EPISODE 101

[INTRODUCTION]

[0:00:10.8] SC: Hello and welcome to another episode of TWiML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

A quick thanks to everyone who participated in last week's TWiML online meetup, it was another great one. If you missed it, the recording will be posted to the meetup page at twimlai.com/meetup. Definitely check it out.

I never cease to be amazed by the generosity and creativity of the TWiML community. I'd like to send a special shout out to listener Shirin Glander for her exception sketch notes. Shirin has been creating beautiful hand-sketch notes of her favorite TWiML episodes and sharing them with the community.

Shirin, we truly love and appreciate what you're doing with those, so please keep up the great work. We'll link to her sketch notes in the show notes for this episode and you should definitely follow her on Twitter @ShirinGlander for more.

This is your last chance to register for the RE•WORK Deep Learning and AI Assistant Summit in San Francisco, which are this Thursday and Friday, January 25th and 26th. These events feature leading researchers and technologist like the ones you heard in our Deep Learning Summit Series last week.

The San Francisco event is headlined by Ian Goodfellow of Google Brain, Daphne Koller of Calico Labs and more. Definitely check it out and use the code TWIMLAI for 20% off of registration.

In this episode, I'm joined by Inmar Givoni, Autonomy Engineering Manager at Uber ATG to discuss her work on the paper Min-Max Propagation, which was presented at NIPS last month in Long Beach. Inmar and I get into a really meaty discussion about graphical models, including what they are, how they're used, some of the challenges they present for both training and inference and where they can be best applied.

Then we jump into an in-depth look at the key ideas behind the min-max propagation paper itself, including the relationship to the broader domain of belief propagation and ideas like affinity propagation, and how all these can be applied to a used case example like the makespan problem.

This was a really fun conversation. Now on to the show.

[INTERVIEW]

[0:02:34.4] SC: All right, everyone. I am on the line with Inmar Givoni. Inmar is the Autonomy Engineering Manager at Uber ATG. That's Uber's Advanced Technology Group in Toronto, Canada. Inmar, welcome to This Week in Machine Learning and AI.

[0:02:48.9] IG: Thank you, Sam. It's great to be here.

[0:02:50.8] SC: It's awesome to have you on the show. We've been trying to get this coordinated for quite a while. I'm glad we're able to make it happen early in the New York.

[0:03:00.3] IG: Absolutely. I'm excited about this.

[0:03:03.0] SC: As is our tradition here, why don't we get started by having you tell the audience a little bit about your background and how you got interested in machine learning.

[0:03:11.8] IG: Sure. For me, it actually started pretty early on in the sense that back in high school I thought I wanted to be a neuroscientist and I was pretty sure that understanding and researching the brain is the most interesting thing that I could apply myself to.

Going into university, I chose the areas of computer science and biology, because I thought of brains as machines and thinking about the computer science and engineering approaches would be useful for understanding them, and biology because it is after all a biological substance.

I spend a lot of my time taking neuroscience courses and talking to a neuroscientist and trying to understand how they approach solving the problem. At the same time towards at the end of my undergraduate degree, so the last year I also took a machine learning course.

First of all, the machine learning course was really interesting. It was such a different way of thinking about how to solve problems that really appeal to me. The mathematics of it were beautiful. It combines a lot of things that I previously learned. They never really clicked into one place, so anything from linear algebra, calculus, probability theory, statistics, graph theory, combinatory, all of it is used in one way or the other in machine learning.

Also, it allows us to solve problems that you can't really solve with traditional programming, or engineering approaches. It's a whole new mathematical way of looking at these problems and then coming up with really beautiful solutions.

As for the neuroscience, I felt like maybe instead of trying to understand the brain, a different way of doing it would be to try and build software that exhibits intelligent behavior. That drove my decision to do a PhD in machine learning. I moved to Toronto, which is one of the best places in the world to do machine learning research and did that for a few years.

While I was in school, I did a few internships and I also realized that I really like working towards a specific product and towards something that is tangible and is out there in the world and people use and is also up for that type of scrutiny.

After graduation, I worked in various companies on applications of machine learning, so real-world products. I like physical things, so in all of them there was some sort of a physical product that you can actually touch. Of course, self-driving cars is one of the most exciting new technologies, or maybe technology itself is not that new or the idea of doing it is not that new. It's been around for about a decade at least, but it's now really coming into full attention from everyone in the world. I thought joining this office would be an incredible opportunity. I've been here for a few months now.

[0:06:10.6] SC: Wow. What specifically do you do there?

[0:06:13.0] IG: The office in Toronto is a research and development office. It's led by Raquel Urtasun, who was one of the world experts in the intersection of self-driving, deep learning and computer vision. There is a research group of researchers who are working on creating new innovative cutting-edge algorithms for using deep learning for self-driving.

My role is helping take these products, these research prototypes and first-phase algorithms and actually get them into the car. Get them into production. I manage a team of applied researchers and software engineers who help with everything that has to happen in order to take a prototype, something that resembles what you would get out of a publication and actually get into a production feature.

[0:07:10.5] SC: Now we've talked about – we did a series on autonomous driving last year, I think in the fall or towards the end of last year. One of the things that jumped out of that was the different approaches and philosophies to doing machine learning for autonomous driving end-to-end versus integrating different systems, and camera-first versus sensor fusion approaches. Does Uber have a – What's Uber's approach to autonomous vehicles in that domain?

[0:07:51.8] IG: We're definitely, as a place that specializes in deep learning, we believe that deep learning approaches for the various aspects of technologies you need to introduce and around perception, understanding where the car is, understanding who the actors are can all benefit from deep learning.

In terms of what kind of architecture specifically, or the sensors and so on, I think this is still something that is an ongoing research and understanding what are the benefits of the different sensors and when is the right time to use them and in what configuration. I would say that's still an open question.

[0:08:35.1] SC: One of the things that we wanted to dig into in this conversation was a paper that you co-authored that was released at NIPS, or was in the NIPS proceedings on min-max propagation. One of the things that I observed at NIPS was a lot of conversations around graphical models and graphical approaches in general for machine learning, and as maybe a segway into that conversation I wanted to get a sense from you, is that – would you say that there was – there is heightened interest in graphical models this year? Or is it a conversation that has been continuing on without – I guess, is there a heightened level of interest in this particular type of modeling approach?

[0:09:31.0] IG: Right. Here is my take on it. Graphical models are very powerful mathematical tools to represent our understanding of the world and our understanding of what we don't

understand and the fact that there is uncertainty and noise. As modeling tools, they're very powerful.

The problem has been that they're very difficult to train and to do learning and inferencing. They are computationally expensive and they do not scale up to full-blown industrial purposes. There has always been – not always, but there's been an ongoing exploration and research in graphical models for a long time.

I entered the field doing research, I would say around 2006 and these were very popular and like a big area of focus. In fact, my thesis focuses on application of graphical models to clustering. It remained within, I would say primarily within academia, with the exception of specific models that were simpler to use than others and were used in industry.

Then we had, over the last few years obviously we had this intense focus on deep learning and neural networks. A lot of people worked on them. Some people shifted from working on graphical models to focusing on innovation in deep learning.

I think now that we are getting to a place where a lot of the ground work has been done and a lot of the – let's call it the low-hanging fruits have been picked, now there is growing interest in looking again at graphical models and then asking –given all we've learned about how to do deep learning at scale and how to use it and how to be able to solve difficult problems with it, can we now merge better the two strategies and can we create new types of models that leverage both, so that maybe graphical models can become as useful as deep learning models?

[0:11:45.4] SC: What are the types of applications that lend themselves most readily to use of graphical models?

[0:11:52.5] IG: There are many different applications. One that is commonly I would say looked at is in, let's say healthcare. Understanding or predicting whether a particular patient has a particular type of cancer, you can describe all that you know about the different measurements you've taken and so on using a graphical model. Then you can try to infer the actual underlying state of their illness if they have learned the type, the stage and so on.

[0:12:27.3] SC: The main idea as I understand it behind the application of graphical models is as opposed to traditional – I guess , traditional, we'll start with calling it traditional, but a typical

data set let's say, where you've got data points and the data set doesn't really express any inherent relationship between the data points, as opposed to a relationship between features within those data points – within what graphical models are really trying to do is identify relationships between the data points and the data set. Is that an accurate assessment?

[0:13:13.8] IG: Yes, yes. They deal with representing all sorts of quantities in the world as variables and either explicitly saying that there are known complex relationships between these variables, or trying to learn the complex relationships between these variables.

[0:13:33.8] SC: When you say explicitly saying that there are known complex relationships is are you there describing, pulling in prior knowledge about the relationships between these variables into the models?

[0:13:49.6] IG: Yeah, that would be one way of doing it, where you represent all sorts of prior knowledge given – with distributions that you believe are well-suited to represent your prior knowledge. Also representing the actual relationships between the variables, so if we're trying to think of a relatively simple example, let's say that I'm trying to represent an image in terms of a graph. Maybe every pixel in the image will be a node in the graph, and every pixel will be connected to the pixel that are around it and these connections, these edges on the graph represent the fact that we think that the value of the pixel is highly correlated to the value of the pixel around it.

For example, if we knock about the pixels around it, we can make guesses as to the value of the pixel, or we can represent the possible values as a distribution that is not uniform. It knows something from its surround. That would be one way of representing relationships between variables.

[0:14:53.9] SC: Are images, is that a common application area for graphical models?

[0:14:59.6] IG: Images used to be. Before, we were able to successfully work with different neural networks. In terms of performance, right now I would say that convolutional neural networks are much better at tasks around prediction and deduction and various tasks that have to do with images.

[0:15:21.0] SC: What have been some of the biggest achievements today of this renewed interest in graphical models with the background of the progress and deep learning over the recent years?

[0:15:38.3] IG: I think one interesting idea is to allow for neural networks to operate over graphs. There is new – a researcher in graph neural networks and graph convolutional neural networks. Other ideas are that within the network you represent some of the new models, some of the network explicitly using some formulation of a graphical model.

[0:16:15.1] SC: What does it mean to have the network operate over a graph?

[0:16:20.0] IG: Again, going back to the example of images, if you think about it as a cross – it's a grid. It's a simple cross. We can very easily do operations on it like convolutional operations, because we just shift them, shift the filter over and it's the same operation. You can also abstract this notion of a convolution to something where it's not a nice grid, but it's still a graph. That requires the mathematics to be developed for it and to be made efficient.

[0:16:54.1] SC: Is the idea then to replicate what a convolutional network is doing, but with graphical models?

[0:17:04.6] IG: Not necessarily. It's just to use the – I guess the body of knowledge and work from graphical models and maybe we can get into inference a little bit in a moment. For doing inference and graphical models that is not really captured within neural networks. Neural networks, people talk about doing, learning and inferencing neural networks but really – I think the – it's come to mean that learning is when you train the algorithm and inference is when at this time you basically run for propagation through a network and come up with a prediction.

It's a correct use of the term, but in the context of neural networks that the inference part is easy because it's just a straightforward calculation. Sometimes computing this prediction, so computing and in the case of [inaudible 0:18:00.5], it's like what is the probability that there is a particular object and damage, let's say, right? Then it spits out probability of whether it's there or not in this particular place, let's say.

In some cases, let's say when the underlying model is a graphical model, computing this probability is in and on itself a difficult set problem. Then there is a large body of work on

computing, running this algorithm. In some cases, and particular in graphical models within training itself, so not just when you're at test time, you're interleaving the learning and the inference part.

The learning is often referred to coming up with estimates over taking variables. We didn't really talk about what hidden variables are, but – sorry, coming with estimates of the parameters governing the behavior of the hidden variables.

The parameters in a neural network would be the weights, and so we're learning we're basically optimizing the weights. In graphical models, the parameters can be for example parameters that govern distribution. I assume that there is a quotient and I don't know what is the mean and the variance of the quotient. That would be the parameters that I'm trying to learn.

Then there is also an inference part, which is computing all sorts of typically predications or probability distributions on the variables of interest that requires running some sort of a sometimes complex computation. When I'm done with that I sometime – it's right back to the learning parts. Given that I've refined my estimate of the probabilities, now I refine my estimate of the learning part, of the parameters and go back and forth between the two.

What I've just described is a very high level in algorithm called the expectation maximization algorithm, or EM algorithm, which is used for doing learning and inference in graphical models like a Gaussian mixture model for instance. If you want, we can talk a little bit about what is the Gaussian mixture model.

[0:20:06.8] SC: Yeah. Before we do that, so the expectation maximization that you were just describing, this is trying to contextualize this – we were talking about training and inference and you started to talk about inference, but this is used as part of training, but includes inference calculations. Is that the –

[0:20:28.8] IG: Yeah. Again, in non-neural network world wherein you're talking about the training procedure itself may include both learning and inference.

[0:20:38.9] SC: Got it.

[0:20:40.3] IG: Maybe the Gaussian mixture model is a good example to run through for that. In Gaussian mixture models, we basically have a whole bunch of data. It's a way of clustering the data, let's say. In clustering, we assume that the data can naturally be divided into groups, or clusters. What makes a good cluster is that the points in the cluster are quite similar to each other and are quite different from other points in other clusters.

Now that I'm given a whole bunch of data, how do I go about actually finding these different clusters? In the Gaussian mixture model I basically make an assumption that there was some process that generated the data, which is let's say that the data happens to have five clusters, so someone flip the coin or roll a dice, or samples the centers of the clusters. Then from each center, a bunch of data points were sampled, but there are some noise involved.

Now if all I get to see is the end-result is I get to see a lot of points and I need to infer, I need to learn the parameter, so I need to learn the means of the clusters, the centers of the clusters and their variances, how noisy the cluster are. Then I need to infer the probability that each point came from any one particular cluster, so that would be a distribution over the clusters.

[0:22:10.8] SC: Are you also inferring the number of latent variables or processes that is producing your data, or is that always an assumption that you make in your modeling?

[0:22:24.8] IG: That's a good question. The traditional algorithm assumes that this is actually given as an input. Somehow, I figured out that there are five clusters. Of course, that's not really a good assumption because I typically don't know.

The next obvious step is to run it with a whole bunch of possible number of clusters and figure out how to measure which one is best, but that's not necessarily easy to do. Then there is a very interesting literature on what's called a nonparametric approaches, or nonparametric Bayesian approaches, which basically say we don't want to make the assumption that we know the number of clusters ahead of time.

We're going to also inject that. Because it's Bayesian, they're not actually inferring the number of clusters. They are integrating over the number of clusters. It gets to a fairly complicated myth, but there is definitely the attempt to try to accommodate for that. Going back to what I was

saying before, that's an example of some really interesting and fairly beautiful mathematics around that area, but it's not yet practical to run these algorithms at scale.

[0:23:41.9] SC: Is that when you're saying, these algorithms in this case, so you're referring to –

[0:23:48.5] IG: Nonparametric Bayesian approaches.

[0:23:49.4] SC: Nonparametric in particular. The Gaussian mixture models are fairly widely used, is that right?

[0:23:55.5] IG: Yeah. If you know the number of clusters, or if you're going to assume that you give it as an input to the algorithm, then that is an algorithm that can be applied quite successfully to data. Going back to this thing, the EM algorithm, what you typically end up doing is iterating between two types of computations, where one is given some estimates of different means and variances of each cluster, I will compute – I will do the inference, so I will compute what is the probability for each point to be associated with each one of the clusters. Then after I'm done computing that, I go back to re-estimating my parameters and refining the estimates of the parameters. I do that until convergence.

[0:24:44.9] SC: Does that mean you're iterating on the cluster centers and the parameters of your distributions for each of the clusters as you're incorporating the points into the clusters?

[0:24:59.2] IG: This is for training time. In training time, in this case I assume I have access to all of my data and I'm just trying to figure out where are these cluster centers and their variances and which point belongs where.

I will train by running the EM algorithm. I will end up with estimates of the cluster center and their variances. Then if I'm getting a new data point, then I can do that prediction of what is the probability that it came from cluster 1, 2, 3, 4?

[0:25:30.2] SC: Right. Where does min-max propagation fit in?

[0:25:33.6] IG: Right. Now we have to – that quite [inaudible 0:25:35.9], because it dove into one particular example. Even with Gaussian external, the inference itself is relatively straightforward, let's say. At a very high-level, this is a very, very high-level algorithm to

approximately solve a particular set of empty heart problems. The algorithm is a new variant of relief propagation and the problems that we're looking at are min-max problems. In particular, we've demonstrated on makespan.

Now, in order to parse through all of that, we have to talk about what each of these things mean, right? I can start talking about it and you can interrupt me with questions. I'll try to take us through all of it.

Empty heart problems are roughly speaking problems for which we don't know if there exist an efficient solution that will run in a reasonable amount of time. We typically know a brute force solution, so the problem with brute force is that as the problem scales, it takes longer and longer in a typically exponential manner, so it's just not practical.

Finding solutions that are approximations to empty heart problems that run in a reasonable amount of time is a pretty big research area in theoretical computer science and other disciplines like the operational research. Okay, so that's empty heart.

Now, if we look about min-max problems, they appear a lot in game theory, decision theory and other math and science domain. The technical definition is that you have a function of two sets of variables, let's call them X and Y and you want to find the mean over X, max over Y of the function XY. That doesn't really give a lot of intuition, so let's try and look at an example.

In the paper, we'll look at a problem called makespan, which should be fairly easy to grasp. Let's say you have a bunch of incoming jobs, and I'm talking about jobs in the computer science sense; some automated tasks that need to be executed on a computer. Each will require some amount of resources, so something like – think about CPU, or memory to run. We have a bunch of machines, like a computing cluster to run it. Maybe each machine has slightly different capabilities. Each will take slightly different time to run each one of these jobs.

We want [inaudible 0:27:59.3] on all of them, so we obviously want to divide up the jobs in a way that everything finishes as soon as possible. If we think about it, that time will be determined by the bottleneck, the machine that will take the longest to process all the jobs that were assigned to that machine.

What we want to do is we want to minimize the maximum time it could possibly take. We want to find a way to split up the jobs across the machines, so that we minimize the maximum time it would possibly take to execute the jobs.

[0:28:29.7] SC: Got it. You called this problem what? Makespan?

[0:28:33.9] IG: Makespan.

[0:28:34.5] SC: Makespan. Okay, sounds like job shop problem?

[0:28:38.8] IG: Yeah. It's related to that. Notice that there is a simple brute force solution, which is try out all the different ways you can divide up the load, right? A problem of course is that this will be exponential in the number of jobs, and so it becomes very intractable very quickly.

Okay, so now we understand at least one example of a min-max problem. The algorithm we're presented is again in the context of it's to solve the min-max problem and the paper demonstrated on the makespan problem, but it's a little bit more – it's more relevant and that it's relevant for a set of min-max problems. It goes into the technical properties of which kind of min-max problems can be addressed. I think we can skip that. Now we can talk about the algorithm.

[0:29:25.4] SC: Well, can you give us a high-level characterization of those types?

[0:29:31.2] IG: Min-max problems?

[0:29:32.1] SC: The types of min-max problems just to get a sense for even what are the dimensions around what you're thinking about characterizing these problems.

[0:29:43.1] IG: They are off the notion of I want to minimize my worst-case scenario. Another example of min-max problems is when you have some sort of a two-player game and you're trying to come up with a strategy so that whatever the other player can do, which is some sort of – I will incur some loss. The maxing loss, I want to minimize that. It's the most abstract formulation for that problem.

[0:30:12.4] SC: Then it sounds like what you've done in paper in terms of characterizing the types of min-max problems to which this particular method applies, it's not necessarily an intuitive characterization. It's more mathematical detail, I guess.

[0:30:31.8] IG: Yeah, exactly. It's a set of – Yeah. Restrictions, let's say.

[0:30:36.2] SC: Okay. You mentioned belief propagation.

[0:30:38.9] IG: Right.

[0:30:39.7] SC: Let's talk a little bit about that.

[0:30:40.9] IG: Now we can talk about the algorithm we actually propose. It is a variant of belief propagation. In order to understand belief propagation, we need to know a little bit about inference and graphical models and message-passing algorithms. We talked already a little bit about inference and graphical models.

In the context of describing – going back again to the notion of we have data, or observation and we assume that there is some noise in this observation, in these measurements and then there is some uncertain and we assume that there are some quantities that we don't have direct way to measure, but are – that have dependencies with relationships with the quantities.

We often talk about this in terms of observe and hidden variables. They are in graphical models, we try to capture the way in which they are related using a graph. Basically every variable, whether it's something that we've observed, or something that we can't observe is represented as a node in the graph. Then the interactions between these variables are represented as edges in the graph.

Once we describe the problem in that manner, we can start borrowing from graph theory and graph algorithms to answer different questions about the graph that we have describe, which is basically our representation of the probability distribution over all of these variables.

Belief propagation is a general algorithm. It has several existing variance that are quite well-known. One of them is the max-product belief propagation, another one is sum-product belief propagation. They are algorithms or recipes for computing these quantities of interest.

Let's say that we have a bunch of these hidden variables and one question we can ask is – let's say that they are binary, so they can either take the value zero or one. We can ask what is the particular setting of all of these variables that yields the highest probability. This is the mode of the distribution. In order to compute that, we can use something like the max-product algorithm.

At an intuitive level, the way these algorithms work is again going back to this graph that we have in mind, they send messages between the nodes. The nodes exchange numerical quantities, sometimes in an iterative session, and then eventually there is some kind of a decision.

[0:33:18.3] SC: The nodes are variables observed and hidden variables. Can you give an example of a scenario that includes both these observed and hidden variables? These hidden variables in the sense of latent variables and Bayesian for instance?

[0:33:38.7] IG: Yeah, latent variables.

[0:33:40.0] SC: Okay. The observed variables are – what's an example where you'd have both observed and hidden variables?

[0:33:50.3] IG: I can actually go back to my graphic Gaussian mixture model and I can represent that as a graphical model where my observed variables is the data that I'm trying to cluster, and the hidden variables are the clusters that are represented by these parameters. Maybe I can talk in order to segway into this particular – that the min-max inference, I'd like to talk about another clustering algorithm, which I think would really focus us on the inference side of the hidden variables.

That's a different clustering algorithm and this was actually the focus of my thesis work, and it's called affinity propagation. This is an algorithm that was first presented by my supervisor Brendan Frey and this graduate student Delbert Dueck. I focused on various extensions of the algorithm and reformulating of the mathematics.

Anyways, the way we represent clustering in that algorithm is we basically say, "Okay, we have a bunch of data points who want to split it into groups that makes sense." Instead of talking about means and variances, we're going to say that each cluster is basically best represented by an exemplar. This is like the representative of the cluster.

Really, the problem becomes can we find the right set of representatives? Then every point that is not a representative just needs to be associated with the representative that is closest to it, that is most similar to it.

The problem of course is you don't know which subset of points is best to represent these clusters, right? When you do this max product belief propagation in the context of this graph, your variables are basically the data points. The hidden variables are for each point which cluster should it be assigned to? That's something you don't know.

[0:36:07.3] SC: For each point – Right. That specifically as opposed to the parameters of the cluster itself.

[0:36:15.6] IG: Right. We're not using parameters anymore or this algorithm. You can see that here, we'll be doing inference. We're going to come up with estimates of for each point what is the cluster it's going to be assigned to without talking about a notion of means and variances. There is no learning curve, it's only inference.

The way the algorithm ends up working is basically again sending messages, so the nodes send messages to each other and the messages come out of a mathematical derivation. If you look at them, you can give them – give an intuitive explanation of what they're really trying to say, which is they – it's an iterative process where first all the nodes send a message to all of their neighbors, so each one of their neighbors is saying, "To what extent do I want you to be my representative, right?"

After collecting all the information from my neighbors, so all my neighbors have told me to what extent they want me to be their representative. Now I send all my neighbors a message back saying to what extend do I want to be a representative. It's right back and forth on these messages until you converge.

Think back a minute, in belief propagation we end up having these messages that are exchanged between the nodes in the graphical model for the purpose of doing inference, for the purpose of computing these quantities that we care about.

I think now we might be ready to talk about the min-max propagation.

[0:37:48.4] SC: Yeah. Just to make sure I'm on the same page, so we're talking about messages and message passing and things like that. This is basically a computational tool or an accounting tool that we're using to just keep track of quantities in this algorithm really as we're trying to do the inference. The side of my brain that thinks about distributed computing is thinking about real things, passing messages and it's like, that's not really what we're doing here.

[0:38:22.9] IG: Right. These are not text messages or JSON messages. These are numerical quantities that are computed and then used, but it's convenient to think about them as there is a numerical quantity computed for each one of these nodes. Then there is another one computed, which relies on the previous computation, so it's almost as if the node sent that numerical representation maybe a vector of numbers to all of its neighbors and said, "This is my message. Now you're ready to do your computation in the next iteration of the algorithm."

Now I think we have all the building blocks. We have these graphical models and we know a little bit about message passing algorithms. The question we ask in this paper is can we take this belief propagation type algorithms? We understand that they can work for four types of question that we're interested in the context of finding the assignment of the variable that gives us the maximum probability, or computing marginals and other tests.

Can we somehow formulate it to solve the min-max problem? It turns out that we can. It's not by running the same algorithm, it's by borrowing the general idea and the – some of the mathematics around it, and writing down the min-max problem as a graphical model. Then deriving the form of the messages that need to be sent in order to compute what should be the solution to the min-max problem.

They are similar in principle to the messages you send when you're doing max product or some product, but of course its own flavor that required figuring out the right derivation. Another interesting thing is what we do when we have interactions that are between quite a few variables.

If we have in our graph an interaction between a subset of variables that is more than just two, some of the computations can on the surface seem like they are exponential in the number of the variables. In order to be able to actually apply this framework to min-max, we have to

recognize that what looks like it's going to be an exponentially expensive computation can be actually done by some – a little bit of cleverness and bookkeeping in non-exponential time zone and something that is reasonable to compute.

[0:41:07.4] SC: Okay. Just summarizing, basically what you did with min-max propagation was you borrowed from this message passing approach that comes out of belief propagation and applied it to this makespan problem. In addition, you've mathematically characterized the more broader set of min-max problems to which this model would also apply.

[0:41:40.8] IG: Yes. Importantly derive the form the messages need to take if you're doing a min-max computation as opposed to a sum-product computation, or a max product computation.

[0:41:53.2] SC: What is that form?

[0:41:55.6] IG: That's in the technical details, so it's basically an equation that describes some form of operations as minimums and maximums that can be computed in let's call it linear time for the purposes of this talk. It's an equation. It's an equation that when you go about implementing the algorithm, you will just writing some code.

[0:42:19.0] SC: Okay. Given the computational requirements of this and the space of potential application, where do you think this paper and algorithm will have the broadest impact?

[0:42:31.2] IG: I think it will be interesting to see what kind of things people use it for, because it's relatively – it's a new formulation. We wanted to look at the case of makespan as a particular application, which is actually it is useful for things like scheduling tasks, it's useful for workload in terms of power consumption on turbines and power plants. In the operational research world, I believe they look at these types of applications.

I think one of the interesting things you get to do as a researcher sometimes focus more on the core algorithm and the mathematics of it, and post it out there for the world to see, so that various people who are looking at different problem domains can recognize that this actually matches to their problem of interest and take it into places.

[0:43:39.8] SC: Right. On that note, it might be worth mentioning that this is all work that you did in the UVT context as opposed to the Uber context, is that right?

[0:43:48.9] IG: Right, right. It's also worth mentioning my co-author, which are Christopher, Siamak and Brendan.

[0:43:58.3] SC: Okay. Awesome. Very cool. It strikes me that this is a little bit of in a side, but I'm not sure that you know but we do a monthly meetup, where we dig into research papers and we've done a bunch of deep learning focus once. It strikes me that this would be an interesting one to maybe have you or one of your co-authors present to the group to really dig into some of the details here. It seems like a lot of it is in the details.

[0:44:30.6] IG: Yeah. I would be happy to talk about that and to look into this opportunity. I think one of the – again, as a sign of – one of the exciting things about deep learning is that they're so successful in terms of how applicable they are to problems that we care about. The one thing that they don't have as much as other algorithms is deep mathematics and challenging representations that you really need a lot of time to wrap your head around. It's really nice to dig into these papers and there is typically quite a bit of math in them as well. I think it will probably be interesting and will challenge people a little bit in a way that's interesting to look into these types of papers.

[0:45:27.7] SC: Awesome. Where do you go from here with this particular work in your research? Is this something that you are finishing up that relates back to your time at Toronto, or is this ongoing work that you're involved in?

[0:45:42.0] IG: I would say this is more of what I call recreational research. It's something that I work with Chris in the past as a graduate student, as well in the lab and he took it from there. For me right now, the main focus is definitely here at work and working on self-driving cars.

[0:46:08.8] SC: Awesome. Awesome. Well, I really appreciate you taking the time to walk through this with me. I know I've had a lot of questions and I still have a lot of questions. This is not a topic that we've gone into in a lot of detail here in the podcast, but I think I did – I had a several conversations on graphical models at NIPS. I forget how many of those came out in our

NIPS series off the top of my head, but we've got a few more to release over the next few weeks or months.

I guess, I was going back to our earlier conversation. I was surprised by this undercurrent of work happening in graphical models at NIPS and how often I heard it. Although, I don't know that that means anything. It was the first time I was at NIPS, so my baseline does not exactly – was not exactly very dialed in. There definitely seems to be a lot of people interested in this area.

[0:47:19.3] IG: Yeah. Absolutely. Again, neural networks have taken frontstage for the last few years. There is always ongoing research on graphical models and on an array of other problems in machine learning. I think it's nice to see things, like the focus going back to some of the models and seeing what can we do with them now.

[0:47:44.8] SC: Absolutely. Well, on that note Inmar, thank you so much for joining us.

[0:47:49.3] IG: Absolutely. Thank you for having me. I hope that it makes for an interesting listening, or at least if not everything was super clear as introduction to go and explore some more.

[0:48:02.3] SC: Absolutely. Absolutely. Thank you.

[0:48:04.1] IG: Thank you. Bye-bye.

[END OF INTERVIEW]

[0:48:09.9] SC: All right everyone, that's our show for today. Thanks so much for listening and for your continued feedback and support. For more information on Inmar or any or the topics covered in this episode, head on over to twimlai.com/talk/101.

Of course, we'd be delighted to hear from you either via a comment on the show notes page or via Twitter directly to me at @samcharrington, or to the show at @twimlai.

Thanks once again for listening, and catch you next time.

