MITOCW | 19. Disease Progression Modeling and Subtyping, Part 2

PROFESSOR: So I'm going to begin by trying to build some intuition for how one might be able to do staging from cross-sectional data, and we'll return to this question of combined staging subtyping only much later.

So imagine that we had data that lived in one dimension. Here, each data point is an individual, we observe their data at just one point in time, and suppose we knew exactly which biomarker to look at. Right? So I gave you an example of that here, when you might look at some antibody expression level, and that might be what I call biomarker A, is if you knew exactly what biomarker to look at, you might just put each person along this line and you might conjecture that maybe on one side of the line, this is the early disease, and that the other sort of line, maybe that's the late disease.

Why might that be a reasonable conjecture? What would be an alternative conjecture? Why don't you talk to your neighbors and see if you guys can come up with some alternative conjectures. Let's go. All right, that's enough. So hopefully simple questions, so I won't give you too much time. All right, so what would be another conjecture? So again, our goal is we have one observation per individual, each individual is in some unknown stage of the disease, we would like to be able to sort individuals and turn it into early and late stages of the disease. I give you one conjecture of how to do that, sorting, what would be another reasonable conjecture? Raise your hand. Yep?

AUDIENCE: That there's the different-- that they have different types of the same diseases. They all have the same disease and it could-- just one of the subtypes might be sort of the--

PROFESSOR: Yeah. So you're going back to the example I gave here where you could conflate these things. I want to stick with a simpler story, let's suppose there's only one subtype of the disease. What would be another way to sort the patients given this data where the data is these points that you see here? Yeah?

AUDIENCE: For any disease in the middle range, and then as you [INAUDIBLE]

PROFESSOR: OK, so maybe early disease is right over here, and when things get bad, the patient-- this biomarker starts to become abnormal, and abnormality, for whatever reason, might be sort of to the right or to the left. Now I think that is a conjecture one could have. I would argue that that's perhaps not a very natural conjecture given what we know about common biomarkers that are measured from the human body and the way that they respond to disease progression. Unless you're in the situation of having multiple disease subtypes where, for example, going to the right marker might correspond to one disease subtype and going to the left marker might correspond to another disease subtype.

What would be another conjecture? You guys are missing the easy one. Yeah, in the back.

AUDIENCE: Well, it might just be one where the high values are [INAUDIBLE] stage and low values are later ones?

PROFESSOR: Exactly. So this might be early disease and that might be late disease.

AUDIENCE: It says vice versa on this slide.

PROFESSOR: Oh, does it really? Oh shoot.

[LAUGHTER]

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right, right, OK, OK. Thank you. Next time I'll take out that lower vice versa.

[LAUGHTER]

That's why you guys aren't saying that. OK. OK, so this is good. Now I think we're all on the same page, and we had some idea of what are some of the assumptions that one might need to make in order to actually do anything here. Like for example, we are making some-- we'll probably have to make some assumption about continuity, that there might be some gradual progression of the biomarker relevance from early to late, and it might be getting larger, it might be getting smaller. If it's indeed the scenario that we talked about earlier where we said like early disease might be here and late disease might be going to either side, in that case, I think one could easily argue that with just information we have here, disease progression-- disease stage is unidentifiable, right? Because you wouldn't know where would it-- where should you-- where should that transition point be? So here, here, here, here, here, here.

In fact, the same problem arises here. Like you don't know, is it early disease-- is it going this way or is it going that way? What would be one way to try to disentangle this just to try to get us all on the same page, right? So suppose it was only going this direction or going that direction, how could we figure out which is which? Yeah?

AUDIENCE: Maybe we had data on low key and other data about how much time we had taken

PROFESSOR: Yeah. No, that's great. So maybe we have data on let's say death information, or even just age. And if we started from a very, very rough assumption that disease stage let's say grows monotonically with age, then-- and if you had made an additional assumption that the disease stages are-- that the people who are coming in are uniformly drawn from across disease stages, with those two assumptions alone, then you could, for example, look at the average age of individuals over here and the average age of individuals over here, and you'd say, the one with the larger average age is the late disease one.

Or you could look at time to death if you had for each-- for each data point you also knew how long until that individual died, you could look at average time to death for these individuals versus those individuals and try to tease it apart in that way. That's what you meant. OK, so I'm just trying to give you some intuition for how this might be possible. What about if your data looked like this? So now you have two biomarkers. So we've only gone up by one dimension only, and we want to figure out where's early, where's late? Already starts to become more challenging, right?

So the intuition that I want you to have is that we're going to have to make some assumptions about disease progression, such as the ones we were just discussing, and we also have to get lucky in some way. So for example, one way of getting lucky would be to have a real lot of data. So if you had a ton, ton of data, and you made an additional assumption that your data lives in some low dimensional manifold where on one side of manifold is early disease and the other side of manifold is late disease, then you might be able to discover that manifold from this data, and you might conjecture that the manifold is something like that, that trajectory that I'm outlining there with my hand.

But for you to be able to do that, of course, you need to have enough data, all right? So it's going to be now a trade-off between just having cross-sectional data, it might be OK so long as you have a real lot of that data so you can sort of fill in the spaces and really identify that manifold. A different approach might be, well maybe you don't have just pure cross-sectional data, maybe you have two or maybe three samples from each patient. And then you can color code this. So you might say, OK, green is patient 1-- or patient A, we'll call it, and this is the first time point from patient A, second time point from patient A, third and fourth time points from patient A. Red is patient B, and you have two time points for patient B, and blue here is patient C, and you have 1, 2, 3 time points from patient C. OK?

Now again, it's not very dense data, we can't really draw curves out, But now we can start to get a sense of the ordering. And again, now we can-- even though we don't-- we're not in a dense setting like we were here, here, we'd still nonetheless be able to figure out that probably the manifold looks a little bit like this, right? And so again, I'm just trying to build intuition around when disease progression modeling for cross-sectional data might be possible, but this is a wide open field.

And so today, I'll be telling you about a few algorithms that try to build on some of these intuitions for doing disease progression modeling, but they will break down very, very easily. They'll break down when these assumptions I gave you don't hold, they'll break down when your data is high dimensional, they'll break down when your data looks like this where you don't just have a single subtype of perhaps a multiple subtypes. and so this is a really very active area of research, and it's an area that I think we can make a lot of progress on in the field in the next several years.

So I'll begin with one case study coming from my own work where we developed an algorithm for learning from cross-sectional data, and we valued it in the context of chronic obstructive pulmonary disorder or COPD. COPD is a condition of the lungs typically caused by air pollution or smoking, and it has a reasonably good staging mechanism. One uses what's called a spirometry device in order to measure the lung function of individual at any one point in time. So for example, you take this spirometry device, you stick it in your mouth, and you breathe in, and then you exhale, and one measure is how long it takes in order to exhale all your air, and that is going to be a measure of how good your lungs are.

And so then one can take that measure of your function and one can stage how severe the person's COPD is, and that goes by what's called the gold criteria. So for example, in stage 1 of the COPD, common treatments involve just vaccinations using a short-acting bronchodilator only when needed. When the disease stage gets much more severe, like stage 4, than often treatment is recommended to be inhaling glucocorticosteroids if there are repeated aspirations of the disease, long-term oxygen If respiratory failure occurs, and so on.

And so this is a disease that's reasonably well-understood because there exists a good staging mechanism. And I would argue that when we want to understand how to do disease staging in a data-driven fashion, we should first start by working with either synthetic data, or we should start with working with a disease where we have some idea of what the actual true disease staging is. And that way, we can look to see what our algorithms would recover in those scenarios, and does it align with what we would expect either from the way the data was generated or from the existing medical literature.

And that's why we chose COPD. Because it is well-understood, and there's a wealth of literature on it, and because we have data on it which is much messier than the type of data that went into the original studies, and we could ask, could we come to the same conclusions as those original studies?

So in this work, we're going to use data from the electronic medical record. We're only going to look at a subset of the EMR, in particular, diagnosis codes that are recorded for a patient at any point in time, and we're going to assume that we do not have access to spirometry data at all. So we don't have any obvious way of staging the patient's disease. The general approach is going to be to build a generative model for disease progression.

At a very high level, this is a Markov model. It's a model that specifies the distribution of the patient's data, which is shown here in the bottom, as it evolves over time. According to a number of hidden variables that are shown in the top, these S variables that denote disease stages, and these X variables that denote comorbidities that the patient might have at that point in time, these X and S variables are always assumed to the unobserved. So if you were to clump them together into one variable, this would look exactly like a hidden Markov model.

And moreover, we're not going to assume that we have a lot of longitudinal data for a patient. In particular, COPD evolves over a 10 to 20 years, and the data that we'll be learning from here has data only over one to three-year time range. The challenge will be, can we take data in this one to three-year time range and somehow stitch it together across large numbers of patients to get a picture of what the 20-year progression of the disease might look like? The way that we're going to do that is by learning the parameters of this probabilistic model. And then from the parameters, we're going to either infer the patient's actual disease stage and thus sort them, or actually simulate data from this model to see what a 20-year trajectory might look like. Is the goal clear?

All right. So now what I'm going to do is I'm going to step into this model piece by piece to tell you what each of these components are, and I'll start out with a very topmost piece shown here by the red box. So this is the model of the patient's disease progression at any one point in time. So this variable, S1, for example, might denote the patient's disease stage on March 2011; S2 might denote the patient's disease stage April 2011; S capital T might denote the patient's disease stage June 2012. So we're going to have one random variable for each observation of the patient's data that we have. And notice that the observations of the patient's data might be at very irregular time intervals, and that's going to be OK with this approach, OK? So notice that there is a one-month gap between S1 and S2, but a four-month gap between St minus 1 and St, OK? So we're going to model the patient's disease stage at the point in time when we have an observation for the patient.

S denotes a discrete disease stage in this model. So S might be a value from 1 up to 4, maybe 1 up to 10 where 1 is denoting a early disease stage and 4 or 10 might denote a much later disease stage. If we have a sequence of observations per patient-- for example, we might have an observation on March and then in April, we're going to denote the disease stage by S1 and S2, what this model is going to talk about is the probability distribution of transitioning from whatever the disease stage at S1 is to whatever the disease stage at S2.

Now because the time interval is between stages are not homogeneous, we have to have a transition distribution that takes into consideration that time gap. And to do that, we use what's known as a continuous time Markov process. Formally, we say that the transition distribution-- so the probability of transitioning from stage I at time t minus 1 to state j at time t, given as input the difference in time intervals-- the difference in time points delta-- so delta is the number of months between the two observations. So this conditional distribution is going to be given by the matrix exponential of this time interval times a matrix Q.

And then here, the matrix Q gives us the parameters that we want to learn. So let me contrast this to things that you might already be used to. In a typical hidden Markov model, you might have asked t goes to St-- or St minus 1 goes to St, and you might imagine just parametrizing St given St minus 1 just by a lookup table. So for example, if the number of states-- for each running variable is 3, then you would have a 3 by 3 table where for each state St minus 1, you have some probability of transition to the corresponding state St, so this might be something like 0.9, 0.9, 0.9, where notice, I'm having a very large value along the diagonal, because if, let's say, a very small period-- so a priori, we might believe that patients stay in the same disease, and then one might imagine that the probably transitioning from state 1 at time t minus 1 to state 2 at time t might be something like 0.09, and the probability of skipping state 2, going directly to state 3 from state 1 might be something much smaller like 0.01, OK?

And we might say something like that the probability-- we might imagine that the probability of going in a backwards direction, going from stage 2 at time t minus 1 to let's say stage 1 at time t, that might be 0 all right? So you might imagine that actually this is the model, and what that's saying is something like you never go in the backwards direction, and you're more likely to transition to the state immediately adjacent to the current stage and very unlikely to skip a stage.

So this would be an example of how you would parametrize the transition distribution in a typical discrete time Markov model, and the story here is going to be different specifically because we don't know the time intervals. So intuitively, if a lot of time has passed between the two observations, then we want to allow for an accelerated process. We want to allow for the fact that you might want to skip many different stages to go to your next time step, to go to the stage of the next time step, because so much time has passed.

And that intuitively is what this scaling of this matrix Q by delta corresponds to. So the number of parameters in this parameterization is actually identical to the number of parameters in this parametrization, right? So you have a matrix Q which is given to you in essence by the number of states squared-- really, the number of states-- there's an additional redundancy there because it has to sum up to 1, but that's irrelevant.

And so the same story here, but we're going to now parametrize the process by in some sense the infinitesimally small time probability of transitioning. So if you were to take the derivative of this transition distribution as the time interval shrinks, and then you were to integrate over the time interval that was observed and the probability of transitioning from any state to any other state with that infinitesimally small probability transitioning, what you get out is exactly this form. And I'll leave-- this paper is in the optional readings for today's lecture, and you can read through it to get more intuition about the continuous time Markov process. Any questions so far? Yep?

AUDIENCE: Those Q are the same for both versions or--

PROFESSOR: Yes. And this model Q is essentially the same for all patients. And you might imagine, if there were disease subtypes, which there aren't in this approach, that Q might be different for each subtype. For example, you might transition between stages much more quickly for some subtypes than for others. Other questions? Yep?

AUDIENCE: So-- OK, so Q you said had like-- it's just like a screen number used beforehand you kind of like specified these stages that you pick [INAUDIBLE]

PROFESSOR: Correct. Yes. So you pre-specify the number of stages that you want to model, and there are many ways to try to choose that parameter. For example, you could look at how about likelihood under this model, which is learned for the different of stages. You could use typical model selection techniques from machine learning as another approach where you try to penalize complexity in some way. Or, what we found here, because of some of the other things that I'm about to tell you, it doesn't actually matter that much.

So similarly to when one does hard [INAUDIBLE] clustering or even K-means clustering or even learning a problematic topic model, if you use a very small number of topics or number of clusters, you tend to learn very coarse-grained topics or clusters. If you use very many more-- if you use a much larger number of topics, you tend to learn much more fine-grained topics. Same story is going to happen here. If you use a small number of disease stages, you're going to learn very coarse-grained notions of disease stages; if you use more disease stages, you're going to learn a fine-grained notion; but the overall sorting of the patients is going to end up being very similar. But to make that statement, we're going to need to make some additional assumptions, which I'm going to show you in a few minutes. Any other questions? These are great questions. Yep?

AUDIENCE: So do we know the staging of the disease because I

PROFESSOR: No, and that's critical here. So I'm assuming that these variables-- these S's are all hidden variables here. And the way that we're going to learn this model is by maximum likelihood estimation where we marginalize over the hidden variables, just like you would do in any EM type algorithm. Any other questions? All right, so what I've just shown you is the topmost part of the model, now I'm going to talk about a horizontal slice. So I'm going to talk about one of these time points.

So if you were to look at the translation-- the rotation of one of those time points, what you would get out is this model. These X's are also hidden variables, and we have pre-specified them to characterize different axes by which we want to understand the patient's disease progression. So in Thursday's lecture, we characterized the patient's disease as subtype by just a single number, and similarly in this example is just by a single number, but we might want to understand what's really unique about each subtype. So for example-- sorry, what's really unique about each disease stage.

So for example, how is the patient's endocrine function in that disease stage? How is the patient's psychiatric status in that disease stage? Has the patient developed lung cancer yet and that disease stage? And so on. And so we're going to ask that we want to be able to read out from this model according to these axes, and this will become very clear at the end of this section where I show you a simulation of what 20 years looks like for COPD according to these quantities. When does the patient typically develop diabetes, when does the patient typically become depressed, when does the patient typically develop cancer, and so on.

So these are the quantities in which we want to be able to really talk about what happens to a patient at any one disease stage, but the challenge is, we never actually observe these quantities in the data that we have. Rather, all we observe are things like laboratory test results or diagnosis codes or procedures that have been formed and so on, which I'm going to call the clinical findings in the bottom.

And as we've been discussing throughout this course, one could think about things as diagnosis codes as giving you information about the disease status of the patient, but they're not one and the same as the diagnosis, because there's so much noise and bias that goes into the assigning of diagnosis codes for patients. And so the way that we're going to model the raw data as a function of these hidden variables that we want to characterize is using what's known as a noisy-OR network.

So we're going to suppose that there is some generative distribution where the observations you see-- for example, diagnosis codes are likely to be observed as a function of whether the patient has these phenotypes or comorbidities with some probability, and that probability can be specified by these edge weights. So for example, a diagnosis code for diabetes is very likely to be observed in the patient data if the patient truly has diabetes, but of course, it may not be recorded in the data for every single visit the patient has to a clinician, there might be some visits to clinicians that have nothing to do with their patients endocrine function and diabetes-- the diagnosis code might not be recorded for that visit. So it's going to be a noisy process, and that noise rate is going to be captured by that edge.

So part of the learning algorithm is going to be to learn that transition distributions-- for example, that Q matrix I showed you in the earlier slide, but the other role-- learning algorithm is to learn all of the parameters of this noisy-OR distribution, namely these edge weights. So that's going to be discovered as part of the learning algorithm. And a key question that you have to ask me is, if I know I want to read out from the model according to these axes, but these axes are never-- I'm never assuming that they're explicitly observed in the data, how do I ground the learning algorithm to give meaning to these hidden variables?

Because otherwise if we left them otherwise unconstrained and you did maximum likelihood estimation just like in any factor analysis-type model, you might discover some factors here, but they might not be the factors you care about, and if the learning problem was not identifiable, as is often the case in unsupervised learning, then you might not discover what you're interested in. So to ground the hidden variables, we introduced-- we used a technique that you already saw in an earlier lecture from lecture 8 called anchors.

So a domain expert is going to specify for each one of the comorbidites one or more anchors, which are observations, which we are going to conjecture could only have arisen from the corresponding hidden variable. So notice here that this diagnosis code, which is for type 2 diabetes, has only an edge from X1. That is an assumption that we're making in the learning algorithm. We are actually explicitly zeroing out all of the other edges from all of the other comorbidities to a 1.

We're not going to pre-specify what this edge rate is, we're going to allow for the fact that this might be noisy, it's not always observed even if the patient has diabetes, but we're going to say, this could not be explained by any of the other comorbidities. And so for each one of the comorbidites or phenotypes that we want to model, we're going to specify some small number of anchors which correspond to a type of sparsity assumption on that graph.

And these are the anchors that we chose for asthma, we chose a diagnosis code corresponding to asthma; for lung cancer, we chose several diagnosis codes correspond to lung cancer; for obesity, we chose a diagnosis code corresponding to morbid obesity; and so on. And so these are ways that we're going to give meaning to the hidden variables, but as you'll see in just a few minutes, it is not going to pre-specify too much of the model. The model's still going to learn a whole bunch of other interesting things.

By the way, the way that we actually came up with this set was by an iterative process. We specified some of the hidden variables to have anchors, but we also left some of them to be unanchored, meaning free variables. We did our learning algorithm, and just like you would do in a topic model, we discovered that there were some phenotypes that really seemed to be characterized by the patient's disease-- that seemed to characterize a patient's disease progression. Then in order to really dig deeper, working collaboratively with a domain expert, we specified anchors for those and we iterated, and in this way, we discovered the full set of interesting variables that we wanted to model. Yep?

AUDIENCE: Did you measure how good an anchor these were? Like are some comorbidities better anchors than others?

PROFESSOR: Great. You'll see-- I think we'll answer that question in just a few minutes when I show you what the graph looks like that's learned. Yep?

AUDIENCE: Were all the other weights in that X to O network 0? They weren't part of it here. So it looks like a pretty sparse--

PROFESSOR: They're explicitly nonzero, actually, it's opposite. So for an anchor, we say that it only has a single parent. Everything that's not an anchor can have arbitrarily many parents. Is that clear? OK. Yeah?

AUDIENCE: Do the anchors that you have in that linear table, you itereated yourself on that or did the doctors say that these are the [INAUDIBLE]?

PROFESSOR: We started out with just a subset of these conditions. As things that we wanted to model-- things that we wanted to understand what happens along disease progression according to these axes, but just a subset of them originally. And then we included a few additional hidden variables that didn't have any anchors associated to them, and after doing unsupervised learning and just a preliminary development stage, they discovered some topics and we realized, oh shoot, we should have included those in there, and then we added them in with corresponding anchors.

And so you could think about this as an exploratory data analysis pipeline. Yep?

AUDIENCE: Is there a chance that these aren't anchors?

PROFESSOR: Yes. So there's definitely the chance that these may not be anchors related to the question was asked a second ago. So for example, there might be some chance that the morbid obesity diagnosis code might be coded for a patient more often for a patient who has, let's say, asthma-- this is a bad example. And in which case, that would correspond to there truly existing an edge from asthma to this anchor, which would be a violation of anchor assumption.

All right, so we chose these to make that unlikely, but it could happen. And it's not easily testable. So this is another example of an untestable assumption, just like we saw lots of other examples already in today's lecture and the causal inference lectures. If we had some ground truth data-- like if we had done chart review for some number of patients and we actually label these conditions, then we could test that anchor assumption. But here, we're assuming that we don't actually know the ground truth of these conditions.

AUDIENCE: Is there a reason why you choose such high-level comorbidites? Like I imagine you could go more specific. Even, say, the diabetes, you could try to subtype the diabetes based on this other model, sort of use that as a single layer, but it seems to-- at least this model seems to choose [INAUDIBLE] high level. I was just curious of the reason.

PROFESSOR: Yes. So that was a design choice that we made. There are many, many directions for follow-up work, one of which would be to use a hierarchical model here. But we hadn't gone that direction. Another obvious direction for follow-up work would be to do something within the subtyping with this staging by introducing another random variable, which is, let's say, the disease subtype, and making everything a function of that.

OK, so I've talked about the vertical slice and I've talked about the topmost slice, but what I still need to tell you about is how these phenotypes relate to the observed disease stage. So for this, we use-- I don't remember the exact technical terminology-- a factored Markov model? Is that right, Pete? Factorized Markov model-- I mean, this is a term that existed in the graphical model's literature, but I don't remember right now.

So what we're saying is that each of these Markov chains-- so each of these X1 up to Xt-- so this, will say, is the first one I call diabetes. This is the second one which I'll say is depression. We're going to assume that each one of these Markov chains is conditionally independent of each other given the disease stage. So it's the disease stage which ties everything together. So the graphical model looks like this, and so on, OK?

So in particular, there are no edges between, let's say, the diabetes variable and the depression variable. All correlations between these conditions is assumed to be mediated by the disease stage variable. And that's a critical assumption that we had to make. Does anyone know why? What would go wrong if we didn't make that assumption? So for example, what would go wrong if we had something look like this, x1-- what was my notation? X1,1, X1,2, X1,3, and suppose we had edges between them, a complete graph, and we had, let's say, also the S variable with edges to everything? What would happen in that case where we're not assuming that the X's are conditionally independent given S?

So I want you to think about this in terms of distributions. So remember, we're going to learn how to do disease progression through learning the parameters of this model. And so if we set this up and-- if we set up the learning problem in a way which is unidentifiable, then we're going to be screwed, we're not going to able to learn anything about disease progression. So what would happen in this situation? Someone who hasn't spoken today ideally.

So any of you remember from, let's say, perhaps an earlier machine learning class what types of distribution's a complete graph-- a complete Bayesian network could represent? So the answer is all distributions, because it corresponds to any factorization of the joint distribution. And so if you allowed these x variables to be fully connected to each other-- so for example, saying that depression depends on diabetes in addition to the stage, then in fact, you don't even need this stage variable in here. The marginal-- you can fit any distribution on these X variables even without the S variable at all. And so the model could learn to simply ignore the S variable, which would be exactly not our goal, because our goal is to learn something about the disease stage, and in fact, we're going to be wanting to make assumptions on the progression of disease stage, which is going to help us learn.

So by assuming conditional independence between these X variables, it's going to force all of the correlations to have to be mediated by that S variable, and it's going to remove some of that unidentifiability that would otherwise exist. It's this subtle but very important point. So the way that we're going to parametrize the conditional distribution-- so first of all, I'm going to assume these X's are all binary. So either the patient has diabetes or they don't have diabetes. I'm going to suppose that-- and this is, again, another assumption we're making, I'm going to suppose that once you already have a comorbidity, then you always have it. So for example, once this is 1, then all subsequent ones are also going to be 1. Hold the questions for just a second.

I'm also going to make an assumption that later stages of the disease are more likely to develop the comorbidity. So in particular, one can formalize that mathematically as probability of X-- I'll just say Xi being 1 given S-- I'll say St equals little s, comma, Xt minus 1 equals 0, and suppose that this is larger than or equal to probability of Xt equals 1 given St equals S prime and Xt minus 1 equals 0 for all S prime less than S, OK? So I'm saying, as you get further along in the disease stage, you're more likely to observe one of these complications.

And again, this is an assumption that we're putting into the learning algorithm, but what we found that these types of assumptions are really critical in order to learn disease progression models when you don't have a large amount of data. And note that this is just a linear inequality on the parameters of the model. And so one can use a convex optimization algorithm during learning-- during the maximum likelihood estimation step with this algorithm, we just put a linear inequality into the convex optimization problem to enforce this constraint. There are a couple of questions.

AUDIENCE: Is there generally like a quick way to check whether a model is unidentifiable or--

PROFESSOR: So there are ways to try to detect to see if a model is unidentifiable. It's beyond the scope of the class, but I'll just briefly mention one of the techniques. So one could-- so you can ask the identifiability question by looking at moments of the distribution. For example, you could talk about it as a function of all of the observed moments of distribution that you get from the data.

Now the observed data here are not the S's and X's, but rather the O's. So you look at the joint distribution on the O's, and then you can ask questions about-- if I was to-- so suppose I was to choose a random set of parameters in the model, is there any way to do a perturbation of the parameters in the model which leave the observed marginal distribution on the O's identical? And often when you're in the setting of non-identifiability, you can take the gradient of a function and see-- and you could sort of find that there is some wiggle space, and then you show that OK, there are actually-- this objective function is actually unidentifiable.

Now that type of technique is widely used when studying what are known as method of moments algorithms or estimation algorithms in learning verbal models, but they would be much, much harder to apply in this type of setting because first of all, these are much more complex models, and estimating the corresponding moments is going to be very hard because they're very high dimensional. And second, because they're-- I'm actually conflating two different things when I talk about identifiability. One statement is the infinite data identifiability, and the second question is your ability to actually learn a good model from a small amount of data, which is a sample complexity. And these constraints that I'm putting in, even if they don't affect the actual identifiability of the model, they could be extremely important for improving the sample complexity of learning algorithm. Is there another question?

So we valued to this using a data set of almost 4,000 patients where, again, each patient we observed for only a couple of years-- one to three years. And the observations that we observed were 264 diagnosis codes, the presence or absence of each of those diagnosis codes during any three-month interval. Overall, there were almost 200,000 observations of diagnosis codes in this data set.

The learning algorithm that we used was expectation maximization. Remember, there are a number of hidden variables here, and so if you want to maximize the likely-- if you want to learn the parameters that maximize the likelihood of those observations O, then you have to marginize over those hidden variables, and EM is one way to try to find a local optima of that likelihood function, with the key caveat that one has to do approximate inference during the E step here, because this model is not tractable, there's no closed form-- for example, dynamic programming algorithm for doing posterior inference in this model given its complexity.

And so what we used was a Gibbs sampler to do approximate inference within that E step, and we used-- we did block sampling of the Markov chains where we combined a Gibbs sampler with a dynamic programming algorithm, which improved the mixing rate of the Markov chain for those of you who are familiar with those concepts. And in the end step of the learning algorithm when one has to learn the parameters of the distribution, the only complex part of this model is the continuous time Markov process, and there's actually been previous literature from the physics community which shows how you can really-- which gives you analytic closed-form solutions for that M step of that continuous time Markov process.

Now if I were to do this again today, I would have done it a little bit differently. I would still think about modeling this problem in a very similar way, but I would do learning using a variational lower bound of the likelihood with a recognition network in order to very quickly get you a lower bound in the likelihood. And for those of you who are familiar with variational autoencoders, that's precisely the idea that is used there for learning variational autoencoders. So that's the way I would approach this if I was to do it again.

There's just one or two other extensions I want to mention. The first one is something which we-- one, more customization we made for COPD, which is that we enforced monotonic stage progression. So we said that-- so here I talked about a type of monotonically in terms of the conditional distribution of X given S, but one could also put an assumption in the-- I already talked about that, but one could also put an assumption on P of S-- S of t given S of t minus 1, which is implicitly an assumption on Q, and I gave you a hint of how one might do that over here when I said that you might put 0's to the left-hand side, meaning you can never go to the left. And indeed, we did something like that here as well, which is another type of constraint.

And finally, we regularize the learning problem by asking about that graph involving the conditions, the comorbidities, and the diagnosis codes be sparse, by putting a beta prior on those edge weights. So here's what one learned. So the first thing I'm going to do is I'm going to show you the-- we talked about how we specified anchors, but I told you that the anchors weren't the whole story. That we were able to infer much more interesting things about the hidden variables given all of the observations we have.

So here I'm showing you several of the phenotypes that were learned by this unsupervised learning algorithm. First, the phenotype for kidney disease. In red here, I'm showing you the anchor variables that we chose for kidney disease, and what you'll notice are a couple of things. First, the weight, which you should think about as being proportional in some way to how often you would see that diagnosis code given that the patient had kidney disease, the weights are all far less than one, all right? So there is some noise in this process of when you observe a diagnosis code for a patient.

The second thing you observe is that there are a number of other diagnosis codes that are observed to be-- which are explained by this kidney disease comorbidity, such as anemia, urinary tract infections, and so on, and that aligns well with what's known in the medical literature about kidney disease. Look at another example for lung cancer. In red here I'm showing you the anchors that we had pre-specified for these, which mean that these diagnosis codes could only be explained by the lung cancer comorbidity, and these are the noise rates that are learned for them, and that's everything else. And here's one more example of lung infection where there was only a signal anchor that we specified for pneumonia, and you see all of the other things that are automatically associated to that as by the unsupervised learning algorithm. Yep?

AUDIENCE: So how do you [INAUDIBLE] for the mobidity, right?

PROFESSOR: Right. So that's what the unsupervised learning algorithm is doing. So these weights are learned, and I'm showing you something like a point estimate of the parameters that are learned by the learning algorithm.

AUDIENCE: And so we--

PROFESSOR: Just like if you're learning a Markov model, you learned some transition and [INAUDIBLE], same thing here. All right. And this should look a lot like what you would see when you do topic modeling on a text copora, right? You would discover a topic-- this is analogous to a topic. It's a discrete topic, meaning it either occurs or it doesn't occur for a patient. And you would discover some word topic distribution. This is analogous to that word topic distribution for a topic in a topic model.

So one could then use the model to answer a couple of the original questions we set out to solve. The first one is given a patient's data, which I'm illustrating here on the bottom, I have artificially separated out into three different comorbidities, and a star denotes an observation of a data type of that one. But this was artificially done by us, it was not given to learning algorithm.

One can infer, when the patient initiated-- started with each one of these comorbidites, and also, when-- so for the full three-year time range that we have data for the patient, what stage was the patient in in the disease at any one point in time? So this model infers that the patient starts out in stage 1, and about half a year through the data collection process, transitioned into stage 2 of COPD.

Another thing that one could do using this model is to simulate from the model and answer the question of what would, let's say, a 20-year trajectory of the disease look like? And here, I'm showing a 10-year trajectory. And again, only one to three years of data was used for any one patient during learning. So this is the first time we see the those axes, those comorbidities really start to show up as being important as the way of reading out from the model. Here, we've thrown away those O's, those diagnosis codes altogether, we only care about what we conjecture is truly happening to the patient, those X variables, which are unobserved during training.

So what we conjecture is that kidney disease is very uncommon in stage 1 of the disease, and increases slowly as you transition from stage 2, stage 3, to stage 4 of the disease, and then really bumps up towards stage 5 and stage 6 of the disease. So you should read this as saying that in stage 6 of the disease, over 60% people have kidney disease. Now the time interval is here. So how I've chosen these-- where to put these triangles, I've chosen them based on the average amount of time it takes to transition from one stage to the next stage according to the learned parameters of the model. And so you see that stages 1, 2, and 3, and 4 take a long period of-- amount of time to transition between those four stages, and then there's a very small amount of time between transitioning from stage 5 to stage 6 on average.

So that's for kidney disease. One could also read this out for other comorbidities. So in orange here-- in yellow here is diabetes, in black here is musculoskeletal conditions, and in red here is cardiovascular disease. And so one of the interesting inferences made by this learning algorithm is that even in stage 1 of COPD, very early in the trajectory, we are seeing patients with large amounts of cardiovascular disease. And again, this is something that one can look at the medical literature to see does it align with what we expect? And it does, so even in patients with mild to moderate COPD, the leading cause of morbidity is cardiovascular disease. Again, this is just a sanity check that what this model is learning for a common disease actually aligns with the medical knowledge.

So that's all I want to say about this probabilistic model approach to disease progression modeling from cross-sectional data. I want you to hold your questions so I can get through the rest of the material and you can ask me after class. So next I want to talk about these pseudo-time methods, which are a very different approach for trying to align patients into early-to-late disease stage. These approaches were really popularized in the last five years due to the explosion in single-cell sequencing experiments in the biology community.

Single-cell sequencing is a way to really understand not just what is the average gene expression, but on a cell-by-cell basis can we understand what is expressed in each cell. So at a very high level, the way this works is you take a solid tissue, you do a number of procedures in order to isolate out individual cells from that tissue, then you're going to extract the RNA from those individual cells, you go through another complex procedure which somehow barcodes each of the RNA from each individual cell, mixes them all together, does sequencing of it, and then deconvolves it so that you can see what was the original RNA expression for each of the individual cells.

Now the goal of these pseudo-time algorithms is to take that type of data and then to attempt to align cells to some trajectory. So if you look at the very top of this figure part figure a that's the picture that you should have in your mind. In the real world, cells are evolving with time-- for example, B cells will have a well-characterized evolution between different cellular states, and what we'd like to be understand, given that you have cross-sectional data-- so you can imagine-- imagine you have a whole collection of cells, each one in a different part, a different stage of differentiation, could you somehow order them into where they were in different stages of differentiation?

So that's the goal. We want to take this-- so there exists some true ordering that I'm showing from dark to light. The capture process is going to ignore what the ordering information was, because all we're doing is getting a collection of cells that are in different stages. And then we're going to use this pseudo-time method to try to re-sort them so that you could figure out, oh, these were the cells in the early stage and these are the cells in the late stage. And of course, there's an analogy here to the pictures I showed you in the earlier part of the lecture.

Once you have this alignment of cells into stages, then you can answer some really exciting scientific questions. For example, you could ask a variety of different genes which genes are expressed at which point in time. So you might see that gene a is very highly expressed very early in this cell's differentiation and is not expressed very much towards the end, and that might give you new biological insights.

So these methods could immediately be applied, I believe, to disease progression modeling where I want you to think about each cell as now a patient, and that patient has a number of observations for this data. The observations are an expression for that cell, but in our data, the observations might be symptoms that we observe for the patient, for example. And then the goal is, given those cross-sectional observations, to sort them.

And once you have that sorting, then you could answer scientific questions, such as, I mentioned, of a number of different genes, which genes are expressed when. So here, I'm showing you sort of the density of when this particular gene is expressed as a function of pseudo-time. Analogously for disease progression modeling, you should think of that as being a symptom. You could ask, OK, suppose there are some true progression of the disease, when do patients typically develop diabetes or cardiovascular symptoms? And so for cardiovascular, going back to the COPD example, you might imagine that there's a peak very early in the disease stage; for diabetes, it might be in a later disease stage. All right? So that-- is the analogy clear?

So this community, which has been developing methods for studying single-cell gene expression data, has just exploded in the last 10 years. So I lost count of how many different methods there are, but if I had to guess, I'd say 50 to 200 different methods for this problem. There was a paper, which is one of the optional readings for today's lecture, that just came out earlier this month which looks at a comparison of these different trajectory inference methods, and this picture gives a really interesting illustration of what are some of the assumptions made by these algorithms.

So for example, the first question, when you try to figure out which method of these tons of methods to use is, do you expect multiple disconnected trajectories? What might be a reason why you would expect multiple disconnected trajectories for disease progression modeling? TAs should not answer.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Different subtexts would be an example. So suppose the answer is no, as we've been assuming for most of this lecture, then you might ask, OK, there might only be a single trajectory, because we're only assuming that a single disease subtype, but do we expect a particular topology? Now everything that we've been talking about up until now has been a linear topology, meaning there's a linear projection, there's such notion of early and late to stage, but in fact, the linear trajectory may not be realistic. Maybe the trajectory looks a little bit more like this bifurcation. Maybe patients look the same very early in the disease stage, but then suddenly something might happen which causes some patients to go this way and some patients to go that way. Any idea what that might be in a clinical setting?

AUDIENCE: A treatment?

PROFESSOR: Treatments, that's great. All right? So maybe these patients got t equals 0, and maybe these patients got t equal as 1, and maybe for whatever reason wouldn't even have good data on what treatments patients got, so we don't actually observe the treatment, right? Then you might want to be able to discover that bifurcation directly from the data, then that might suggest, going back to the original source of the data, to ask, what differentiated these patients at this point in time? And you might discover, oh, there was something in the data that we didn't record, such as treatment, all right?

So there are a variety of methods to try to infer these pseudo-times under a variety of different assumptions. What I'll do in the next few minutes is just give you an inkling of how two of the methods work. And I chose these to be representative examples. The first example is an approach based on building a minimum spanning tree. And this algorithm I'm going to describe goes by the name of Monocle. It was published in 2014 in this paper by Trapnell et al, but it builds very heavily on an earlier published paper from 2003 that I mostly citing here.

So the way that this algorithm works is as follows. It starts with, as we've been assuming all along, cross-sectional data, which lives in some high-dimensional space. I'm drawing that in the top-left here. Each data point corresponds to some patient or some cell. The first step of the algorithm is to do dimensionality reduction. And there are many ways to do dimensionality reduction. You could do principal components analysis, or, for example, you could do independent components analysis. This paper uses the independent component analysis. What ICA is going to do, it's going to attempt to find a number of different components that seem to be as independent from one another as possible.

Then you're going to represent the data now in this low-dimensional space, and in many of these papers, it's quite astonishing to me, they actually use dimension 2. So they'll go all the way down to two-dimensional space where you can actually plot all of the data. It's not at all obvious to me why you would want to do that, and for clinical data, I think that might be a very poor choice. Then what they do is they build a minimum spanning tree on all of the patient or cells.

So the way that one does that is you create a graph by drawing an edge between every pair of nodes where the weight of the edge is the Euclidean distance between those two points. And then-- so for example, there is this edge from here to here, there's an edge from here to here and so on. And then given that weighted graph, we're going to find the minimum spanning tree of that graph, and what I'm showing you here is the minimum spanning tree of the corresponding graph, OK?

Next, what one will do is go look for the longest path in that tree. Remember, finding the longest path in a graph-- in an arbitrary graph has a name, it's called the traveling salesman problem and it's the NP-hard problem. How has that gotten around here? Well we're not-- this is not an arbitrary graph, this is actually a tree. So here's that poor-- here's a algorithm for finding the longest path. I won't talk about that.

So one finds along this path in a tree-- in the tree, and then what one does is one says, OK, one side of the path corresponds to, let's say, early disease stage and the other side of the path corresponds to late disease stage, and it allows for the fact that there might be some bifurcation. So for example, you see here that there is a bifurcation over here. And as we discussed earlier, you have to have some way of differentiating what the beginning is and what the end should be, and that's where some side information might become useful.

So here's an illustration of applying that method to some real data. So every point here is a cell after doing dimensionality reduction. The edges between those points correspond to the edges of the minimum spanning tree, and now what the authors have done is they've actually used some side information that they had in order to color each of the nodes based on what part of the cell differentiation process is believed-- that cell is believed to be in. And what one discovers, that in fact, this is very sensible, that all of these points are in a much earlier disease stage than [INAUDIBLE] than these points, and this is a sensible bifurcation

Next I want to talk about a slightly different approach to-- this is the whole story, by the way, right? It's conceptually a very, very simple approach. Next I want to talk about a different approach, which now tries to return back to the probabilistic approaches that we had earlier in the lecture. This new approach is going to be based on Gaussian processes. So Gaussian processes have come up a couple of times in lecture, but I've never actually formally defined them for you. So in order for what I'm going to say next to make sense, I'm going to formally define for you what a Gaussian process is.

A Gaussian process mu for a collection of time points, T1 through T capital N, is defined by a joint distribution, mu, of those time points, which is a Gaussian distribution. So we're going to say that the function value for these T different time points is just a Gaussian, which for the purpose of today's lecture I'm going to assume is zero mean, and where the covariance function is given to you by this capital K, it's a covariance function of the time points-- of the input points. And so if you look-- this has to be a matrix of dimension capital N by capital N. And if you look at the I1 and I2 of the N tree, if you look at the N tree of that matrix, we're defining it to be given to by the following kernel function. It looks at the exponential of the negative Euclidean distance squared between those two time points.

Intuitively what this is saying is that if you have two time points that are very close to one another, then this kernel function is going to be very large. If you have two time points that are very far from one another, then this is very large-- it's a very large negative number, and so this is going to be very small. So the kernel function for two inputs that are very far from another are very small; the kernel function for inputs that are very close to each other is large; and thus, what we're saying here is that there's going to be some correlation between nearby data points.

And that's the way which we're going to specify a distribution of functions. If one were to sample from this Gaussian with a covariance function specified in the way I did, what one gets out is something that looks like this. So I'm assuming here that every-- that these curves look really dense, and that's because I'm assuming the N is extremely large here. If and what small, let's say 3, there'd only be three time points here, OK?

And so if you can make this distribution of functions be arbitrarily complex by playing with this little l-- so for example, if you made a little l be very small, then what you get are these really spiky functions that I'm showing you in a very light color. If you make a little l be very large, you get these very smooth functions, right? So this is a way to get a function-- this is a way to get a distribution over functions just by sampling from this Gaussian process.

What this paper does from Campbell and Yau published two years ago in *Computational Biology* is they assume that the observations that you have are drawn from a Gaussian distribution whose mean is given to you by the Gaussian process. So if you think back to the story that we drew earlier, suppose that the data lived in just one dimension, and suppose we actually knew the sorting of patients. So we actually knew which patients are very early in time, which patients are very late in time.

You might imagine that that single biomarker, biomarker A, you might imagine that the function which tells you what the biomarker's value is as a function of time might be something like this, right? Or maybe it's something like that, OK? So it might be a function that's increasing or a function that's decreasing. And this function is precisely what this mu, this Gaussian process is meant to model. The only difference is that now, one can model the Gaussian process-- instead of just being a single dimension, one could imagine having several different dimensions. So this P denotes the number of dimensions, right? Which corresponds to, in some sense, to the number of synthetic biomarkers that you might conjecture exist.

Now here, we truly don't know the sorting of patients into early versus late stage. And so the time points T are themselves assumed to be latent variables that are drawn from a truncated normal distribution that looks like this. So you might make some assumption that the time intervals for when a patient comes in might be, maybe patients come in really typically very in the middle of the disease stage, or maybe you're assuming it's something flat, an so patients come in throughout the disease stage.

But the time point itself is latent. So now the generative process for the data is as follows. You first sample a time point from this truncated normal distribution, then you look to see-- oh, and you sample from the very beginning your sample this curve mu, and then you look to see, what is the value of mu for the sample time point, and that gives you the expected value you should expect to see for that patient.

And the one, then, jointly optimizes this to try to find the most-- the curve, the curve mu which has highest posterior probability, and that is how you read out from the model both what the latent progression looks like, and if you look at the posterior distribution over the T's that are inferred for each individual, you get the inferred location along the trajectory for each individual.

And I'll stop there. I'll post the slides online for this last piece, but I'll let you read the paper on your own.