These transcriptions may contain errors, especially in spelling of names. These are unfortunate, and we regret that we do not have the resources to fix these errors. Still we believe these transcripts will be valuable to many users.

Link Mining

Lise Getoor, University of Maryland

>> So, I'm coming from the computer science perspective and I was going to say this was going to be now for something totally different, but I think, actually, part of my message is going to be very resonant with the message of drawing a picture and kind of understanding, you know, what your domain is. And I realize that, you know, this is probably the most sleepy time talk [laughter] flat there is. So, I hope that we can make it interactive like any good academic, I have way too many slides, but I don't have any compunction to get through all of them. So, I don't--I don't know you as an audience that well, so as much as you can give me feedback about, you know, where you have questions, where I'm being unclear, where something matches with something that you're interested in, that will help to keep us all awake. So, Link Mining. So, alternate title for this talk is really, okay, what can machine learning, statistics, data mining, what can it bring to you? What kind of toolkits are out there? And what should you know about them so that you know what's easy to do and what's hard to do, what's research and what you can just get something off the shelf to do? And so, obviously, in an hour talk, I'm not going to be able to kind of give you all of the details of this, but what I'm going to try and do is give you kind of a sense of the lay of the land so that you know where to potentially go look when you have a problem with your data, when you have certain kinds of issues. For example, the kinds of things that you can do with all of these kind of tools are first off, make predictions. And so, clearly, I mean, in the talks already, there's been some focus on how do I predict things about users, how do I predict things about content and so on. Also, if your data for some reason has some missing values, which is very common, you can actually use similar kinds of techniques to fill in the missing values, and that may make better predictions and make your tools and your research more useful. Another important aspect is figuring out what's weird in your data. Now, what's weird in your data could be really interesting because it's a scientific discovery. So, now, I found this anomaly, or, unfortunately, about 90 percent of the time, it's really that there's some error in your data, and it's really good to look for those errors in the data so that you don't get that the average age of your musician is a 150 years old, or understand why that makes sense in your data. Then there's other things like finding patterns, identifying clusters that are all kind of useful things that machine learning can do for you. And in general, traditionally, in machine learning, it kind of group the top three as things that are--go by the name of supervised learning. So, the notion that you have some training data that has the correct labels. From that, you're going to learn a model. Then you're going to apply the model to new data and make predictions. That's--it generally goes under the term supervised learning. There's also a full area that goes under the term unsupervised learning where nobody gives you--there's no teacher. There's nobody telling you the right answer, it's just you kind of exploring the data trying to find patterns. And, interestingly enough, if you go to machine learning class, an intramachine learning class, it will make a strong distinction between these thing like either you're in case one or you're in case two. But in machine learning research now, there's a real interest in kind of how do you mix these things, how do you kind of get a few labels, maybe crowd source labels and mix that together with your pattern lining thing and so all of those are very active areas of research. But just kind of get a sense of where you guys are in terms of your familiarity of machine learning, I saw at least one person nodding when I said supervised machine learning, so that's good. So, if someone said, "What are the top five machine learning algorithms?" What would you throw out?

>> Do you mean clustering?

>> Clustering. That's good. Say again?

>> Give sampling.

>> Give sampling. Give sampling is a method for doing imprints, but it very much fits with the prediction. It's a way of making predictions. Yeah. Yeah.

>> LDA.

>> LDA. So, which LDA are you talking about? So, LDA-- [Inaudible Remark] Say again? [Inaudible Remark] Okay. Does anybody know the other LDA? [Inaudible Remark] Linear Discriminate Analysis. So, we have two models that have the same initials that are very popular in their respective communities. Other things? [Inaudible Remark] Yeah. Two big ones. Neural Nets and Support Vector Machines. Any others? [Inaudible Remark] Yeah. Near and dear to my heart. My thesis is in graphical models, and so on. So, yeah. So, there's tons and tons of examples of different models that people use. Probably, the big one that we didn't mention that actually is very popular is decision trees. And decision trees have the nice property that they can be very interpretable. But there's a whole class of things. But most of the work in machine learning tends to assume that you have a flat representation coming in. So, basically, that you have input vector and then you're trying to predict something on it or you're trying to cluster these input vectors. So, what I mean by link mining is link mining is applying these machine learning methods to settings where you have graphs or networks. And the important thing when you have graphs or networks, you really need to think about what's in your domain. So, what are the different kinds of entities? Usually, there's people in there somewhere or users, and so on. But what are the other kinds of things? What are the messages? What are the documents? What are the places that people can go? What are the groups that people can belong to? What are the organizations and so on? All of that is actually part of this picturing your domain and really thinking it through. Then, of course, there's the relationships. So, what are those links? Of course, friends, followers and so on. So, the fact that your data is this giant heterogeneous mess of different kinds of entities and different kinds of relationships and, you know, in computer science, we call this--one thing you could call it is a data model. So, thinking of what the data model is or the area that you're doing researching I think can be a great benefit to building better models and understanding the kinds of questions you can ask and so on. And so, here is a little example, it's just a fragment of--in social media. You know, what are the kinds of relationships that you can have? Well, there's all kinds of user-user relationships that you could model. Some of them are directed, some of them are undirected, some of them are kind of constructed. So, this notion of things like co-edits and co-mentions, two people that have mentioned the same thing, two people that have edited the same document, even though you wouldn't necessarily have that initially in your data, it may be something really useful to construct because those kinds of things can lead to very predictive models. And then, of course you can have relationships between user documents and they don't have--even though we like to draw them as binary because it's easier in our graphs oftentimes they are triples so a user, a query, and a URL, a user, a tag, and a document. So, you shouldn't restrict yourself to having to have just binary relationships, things between two entities. So, I encourage you in your domain, try and kind of build a model, a picture like this, and that may help you. So, I'm going to talk about a couple link mining tasks and algorithms but first, I want to kind of review collection of them. And the simplest one, I'm going to call node labeling. And node labeling is a notion that you have some entity, you're trying to predict some label for it. So, in this example, I have Harry. And what I'm trying to predict for Harry is his political persuasion. And so, I want to kind of go through and think about like what are all the kinds of features? What are all the kinds of information that you could get from social media that could help you make this kind of prediction? So, what do you guys think? What would you--what would you look at?

>> [Inaudible] political persuasion with his friends.

>> Yes, that's definitely a big one. So, looking at political persuasion of friends and we're going to see over and over this with what you guys I'm sure are familiar with is homophily kind of principle, and how do you decode homophily in different ways in these kinds of models? That's it? Yup?

>> The text are dependent on that person's posts?

>> Exactly. Yeah, so-- [Inaudible Remark] Say again? [Inaudible Remark] Yes. [Inaudible Remark] Exactly, definitely. And, if those pages have with them a label of whether what political party they are associated with and so on, yeah. Other-- [Inaudible Discussion] And so, all of these things, what I want to kind of give you is the tools to think about. I don't have to make a model that necessarily assumes it's just one of these things. I can build models that can stick all of these in and then see which one ends up being most predictive. So, maybe the music groups is the most predictive and I don't even have to worry about going and crawling to get the friends and so on. And being able to kind of explore those things and tradeoff and see which one is going to be most useful and which one is easiest for me to get the data for. So, things like their friends, TV shows they watch, people they don't like. And then this helps me kind of use the network context to predict an attribute. But now I can do something where I actually can then once I predict an attribute for Harry, that may give me the information that helps me then predict an attribute for another person in the network. So, there's this ability to kind of cascade where I make a prediction for one and then that helps infer a prediction for another one and I'm going to talk about methods for doing that. So, node labeling is definitely a big one. And you can think of sentiment. Sentiment is also in this theme. So, you're trying to label the sentiment of a document in a document collection and so on. The other--or another one is let's change things a little bit and say, "Well, we're trying to predict whether or not these guys are friends." And, of course, you could use this as a recommendation. So, recommending who you should become friends with because the algorithm is saying, "Well, you should like this person." So, what kinds of features would you guys use here? So, here the distinction as you now have two entities and you're trying to predict something about those two entities together.

>> Network structure?

>> And, what kinds of things about network structure?

>> You would look for holes. So, for example, there is a nearly complete collection [inaudible].

>> Yeah, so all kinds of things and look at the net--local network structure between those two nodes can be predictive. Other things?

>> We just call them attributes.

>> Yeah. So, looking at matching attributes. So, for example, you know, are they of the same political persuasion? It can be something that predicts whether or not there should be a link. Other things?

[Inaudible Remark]

Yeah, so, definitely, things like preferential attachment take into account the centrality of the nodes. So--and it's interesting and some domains, two central nodes maybe more likely to be linked, but in other domains, like the one that I'm thinking of is a web page classification. Web pages--professor web pages, they will point to student web pages but they never point to other professor web pages. So, you know, taking into account, depending on your domain, the structure can fit in, in different ways. So, it could be that they're members of the same group. It could be that they went to the same school. It could be that they have lots of friends in common. All of these are things that you should be able to encode in a model

that predicts whether or not there should be a relationship. Then the next one that I want to talk about is one that maybe you guys won't be so familiar with but I actually think it's hugely important and if anything in this talk, this is someone that I want you to remember and think about if it happens in your data because it's really important. And, it's kind of a funny problem because it's entity resolution, and I'll explain that in a sec, but it goes by a bunch of different names and different areas which is really funny because entity resolution is about resolving names to entities and figuring out if they're the same and the fact that they can even get the same name for the concept is really entertaining. [Laughter] But the abstract version on the problem is, you know, in the digital world, we have all these different, you know, representations for people. You know, it could be your user IDs on the different social media sites and so on. And in the real world, there are different actual identities. And now, I know I'm treading on kind of shaking ground here because the whole notions of identities and having multiple identities and exactly how you model this but this is a computer science view. We think there's just one identity. [Laughter] But there's a couple of variations of the problem. So, one is just that, okay, you have all these things and it's a duplication problem or a clustering problem. So, I want to cluster all of the mentions that are referring to the same underlying individual. In other settings, it's more natural. And this is the one that I'm going to talk about in algorithm for how to do this. In other problems, it's more that you have two databases and you know that there aren't any duplicates within the database so all you have to do is do the matching problem across the database. And those--and they're turning into a slightly different algorithms. So, it's good to kind of think which setting are you in. Are you in that setting where you're clustering mentions? Or are you in a setting where you're matching mentions? And then another variation of the problem is that you have something like a dictionary that's clean and that one does tell you the set of real world people out there and all you have to do is match to that. That makes the problem much easier. This is something that comes up all the time in catalogs, products. If you have one clean hierarchy and then you're looking at, you know, eBay and trying to match to that, that can make the problem easier. So, thinking about what setting, but let me try and illustrate how important the problem is. And this is my favorite example. Ben's probably seen it ten times already, but this is actually is a co-autograph. So, the nodes are authors and the links are the fact that they coauthored the paper together. And this is actually from the InfoVis Challenge in 2004. They were using it for data visualization challenge. And the thing that was interesting about this data is that they said that it had been hand cleaned extensively. They were not suppose to be any issues in the data. But if you look at the data for a little bit, you start seeing that there actually are a fair number of problems with the data. And, in fact, if you look at like the before entity resolution network and the after one, they are completely different networks. So, you think of any of your network analysis methods, any network statistics that you compute on this graph are going to be completely bogus, you know, the degree centrality, the path lengths, everything is wrong about it. And then on top of it, just in terms of the story it tells, this one kind of looks like a giant spaghetti mess versus this one tells a really clean story, you know? Here is this nice, clean coauthored [inaudible]. And so, one thing I really want to emphasize is you should think about your data whether or not there's a chance that it has this property that you have duplicate nodes referring to the same real world entity and take care of that before you do any further network analysis. And then the last kind of task that I'll mention in general, but I won't go into details is this clustering problem our group detection community detection and network [inaudible]. Another whole class of algorithms. And I'm sure between the four that I mentioned, node labeling, link prediction, entity resolution and group detection there's lots of other ones I haven't talked about dynamic networks and so on. So, now I'm giving you the 30,000 foot view. So, any questions at this point? Yeah?

>> I'm curious, what are the underlying structures that you use when you do this kind of analysis, crafted basis, use [inaudible] basis?

>> So I'm going to go into the methods a little bit, but still, I'm abstracting so that people use all of the above that there's a lot of folks that use relational databases but then they figure out that maybe they should use a graphed database because when you're doing these things, where you're doing a lot of falling

hops self-joins, self-joins are not good to do in relational databases. So, you should think about the features that you're going to use. I'm going to talk a lot about the features. If you have a lot of self-joins, that's a signal that you might want to think about using some special data structures for it. Yeah?

>> This maybe out of order, when you're trying to mix methods that is you have certain number of attributes available for you to--for each node and you have an [inaudible] structure, instead you could try to do both at once. Do you flatten the problem and [inaudible]--

>> Actually, you totally are getting to actually my next slide, but this is the interesting aspect exactly how to flatten and I'm going to talk about first, the, probably, the most naïve way of flattening and then get into--actually, a lot of my work is how to interleave all of these things but I won't get to that I'm sure. Okay, so for node labeling and link prediction, I'm going to do these algorithms for these at the same time. They're closely related enough that--and I want to repeat it for each of them. And I'm going to start of with the simplest version which is the relational classifier and then I'm going to talk about something called collective classification. So, in this slide, it's a very abstract slide. I'm going to try and go through it slowly. I want to make sure that you understand it though so ask me questions as we go along. It's an attempt to, like, in one slide, capture this flattening process exactly. So, in this, this is kind of an abstract version of a problem where I have three different kinds of entities and I have someone just between them. And I need to take this and smoosh into something that I can feed into a machine learning algorithm, in particular, a machine learning algorithm that takes this input vectors or arrays. So, what I'm going to focus on is predicting an attribute, first, the node labeling problems. So, predicting an attribute of some of the entities. And in particular, I'm going to talk about predicting an attribute of the green nodes. So, everybody's following the color sees five nodes or those five lines in the center. And the pink things at the end with the question marks, those are the things I'm trying to predict. So, I'm not--example from before, it might be that I'm trying to predict political persuasion. And I'm going to basically take the network structure and flatten it into a set of local features and the local features when I mean they are just any kind of attribute that's associated with the node itself. So, for political persuasion, maybe gender, maybe income level, things that are associated with the person itself. And that kind of is vanilla Machine Learning to take in the attributes of the entity. Now, I'm going to do this thing where I'm going to construct the relational features. And the relational features are the parts that use the network structure. And those thing can be as simple as counting the number of neighbors, counting the average value of some attribute that's linked by some neighbor, but I'm going to compute a bunch of these things and then I'm going to have a flat vector for each of these five entities that I want to predict something about. So, questions here? I know this is very abstract. So, this is one version of the problem. Now I can do a similar thing for my link prediction case. So, suppose I'm trying to predict links between the green nodes, well, what I'm going to do is I'm going to take the cross product of every green node with every other green node. I'm again going to construct this vector. This vector can again have the local attributes associated with each entity, but in addition, it can have what--I'll call this matching attributes. So, that was discussed earlier when we are talking about potential attribute types. And they can be things like how close the attribute values are, whether they're the same. And then other kinds of relational features like counting the number of shared neighbors, the number of shared friends and so on. So, I'm going to get out again a vector but now it corresponds to these two things and that pink box there is predicting whether or not a link exists between them. So, once I've done that, then I've represented these things as stifling features. The instances are treated independently of each other which is probably not correct, but, actually, it still works pretty well so it's a good thing to start off with. I compute these relational features and then the cool thing is I can throw any classifier at it. So, of the things that we talked about, [inaudible], Neural Nets, Support Vector Machines, Decision Trees. Once I have done this flattening, I just use existing machinery for this. So, that's really cool. The art however in all of this is the construction of the features. So, being the domain expert that knows what are useful features to encode is really important. So, while a lot of work in machine learning is very focused on the different exact method, you know, which mathematical optimization are you using, in reality I've seen this over and over again. You

look at the two systems and it's while one was using this feature, and if you did that feature to the other algorithm, it will do just as well. So, something to keep in mind. How am I doing in time?

>> We got plenty, 20 minutes and then time for questions.

>> Okay. So, just briefly to say two studies--two examples of many out there that do this kind of relational classification, one is actually predicting at click-through rate on web pages. And this is based off of a paper. I'm not going to go into all the details and as I was looking through the slide this morning I was realizing that I actually don't remember some of the details about it. So, don't probe me too hard on this. But, hopefully, it gives you a sense of the kinds of things that you can do in a web domain where we have an ad here and we're trying to predict the click-through rate. So, how many people will click on this ad? How do we estimate this? Well, we're going to kind of view this whole network of how can we draw in information that will help us do a good job at predicting this. And the standard way of doing this is using the bid terms for this add to then get to other ads that have the same bid terms. And, for example, doing something like averaging their click-through rate using that as some input to the--making this prediction. Using the average click-through rate of related things through synonyms and so on, using information about the number of web pages that contain these bid terms, the number of queries that match these things. And you can kind of construct these rich information that you can then feed into your algorithm, try and fit a model that will predict this. And this would be probably a regression model, so it would be predicting the number of click-throughs having done this flattening. Another case study is we're doing link prediction. And this is predicting friendships. It's actually join work with Jen Vulvach [phonetic] from several years ago where we we're looking at pet works. So, Jen has done a lot of work in pet social networks. So, Dogster, Catster, and Hamsterster or something like that. [Laughter] I always remember that the distinguishing thing that she said about Hamsterster was that most of the hamsters were dead. [Laughter] But anyways, so you can say what you want, these are really human constructed pet works and so on. But what we were looking at is trying to predict whether there would be a friendship link between two pets. And you can do things like do these matching predicts like say, well, they're at the same breed maybe there's more likely to be pets. But then you can do these all--these kind of structural features like looking at first not just the count of the number of friends they have, looking at account and the overlap of their friends. And these networks, they also were members of families, so you could say something about, you know, how friendly they were with their families and add that in as a feature. And you could finally, kind of--often times what you want to do is look at kind of the proportion overlap, kind of the correlation, coefficient or the Jaccard Similarity between the links that actually exist in the network versus the links that could exist. So, that's a common feature. But these relational features feed into a link prediction algorithm that works surprisingly well at predicting friendships between pets. So, the key idea in this relational classifier work is a construction of these features. And so the features can be attributes of entities, match predicates attributes or related entities, some sort of structural features or based on overlaps in sets. And you can go a long way with just this simple setting. But the next idea that I wanted to get to is collective classifiers. And this is the idea where we're going to extend our relational classifiers by basically being able to propagate information. So, I'm going to make a prediction, once I've made that prediction, that's going to feed into other predictions that I make. And let me illustrate through a cartoon how this works. So, in this setting, I'm trying to denote a case where I'm--the labels of the nodes are these three colors up here. So, think of it as a document classification problem where I'm trying to figure out the topic of the documents either with pink, purple or blue. And the way the algorithms work is first off you give me something that's fully labeled. And from this, I'm going to learn a model and then I'm going to apply it to a new unlabeled network. Now, that is not always a realistic setting. There's a lot of work in kind of doing this more semi-supervised, but let's keep things simple here and say, we have one fully labeled data set. I'm going to learn a model there. I'm going to apply it in a new setting. So, when I apply it in a new setting, I have this new network that has no labels. And so I'm going to kind of bootstrap and get some initial labels for this using just the local features, none of the network information, to get assignments. And so the local features that I would use here would probably be the

words in the documents. So, just the document content figure out what initial topic labelings I get. And I would go through and say, "Oh, I get this assignment choosing the most probable label." Then what I do to actually start propagating information is I'm going to iteratively update the category each entity, but now I can use the predicted labels of its neighbors to make that prediction. So, I can go here. Again, I might be looking at the categories of my neighbors to make the predictions and I iterate until I reach some sort of fixed point where there's no more changes in the labels. And so this kind of algorithm while simple is a powerful way of improving the accuracy of a relational classifier. And making it so that it's not making each decision independently, but it's actually more of a joint model, yet at the same time it's very tractable to do something like this. So, the key idea here is to have this ability to propagate things. There's a lot of variations. So, here's where the Gibbs sampling comes up. There's a Gibbs sampling approach to doing this. And there's a lot of work going on in this area. So, questions about collective classification? You get the basic idea behind it, right? This propagation that's happening. Yeah?

>> Why would you choose to use the propagation model versus the flattening model?

>> So the flat model is always the baseline that I would try using. So, I--because it's simpler, I would start with that. In the case though where you have a lot of--where you really have this whole unlabeled area of the network that you want to make predictions on, that's the case where the collective classification is really going to help you. And it's important to realize some domains that's important to use and some domains you can just get away with doing the relational classifier. Okay, so, entity resolution. So, I want to talk briefly about entity resolution. So, this is the problem of mapping the dimensions to real world entities. And the two kind of things to distinguish here are there's an identification problem which is basically figuring out all of the aliases that refer to the same underlying entity. But then there's also an interesting disambiguation problem. The disambiguation problem is the fact that in some cases I may have exactly the same string. So, I have this Jay Smith here, and this Jay Smith here, in one case it refers to one entity and in another case it refers to another entity. You know, how am I going to figure that out? I need to encode enough context to make this decision. And the kind of information I can use--this is actually from that little fragment from before where the square nodes denote the two nodes that I'm trying to decide. Are they referring to the same entity or not? And then the way that I've drawn this, this is based on work with Ben Shneiderman and colleagues on D-Dupe, and taking it out of context as a tool but it's to show the shared relational context in the center. So, these are the coauthors that are in common. And then on the sides are the coauthors that are not in common between the two or at least that I, at this point, think are not in common. And so in this I can see, okay, here's two names. There's one character difference between them. They have some shared coauthors. And in fact it turns out that they do refer to the same entity. Here's another taste where I have two names, one character difference between them and they have no shared coauthors. So, I can quickly kind of see, okay, these probably aren't referring to the same entity and it turns out in fact they're not. But then actually the most interesting thing kind of like the collective classification is there's also a propagation effect that can happen here where I can figure out say that these two Elman Dorf references are the same. And then once I merge those, that gives me additional evidence that helps me merge the singer references where the singer references, you know, maybe those names are so common, I won't merge them without having this additional information. So, again there are algorithms which allow us to do this kind of collective entity resolution propagating information. And the way they work--I think I'm going to skip through some of this animation for time is to basically form--and I'm sorry of course, I leave on the math slide but the idea is simple. I'm going to use these kind of features that I talked about before. These relational features and I'm going to basically predict a link where that link is predicting whether or not they refer to the same entity. And I'm going to go through and I'm going to measure the--I'm going to do a clustering based on a similarity function I defined on these attributes. And I'm going to perform a greedy agglomerative clustering where I tradeoff the similarity of--where did my mouse go? Okay. The similarity of the attributes in a cluster, so, the set of mentions, and the similarity of the neighborhood for the things that they are connected to the coauthors that they are connected to. And then I'm going to

basically go through and repeatedly merge things that are most similar to get out my clusters. So, this will solve the entity resolution problem. And we have some evaluation data sets that I'm not going to go into the details here just to say, one is a computer science data set, one is a physics dataset and one is a biology data set. So, put those in your mind and think about your stereotypes of these things. We're going to compare several algorithms and, you know, we're computer scientists and we're talking about algorithms, so of course, the point of this slide is to say, well, there's these baseline algorithms and our relational clustering, one does the best and I haven't told you what this measure is, but it's something related to accuracy. So, that's cool. [Laughter] This is your pattern for a computer science machine learning talk. But the question I have for you guys is forget looking at it as an algorithm comparison. Look at it and instead as a column comparison, we're basically going from something that only uses attributes to something that uses the network structure going down the column. In the first case, computer science, we get a little bump up from using the network structure. For the physics data set, actually, it's a big enough dataset that--that doesn't look like it's accurate but it does give you a lot more example, versus the physics data--or, sorry, the biology dataset. That's the one where we're really getting a big benefit from a network structure. So, any theories for why?

[Pause]

>> The nature of the names of the authors.

>> So, there is definitely something about the nature of the names of the authors.

>> The number of coauthors in biology.

>> Number of coauthors. And those are the kind of two big things. So, it turns out that this computer science dataset is just not that ambiguous in the first place. And computer scientists are loners. They publish with like two people, three people [inaudible]. I say but in humanities I had heard that you're--you have to publish alone but since--sorry. Anyway-- [Laughter] Physics, there are more coauthors, but biology has the two things, the number of coauthors, the average number of coauthors is much more. I think it was like ten, average. But then they made this a challenge dataset by focusing on Asian authors and initializing the first name. So, by the time you initialize an Asian first name, you've really made it that you have to use the network structure to disambiguate. But the point here is understand your domain. Understand whether you are more in the first setting where, oh, you can just use local information, or you're in the second setting where you really need the network structure or the third one where you absolutely have to have it. So, let me briefly say something about the flip side. All this is wonderful if you're trying to do personalization. So, I'm trying to know as much as I can of value to get better recommendations for music and search results and so on. But it's also kind of creepy if you think of it from the other side. Well, all of these things basically correspond to something that I can predict about you. And one of the one's entity resolution is basically identity disclosure. So, how easy is it for me to map from you're user ID for your anonymous query to the real world person that you correspond to? Attribute disclosure, you know, political persuasion can be a sensitive thing. And certainly sexual orientation and so on, there have been lots of studies that show these are very easy to infer even if you've hidden your attributes. One piece of work that we did--so you're familiar obviously with the homophily one, that if you know this label for you friends, you can infer it. But even on a site like Facebook where you can control the privacy of your attributes at least used to be the case that one of the things you didn't have control over is if you were a member of a group. It was the group owner that could publish who was a member of the group. And so even if you've carefully hidden all your attribute information, by the fact that you're a member of a group, if that group is a group that's very homogenous in a particular sensitive attribute, it's very trivial to infer that so taking that into account. And so, there's a variety of mappings from link mining to these kinds of things. So, in my group we do a lot of stuff that actually has to do with networks and analyzing them more than just these tools. If you

want to look at them, go to my web page. In particular I feel like I can't go to an ACIL supported symposium without mentioning. We do a little bit of work in visual analytics usually with a lot of help from people. So, I'm very interested in kind of where machine learning meets visual analytics to have confidence in these predictions because a lot of times the statistical confidence you have in these things is really low. And so, yeah, it can help you rank something, but then present it to the users so they can say, you know, "Oh, you just totally screwed up on that one." And then feed that back into your algorithm to improve your algorithm. And so the conclusion is that link mining algorithms can be very useful in this space. So, be aware of them. There's a lot of active work in this space. And despite the fact that sometimes you still see algorithms that really talk about just a single kind of node and a single kind of link. We really need things that support multiple kinds of relationship, multiple kinds of entity and so on. And these kind of collective algorithms are interesting because they can propagate information in useful ways. And there is a lot of pitfalls to be aware of. So, the statistical confidence you have and the privacy, but then there is this kind of tradeoff with the benefits of improved accuracy of models. And so, one needs to be kind of cognizant of these two things going on. So, thanks.

[Applause]

>> While you're coming up with one I have one that, you know, you said that you show the results to the domain expert and then they can say, "Oh, you got that one wrong." But the most common thing that we see when we show data mining results or statistical analysis is the domain expert says, "Huh, how come you got that result?" And it's very unsatisfying to say, "Well, the statistics did it." So, Decision Tree models gives you a handle on it, but the methods that really work in the interfaces I think are most successful which I'd like to encourage are ones which enable the explanation of the sensitivity analysis of the variables and which factors led to--most led to a particular selection.

>> Yeah, yeah.

>> So which of the methods are amenable to providing the kind of feedback which gives understanding not just results?

>> So, I totally agree with this and I think that to cast dispersion on my colleagues that a lot of people in machine learning, you know, they want to twiddle with the math and they're not interested in this aspect. But, for example--and it's a hard area to get into because you need so many different pieces, but, you know, this is selling my own stuff but this G pair--

>> Right.

>>--is very much in that--this notion of being able to have multiple models, be able to compare the output of the models, be able to see when they agree, when they disagree and then be able to drill down and the interesting thing in these collective algorithms is also seeing like cascades of errors and being able to kind of have something that allows you to follow these. In some of the models there is something that you can interpret as there'll be a wait on the features.

>> Yeah.

>> So you can interpret that. And that has pluses and minuses. I mean Decision Tree is an interesting case where its interpretable fault but those are not very stable models. So, if you change the data even slightly, you get a completely different Decision Tree. And so this notion of being able to do sensitivity analysis is important. So, I guess what I'm saying is that I agree with you. I don't think that you can necessarily point to

a model and say it's bad and say there's no chance at making it more interpretable but there is needs to be more work in making these.

>> Yeah. Okay, I'd [inaudible] go there.

>> This is a sideways of what you're asking. Having do a privacy [inaudible] you might as well take it. Can the system be turned on its head and not having done the analysis, having practically both the privacy for a database, can you now say, okay, knowing the various learning--machine learning techniques that allow us to break the privacy, can we figure out how much we have to fudge the data, how much noise we have to go in so that what we pass along can be passed along safely without anybody figuring out that I'm [inaudible] or texting while I'm driving? [Laughter]

>> Yeah, so there is a variety of approaches to how do you add noise into the data to make it so that it's no longer that I can--for identity disclosure go to exactly the individual and say that's one of those guys in this group. The thing that I think nobody has been able to give a good answer that I'm aware of is all of those methods make some very strong assumptions about having a closed world and not being able to go out and get additional information to join in to then update it. So, while there is a lot of work that started off just with a single table [inaudible] anonymity and so on and now they're starting to be work that's in a relational setting or the graph setting. This whole notion of, well, what's realistic in terms of being able to say and they can't go out and buy another database to join against this to fit in. That's a place where I think the machine learning work is useful to kind of give you pause and say, "Here's what could happen." In terms of really fixing the problem, it's a lot more murky in terms of like where our policy goes and so on.

>> Other question? Let's take--well, we'll to take these two quickly. Yes?

>> So, it's seems like for all the common work is to find ways to flatten the data so that we can give it to these algorithms that we already have. Is there much work or do you see a future for work in developing machine learning algorithms that can natively work in relational environment?

>> Yes. And to be honest, I actually do a lot of work in that space. And so with the whole graphical models community is about methods that work with the graph. Although, still, there's things that need to be drawn in there to make--allow them to deal with multiple graphs and changing graph structures. So, there--so here I'm trying to give you an approach to--you have off the shelf things how to approach them. The interesting thing is there is a lot of people that advocate having these big, joint models and often times these ones feed it. So, it's good to know about these and start with these before going to these.

>> And [inaudible] last word.

>> So, you ended with showing us some tradeoffs of the same--well, we have better understanding personalization and also and there are policy violation. So, tendency is kind of being resolved or we'll say always [inaudible] and I mean I have thought of about a few more [inaudible] that you familiar [inaudible], you know?

>> Yeah.

>> Well, so we leave that about other way, but, for example another thing is cope this machine algorithm technique to why that information overload but then we know from experience and sometimes [inaudible] which run exactly at the other end turn out to be really is the most important elements in our life.

>> Yeah, it is.

>> So, can you see sort of, you know, that this is tradeoffs can be resolved or will say always be with us?

>> I think it's important to be aware of them and exactly these information overload is a perfect example. So, it's like I can learn this class where it tells me, "Oh, I always answer e-mail from so on so quickly, although mine would say I don't answer it quickly [inaudible]." But then it's going to miss out that I got a new e-mail from some person that I've never heard of that's, you know, offering me some amazing opportunity and so on. So, this--are you trying to predict to the common case? So, you're finding patterns or are you trying to detect anomalies and alert to anomalies? Somehow, I think that that tension is always going to be there, but having a system that can allow you to flip back and forth and say, "Okay, turn my e-mail answer ranker thing on autopilot and let me quickly go through these" and then switch it over and say, "Oh, you know, tell me what's unusual in my e-mail today" and kind of go through those and they have people be aware of these patterns and potentially be able to inspect them more. So, the notion of actually having your ranker that I could go in and look at and I see, well, these are all the things that's using. And then I'd have a better understanding of am "I okay with the things that it's going to miss?" You know, in certain cases, I might be. So, that's some parts of the [inaudible].

>> Thank you Lise. [Applause] All right.