Old

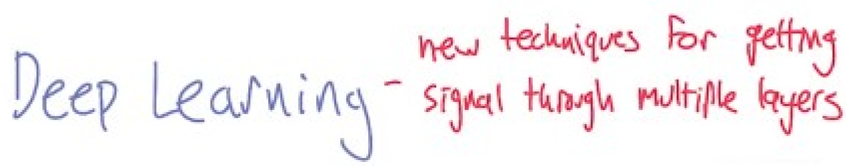
New and Interesting

1. Hi Michael. >> Hi Charles, how are you? >> I'm doing fine, thanks for asking, how are you doing? >> I'm doing well, I'm exciting to be kind of pulling the course together and wrapping it up and putting it in a box and sending it out into the world. >> Yes, me too, it's actually been fun so far. As we tape this we are about halfway through, I would say, of the first iteration of the course. And Michael kindly agreed to come down Atlanta to talk to me again. So that we could mention a few other things that we didn't get a chance to cover in the class, because we basically just didn't have time. It's not that they weren't interesting, but because this was a survey kind of course, we wanted to make certain that you had the basics. And that from there you could go off and learn things and really understand why they're interesting. So, what we're going to do is just very quickly, talk about some of the things that are either new and interesting now, or perhaps old that we just didn't have time to cover, although, may still show up on the final exam. >> [LAUGH] And plus, I believe that everyone now has a really good background for being able to appreciate those things. >> Right, I believe that's important. The main goal of the course was to give people enough of a breadth, and just enough depth. So that they can go out into the world and understand why something like deep neural networks are interesting or different, or the same, as the way we've been thinking about these problems for years. So maybe that's a good one to start with. So, maybe we can just talk about some of the things we didn't cover. So, there are two big things that are popular right now. Or that a lot of people would have heard of. >> Buzz words. >> Buzz words but in a good way. >> [LAUGH] >> As opposed to just made up stuff. Deep neural networks. >> Okay. Sure. >> Or deep learning, which I think is how most people talk about it. >> Deep learning. Sure. >> And big data. So, those are two kind of things that are out there in 2014 that people talk about. We didn't really get a chance to mention, how would you describe them in 15 seconds?

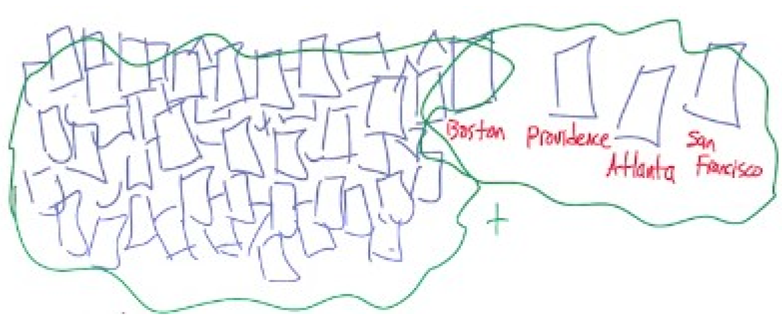
Big Data - algorithmic challenges from gigantic datasets. Linear too slow...

>> So, big data is the issues that come up when you actually have a tremendous amount of data. So more and more now it's possible to attach sensors to the world and get tremendous amount of information, either web pages or biological data. I've got a little thing that measures my steps every day. You start to pull this data off over enough people, there's a tremendous resource, but also challenges in extracting good information from it. >> Right. And a lot of those challenges are algorithmic. We talk about things like the curse of dimensionality and now we deal with exponentially. Well, when you have big enough data, the truth is even linear can be slow. Quadratic is unacceptable. >> Linear in the amount of data. >> Yeah, linear in the amount of data, because you've got so much data coming in. And, by the way, you're never going to be able to look at it again. Because, by the time you look at the first set of data, another day has passed, and you've got another several petabytes

of data that are coming in. What's really nice about big data, which is worth mentioning, I think, without going into too much detail. >> Yeah, it's only been 15 seconds. So that's it. >> Only been 15 seconds? >> Yeah. >> We got another 3 or 4, is that [LAUGH] the data, because we have access to all this data, it's fundamentally changed the way science works. So it used to be people talked about experiments and theory and I'm sure there was something else, about how science was done. >> [LAUGH] Something else. Like scientific method. >> But now one of the new pillars of science, which is finally agreed upon by the scientific community is Computation. The ability to do simulation the ability to look at the data in order to validate models in a way we couldn't do even 15 years ago. And that's really amazing, it's one of the reasons why my guess is most of the people who are looking at this first heard about machine learning. It's one of the reasons why it's so popular because it's applicable to these big science problems that we care about. Okay, what about deep learning?

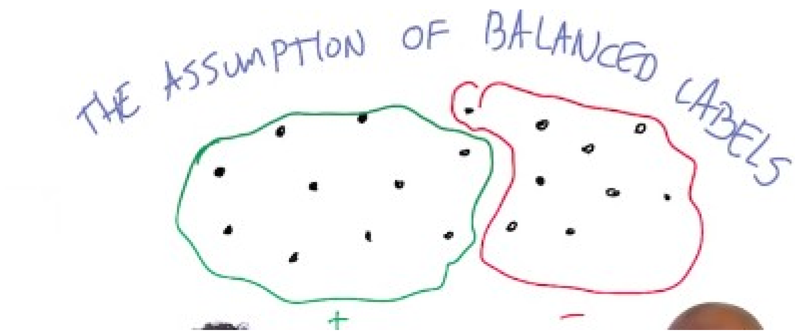


>> Yeah, so Deep Learning or Deep Neural Nets. I think in some ways it's a reboot. Is that a word that we use now? >> Reboot is a world. >> A prequel. No, it's a reboot. I'm going to go with reboot. >> Okay. Let's go with reboot. >> Of neural nets, so back in the 80s, there was this idea that you could train up neural nets with things like back prop. But it was not so clear that you could do better with multiple layers than you could with just a single layer. Even though we know that brains are organized into lots and lots of layers. >> Right. >> So it kind of fell out of favor for while, but it's back. And I think part of it is there's a new set of techniques that can be used to organize the computation in a way that you can actually get valuable information, signal, coming through at each of the different layers of the Deep Neural Net. >> Right. Does it hold onto a clever representational trick or is there more to it than that? >> I think there's a bunch of different tricks. And one of the reasons we didn't talk about them, is we don't really know them. >> Yeah, and it would take too long. And. >> Yes, sorry. It would take too long for us to learn them. >> Right. >> Teach them. >> Yeah to teach them. It certainly would take too long, because we understand them completely. Okay. >> We have deep neural nets. >> We do. Well, maybe we think we do. I guess we got things to do. Is that what we actually think we do? Isn't there a biological plausibility. I can't remember. >> Sure.



>> Yeah, let's go with that. Okay, so let's see. What else haven't we covered that would be, how about semi-supervised learning? This is something that I really wish we could have gotten into the class and maybe in a later iteration of this we'll find a way to work. >> We can put it half way between supervised and unsupervised. >> Or we could stick it in unsupervised learning, like we did randomized

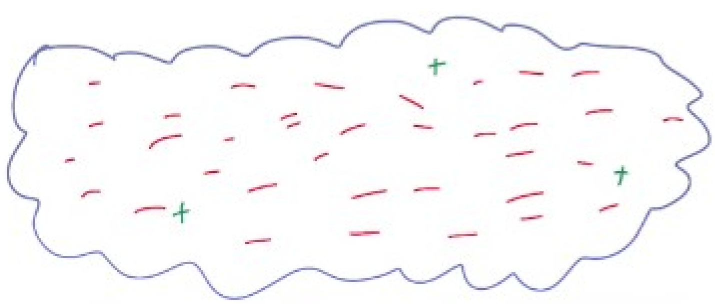
optimization. >> Yeah, it kind of fleshes it out a little bit. But anyway, the idea of semi-supervised learning is really cool. Like it's a slightly different problem. It has elements of both supervised and unsupervised. >> Like what? >> So, imagine for example, that you've got tons of web pages that have on them information about cities. >> Okay. >> And you label them, you say this webpage has information about this city. And this webpage has information about this city. And you eventually you get tired of doing that because labeling is human expensive. >> Yes. >> So if you've got a billion web pages and you've only have time to label a million of them, then there's still a factor of a thousand. >> Sure let's go with that. >> Yeah, of pages that you have information on them but there's no labels. And so I think in the early days people thought since we don't have labels, there's no really valuable information there. The idea of semi supervised learning is that you can actually extract using unsupervised methods, enough structure from the unlabeled data that when combined with the labelled data, its like having the power of a lot more label data than you actually started with. >> Yeah, that was a big deal in the late 80s, early 90s in particular.



2. So I think there's one more class of problems that we didn't really get a chance get. There are tons by the way just to be clear, algorithms and representation that we didn't really get into. A lot of them actually coming out of big data like k-d trees for example. But there's a whole class of problems we didn't really discuss that's actually quite important. And that's because it reveals one of the assumptions that we've had. I'm going to call them the assumption of balanced labels. >> Balanced labels. >> Yeah, which is related to cost-sensitive learning and a variety of other things. So implicit in a lot of the work that we talked about was this idea that if you're a big data set, about half of them are labeled negative and about half of them are labeled positive. >> Sure. >> Right? But in practice, the world doesn't work that way, and you have to actually take that into account.

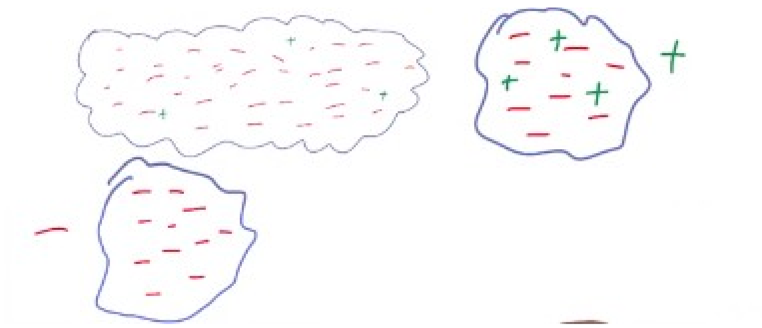
>> So, what would be an example? >> So, here's my favorite example. So imagine you have a camera. >> [LAUGH] >> Close your eyes. Now imagine you have a camera, and let's say that it's in an airport, like we are now. And people are going through constantly and in fact, you're in Atlanta Airport, which is the world's busiest airport, and there are cameras constantly taking pictures of you, and there are, and we're trying to find terrorists. So we're taking pictures of people, and of their faces say, and maybe

what they're wearing, and we just want to label which ones are terrorists and which ones aren't. Well the vast majority of people are not terrorists, we hope, even the ones going through airports. And so let's say 99% of the people that come through, you would label negative, and less than 1% you would label positive. Does that make sense as a problem? >> Yeah, though I'm thinking about this issue about most people aren't terrorists. Like I feel like. >> That most people are? >> Are there really any non-terrorists, or are we all just terrorists who just haven't gotten inspired yet? >> You've met my son. >> [LAUGH] >> So let's say terror is by some definition ruling to accept. >> Yeah, yeah. Let's talk- >> How about a threat? A threat- >> That's where I was going to go. Yes, forget whether they're terrorists or not, just In the airport, the airport terrorists, this person is threatening to the safety of the airport. >> And in fact we can back it up a little bit and say it was not even terrorists, it's just someone's that's going to cause some kind of trouble as defined by the airports. So you know we'll get irritated as they go through the line. >> Okay, that explains why the last time that we got together in Atlanta I was chased by police. >> You were chased by police. That's a funny story, he was literally chased by the police in the Atlanta airport. You got outta that one pretty well I thought. >> Thanks. >> Eventually. >> Yeah. >> If we're not happy with- >> [LAUGH] >> Anyway. >> So anyway, let's say maybe 3% of people cause trouble in a day. >> [LAUGH] >> Michael. >> Yeah. >> And let's say the other 97% don't, so if you were going to throw this at a learner, what would happen. >> Yeah. >> The best learner is probably the one that just says. >> No one will cause trouble. >> No one will cause trouble. >> because it could be wrong sometimes, yeah that's right. It'll be wrong sometimes, but very rarely. >> Yeah, 97% is pretty good, well that's the problem. That's what's called the majority classifier. >> Yes. >> And it's a baseline, which we've talked about a little bit. And the problem is that when your algorithm does 97%, that's actually not very impressive on the one hand. But there's another issue which is that. >> On this date it's not so impressive. >> On this date it's not so impressive. But this other issue which is the cost of being wrong. When somebody really is going to cause trouble is much higher. So it's actually very important that you get that 3%, you can't just treat it as 3%. So I'm just going to just very briefly go through this and then we're going to move on. There's a bunch of approaches to this. You could re-weight the way you do error, there's all kinds of clever things you can do. But there's one that I really like which is do to Paul Viola and Mike Jones, it was applied to a vision problem. It's called cascade learning. I like this because, it's a seminal paper. I like this because it is rally good work. And also because Paul Viola is one of my advisers and Mike Jones was my office mate when I was in grad school. They did all of this while I was sitting there and didn't put me on their paper. >> [LAUGH] You think you should be on the paper just by virtue of the fact you were there inspiring them with your presence. >> My [SOUND] agent lets them go up by one. >> Yeah. >> And that's important.

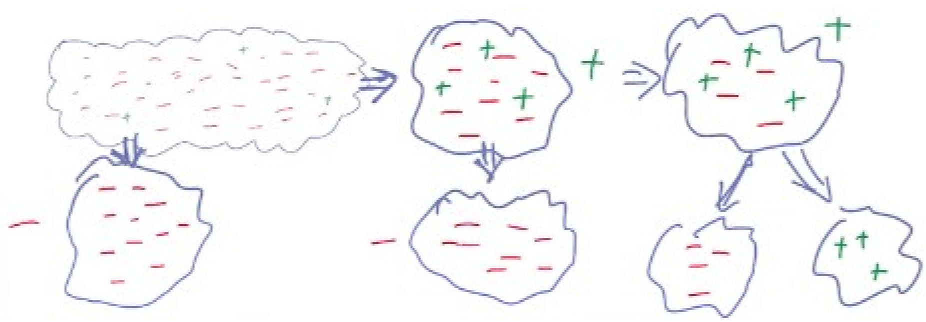


3. All right. >> Okay, so here's the idea. So just close your eyes in a minute, well don't close your eyes, because I'm going to show you this but imagine, >> [LAUGH] >> Imagine you have cloud of data. >> All right. >> Got it? >> I do. >> And let's say some of it's labeled minus. >> Mm-hm. >> In fact the vast majority is labeled minus, and just a few are labeled plus. Well if we do it as we've done before as requiring learning algorithms do, we were basically going to say all of them were

negative >> Okay, as you've said.

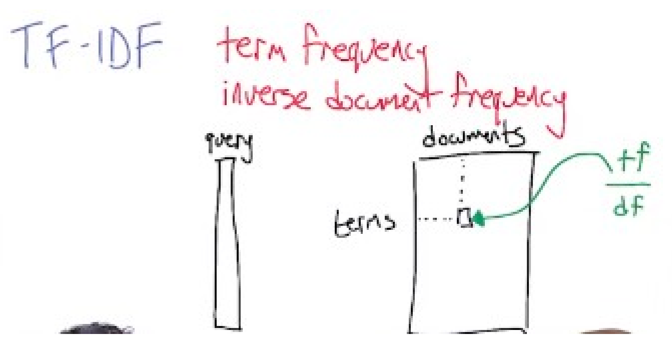


>> So, here's what were going to do instead. We're going to build a learner, that's going to label about half of them negative and half of them positive (negative 表示不是 terrorist, positive 表示是 terrorist). >> I can make one of those. >> But, it's going to have another property. >> Doh! >> So you take that cloud that I had. >> Okay. >> And half of them are negative, let's say the negative ones come down here, and the positive ones go this way. Now, here's the thing. All the negative ones are actually true negatives, okay? >> I see. >> The problem was we'll include all the positives, but also some of the negatives. So I'll have no false negatives, and I'll have many false positives. >> So in the causing trouble in the airport example, you would. Like half the people you would say these people are not a threat, and you'd just be right on those. >> Yes. >> But the other half you might say these could be a threat- >> Or even say that they are a threat and you'll be wrong for most of them, but all the ones who are threats will survive that filter.



>> I see. >> So now, that I've got the cloud, and it's over here, and I've got mostly negatives, and a few positives. I keep doing the same thing. >> Mm-hm. >> So I shove the negatives off, and I'm never wrong, and shove the positives, and it goes on and on and on. >> It sort of vaguely reminds me of boosting. >> It does, and in fact, the way that they did this in the original paper is they did boosting over very simple learners. But what's interesting here, there's a couple of points. One is, I started out with this big cloud, mostly negatives and a few positives. I kept cutting in half, and by the time I got over here, I have a much smaller set, and now it's balanced. So now half of them are negative, and half of them are positive. >> because we keep separating the chaff from the wheat. >> Right, yeah, let's go with that. And so the year with the sort of small set, and now I can apply my learner here, with the sort of 50/50 split let's say, and I actually do a pretty good job. Now there's two things that are worth pointing out here. >> Okay. >> Well, there's a zero thing, which is- >> It's like you haven't pointed out anything so far. >> I'm pointing everywhere. >> Yeah. >> Now, here's the other thing, here's the cool thing. Notice that, as I go from this data over here to this smaller, and smaller. If I'm right, and I get about half the data surviving the filter, then I can actually put twice as much learning energy in each

level. >> I'm pretty sure we didn't study learning energy. >> Well I'm thinking computational effort, right? >> All right. I see, to make it faster. >> Right. So, if I keep going down my hill- >> Because the expected amount of computation is going to be quite small, because most people come in and they get filtered in the first level. >> Right. >> It's like a log thing. >> Yeah, so the size of the data that I'm looking at is going down [CROSSTALK] >> But that means is that if I, or by a factor of two. Which means that, if I can put in a factor of two more of computational effort. >> I see. >> And big O, it's going to be the same. >> Neat. >> So. I can basically do something stupid, less stupid, and by the time it gets in I can do something actually rather sophisticated. Which I couldn't apply over here, because it would take too much computational effort. >> I see, again, the expected cost would be tremendous, but here the change in expected cost is very small, because very little of the data actually makes it all the way through that pipeline. >> Right, so that's point one. Point two is, I now have this series of learner. And over here I've got a nice learner, that does a nice job of separating everything out. But you'll notice that if I took that learner. You might say, well let's just take this learner, and I can just apply it from the very beginning (to some other data). I mean, after all, now that I've learned the learner, it's not computationally painful, I can, usually I can [CROSSTALK] >> I see because the classifier is fast, [CROSSTALK] the learning process was, okay? And that's really good, except it will do terribly, because why? >> It has a different distribution data. >> It has a different distribution. It learned on this distribution, not this distribution. >> Got it. >> So you still have to do this cascade of learners. That's why it's called cascade learners. >> I thought it was because like detergent. >> It is like detergent, it scrubs out. I don't know enough about the Cascade commercials to make a terrible pun here, but. >> All right,, let's substitute the terrible pun here. >> Leaves drops the spot, right, that's the whole [CROSSTALK] >> Yeah. >> So all the spotting stuff comes over here, and you think of those that drops the spot and you get spotting [INAUDIBLE]. >> [LAUGH] >> I'm glad we went through that effort. Anyway, so this sort of casting learning thing works really well, and it gets you to this nice little place where you've got this [INAUDIBLE]. >> Good. >> And there's all kinds of things out there like that in the supervised learning world. >> Can you name something that's had an impact on I don't know, things that people have experience with? >> What do you mean? >> My understanding is that they use some version of this, in hand held cameras, when they actually put a little box around your face. >> I didn't realize that. Although it makes sense, because the original work in fact was in exactly that space. >> Yeah. >> It was sort of doing face finding- >> Finding a face. Yeah. >> Where they used very simple kind of pixel based- >> because most regions of an image don't have a face in them. >> Right. >> Yeah. Finding a face. >> Yeah. >> Well it depends on the image that you have. I'm with you. Okay, so any way there are tons of things like that you should explore, and hopefully you are now prepared to not just explore them, but to even understand them. >> Cool. >> So that's. >> Nice.

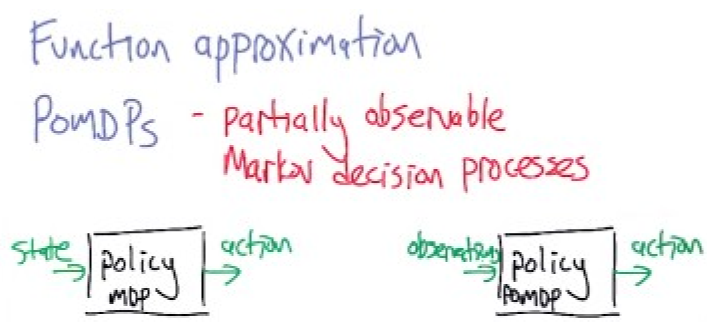


TF-IDF 就是 term frequency inverse document frequency 的縮寫.

4. All right so that's supervised learning, Michael, and there's a whole bunch of stuff that we covered.

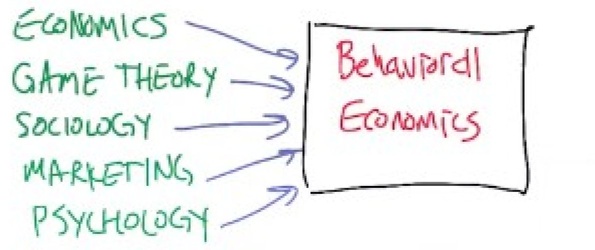
What about unsupervised learning? Is there anything there in that section that you wish we had been able to talk about, too? >> It certainly seems like there ought to be, in the sense that it was a lot shorter than the other sections. >> A lot shorter than the other sections. >> Yeah, one thing that I thought would be worth bringing up in this context is tf-idf. So tf-idf is something that I think is in one version of this class that you teach. But we didn't get a chance to talk about it in here. And so, what does it have to do with unsupervised learning, or randomized optimization? >> I don't know, Michael. What does it have to do with it? >> All right, so tf-idf, it stands for term frequency inverse document frequency. And it is a way of applying weight when you're doing, kind of a nearest neighbor operation in textual data. So, if you have a big collection of text like, web pages and queries, and you're making queries against the webpages, you might want to know how close is this query is to all the various documents that are in your collection, all of the various webpages you might want to return. And it turns out that there's lots of different similarity measures that you can use. You can use Euclidian distance, you can use some kind of dot product-y thing. >> You can use the cosine of the angle between documents. >> Yeah there we go. As a way of measuring how similar this query is to a document so we can return the most similar ones. But it turns out that there's better and worse ways of doing this waiting. So it has become apparent through the years, I think back into the 60's actually with Jerry Salton, that a really good way of doing this that's quite simple but incredibly powerful, is to say that the amount of weight that you put on the appearance of a term in a document, should be proportional, or positively related to the term frequency. The number of times that word that term appears. So if I'm searching for things about snow, and we have a webpage that just mentions snow in passing, that's not as important. As if we have a document that just mentions snow all over the place. That the importance of that word grows with the number of times that it appears in the document. >> Well that makes sense, but then that would imply that probably the most important word is the. >> Aye, yes. Well first of all, no. It's not really that important. At least as far as determining whether some document is relevant. So, exactly so. So that's the tf part is the term frequency. The idf part, inverse document frequency, says, well how many documents in your entire collection have that word in them? If it appears indiscriminately across a large number of documents then we want to down-weight it. >> Like the. >> Like the, so the appears almost everywhere. And so the document frequency, the number of documents it appears in, is huge. The inverse document frequency is therefore very small, so that gets very little weight when you're doing this kind of comparison. >> That makes sense, heck did a whole thesis about this. >> Oh, so maybe you've heard of this before. >> I have heard of this before, and it's actually one of these things that are just completely accepted. And it makes sense in the unsupervised learning case because you're effectively in the ad hoc retrieval task, as they call it. We would probably call it the Google task if it had been invented. >> You know what was funny? I worked in information retrieval before Google, and what was funny is- >> They stole all your ideas? >> No, No, No, No, No, No, No, No, Yes. No it was hard sometimes writing papers to make the case that it was important to be able to do information retrieval. Like what are we retrieving on? There's only ten documents in the world. Like it was a pretty funny thing and now its so obvious to everyone that this is important. In fact, they don't even think about it anymore because- >> It changed the world. >> It did. Okay, so that was something that we could have talked about, but mainly we didn't have time and we were tired. Let's see, we've talked about semi supervised learning already, and that is something that could have made sense in the section. And I guess, but that's all sort of the unsupervised learning space. Spectral clustering, where as different kinds of clustering we could've talked about but, we didn't >> I like spectral clustering >> Yeah, it's kind of a neat thing, and pretty good right, it does pretty well in the real world, right? >> Yeah, I think so >> Okay, so you should look that up >> What's neat about it is it actually doesn't get stuck in local optima. So it's a whole series of methods that are based more on linear algebra, where you can kind of invert matrixes and get one answer, than it is on search, and EM, and gradient decent, and things like that. >> Yes, there's a claim you could make that a lot of machine learning, particularly the unsupervised learning space, is effectively linear algebra. At bottom. >> Well, and spectral clustering, they fess up to that. >> Right, anything else we, we didn't say anything new. Is there anything in randomized optimization that we didn't really have a chance for talking about- >> I think it's worth at least mentioning cross entropy, which is a method that's really simple to implement. It's incredibly effective. Students that I work with have implemented it in a number of settings, and it just always does well. It does better than our friends simulated and kneeling and hill climbing and things like that. It does really well. Well, so here's the thing, I think it's an awful lot like MIMIC. [CROSSTALK] It's a MIMIC mimic. Yeah, very nice. But it seems to me that the two communities weren't really aware of each other. So maybe we need to put these two things together. You can be MIMIC, and I can be cross-entropy and we can figure out how they relate to each other. >> And we can cross fertilize each other. I don't like the sound of that at all. >> [LAUGHTER] Okay, so on that note, we'll leave unsupervised learning and randomized optimization. >> Yes, see and this is what happens when we're not supervised. >> [LAUGHTER] Well done.

5. Okay, Michael. So, we did supervised learning and we did unsupervised learning in randomized optimization. >> Yep. >> And the last section of the course was on reinforcement learning. >> Nice. >> Now there's a lot of stuff that we didn't cover in that section of the course. Which sort of makes me sad, because that's what I do for a living. I do reinforcement learning, a little bit of game theory. You do a lot of game theory and a lot of reinforcement learning. And there were a bunch of things that we didn't cover. Can you think of anything in particular you wish you did mention? >> Well I think the thing that most people want to know about when they hear about reinforcement learning is function approximation. The idea of can you apply this idea to something other than a three by three grid world right? So to do that you often have to use the ideas that we talked about in the other sections of the class. So use ideas from supervised learning to learn something about the environment that you are acting in. >> Or even things from the unsupervised learning section of the class. >> Like feature selection. >> Like feature selection. And in fact, the notion of function approximation is sort of a a special case, maybe, the right term. Attraction, a particular state of attraction, action attraction, this sort of notion of being able to learn about one part of the space and have it teach you about other parts of the space.

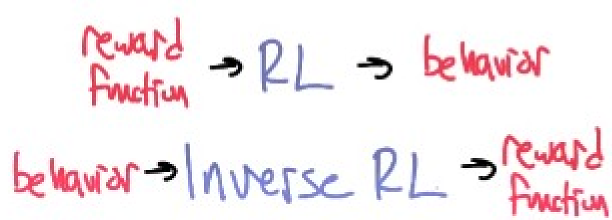


>> Generalization? >> Generalization, in general, that's a big part of making reinforcement learning work in the real world. Speaking of which, there's POMDPs. We didn't talk about POMDPs- >> Yeah, I get asked about that a lot. So, should I say what it is? >> People on the street walk up to you and say POMDPs what's that all about? >> It's surprising. >> So why don't you explain what POMDPs are? >> Usually it's after class, when I'm talking about something else. >> Oh, well that I believe. >> So partially observable Markov decision processes, right. So when we talked about without the PO. MDP, yeah, all right. So when you talk about MDPs, the agent always has complete information about what the current state is. That's the state is what you're using to decide what to do. The policy maps state to action. But in reality, you don't really have complete state information. If you're a helicopter and

you're flying through the air, you have to decide what to do based on what you can sense right now. And there may be even uncertainty about what's actually going on around you. >> Right. >> And so, you can't just use state information, because you don't have it. In the POMDP world, there's a separation between what the actual world has as its current state, and what the decision maker knows to be its perception of the state, and those can be out of whack.



>> And that makes sense, I mean we live in a world where we don't know what is happening for every atom in the universe that might have actually met. So you know there's a whole other class of problems that I wish we could have talked about that we didn't. And those are all the ones that involve humans. >> Yes. >> So a lot of game theory is mathematical and abstract. But really it's about trying to understand human behavior. Right. In fact there's been a bunch of work done on bringing in behavior into the game theory and game theory into behavior and marrying economics, game theory, sociology, marketing, and- >> Psychology, right, yeah. I mean, behavioral game theory is one of the names for that. Though neuroeconomics also comes up as a word that just seems like it shouldn't be a word. >> Yeah, that feels kind of made up. Except of course- >> I will pay you in brain cells. >> [LAUGHTER] I'll get those brains. >> That's the currency that zombies use. >> [LAUGHTER] Okay, so there's a lot about that with people with game theory, and even in reinforcement learning, there's been a huge move. Actually, dating back decades, but really has exploded over the last four or five years. >> Well in fact, what is the name of school that you're in at Georgia Tech? >> The School of Interactive Computing. >> Yeah. That's it. Interaction turned out to be a really important thing to some people. >> Right. So, a lot of my work and reinforcement learning, in particular, has been about bringing people into the loop. Learning from people, watching what people do in order to do feature selection, or state abstraction, or just simple learning. >> Some of my best friends care about people. >> Are you saying I'm your best friend, Michael? [SOUND] Very good, very good. Okay, so there's a whole bunch of stuff there. [CROSSTALK] >> No, I said some of my best friends. >> [LAUGH] >> You are some of my best friends. >> I am some of your best friends? >> I don't know. But the point is that there's this interesting line that happens when you start thinking about how these learning systems are interacting with people either once they start to behave in the world or just during the learning process itself. >> Right, because in fact, in practice that's how we teach it. So obviously we can talk about this forever. >> Yeah. I'll just mention one, let me just mention one buzzword. >> Okay.



>> So, inverse reinforcement learning is one that I particularly like. We talked about reinforcement learning in the class, where you take a reward function and an interaction with an environment and you create behavior. An inverse reinforcement learning goes in the other direction. No, you start from

observations of behavior, interacting in an environment that some expert is doing, and you try to guess what the reward function was that, that expert would have been using. >> Or a reward function that's consistent with that behavior. >> Exactly. because you can't know, really how rewarding is it to say, get groaned at when you make a pun. But you can tell that it's obviously more rewarding than not. >> Right. But reward function, and reward function times seven is basically the same thing. >> Yeah, because you can scale them and it doesn't change what the behavior looks like. That's right. So what we take away from this is some representation of the motivations, desires, of the individual who demonstrated the behavior, that we can then transfer into new environments and get good behavior out of that.

>> You know, I think it's broader than that. It's even broader than that, which is really you can think of this whole learning framework as kind of programming framework. Software engineering. Where reinforcement signals are the mechanism by which you program the agent in order to get some particular behavior as opposed to simple function calls. >> I like that topic, in fact, I think we should write a grant proposal about that. >> Let's write a grant proposal. Maybe a paper in triple x. >> Nice. >> Let's do that. So look I think it's obvious we can talk about reinforcement learning forever even though we didn't spend a lot of time on it in the class. Because there's all these basic things you really have to get about supervisory and unsupervised. >> But how can we get all this information out there? >> I can think of one way. >> Mm-hm. Direct brain interface. >> No. >> Okay. >> I mean yes, but no. >> Okay. >> We could do another class. >> You mean, like together? >> Yes, we could do a whole class on reinforcement learning, game theory and explore all these ideas. >> But who would even want to come to such a class? >> Everyone because we're going to make it required [CROSSTALK] >> Oh! That's great! >> Mm-hm. >> Let's do that. >> Let's do that. >> Let's make a class on reinforcement learning. >> Yes, we'll do a nice three hour course on reinforcement learning and game theory, and we can come back and do all the same thing. We can use the same editor. >> Yeah, though it's very possible that he'll cut this out. If we even talk about it. >> [LAUGH] No we'll use the same edit, push car will be there. It'll, we'll get the band back together. >> Get the band back together again, that sounds really fun. And the groupies can be the same. >> Yes. We can have exactly the same set. So, you want to do it? >> I think I would. >> Let's shake on it. >> Let's do this thing. >> Done. >> Alright. >> Well, then, see you in the next class. >> Yeah, we'll get started next week.