairs at once. How can I do that? Well, it depends on the question I got there, essentially. What data do I observe? Or how do I get to know the rewards of the S&A pairs? We'll come back to that.

So in the next step, I have to update everything again. So it's the previous Q value, which was minus 0.04 for a lot of things, then plus the immediate reward, which was this RT. And I have to keep going. So the dominant thing for the table this time was that the best Q value for almost all of these boxes was minus 0.04.

So essentially I will add the immediate reward plus that almost everywhere. What is interesting, though, is that here, the best Q value was 0.96. And it will remain so. That means that the best Q value for the adjacent states-- we look at this max here and get 0.96 out. And then add the immediate reward.

Getting to here gives me 0.96 minus 0.04 for the immediate reward. And now we can figure out what will happen next. These values will spread out as you go further away from the plus 1. I don't think we should go through all of this, but you get a sense, essentially, how information is moved from the plus 1 and away. And I'm sure that's how you solved it yourself, in your head. But this makes it clear why you can do that, even if you don't know where the terminal states are or where the value of the state-action pairs are.

AUDIENCE: Doesn't this calculation assume that if you want to move in a certain direction, you will move in that direction?

FREDRIK D. JOHANSSON: Yes. Sorry. Thanks for reminding me. That should have been in the slide, yes. Thank you.

I'm going to skip the rest of this. I hope you forgive me. We can talk more about it later.

Thanks for reminding me, Pete, there, that one of the things I exploited here was that I assume just deterministic transitions. Another thing that I relied very heavily on here is that I can represent this Q function as a table. I drew all these boxes and I filled the numbers in. That's easy enough.

But what if I have thousands of states and thousands of actions? That's a large table. And not only is it a large table for me to keep in memory-- it's also very bad for me statistically. If I want to observe anything about a state-action pair, I have to do that action in that state.

And if you think about treating patients in a hospital, you're not going to try everything in every state, usually. You're also not going to have infinite numbers of patients. So how do you figure out what is the immediate reward of taking a certain action in a certain state? And this is where a function approximation comes in.

Essentially, if you can't represent your data set table, either for statistical reasons or for memory reasons, let's say, you might want to approximate the Q function with a parametric or with a non-parametric function. And this is exactly what we can do. So we can draw now an analogy to what we did last week. I'm going to come back to this, but essentially instead of doing this fixed-point iteration that we did before, we will try and look for a function Q theta that is equal to R plus gamma max Q.

Remember before, we had the Bellman inequality? We said that q-star S, A is equal to R S, A, let's say, plus gamma max A prime q star S prime A prime, where S prime is the state we get to after taking action A in state S. So the only thing I've done here is to take this equality and make it instead a loss function on the violation of this equality. So by minimizing this quantity, I will find something that has approximately the Bellman equality that we talked about before.

This is the idea of fitted Q-learning, where you substitute the tabular representation with the function approximations, essentially. So just to make this a bit more concrete, we can think about the case where we have only a single step. There's only a single action to make, which means that there is no future part of this equation here. This part goes away, because there's only one stage in our trajectory. So we have only the immediate reward. We have only the Q function.

Now, this is exactly a regression equation in the way that you've seen it when estimating potential outcomes. RT here represents the outcome of doing action A and state S. And Q here will be our estimate of this RT.

Again, I've said this before-- if we have a single time point in our process, the problem reduces to estimating potential outcomes, just the way we saw it last time. We have curves that correspond outcomes under different actions. And we can do regression adjustment, trying to find an F such that this quantity is small so that we can model each different potential outcomes. And that's exactly what happens with the fitted Q iteration if you have a single timestep, too.

So to make it even more concrete, we can say that there's some target value, G hat, which represents the immediate reward and the future rewards that is the target of our regression.

And we're fitting some function to that value.

So the question we got before was how do I know the transition matrix? How do I get any information about this thing? I say here on the slide that, OK, we have some target that's R plus future Q values. We have some prediction and we have an expectation of our transitions here.

But how do I evaluate this thing? The transitions I have to get from somewhere, right? And another way to say that is what are the inputs and the outputs of our regression? Because when we estimate potential outcomes, we have a very clear idea of this. We know that y, the outcome itself, is a target. And the input is the covariates, x.

But here, we have a moving target, because this Q hat, it has to come from somewhere, too. This is something that we estimate as well. So usually what happens is that we alternate between updating this target, Q, and Q theta. So essentially, we copy Q theta to become our new Q hat and we iterate this somehow. But I still haven't told you how to evaluate this expectation.

So usually in RL, there are a few different ways to do this. And either depending on where you coming from, essentially, these are varyingly viable. So if we look back at this thing here, it relies on having tuples of transitions-- the state, the action, the next state, and the reward that I got. So I have to somehow observe those. And I can obtain them in various ways.

A very common one when it comes to learning to play video games, for example, is that you do something called on-policy exploration. That means that you observe data from the policy that you're currently optimizing. You just play the game according to the policies that you have at the moment. And the analogy in health care would be that you have some idea of how to treat patients and you just do that and see what happens. That could be problematic, especially if you've got that policy-- if you randomly initialized it or if you got it for some somewhere very suboptimal.

A different thing that we're more, perhaps, comfortable with in health care, in a restricted setting, is the idea of a randomized trial, where, instead of trying out some policy that you're currently learning, you decide on a population where it's OK to flip a coin, essentially, between different actions that you have. The difference between the sequential setting and the one-step setting is that now we have to randomize a sequence of actions, which is a little bit unlike the clinical trials that you have seen before, I think.

The last one, which is the most studied one when it comes to practice, I would say, is the one that we talk about this week-- is off-policy evaluation or learning, in which case you observe health care records, for example. You observe registries. You observe some data from the health care system where patients have already been treated and you try to extract a good policy based on that information.

So that means that you see these transitions between state and action and the next state and the reward. You see that based on what happened in the past and you have to figure out a pattern there that helps you come up with a good action or a good policy. So we'll focus on that one for now.

The last part of this talk will be about, essentially, what we have to be careful with when we learn with off-policy data. Any questions up until this point? Yeah.

AUDIENCE: So if [INAUDIBLE] getting there for the [INAUDIBLE], are there any requirements that has to be met by [INAUDIBLE], like how we had [INAUDIBLE] and cause inference?

FREDRIK D. JOHANSSON: Yeah, I'll get to that on the next set of slides. Thank you. Any other questions about the Q-learning part? A colleague of mine, Rahul, he said-- or maybe he just paraphrased it from someone else. But essentially, you have to see RL 10 times before you get it, or something to that effect. I had the same experience. So hopefully you have questions for me after.

AUDIENCE: Human reinforcement learning.

FREDRIK D. JOHANSSON: Exactly. But I think what you should take from the last two sections, if not how to do Q-learning in detail, because I glossed over a lot of things. You should take with you the idea of dynamic programming and figuring out, how can I learn about what's good early on in my process from what's good late? And the idea of moving towards a good state and not just arriving there immediately. And there are many ways to think about that.

OK, we'll move on to off-policy learning. And again, the set-up here is that we receive trajectories of patient states, actions, and rewards from some source. We don't know what these sources necessarily-- well, we probably know what the source is. But we don't know how these actions were performed, i.e., we don't know what the policy was that generated these trajectories. And this is the same set-up as when you estimated causal effects last week, to a large extent.

We say that the actions are drawn, again, according to some behavior policy unknown to us. But we want to figure out what is the value of a new policy, pi. So when I showed you very early on-- I wish I had that slide again.

But essentially, a bunch of patient trajectories and some return. Patient trajectories, some return. The average of those, that's called a value. If we have trajectories according to a certain policy, that is the value of that policy-- the average of these things. But when we have trajectories according to one policy and want to figure out the value of another one, that's the same problem as the covariate adjustment problem that you had last week, essentially. Or the confounding problem, essentially.

The trajectories that we draw are biased according to the policy of the clinician that created them. And we want to figure out the value of a different policy. So it's the same as the confounding problem from the last time.

And because it is the same as the confounding from last time, we know that this is at least as hard as doing that. We have confounding-- I already alluded to variance issues. And you mentioned overlap or positivity as well. And in fact, we need to make the same assumptions but even stronger assumptions for this to be possible.

These are sufficient conditions. So, under very certain circumstances, you don't need them. I should say, these are fairly general assumptions that are still strict-- that's how I should put it.

So last time, we looked at something called strong ignorability. I realized the text is pretty small in here. Can you see in the back? Is that OK? OK, great.

So strong ignorability said that the potential outcomes-- Y0 and Y1-- are conditionally independent of the treatment, t, given the set of variables, x, or the variable, x. And that's saying that it doesn't matter if we know what treatment was given. We can figure out just based on x what would happen under either treatment arm, where we should treat this patient, with t equals 0, t equals 1.

We had an idea of-- or an assumption of-- overlap, which says that any treatment could be observed in any state or any context, x. That's what that means. And that is only to ensure that we can estimate at least a conditional average treatment effect at x. And if we want to estimate the average treatment effect in a population, we would need to have that for every x in that population.

So what happens in the sequential case is that we need even stronger assumptions. There's some notation I haven't introduced here and I apologize for that. But there's a bar here over these Ss and As-- I don't know if you can see it. That usually indicates in this literature that you're looking at the sequence, up to the index here. So all the states up until t have observed and all the actions up until t minus 1.

So in order for the best policy to be identifiable-- or the value of a positive to be identifiable-- we need this strong condition. So the return of a policy is independent of the current action, given everything that happened in the past. This is weaker than the Markov assumption, to be clear, because there, we said that anything that happens in the future is conditionally independent, given the current state. So this is weaker, because we now just need to observe something in the history. We need to observe all confounders in the history, in this instance.

We don't need to summarize them in S. And we'll get back to this in the next slide. Positivity is the real difficult one, though, because what we're saying is that at any point in the trajectory, any action should be possible in order for us to estimate the value of any possible policy. And we know that that's not going to be true in practice.

We're not going to consider every possible action at every possible point in the health care setting. There's just no way. So what that tells us is that we can't estimate the value of every possible policy. We can only estimate the value of policies that are consistent with the support that we do have.

If we never see action 4 at time 3, there's no way we can learn about a policy that does that-- that takes action 4 at time 3. That's what I'm trying to say. So in some sense, this is stronger, just because of how sequential settings work. It's more about the application domain than anything, I would say.

In the next set of slides, we'll focus on sequential randomization or sequential ignorability, as it's sometimes called. And tomorrow, we'll talk a little bit about the statistics involved in or resulting from the positivity assumption and things like importance weighting, et cetera. Did I say tomorrow? I meant Thursday.

So last recap on the potential outcome story. This is a slide-- I'm not sure if he showed this one, but it's one that we used in a lot of talks. And it, again, just serves to illustrate the idea of a one-timestep decision.

So we have here, Anna. A patient comes in. She has high blood sugar and some other properties. And we're debating whether to give her medication A or B. And to do that, we want to figure out what would be her blood sugar under these different choices a few months down the line?

So I'm just using this here to introduce you to the patient, Anna. And we're going to talk about Anna a little bit more. So treating Anna once, we can represent as this causal graph that you've seen a lot of times now. We had some treatment, A, we had some state, S, and some outcome, R. We want to figure out the effect of this A on the outcome, R.

Ignorability in this case just says that the potential outcomes under each action, A, is conditionally independent of A, given S. And so we know that ignorability and overlap is sufficient conditions for identification of this effect. But what happens now if we add another time point? OK, so in this case, if I have no extra arrows here-- I just have completely independent time points-- ignorability clearly still holds. There's no links going from A to R, there's no from S to R, et cetera. So ignorability is still fine.

If Anna's health status in the future depends on the actions that I take now, here, then the situation is a little bit different. So this is now not in the completely independent actions that I make, but the actions here influence the state in the future. So we've seen this. This is a Markov decision process, as you've seen before. This is very likely in practice.

Also, if Anna, for example, is diabetic, as we saw in the example that I mentioned, it's likely that she will remain so. This previous state will influence the future state. These things seem very reasonable, right? But now I'm trying to argue about the sequential ignorability assumption. How can we break that? How can we break ignorability when it comes to the sequential, say?

If you have an action here-- so the outcome at a later point depends on an earlier choice. That might certainly be the case, because we could have a delayed effect of something. So if we measure, say, a lab value which could be in the right range or not, it could very well depend on medication we gave a long time ago. And it's also likely that the reward could depend on a state which is much earlier, depending on what we include in that state variable. We already have an example, I think, from the audience on that.

So actually, ignorability should have a big red cross over it, because it doesn't hold there. And it's luckily on the next slide. Because there are even more errors that we can have, conceivably, in the medical setting. The example that we got from Pete before was, essentially, that if we've tried an action previously, we might not want to try it again. Or if we knew that something worked previously, we might want to do it again.

So if we had a good reward here, we might want to do the same thing twice. And this arrow here says that if we know that a patient had a symptom earlier on, we might want to base our actions on it later. We've known that the patient had an allergic reaction at some point, for example. We might not want to use that medication at a later time.

AUDIENCE: But you can always put everything in a state.

FREDRIK D. JOHANSSON: Exactly. So this depends on what you put in the state. This is an example where I should introduce these arrows to show that, if I haven't got that information here, then I introduce this dependence. So if I don't have the information about allergic reaction or some symptom before in here, then I have to do something else.

So exactly that is the point. If I can summarize history in some good way-- if I can compress all of these four variables into some variable age standard for the history, then I have ignorability, with respect to that history, H. This is your solution and it introduces a new problem, because history is usually a really large thing. We know that history grows with time, obviously. But usually we don't observe patients for the same number of time points.

So how do we represent that for a program? How do we represent that to a learning algorithm? That's something we have to deal with. You can pad history with 0s, et cetera, but if you keep every timestep and repeat every variable in every timestep, you get a very large object. That might introduce statistical problems, because now you have much more variance if you have new variables, et cetera.

So one thing that people do is that they look some amount of time backwards-- so instead of just looking at one timestep back, you now look at a length k window. And your state essentially grows by a factor, k. And another alternative is to try and learn a summary function. Learn some function that is relevant for predicting the outcome that takes all of the history into account, but has a smaller representation than just t times the variables that you have.

But this is something that needs to happen, usually. Most health care data, in practice-- you have to make choices about this. I just want to stress that that's something you really can't avoid.

The last point I want to make is that unobserved confounding is also a problem that is not avoidable just due to summarizing history. We can introduce new confounding. That is a problem, if we don't summarize history well. But we can also have unobserved confounders, just like we can in the one-step setting.

One example is if we have an unobserved confounded in the same way as we did before. It impacts both the action at time 1 and the reward at time 1. But of course, now we're in the sequential setting. The confounding structure could be much more complicated.

We could have a confounder that influences an early action and a late reward. So it might be a little harder for us to characterize what is the set of potential confounders? So I just wanted to point that out and to reinforce that this is only harder than the one-step setting.

So we're wrapping up now. I just want to end on a point about the games that we looked at before. One of the big reasons that these algorithms were so successful in playing games was that we have full observability in these settings. We know everything from the game board itself-- when it comes to Go, at least. We can debate that when it comes to the video games.

But in Go, we have complete observability of the board. Everything we need to know for an optimal decision is there at any time point. Not only can we observe it through the history, but in the case of Go, you don't even need to look at history. We certainly have Markov dynamics with respect to the board itself.

You don't ever have to remember what was a move earlier on, unless you want to read into your opponent, I suppose. But that's a game theoretic notion we're not going to get into here. But more importantly, we can explore the dynamics of these systems almost limitlessly, just by simulation and self-play. And that's true regardless if you have full observability or not-- like in StarCraft, you might not have full observability. But you can try your things out endlessly.

And contrast that with having, I don't know, 700 patients with rheumatoid arthritis or something like that. Those are the samples you have. You're not going to get new ones. So that is an amazing obstacle for us to overcome if we want to do this in a good way.

The current algorithms are really inefficient with the data that they use. And that's why this limitless exploration or simulation has been so important for these games. And that's also why the games are the success stories of this.

A last point is that typically for these settings that I put here, we have no noise, essentially. We get perfect observations of actions and states and outcomes and everything like that. And that's really true in any real-world application.

All right. I'm going to wrap up. Tomorrow-- nope, Thursday, David is going to talk about more explicitly if we want to do this properly in health care, what's going to happen? We're going to have a great discussion, I'm sure, as well. So don't mind the slide. It's Thursday. All right. Thanks a lot.

[APPLAUSE]