DAVID SONTAG: So we're done with our segment on causal inference and reinforcement learning. And for the next week, today and Tuesday's lecture, we'll be talking about disease progression modeling and disease subtyping. This is, from my perspective, a really exciting field.

It's one which has really a richness of literature going back to somewhat simple approaches from a couple of decades ago up to some really state of the art methods, including one which is in one of your readings for today's lecture. And I could spent a few weeks just talking about this topic. But instead, since we have a lot to cover in this course, what I'll do today is give you a high-level overview of one approach to try to think through these questions.

The methods in today's lecture will be somewhat simple. They're meant to illustrate how simple methods can go a long way. And they're meant to illustrate, also, how one could learn something really significant about clinical outcomes and about predicting these progression from these simple methods. And then in Tuesday's lecture, I'll ramp it up quite a bit. And I'll talk about several more elaborate approaches towards this problem, which tackle some more substantial problems that we'll really elucidate at the end of today's lecture.

So there's three types of questions that we hope to answer when studying disease progression modeling. At a high level, I want you to think about this type of picture and have this in the back of your head throughout today and Tuesday's lecture. What you're seeing here is a single patient's disease trajectory across time. On the x-axis is time. On the y-axis is some measure of disease burden.

So for example, you could think about that y-axis as summarizing the amount of symptoms that a patient is reporting or the amount of pain medication that they're taking, or some measure of what's going on with them. And initially, that disease burden might be somewhat low, and maybe even the patient's in an undiagnosed disease state at that time. As the symptoms get worse and worse, at some point the patient might be diagnosed. And that's what I'm illustrating by this gray curve. This is the point in time which the patient is diagnosed with their disease.

At the time of diagnosis, a variety of things might happen. The patient might begin treatment. And that treatment might, for example, start to influence the disease burden. And you might see a drop in disease burden initially.

This is a cancer. Unfortunately, often we'll see recurrences of the cancer. And that might manifest by a uphill peak again, where it is burden grows. And once you start second-line treatment, that might succeed in lowering it again and so on. And this might be a cycle that repeats over and over again.

For other diseases for which have no cure, for example, but which are managed on a day-to-day basis-- and we'll talk about some of those-- you might see, even on a day-by-day basis, fluctuations. Or you might see nothing happening for a while. And then, for example, in autoimmune diseases, you'll see these flare-ups where the disease burden grows a lot, then comes down again. It's really inexplicable why that happens.

So the types of questions that we'd like to really understand here are, first, where is the patient in their disease trajectory? So a patient comes in today. And they might be diagnosed today because of symptoms somehow crossing some threshold and them coming into the doctor's office. But they could be sort of anywhere in this disease trajectory at the time of diagnosis.

And a key question is, can we stage patients to understand, for example, things like, how long are they likely to live based on what's currently going on with them? A second question is, when will the disease progress? So if you have a patient with kidney disease, you might want to know something about, when will this patient kidney disease need a transplant?

Another question is, how will treatment effect that disease progression? That I'm sort of hinting at here, when I'm showing these valleys that we conjecture to be affected by treatment. But one often wants to ask counterfactual questions like, what would happen to this patient's disease progression if you did one treatment therapy versus another treatment therapy?

So the example that I'm mentioning here in this slide is a rare blood cancer named multiple myeloma. It's rare. And so you often won't find data sets with that many patients in them. So for example, this data set which I'm listening in the very bottom here from the Multiple Myeloma Research Foundation CoMMpass study has roughly 1,000 patients.

And it's a publicly available data set. Any of you can download it today. And you could study questions like this about disease progression. Because you can look at laboratory tests across time. You could look at when symptoms start to rise. You have information about what treatments a patient is on. And you have outcomes, like death.

So for multiple myeloma, today's standard for how one would attempt to stage a patient looks

a little bit like this. Here I'm showing you two different staging systems. On the left is a Durie-Salmon Staging System, which is a bit older. On the right is what's called the Revised International Staging System.

A patient walks into their oncologist's office newly diagnosed with multiple myeloma. And after doing a series of blood tests, looking at quantities such as their hemoglobin rates, amount of calcium in the blood, also doing, let's say, a biopsy of the patient's bone marrow to measure amounts of different kinds of immunoglobulins, doing gene expression assays to understand various different genetic abnormalities, that data will then feed into a staging system like this.

So in the Durie-Salmon Staging System, a patient who is in stage one is found to have a very low M-component production rate. So that's what I'm showing over here. And that really corresponds to the amount of disease activity as measured by their immunoglobulins. And since this is a blood cancer, that's a very good marker of what's going on with the patient.

So at sort of this middle stage, which is called neither stage one nor stage three, is characterized by, in this case-- well, I'm not going to talk with that. If you go to stage three for here, you see that the M-component levels are much higher. If you look at X-ray studies of the patient's bones, you'll see that there are lytic bone lesions, which are caused by the disease and really represent an advanced status of the disease. And if you were to measure for the patient's urine the amount of light-chain production, you see that it has much larger values as well.

Now, this is an older staging system. In the middle, now I'm showing you a newer staging system, which is both dramatically simpler and involves some newer components. So for example, in stage one, it looks at just four quantities. First it looks at the patient's albumin and beta-2 microglobulin levels. Those are biomarkers that can be easily measured from the blood. And it says no high-risk cytogenetics. So now we're starting to bring in genetic quantities in terms of quantifying risk levels.

Stage three is characterized by significantly higher beta-2 microglobulin levels, translocations corresponding to particular high-risk types of genetics. This will not be the focus of the next two lectures, but Pete is going to go much more detail in two genetic aspects of precision medicine in a week and a half now. And in this way, each one of these stages represents something about the belief of how far along the patient is and is really strongly used to guide treatment therapy. So for example, patient is in stage one, an oncologist might decide we're

not going to treat this patient today.

So a different type of question, whereas you could think about this as being one of characterizing on a patient-specific level-- one patient walks in. We want to stage that specific patient. And we're going to look at some long-term outcomes and look at the correlation between stage and long-term outcomes. A very different question is a descriptive-type question. Can we say what will the typical trajectory of this disease look like?

So for example, we'll talk about Parkinson's disease for the next couple of minutes. Parkinson's disease is a progressive nervous system disorder. It's a very common one, as opposed to multiple myeloma. Parkinson's affects over 1 in 100 people, age 60 and above. And like multiple myeloma, there is also disease registries that are publicly available and that you could use to study Parkinson's.

Now, various researchers have used those data sets in the past. And they've created something that looks a little bit like this to try to characterize, at now a population level, what it means for a patient to progress through their disease. So on the x-axis, again, I have time now. On the y-axis, again, it denotes some level of disease disability. But what we're showing here now are symptoms that might arise at different parts of the disease stage.

So very early in Parkinson's, you might have some sleep behavior disorders, some depression, maybe constipation, anxiety. As the disease gets further and further along, you'll see symptoms such as mild cognitive impairment, increased pain. As the disease goes further on, you'll see things like dementia and an increasing amount of psychotic symptoms.

And information like this can be extremely valuable for a patient who is newly diagnosed with a disease. They might want to make life decisions like, should they buy this home? Should they stick with their current job? Can they have a baby? And all of these questions might really be impact-- the answer to those questions might be really impacted by what this patient could expect their life to be like over the next couple of years, over the next 10 years or the next 20 years. And so if one could characterize really well what the disease trajectory might look like, it will be incredibly valuable for guiding those life decisions.

But the challenge is that-- this is for Parkinson's. And Parkinson's is reasonably well understood. There are a large number of diseases that are much more rare, where any one clinician might see a very small number of patients in their clinic. And figuring out, really, how do we combine the symptoms that are seen in a very noisy fashion for a small number of

patients, how to bring that together to a coherent picture like this is actually very, very challenging. And that's where some of the techniques we'll be talking about in Tuesday's lecture, which talks about how do we infer disease stages, how do we automatically align patients across time, and how do we use very noisy data to do that, will be particularly valuable.

But I want to emphasize one last point regarding this descriptive question. This is not about prediction. This is about understanding, whereas the previous slide was about prognosis, which is very much a prediction-like question.

Now, a different type of understanding question is that of disease subtyping. Here, again, you might be interested in identifying, for a single patient, are they likely to progress quickly through their disease? Are they likely to progress slowly through their disease? Are they likely to respond to treatment? Are they not likely to respond to treatment?

But we'd like to be able to characterize that heterogeneity across the whole population and summarize it into a small number of subtypes. And you might think about this as redefining disease altogether. So today, we might say patients who have a particular blood abnormality, we will say are multiple myeloma patients. But as we learn more and more about cancer, we increasingly understand that, in fact, every patient's cancer is very unique.

And so over time, we're going to be subdividing diseases, and in other cases combining things that we thought were different diseases, into new disease categories. And in doing so it will allow us to better take care of patients by, first of all, coming up with guidelines that are specific to each of these disease subtypes. And it will allow us to make better predictions based on these guidelines. So we can say a patient like this, in subtype A, is likely to have the following disease progression. A patient like this, in subtype B, is likely to have a different disease progression or be a responder or a non-responder.

So here's an example of such a characterization. This is still sticking with the Parkinson's example. This is a paper from a neuropsychiatry journal. And it uses a clustering-like algorithm, and we'll see many more examples of that in today's lecture, to characterize patients into, to group patients into, four different clusters. So let me walk you through this figure so you see how to interpret it.

Parkinson's patients can be measured in terms of a few different axes. You could look at their motor progression. So that is shown here in the innermost circle. And you see that patients in

Cluster 2 seem to have intermediate-level motor progression. Patients in Cluster 1 have very fast motor progression, means that their motor symptoms get increasingly worse very quickly over time.

One could also look at the response of patients to one of the drugs, such as levodopa that's used to treat patients. Patients in Cluster 1 are characterized by having a very poor response to that drug. Patients in Cluster 3 are characterized as having intermediate, patients in Cluster 2 as having good response to that drug.

Similarly one could look at baseline motor symptoms. So at the time the patient is diagnosed or comes into the clinic for the first time to manage their disease, you can look at what types of motor-like symptoms do they have. And again, you see different heterogeneous aspects to these different clusters. So this is one means-- this is a very concrete way, of what I mean by trying to subtype patients.

So we'll begin our journey through disease progression modeling by starting out with that first question of prognosis. And prognosis, from my perspective, is really a supervised machine-learning problem. So we can think about prognosis from the following perspective.

Patient walks in at time zero. And you want to know something about what will that patient's disease status be like over time. So for example, you could ask, at six months, what is their disease status? And for this patient, it might be, let's say, 6 out of 10. And where these numbers are coming from will become clear in a few minutes.

12 months down the line, their disease status might be 7 out of 10. 18 months, it might be 9 out of 10. And the goal that we're going to try to tackle for the first half of today's lecture is this question of, how do we take the data, what I'll call the x vector, available for the patient at baseline and predict what will be these values at different time points?

So you could think about that as actually drawing out this curve that I showed you earlier. So what we want to do is take the initial information we have about the patient and say, oh, the patient's disease status, or their disease burden, over time is going to look a little bit like this. And for a different patient, based on their initial covariance, you might say that their disease burden might look like that.

So we want to be able to predict these curves in this-- for this presentation, there are going to actually be sort of discrete time points. We want to be able to predict that curve from the

baseline information we have available. And that will give us some idea of how this patient's going to progress through their disease.

So in this case study, we're going to look at Alzheimer's disease. Here I'm showing you two brains, a healthy brain and a diseased brain, to really emphasize how the brain suffers under Alzheimer's disease. We're going to characterize the patient's disease status by a score. And one example of such a score is shown here. It's called the Mini Mental State Examination, summarized by the acronym MMSE. And it's going to look as follows.

For each of a number of different cognitive questions, a test is going to be performed, which-- for example, in the middle, what it says is registration. The examiner might name three objects like apple, table, penny, and then ask the patient to repeat those three objects. All of us should be able to remember a sequence of three things so that when we finish the sequence, you should be able to remember what the first thing in the sequence was. We shouldn't have a problem with that. But as patients get increasingly worse in their Alzheimer's disease, that task becomes very challenging.

And so you might give 1.4 correct for each correct. And so if the patient gets all three, if they repeat all three of them, then they get three points. If they can't remember any of them, zero points.

Then you might continue. You might ask something else like subtract 7 from 100, then repeat some results, so some sort of mathematical question. Then you might return back to that original three objects you asked about originally. Now it's been, let's say, a minute later. And you say, what were those three objects I mentioned earlier? And this is trying to get at a little bit longer-term memory and so on.

And one will then add up the number of points associated with each of these responses and get a total score. Here it's out of 30 points. If you divide by 3, you get the story I give you here.

So these are the scores that I'm talking about for Alzheimer's disease. They're often characterized by scores to questionnaires. But of course, if you had done something like brain imaging, the disease status might, for example, be inferred automatically from brain imaging. If you had a smartphone device, which patients are carrying around with them, and which is looking at mobile activity, you might be able to automatically infer their current disease status from that smartphone. You might be able to infer it from their typing patterns. You might be able to infer it from their email or Facebook habits.

And so I'm just trying to point out, there are a lot of different ways to try to get this number of how the patient might be doing at any one point in time. Each of those an interesting question. For now, we're just going to assume it's known. So retrospectively, you've gathered this data for patients, which is now longitudinal in nature. You have some baseline information. And you know how the patient is doing over different six-month intervals. And we'd then like to be able to predict to those things.

Now, if this were-- we can now go back in time to lecture three and ask, well, how could we predict these different things? So what are some approaches that you might try? Why don't you talk to your neighbor for a second, and then I'll call on a random person.

[SIDE CONVERSATION]

OK. That's enough. My question was sufficiently under-defined that if you talk longer, who knows what you'll be talking about. Over here, the two of you-- the person with the computer. Yeah. How would you tackle this problem?

AUDIENCE: Me? OK.

DAVID SONTAG: No, no, no. Over here, yeah. Yeah, you.

AUDIENCE: I would just take, I guess, previous data, and then-- yeah, I guess, any previous data with records of disease progression over that time span, and then treated [INAUDIBLE].

DAVID SONTAG: But just to understand, would you learn five different models? So our goal is to get these-- here I'm showing you three, but it might be five different numbers at different time points. Would you learn one model to predict what it would be at six months, another to predict what would be a 12 months? Would you learn a single model?

Other ideas? Somewhere over in this part of the room. Yeah. You.

AUDIENCE: [INAUDIBLE]

DAVID SONTAG: Yeah. Sure.

AUDIENCE: [INAUDIBLE]

DAVID SONTAG: So use a multi-task learning approach, where you try to learn all five at that time and use what? What was the other thing?

AUDIENCE: So you can learn to use these datas in six months and also use that as your baseline [INAUDIBLE].

DAVID SONTAG: Oh, that's a really interesting idea. OK. So the suggestion was-- so there are two different suggestions, actually. The first suggestion was do a multi-task learning approach, where you attempt to learn-- instead of five different and sort of independent models, try to learn them jointly together. And in a second, we'll talk about why it might make sense to do that.

The different thought was, well, is this really the question you want to solve? For example, you might imagine settings where you have the patient not at time zero but actually at six months. And you might want to know what's going to happen to them in the future. And so you shouldn't just use the baseline information. You should recondition on the data you have available for time.

And a different way of thinking through that is you could imagine learning a Markov model, where you learn something about the joint distribution of the disease stage over time. And then you could, for example, even if you only had baseline information available, you could attempt to marginalize over the intermediate values that are unobserved to infer what the later values might be. Now, that Markov model approach, although we will talk about it extensively in the next week or so, it's actually not a very good approach for this problem. And the reason why is because it increases the complexity.

So when you are learn-- in essence if you wanted to predict what's going on at 18 months, and if, as an intermediate step to predict what goes on at 18 months, you have to predict what's going to go on at 12 months, and then the likelihood of transitioning from 12 months to 18 months, then you might incur error in trying to predict what's going on at 12 months. And that error is then going to propagate as you attempt to think about the transition from 12 months to 18 months. And that propagation of error, particularly when you don't have much data, is going to really hurt the [INAUDIBLE] of your machine learning algorithm.

So the method I'll be talking about today is, in fact, going to be what I view as the simplest possible approach to this problem. And it's going to be direct prediction approach. So we're directly going to predict each of the different time points independently. But we will tie together the parameters of the model, as was suggested, using a multi-task learning approach.

And the reason why we're going to want to use a multi-task learning approach is because of data sparsity. So imagine the following situation. Imagine that we had just binary indicators here. So let's say patient is OK, or they're not OK. So the data might look like this-- 0, 0, 1.

Then the data set you might have might look a little bit like this. So now I'm going to show you the data. And one row is one patient. Different columns are different time points. So the first patient, as I showed you before, is 0, 0, 1. Second patient might be 0, 0, 1, 0. Third patient might be 1, 1, 1, 1. Next patient might be 0, 1, 1, 1.

So if you look at the first time point here, you'll notice that you have a really imbalanced data set. There's only a single 1 in that first time point. If you look at the second time point, there are two. It's more of a balanced data set. And then in the third time point, again, you're sort of back into that imbalanced setting. What that means is that if you were to try to learn from just one of these time points by itself, particularly in the setting where you don't have that many data points alone, that data sparsity and in outcome label is going to really hurt you. It's going to be very hard to learn any interesting signal just from that time point alone.

The second problem is that the label is also very noisy. So not only might you have lots of imbalance, but there might be noise in the actual characterizations. Like for this patient, maybe with some probability, you would calculate 1, 1, 1, 1. With some other probability, you would observe 0, 1, 1, 1.

And it might correspond to some threshold in that score I showed you earlier. And just by chance, a patient, on some day, passes the threshold. On the next day, they might not pass that threshold. So there might be a lot of noise in the particular labels at any one time point. And you wouldn't want that noise to really dramatically affect your learning algorithm based on some, let's say, prior belief that we might have that there might be some amount of smoothness in this process across time.

And the final problem is that there might be censoring. So the actual data might look like this. For much later time points, we might have many fewer observations. And so if you were to just use those later time points to learn your predictive model, you just might not have enough data. So those are all different challenges that we're going to attempt to solve using a multi-task learning approach.

Now, to put some numbers to these things, we have these four different time points. We're

going to have 648 patients at the six-month time interval. And at the four-year time interval, there will only be 87 patients due to patients dropping out of the study.

So the key idea here will be, rather than learning these five independent models, we're going to try to jointly learn the parameters corresponding to those models. And the intuitions that we're going to try to incorporate in doing so are that there might be some features that are useful across these five different prediction tasks. And so I'm using the example of biomarkers here as a feature. Think of that like a laboratory test result, for example, or an answer to a question that's available baseline.

And so one approach to learning is to say, OK, let's regularize the learning of these different models to encourage them to choose a common set of predictive features or biomarkers. But we also want to allow some amount of flexibility. For example, we might want to say that, well, at any one time point, there might be couple of new biomarkers that are relevant for predicting that time point. And there might be some small amounts of changes across time.

So what I'll do right now is I'll introduce to you the simplest way to think through multi-task learning, which-- I will focus specifically on a linear model setting. And then I'll show you how we can slightly modify this simple approach to capture those criteria that I have over there.

So let's talk about a linear model. And let's talk about regression. Because here, in the example I showed you earlier, we were trying to pick the score that's a continuous value number. We want to try to predict it. And we might care about minimizing some loss function.

So if you were to try to minimize a squared loss, imagine a scenario where you had two different prediction problems. So this might be time point 0, and this might be time point 12, for six months and 12 months. You can start by summing over the patients, looking at your mean squared error at predicting what I'll say is the six-month outcome label by some linear function, which, I'm going to have it as subscript 6 to denote that this is a linear model for predicting the six-month time point value, dot-producted with your baseline features.

And similarly, your loss function for predicting, this one is going be the same. But now you'll be predicting the y12 label. And we're going to have a different weight vector for predicting that. Notice that x is the same. Because I'm assuming in everything I'm telling you here that we're going to be predicting from baseline data alone.

Now, a typical approach and try to regularize in this setting might be, let's say, to do L2

regularization. So you might say, I'm going to add onto this some lambda times the weight vector 6 squared. Maybe-- same thing over here.

So the way that I set this up for you so far, right now, is two different independent prediction problems. The next step is to talk about how we could try to tie these together. So any idea, for those of you who have not specifically studied multi-task learning in class? So for those of you who did, don't answer. For everyone else, what are some ways that you might try to tie these two prediction problems together? Yeah.

AUDIENCE: Maybe you could share certain weight parameters, so if you've got a common set of biomarkers.

DAVID SONTAG: So maybe you could share some weight parameters. Well, I mean, the simplest way to tie them together is just to say, we're going to-- so you might say, let's first of all add these two objective functions together. And now we're going to minimize-- instead of minimizing just-- now we're going to minimize over the two weight vectors jointly.

So now we have a single optimization problem. All I've done is I've now-- we're optimizing. We're minimizing this joint objective where I'm summing this objective with this objective. We're minimizing it with respect to now two different weight vectors. And the simplest thing to do what you just described might be to say, let's let W6 equal to W12.

So you might just add in this equality constraint saying that these two weight vectors should be identical. What would be wrong with that? Someone else, what would be wrong with-- and I know that wasn't precisely your suggestion. So don't worry.

AUDIENCE: I have a question.

DAVID SONTAG: Yeah. What's your question?

AUDIENCE: Is x-- are those also different?

DAVID SONTAG: Sorry. Yeah. I'm missing some subscripts, right. So I'll put this in superscript. And I'll put subscript i, subscript i. And it doesn't matter for the purpose of this presentation whether these are the same individuals or different individuals across these two problems. You can imagine they're the same individual.

So you might imagine that there are n individuals in the data set. And we're summing over the

same n people for both of these sums, just looking at different outcomes for each of them. This is the six-month outcome. This is the 12-month outcome. Is that clear?

All right. So the simplest thing to do would be just to not-- now that we have a joint optimization problem, we could constrain the two weight vectors to be identical. But of course, this is a bit of an overkill. This is like saying that you're going to just learn a single prediction problem, where you sort of ignore the difference between six months and 12 months and just try to predict-- you put those under there and just predict them both together. So you had another suggestion, it sounded like.

AUDIENCE: Oh, no. You had just asked why that was not it.

DAVID SONTAG: Oh, OK. And I answered that. Sorry. What could we do differently? Yeah, you.

AUDIENCE: You could maybe try to minimize the difference between the two. So I'm not saying that they need to be the same. But the chances that they're going to be super, super different isn't really high.

DAVID SONTAG: That's a very interesting idea. So we don't want them to be the same. But I might want them to be approximately the same, right?

AUDIENCE: Yeah.

DAVID SONTAG: And what's one way to try to measure how different these two are?

AUDIENCE: Subtract them.

DAVID SONTAG: Subtract them, and then do what? So these are vectors. So you--

AUDIENCE: Absolute value.

DAVID SONTAG: So it's not absolute value of a vector. What can you do to turn a vector into a single number?

AUDIENCE: Take the norm [INAUDIBLE].

DAVID SONTAG: Take a norm of it. Yeah. I think what you meant. So we might take the norm of it. What norm should we take?

AUDIENCE: L2?

DAVID SONTAG: Maybe the L2 norm. OK. And we might say we want that. So if we said that this was equal to 0,

then, of course, that's saying that they have to be the same.

But we could say that this is, let's say, bounded by some epsilon. And epsilon now is a parameter we get to choose. And that would then say, oh, OK, we've now tied together these two optimization problems. And we want to encourage that the two weight vectors are not that far from each other. Yep?

AUDIENCE: You represent each weight vector as-- have it just be duplicated and force the first place to be the same and the second ones to be different.

DAVID SONTAG: You're suggesting a slightly different way to parameterize this by saying that W12 is equal to W6 plus some delta function, some delta difference. Is that you're suggesting?

AUDIENCE: No, that you have your-- say it's n-dimensional, like each vector is n-dimensional. But now it's going to be 2n-dimensional. And you force the first n dimensions to be the same on the weight vector. And then the others, you--

DAVID SONTAG: Now, that's a really interesting idea. I'll return to that point in just a second. Thanks.

Before I return to that point, I just want to point out this isn't the most immediate think optimize. Because this is now a constrained optimization problem. What's our favorite algorithm for convex optimization in machine learning, and non-convex optimization? Everyone say it out loud.

AUDIENCE: Stochastic gradient descent.

DAVID SONTAG: TAs are not supposed to answer.

AUDIENCE: Just muttering.

DAVID SONTAG: Neither are faculty. But I think I heard enough of you say stochastic gradient descent. Yes. Good. That's what I was expecting.

And well, you could do projected gradient descent. But it's much easier to just get rid of this. And so what we're going to do is we're just going to put this into the objective function. And one way to do that-- so one motivation would be to say we're going to take the Lagrangian of this inequality. And then that'll bring this into the objective.

But you know what? Screw that motivation. Let's just erase this. And I'll just say plus

something else. So I'll call that lambda 1, some other hyper-parameter, times now W12 minus W6 squared.

Now let's look to see what happens. If we were to push this lambda 2 to infinity, remember we're minimizing this objective function. So if lambda 2 is pushed to infinity, what is the solution of W12 with respect to W6? Everyone say it out loud.

AUDIENCE: 0.

DAVID SONTAG: I said "with respect to." So there, 1 minus other is 0. Yes. Good.

All right. So it would be forcing them that they be the same. And of course, if lambda 2 is smaller, then it's saying we're going to allow some flexibility. They don't have to be the same. But we're going to penalize their difference by the squared difference in their norms.

So this is good. And so you raised a really interesting question, which I'll talk about now, which is, well, maybe you don't want to enforce all of the dimensions to be the same. Maybe that's too much. So one thing one could imagine doing is saying, we're going to only enforce this constraint for-- [INAUDIBLE] we're only going to put this penalty in for, let's say, dimensions-- trying to think the right notation for this. I think I'll use this notation. Let's see if you guys like this.

Let's see if this notation makes sense for you. What I'm saying is I'm going to take the-- d is the dimension. I'm going to take the first half of the dimensions to the end. I'm going to take that vector and I'll penalize that. So it's ignoring the first half of the dimensions.

And so what that's saying is, well, we're going to share parameters for some of this weight vector. But we're not going to worry about-- we're going to let them be completely dependent of each other for the rest. That's an example of what you're suggesting.

So this is all great and dandy for the case of just two time points. But what do we do if then we have five time points? Yeah?

AUDIENCE: There's some percentage of shared entries in that vector. So instead of saying these have to be in common, you say, treat all of them [INAUDIBLE].

DAVID SONTAG: I think you have the right intuition. But I don't really know how to formalize that just from your verbal description. What would be the simplest thing you might think of? I gave you an

example of how to do, in some sense, pairwise similarity. Could you just easily extend that if you have more than two things? You have idea? Nope?

AUDIENCE: [INAUDIBLE]

DAVID SONTAG: Yeah.

AUDIENCE: And then I'd get y1's similar to y2, and y2 [INAUDIBLE] y3. And so I might just--

DAVID SONTAG: So you might say w1 is similar to w2. w2 is similar to w3. w3 is similar to w4 and so on. Yeah. I like that idea.

I'm going to generalize that just a little bit. So I'm going to start thinking now about graphs. And we're going to now define a very simple abstraction to talk about multi-task learning. I'm going to have a graph where I have one node for every task and an edge between tasks, between nodes, if those two tasks, we want to encourage their weights to be similar to another.

So what are our tasks here? W6, W12. So in what you're suggesting, you would have the following graph. W6 goes to W12 goes to W24 goes to W36 goes to W48.

Now, the way that we're going to transform a graph into an optimization problem is going to be as follows. I'm going to now suppose that I'm going to let-- I'm going to define a graph on V comma E. V, in this case, is going to be the set 6, 12, 24, and so on.

And I'll denote edges by s comma t. And E is going to refer to a particular two tasks. So for example, the task of six, predicting at six months, and the task of predicting at 12 months.

Then what we'll do is we'll say that the new optimization problem is going to be a sum over all of the tasks of the loss function for that task. So I'm going to ignore what is. I'm just going to simply write-- over there, I have two different loss functions for two different tasks. I'm just going to add those together. I'm just going to leave that in this abstract form.

And then I'm going to now sum over the edges s comma t in E in this graph that I've just defined of Ws minus Wt squared. So in the example that I go over there in the very top, there were only two tasks, W6 and W12. And we had an edge between them. And we penalized it exactly in that way.

But in the general case, one could imagine many different solutions. For example, you could imagine a solution where you have a complete graph. So you may have four time points. And

you might penalize every pair of them to be similar to one another. Or, as was just suggested, you might think that there might be some ordering of the tasks. And you might say that you want that-- instead of a complete graph, you're going to just have a chain graph, where, with respect to that ordering, you want every pair of them along the ordering to be close to each other.

And in fact, I think that's probably the most reasonable thing to do in a setting of disease progression modeling. Because, in fact, we have some smoothness type prior in our head about these values. The values should be similar to one another when they're very close time points.

I just want to mention one other thing, which is that from an optimization perspective, if this is what you had wanted to do, there is a much cleaner way of doing it. And that's to introduce a dummy node. I wish I had more colors. So one could instead introduce a new weight vector. I'll call it W. I'll just call it W with no subscript. And I'm going to say that every other task is going to be connected to it in that star.

So here we've introduced a dummy task. And we're connecting every other task to it. And then, now you'd have a linear number of these regularization terms in the number of tasks. But yet you are not making any assumption that there exists some ordering between them in the task. Yep?

AUDIENCE: Do you--

DAVID SONTAG: And W is never used for prediction ever. It's used during optimization.

AUDIENCE: Why do you need a W0 instead of just doing it based on like W1?

DAVID SONTAG: Well, if you do it based on W1, then it's basically saying that W1 is special in some way. And so everything sort of pulled towards it, whereas it's not clear that that's actually the right thing to do. So you'll get different answers. And I'd leave that as an exercise for you to try to derive.

So this is the general idea for how one could do multi-task learning using linear models. And I'll also leave it as an exercise for you to think through how you could take the same idea and now apply it to, for example, deep neural networks. And you can believe me that these ideas do generalize in the ways that you would expect them to do.

And it's a very powerful concept. And so whenever you are tasked with-- when you tackle

problems like this, and you're in settings where a linear model might do well, before you believe that someone's results using a very complicated approach is interesting, you should ask, well, what about the simplest possible multi-task learning approach?

So we already talked about one way to try to make the regularization a bit more interesting. For example, we could attempt to regularize only some of the features' values to be similar to another. In this paper, which was tackling this disease progression modeling problem for Alzheimer's, they developed a slightly more complicated approach, but not too much more complicated, which they call the convex fused sparse group lasso.

And it does the same idea that I gave here, where you're going to now learn a matrix W. And that matrix W is precisely the same notion. You have a different weight vector per task. You just stack them all up into a matrix.

L of W, that's just what I mean by the sum of the loss functions. That's the same thing. The first term in the optimization problem, lambda 1 times the L1 norm of W, is simply saying-- it's exactly like the sparsity penalty that we typically see when we're doing regression. So it's simply saying that we're going to encourage the weights across all of the tasks to be as small as possible. And because it's an L1 penalty, it adds the effect of actually trying to encourage sparsity. So it's going to push things to zero wherever possible.

The second term in this optimization problem, this lambda 2 RW squared, is also a sparsely penalty. But it's now pre-multiplying the W by this R matrix. This R matrix, in this example, is shown by this. And this is just one way to implement precisely this idea that I had on the board here.

So what this R matrix is going to say it is it's going to say for-- it's going to have one-- you can have as many rows as you have edges. And you're going to have-- for the corresponding task which is S, you have a 1. For the corresponding task which is T, you have a minus 1.

And then if you multiply this R matrix by W transpose, what you get is precisely these types of pair-wise comparisons out, the only difference being that here, instead of using a L2 norm, they penalized using an L1 norm. So that's what that second term is, lambda 2 RW transposed. It's simply an implementation of precisely this idea.

And that final term is just a group lasso penalty. It's nothing really interesting happening there. I just want to comment-- I had forgotten to mention this. The loss term is going to be precisely

a squared loss. This F refers to a Frobenius norm, because we've just stacked together all of the different tasks into one.

And the only interesting thing that's happening here is this S, which we're doing an element-wise multiplication. What that S is is simply a masking function. It's saying, if we don't observe a value at some time point, like, for example, if either this is unknown or censored, then we're just going to zero it out. So there will not be any loss for that particular element. So that S is just the mask which allows you to account for the fact that you might have some missing data.

So this is the approach used in that KDD paper from 2012. And returning now to the Alzheimer's example, they used a pretty simple feature set with 370 features. The first set of features were derived from MRI scans of the patient's brain. In this case, they just derived some pre-established features that characterize the amount of white matter and so on. That includes some genetic information, a bunch of cognitive scores.

So MMSE was one example of an input to this model, at baseline is critical. So there are a number of different types of cognitive scores that were collected at baseline, and each one of those makes up some feature, and then a number of laboratory tests, which I'm just noting as random numbers here. But they have some significance.

Now, one of the most interesting things about the results is if you compare the predictive performance of the multi-task approach to the independent regressor approach. So here we're showing two different measures of performance. The first one is some normalized mean squared error. And we want that to be as low as possible. And the second one is R, as in R squared. And you want that to be as high as possible. So one would be perfect prediction.

On this first column here, it's showing the results of just using independent regressors-- so if instead of tying them together with that R matrix, you had R equal to 0, for example. And then in each of the subsequent columns, it shows now learning with this objective function, where we are pumping up increasingly high this lambda 2 coefficient. So it's going to be asking for more and more similarity across the tasks.

So you see that even with a moderate value of lambda 2, you start to get improvements between this multi-task learning approach and the independent regressors. So the average R squared, for example, goes from 0.69 up to 0.77. And you notice how we have 95% confidence intervals here as well. And it seems to be significant. As you pump that lambda value larger, although I won't comment about the statistical significance between these

columns, we do see a trend, which is that performance gets increasingly better as you encourage them to be closer and closer together.

So I don't think I want to mention anything else about this result. Is there a question?

AUDIENCE: Is this like a holdout set?

DAVID SONTAG: Ah, thank you. Yes. So this is on a holdout set. Thank you. And that also reminded me of one other thing I wanted to mention, which is critical to this story, which is that you see these results because there's not much data.

If you had a really large training set, you would see no difference between these columns. Or, in fact, if you had a really data set, these results would be worse. As you pump lambda higher, the results will get worse. Because allowing flexibility among the different tasks is actually a better thing if you have enough data for each task. So this is particularly valuable in the data-poor regime.

When it goes to try to analyze the results in terms of looking at the feature importances as a function of time, so one row here corresponds to the weight vector for that time point's predictor. And so here we're just looking at four of the time points, four of the five time points. And the columns correspond to different features that were used in the predictions. And the colors correspond to how important that feature is to the prediction. You could imagine that being something like the norm of the corresponding weight in the linear model, or a normalized version of that.

What you see are some interesting things. First, there are some features, such as these, where they're important at all different time points. That might be expected. But then there also might be some features that are really important for predicting what's going to happen right away but are really not important to predicting longer-term outcomes. And you start to see things like that over here, where you see that, for example, these features are not at all important for predicting in the 36th time point but were useful for the earlier time points.

So from here, now we're going to start changing gears a little bit. What I just gave you is an example of a supervised approach. Is there a question?

AUDIENCE: Yes. If a faculty member may ask this question.

DAVID SONTAG: Yes. I'll permit it today.

AUDIENCE: Thank you. So it's really two questions. But I like the linear model, the one where Fred suggested, better than the fully coupled model. Because it seems more intuitively plausible to--

DAVID SONTAG: And indeed, it's the linear model which is used in this paper.

AUDIENCE: Ah, OK.

DAVID SONTAG: Yes. Because you noticed how that R was sort of diagonal in--

AUDIENCE: So it's-- OK. The other observation is that, in particular in Alzheimer's, given our current state of inability to treat it, it never gets better. And yet that's not constrained in the model. And I wonder if it would help to know that.

DAVID SONTAG: I think that's a really interesting point. So what Pete's suggesting is that you could think about this as-- you could think about putting an additional constraint in, which is that you can imagine saying that we know that, let's say, yi6 is typically less than yi12, which is typically less than yi24 and so on. And if we were able to do perfect prediction, meaning if it were the case that your predicted y's are equal to your true y's, then you should also have that W6 dot xi is less than W12 dot xi, which should be less than W24 dot xi.

And so one could imagine now introducing these as new constraints in your learning problem. In some sense, what it's saying is, well, we may not care that much if we get some errors in the predictions, but we want to make sure that at least we're able to sort the patients correctly, a given patient correctly. So we want to ensure at least some monotonicity in these values.

And one could easily try to translate these types of constraints into a modification to your learning algorithm. For example, if you took any pair of these-- let's say, I'll take these two together. One could introduce something like a hinge loss, where you say you want that-- you're going to add a new objective function, which says something like, you're going to penalize the max of 0 and 1 minus-- and I'm going to screw up this order. But it will be something like W-- so I'll derive it correctly.

So this would be W12 minus W24 dot product with xi, we want to be less than 0. And so you could look at how far from 0 is it. So you could look at W12-- do, do, do. You might imagine a loss function which says, OK, if it's greater than 0, then you have problem. And we might penalize it at, let's say, a linear penalty however greater than 0 it is.

And if it's less than 0, you don't penalties at all. So you say something like this, max of W12 minus W24 dot product xi. And you might add something like this to your learning objective. That would try to encourage-- that would penalize violations of this constraint using a hinge loss-type loss function. So that would be one approach to try to put such constraints into your learning objective.

A very different approach would be to think about it as a structured prediction problem, where instead of trying to say that you're going to be predicting a given time point by itself, you want to predict the vector of time points. And there's a whole field of what's called structured prediction, which would allow one to formalize objective functions that might encourage, for example, smoothness in predictions across time that one could take advantage of. But I'm not going to go more into that for reasons of time.

Hold any more questions to the end of the lecture. Because I want to make sure I get through this last piece.

So what we've talked about so far is a supervised learning approach to trying to predict what's going to happen to a patient given what you know at baseline. But I'm now going to talk about a very different style of thought, which is using an unsupervised learning approach to this. And there are going to be two goals of doing unsupervised learning for tackling this problem.

The first goal is that of discovery, which I mentioned at the very beginning of today's lecture. We might not just be interested in prediction. We might also be interested in understanding something, getting some new insights about the disease, like discovering that there might be some subtypes of the disease.

And those subtypes might be useful, for example, to help design new clinical trials. Like maybe you want to say, OK, we conjecture that patients in this subtype are likely to respond best to treatment. So we're only going to run the clinical trial for patients in this subtype, not in the other one.

It might be useful, also, to try to better understand the disease mechanism. So if you find that there are some people who seem to progress very quickly through their disease and other people who seem to progress very slowly, you might then go back and do new biological assays on them to try to understand what differentiates those two clusters. So the two clusters are differentiated in terms of their phenotype, but you want to go back and ask, well, what is different about their genotype that differentiates them?

And it might also be useful to have a very concise description of what differentiates patients in order to actually have policies that you can implement. So rather than having what might be a very complicated linear model, or even non-linear model, for predicting future disease progression, it would be much easier if you could just say, OK, for patients who have this biomarker abnormal, they're likely to have very fast disease progression. Patients who are likely have this other biomarker abnormal, they're likely to have a slow disease progression.

And so we'd like to be able to do that. That's what I mean by discovering disease subtypes. But there's actually a second goal as well, which-- remember, think back to that original motivation I mentioned earlier of having very little data.

If you have very little data, which is unfortunately the setting that we're almost always in when doing machine learning in health care, then you can overfit really easily to your data when just using it strictly within a discriminative learning framework.

And so if one were to now change your optimization problem altogether to start to bring in an unsupervised loss function, then one can hope to get much more out of the limited data you have and save the labels, which you might overfit on very easily, for the very last step of your learning algorithm. And that's exactly what we'll do in this segment of the lecture.

So for today, we're going to think about the simplest possible unsupervised learning algorithm. And because the official prerequisite for this course was 6036, and because clustering was not discussed in 6036, I'll spend just two minutes talking about clustering using the simplest algorithm called K-means, which I hope almost all of you know. But this will just be a simple reminder.

How many clusters are there in in this figure that I'm showing over here? Let's raise some hands. One cluster? Two clusters? Three clusters? Four clusters? Five clusters? OK.

And are these red points more or less showing where those five clusters are? No. No, they're not. So rather there's a cluster here. There's a cluster here, there, there, there.

All right. So you were you are able to do this really well, as humans, looking at two dimensional data. The goal of algorithms like K-means is to show how one could do that automatically for high-dimensional data. And the K-means algorithm is very simple. It works as follows.

You hypothesize a number of clusters. So here we have hypothesized five clusters. You're

going to randomly initialize those cluster centers, which I'm denoting by those red points shown here. Then in the first stage of the K-means algorithm, you're going to assign every data point to the closest cluster center.

And that's going to induce a Voronoi diagram where every point within this Voronoi cell is closer to this red point than to any other red point. And so every data point in this Voronoi cell will then be assigned to this data point. Every data point in this Voronoi cell will be assigned to that data point and so on.

So we're going to now assign all data points to the closest cluster center. And then we're just going to average all the data points assigned to some cluster center to get the new cluster center. And you repeat. And you're going to stop this procedure when no point in time is changed.

So let's look at a simple example. Here we're using K equals 2. We just decided there are only two clusters. We've initialized the two clusters shown here, the two cluster centers, as this red cluster center and this blue cluster center. Notice that they're nowhere near the data. We've just randomly chosen. They're nowhere near the data. It's actually pretty bad initialization.

The first step is going to assign data points to their closest cluster center. So I want everyone to say out loud either red or green, to which cluster center it's going to point to, what it is going to be assigned to this step.

[INTERPOSING VOICES]

AUDIENCE: Red. Blue. Blue.

DAVID SONTAG: All right. Good. We get it.

So that's the first assignment. Now we're going to average the data points that are assigned to that red cluster center. So we're going to average all the red points. And the new red cluster center will be over here, right?

AUDIENCE: No.

DAVID SONTAG: Oh, over there? Over here?

AUDIENCE: Yes.

DAVID SONTAG: OK. Good. And the blue cluster center will be somewhere over here, right?

AUDIENCE: Yes.

DAVID SONTAG: OK. Good. So that's the next step. And then you repeat. So now, again, you assign every data point to its closest cluster center. By the way, the reason why you're seeing what looks like a linear hyperplane here is because there are exactly two cluster centers.

And then you repeat. Blah, blah, blah. And you're done. So in fact, I think I've just shown you the convergence point.

So that's the K-means algorithm. It's an extremely simple algorithm. And what I'm going to show you for the next 10 minutes of lecture is how one could use this very simple clustering algorithm to better understand asthma.

So asthma is something that really affects a large number of individuals. It's characterized by having difficulties breathing. It's often managed by inhalers, although, as asthma gets more and more severe, you need more and more complex management schemes.

And it's been found that 5% to 10% of people who have severe asthma remain poorly controlled despite using the largest tolerable inhaled therapy. And so a really big question that the pharmaceutical community is extremely interested in is, how do we come up with better therapies for asthma? There's a lot of money in that problem.

I first learned about this problem when a pharmaceutical company came to me when I was a professor at NYU and asked me, could they work with me on this problem? I said no at the time. But I still find it interesting.

[CHUCKLING]

And at that time, the company pointed me to this paper, which I'll tell you about in a second. But before I get there, I want to point out what are some of the big picture questions that everyone's interested in when it comes to asthma. The first one is to really understand what is it about either genetic or environmental factors that underlie different subtypes of asthma. It's observed that people respond differently the therapy. It is observed that some people aren't even controlled with therapy. Why is that?

Third, what are biomarkers, what are ways to predict who's going to respond or not respond to

any one therapy? And can we get better mechanistic understanding of these different subtypes? And so this was a long-standing question. And in this paper from the *American Journal of Respiratory Critical Care Medicine,* which, by the way, has a huge number of citations now-- it's sort of a prototypical example of subtyping. That's why I'm going through it. They started to answer that question using a data-driven approach for asthma.

And what I'm showing you here is the punch line. This is that main result, the main figure over the paper. They've characterized asthma in terms of five different subtypes, really three type. One type, which I'll show over here, was sort of inflammation predominant; one type over there, which is called early symptom predominant; and another here, which is sort of concordant disease. And what I'll do over the next few minutes is walk you through how they came up with these different clusters.

So they used three different data sets. These data sets consisted of patients who had asthma and already had at least one recent therapy for asthma. They're all nonsmokers. But they were managed in-- they're three disjoint set of patients coming from three different populations. The first group of patients were recruited from primary care practices in the United Kingdom.

All right. So if you're a patient with asthma, and your asthma is being managed by your primary care doctor, then it's probably not too bad. But if your asthma, on the other hand, were being managed at a refractory asthma clinic, which is designed specifically for helping patients manage asthma, then your asthma is probably a bit more severe. And that second group of patients, 187 patients, were from that second cohort of patients managed out of an asthma clinic.

The third data set is much smaller, only 68 patients. But it's very unique because it is coming from a 12-month study, where it was a clinical trial, and there were two different types of treatments applied given to these patients. And it was a randomized control trial. So the patients were randomized into each of the two arms of the study.

I'll describe to you what the features are on just the next slide. But first I want to tell you about how their pre-processes to use within the K-means algorithm. Continuous-valued features where z-scored in order to normalize their ranges. And categorical variables were represented just by a one-hot encoding.

Some of the continuous variables were furthermore transformed prior to clustering by taking

the logarithm of the features. And that's something that can be very useful when doing something like K-means. Because it can, in essence, allow for that Euclidean distance function, which is using K-means, to be more meaningful by capturing more of a dynamic range of the feature.

So these were the features that went into the clustering algorithm. And there are very, very few, so roughly 20, 30 features. They range from the patient's gender and age to their body mass index, to measures of their function, to biomarkers such as eosinophil count that could be measured from the patient's sputum, and more. And there a couple of other features that I'll show you later as well. And you could look to see how did these quantities, how did these populations, differ.

So on this column, you see the primary care population. You look at all of these features in that population. You see that in the primary care population, the individuals are-- on average, 54% percent of them are female. In the secondary care population, 65% of them are female. You notice that things like-- if you look at to some measures of lung function, it's significantly worse in that secondary care population, as one would expect. Because these are patients with more severe asthma.

So next, after doing K-means clustering, these are the three clusters that result. And now I'm showing you the full set of features. So let me first tell you how to read this. This is clusters found in the primary care population. This column here is just the average values of those features across the full population.

And then for each one of these three clusters, I'm showing you the average value of the corresponding feature in just that cluster. And in essence, that's exactly the same as those red points I was showing you when I describe to you K-means clustering. It's the cluster center. And one could also look at the standard deviation of how much variance there is along that feature in that cluster. And that's what the numbers in parentheses are telling you.

So the first thing to note is that in Cluster 1, which the authors of the study named Early Onset Atopic Asthma, these are very young patients, average of 14, 15 years old, as opposed to Cluster 2, where the average age was 35 years old-- so a dramatic difference there. Moreover, we see that these are patients who have actually been to the hospital recently. So most of these patients have been to the hospital. On average, these patients have been to hospital at least once recently.

And furthermore, they've had severe asthma exacerbations in the past 12 months, at least, on average, twice per patient. And those are very large numbers relative to what you see in these other clusters. So that's really describing something that's very unusual about these very young patients with pretty severe asthma. Yep?

AUDIENCE: What is the p-value [INAUDIBLE]?

DAVID SONTAG: Yeah. I think the p-value-- I don't know if this is a pair-wise comparison. I don't remember off the top of my head. But it's really looking at the difference between, let's say-- I don't know which of these cl-- I don't know if it's comparing two of them or not. But let's say, for example, it might be looking at the difference between this and that. But I'm just hypothesizing. I don't remember.

Cluster 2, one other hand, was predominately female. So 81% of the patients were female there. And they were largely overweight. So their average body mass index was 36, as opposed to the other two clusters, where the average body mass index was 26. And Cluster 3 consisted of patients who really have not had that severe asthma. So the average number of previous hospital admissions and asthma exacerbations was dramatically smaller than in the other two clusters.

So this is the result of the finding. And then you might ask, well, how does that generalize to the other two populations? So they then went to the secondary care population. And they reran the clustering algorithm from scratch. And this is a completely disjoint set of patients.

And what they found, what they got out, is that the first two clusters exactly resembled Clusters 1 and 2 from the previous study on the primary care population. But because this is a different population with much more severe patients, that third cluster earlier of benign asthma doesn't show up in this new population. And there are two new clusters that show up in this new population.

So the fact that those first two clusters were consistent across two very different populations gave the authors confidence that there might be something real here. And then they went and they explored that third population, where they had longitudinal data. And that third population they were then using to ask, does it not-- so up until now, we've only used baseline information.

But now we're going to ask the following question. If we took the baseline data from those 68

patients and we were to separate them into three different clusters based on the characterizations found in the other two data sets, and then if we were to look at long-term outcomes for each cluster, would they be different across the clusters? And in particular, here we actually looked at not just predicting a progression, but we're also looking at prediction-- we're looking at differences in treatment response. Because this was a randomized-control trial.

And so there are going to be two arms here, what's called the clinical arm, which is the standard clinical care, and what's called the sputum arm, which consists of doing regular monitoring of the airway inflammation, and then tight trading steroid therapy in order to maintain normal eosinophil counts. And so this is comparing two different treatment strategies. And the question is, do these two treatment strategies result in differential outcomes?

So when the clinical trial was originally performed and they computed the average treatment effect, which, by the way, because the RCT was particularly simple-- you just averaged outcomes across the two arms-- they found that there was no difference across the two arms. So there was no difference in outcomes across the two different therapies.

Now what these authors are going to do is they're going to rerun the study. And they're going to now, instead of just looking at the average treatment effect for the whole population, they're going to use-- they're going to look at the average treatment each of the clusters by themselves. And the hope there is that one might be able to see now a difference, maybe that there was heterogeneous treatment response and sometimes that therapy worked for some people and not for others.

And these were the results. So indeed, across these three clusters, we see actually a very big difference. So if you look here, for example, the number of commenced on oral corticosteroids, which is a measure of an outcome-- so you might want this to-- I can't remember, small or large. But there was a big difference between these two clusters. And this cluster, the number commenced under the first arm is two; in this other cluster for patients who got the second arm, nine; and exactly the opposite for this third cluster. The first cluster, by the way, had only three patients in it. So I'm not going to make any comment about it.

Now, since these go in completely opposite directions, it's not surprising that the average treatment effect across the whole population was zero. But what we're seeing now is that, in fact, there is a difference. And so it's possible that the therapy is actually effective but just for a

smaller number of people.

Now, this study would've never been possible had we not done this clustering beforehand. Because it has so few patients, only 68 patients. If you attempted to both search for the clustering at the same time as, let's say, find clusters to differentiate outcomes, you would overfit the data very quickly. So it's precisely because we did this unsupervised sub-typing first, and then use the labels not for searching for the subtypes but only for evaluating the subtypes, that we're actually able to do something interesting here.

So in summary, in today's lecture, I talked about two different approaches, a supervised approach for predicting future disease status and an unsupervised approach. And there were a few major limitations that I want to emphasize that we'll return to in the next lecture and try to address. The first major limitation is that none of these approaches differentiated between disease stage and subtype.

In both of the two approaches, we assumed that there were some amount of alignment of patients at baseline. For example, here we assume that the patients at time zero were somewhat similar to another. For example, they might have been newly diagnosed with Alzheimer's at that point in time.

But often we have a data set where we have no natural alignment of patients in terms of disease stage. And if we attempted to do some type of clustering like I did in this last example, what you would get out, naively, would be one cluster for disease stage. So patients who are very early in their disease stage might look very different from patients who are late in their disease stage. And it will completely conflate disease stage from disease subtype, which is what you might actually want to discover.

The second limitation of these approaches is that they only used one time point per patient, whereas in reality, such as you saw here, we might have multiple time points. And we might want to, for example, do clustering using multiple time points. Or we might want to use multiple time points to understand something about disease progression.

The third limitation is that they assume that there is a single factor, let's say disease subtype, that explained all variation in the patients. In fact, there might be other factors, patient-specific factors, that one would like to use in your noise model. When you use an algorithm like K-means for clustering, it presents no opportunity for doing that, because it has such a naive distance function.

And so in next week's lecture, we're going to move in to start talking a probabilistic modeling approaches to these problems, which will give us a very natural way of characterizing variation along other axes. And finally, a natural question you should ask is, does it have to be unsupervised or supervised? Or is there a way to combine those two approaches.

All right. We'll get back to that on Tuesday. That's all.