Shane Greenstein:

I'm Professor Shane Greenstein, and you're listening to the Harvard Business School digital initiative seminar. A premier seminar series that hosts thinkers and scholars who are pushing forward research on the digital transformation of the economy. By conducting and connecting with cutting edge leaders, equipping leaders, and building community, the digital initiative seeks not just to study, but also to shape digital transformation. To learn more, check out digital.hbs.edu.

Shane Greenstein:

Before we get started, as we normally do, we like to go around the room, and not who's here. And, so we'll go this way this time. I'll start. Shane Greenstein from the technology, operations, and management, or TOM for short.[crosstalk 00:00:51]

Speaker 4:

You did great.

Speaker 2:

Chiara Farronato, TOM.

Speaker 3:

Feng Zhu, TOM.

Speaker 4:

John Bakeman, Monfred.

Speaker 5:

[00:01:13]

Speaker 6:

Allison Mnookin, TOM unit.

Speaker 7:

Diane Williams, ed-tech entrepreneur, alum Sloan and Harvard.

Speaker 8:

Dave Homa the director of the digital initiative.

Speaker 9:

Shrikan Dagabatala, TOM unit.

Speaker 10:

Colin Maclay, digital initiative.

Speaker 11:

Ayelet Israeli, marketing.

Speaker 12:
Donald Way, marketing.

Speaker 13:
Tomo Amano, marketing.

Speaker 14:
Yael-Grushka Cockayne, TOM.

Speaker 15:
Ryan Buell, TOM.

Speaker 16:
Jan Hammond, TOM.

Speaker 17:
[inaudible 00:01:41]Schiller

Speaker 18:
[inaudible 00:01:41]Schubert

Speaker 19:
[inaudible 00:01:46]Also Ph. D design

Speaker 20:
Sam Light Ph. D student design [inaudible 00:01:54]

Speaker 21:
Roy Engles a Ph. D student.

Speaker 22:
Erica Meskasky Ph.D design

Speaker 23:
Frank Denser Ph. D design

Speaker 24:
Himbindu Lakkaraju, TOM.

Speaker 25:
I'm Mike Smith [inaudible 00:02:01] and applied sciences.

Speaker 26:

Janie Thomas with the digital initiative.

Speaker 27:

Mike Tushman, OB

Speaker 28:

Neil Thompson, visiting professor here and research scientist at the computer science MI log at MIT.

Speaker 29:

[inaudible 00:02:13]

Speaker 30:

Chuck Bolser Organizational Behavior.

Speaker 31:

Mike Toffel, TOM.

Speaker 32:

[crosstalk 00:02:23]

David Parkes:

Well what a lovely treat to be here, thank you very much for the invitation. And, it's just incredible to, not only realize how many people I know around the room, but also to see how effective you're being in pulling together a really... eclectic mix of different people. I want to tell you all that I love interruptions, I love questions. I was asked to set ground rules, the ground rules are there are no ground rules. Interrupt, ask questions. This is work that we've been doing now for around two years. I think it's work that I consider to be really squarely in the new possibilities for the digital economy. It's work where we're taking, I think quite an ambitious view, we're asking wherever we can use machine learning techniques, in a completely new and different way, to design market systems, Denovo, essentially without needing to use any theory. And I saw that as a, as somebody who's definitely not anti-theoretical. I do a lot of theory work. And quite a lot of my work has related to auction design, mechanism design, more broadly market design questions. And I and other people who work in that field have over time become somewhat painfully aware that these problems where we once had tried to design optimal economic systems to solve problems we care about, are incredibly analytically challenging.

David Parkes:

And, the question is whether we can use new techniques to design. And if you want to have an analogy in mind, you might think about, just as people are using deep learning pipelines to try to discover new machinery- new materials or new drugs, can we use deep learning pipelines to try to discover new designs for economic systems? That's what this talk is about.

David Parkes:

Just to set the scene, a little bit of background from economic theory. William Vickrey, 1961 tells the world how to sell one item. He says if you want to sell one item, in his case to maximize the welfare created, he says you should use a second price auction. He says that you should collect the bids, say twelve, ten, and four, in this case he would sell to the bidder who's value is twelve, four, ten. So that was Vickrey's idea, a second price auction looks a little bit strange if you haven't seen this before, but if you think about it a little bit, it's similar to the way eBay sells things. There are a lot of reasons why it's not the same, but it's similar in style. And just a little bit of notation, we're going to try to avoid notation as much as we can today, but I do want to set the scene a little bit by using these simple examples.

David Parkes:

So in this case, the bids would be a vector B it's an uni-dimensional vector. We have an allocation rule G, which in this I'm assuming the bids are sources and the bid from the first bidder is the highest, so the victor allocation rule G subeye B will be 1. If I'm the bigger in position 1, 0 other wise I don't get the item. And the payment rule will be, I pay the second highest bid B sub 2, if I'm allocated the item, 0 otherwise. Notice, that you already see the structure of two functions. The allocation function and the payment function.

David Parkes:

What Vickrey showed us is that this design is incentive compatible to a very special form of that, it's strategy proof. Strategy proof means that your dominant strategy, if you're a bidder playing this game, is to big your true value. If you think about it, if you're the bidder who's value is 12, you may as well just say 12. If you say 11, you'll still pay 10 anyway. If you say 11, not knowing how other people will bid, you may forfeit the chance to win at something that you would have been willing to pay. So, the key insight for strategy proofness of the second price auction is the decoupling between the amount you bid, and the amount you pay.

David Parkes:

Okay, so that was Vickrey's, 61. Roger Myerson, 81, 20 years later says, "Okay, you've got an item to sell, and now, you don't want to just give the item to he or she who values it the most. You want to maximize expective revenue.", that's what Myerson says. So, he sets up an optimization problem. He says that we want to find the G and the T, those two functions, they'll maximize the expected total revenue. So I'm just taking the sum over the bidders and them I'm going to impose the incentive compatibility. I'm going to say that I need my rules of the game to be incentive compatible. Myerson solves this problem. He solves it using the idea of virtual valuations. He's going to use knowledge of the distribution function. This is the distribution function from which agent eyes values are sampled. And, he's going to say that we should calculate what he calls a virtual value. Details of that with the form of that function no matter, notice that he uses the distribution function. He's going to allocate the item to the agent with the highest virtual value, and then he's going to charge the winner their critical value, which is the analogy to Vickery's idea. It's the smallest amount such that they would have still been ranked in position one.

David Parkes:

So notice you have the independence again, he proves that this si revenue optimal. And by the way, it's also strategy proof.

David Parkes:

So then the question is, how do we sell two items to maximize expected revenue? And, the answer is that even though this has been a problem that has been very widely studied, the 27 years after Myerson we don't have a complete analytical understanding of this question. Hard to believe, but true. [crosstalk 00:08:59] Let me tell you some thinks that we do know. We do know from Minnelli Vincennes, 2006, that if there's a single bidder, okay, what does it mean to have an auction with one bidder? [crosstalk 00:09:17] Well, you can still ask, if there's one bidder and I'm selling to Jan, and I know something about her distribution over values, I can still ask, how can I maximize expected revenue? The structure of what I'm going to do is basically going to make you offer. SO you're going to decide what you want.

David Parkes:

So what Minnelli Vincennes says, suppose we have one bidder, two items, and that the bidders value for each item is IID uniform 01. Then, here's the optimal allocation rule. The X axis is the value of the bidder for item one Y axis the the value for the bidder for item two, there's a region here where we don't allocate, there's a region here where we allocate both items. Region here where are allocate one but not two up here, we allocate two but not one. And you can pull the payments out from this picture as well. So there's one place where we have an answer. And there are other optimal design results as well. In recent years, computer scientists have been getting in the game as well.

David Parkes:

For one bidder, two additive item variations. Now what does it mean, what variations? Not uniform 01. Different distributions. There are interesting intricate analytical results. For one bidder two item unit demands, unit demand is where Jan only wants one item. And she has a value for item one, a value for item two, if I giver her both, the value she realizes is the maxim of the two. That was solved in 2011. But okay, what about selling to two bidders? Only late 2017, a theoretical computer scientist, Andy Yaw, wrote a paper that gave the first analytical revenue optimality results for selling two items to two bidders. And, to simplify the problem enough to be able to solve the problem, he said that he was going to assume that the support of the values were either .5 or 1. Now we'd say two element support distribution.

David Parkes:

But then, with that he was able to show us how to optimize expected revenue. So, this is roughly the state of the eye. Please?

Crowd questions:

What do you mean by, it's hard to, we don't know? Does it mean that you can't compute results of the optimization problem? Or we don't know a solution to the optimization problem? Or are you looking for some analytic whatever you need?

David Parkes:

Yeah, yeah. SO I think the two things to think about. One is the design problem. SO, Myerson formulated that, "Find the G and the T that maximize expected revenue." That's one problem that could be hawed in some sense. And the other problem to think about it, okay, you've designed your auction. Now is it hard to somehow run the auction? And I've done work on that, if you have auction for multiple girths and combinatorial practices, that gets hard. But that's not what we're talking about today. We're talking about the first problem. We're talking about the design problem being hard.

David Parkes:

What do I mean by hard and why is it hard? First of all, why is it hard? The why is because you have to have the least incentive compatibility requirements. There are things that somehow- that say things like, a bidder of this type cannot benefit by pretending to be a bidder with another type. And these requirements get very hard to handle. The way I think about it is they are global constraints on your design. They're there because it has to hold between all pairs of inputs. And another more technical thing to say, those of you who work with economic theory will know that there's a big qualitative difference in terms of analytical attractibility, between settings where your private information is one number, and setting where your private information is more than one number. This is called the one-dimensional vs. multidimensional distinction in the mechanism design, auction design literature. And, we have good, useful characterizations for how to solve that optimal design problem. In the one-dimensional case. That was Myerson's results.

David Parkes:

He said if it's monotone in the right way, then things will work. We have generalized characterizations for the multidimensional settings. However, they're very hard to work with. SO it becomes analytically intractable. There are also computational complexity results in the literature as well. So back in 2002, Vince Condens Thomas Santon at CMU, wrote a paper that introduced the idea of using computation to solve the design problem. And this, in a sense, is the latest version of that idea. And what they said was something like "Let's imagine that we can discretize all the possible inputs and write down a huge integer program to represent, explicitly, the functions that we're trying to search over. And you can imagine that doesn't scale, and you can write down, NP hard style complexity results as well.

Crowd questions:

And so, then I should think of this as the constraint that doesn't have an occasion inscription.

David Parkes:

Yes. I think that's exactly the right way to think about it. Yup, any other questions?

Crowd questions:

Just to point a, rank options, like the Google options... [crosstalk 00:15:08]

David Parkes:

Yes.

Crowd questions:

Is the second price. Quality waiting, I mean I have a sum intuition on how that one works, but we're not thinking that, or is it a special case here, it only works under very narrow circumstances?

David Parkes:

Yes. That's great, so the auctions that are used for sponsored search are, even though they may not appear to be it, they're one dimensional problems. Why is that? The reason is, you bid your value for a click, an then what I do is I project that one number down to an expected value for the positions on the page. And because your private information is just that one number, then I can use Myerson's theory.

And if I wanted to design the revenue optimal solution to that problem, I know how to do it. It's not what GSP does, but if I wanted to, I could.

Crowd questions:

I see. And it's the rank that makes it...

David Parkes:

Yeah, and the crucial thing there, and your question let's me also take the opportunity to make the following points, machine learning has been used a lot in the design of markets. But it's been used typically in that narrow way. Which is that I'm going to use machine learning to predict how to map that one number to these multiple numbers. And that's not what this talks about. This talks about designing the whole thing.

Crowd questions:

Oh, I get it.

David Parkes:

Using machine learning.

Crowd questions:

That was very helpful.

Crowd questions:

Can you give us examples of in what settings you need this machinery that you technically don't have the machinery for. In the real world.

David Parkes:

Yeah. Well, to give you one concrete example, I had a student, sorry I should have said that the main Ph. D student on this work is Jer Fong who is a Ph. D student at Harvard, he spent last summer at Facebook, and they're looking to see whether they can use this machinery to think about how to design... he's in the research group, so I don't want to overstate it to you, but they're looking to understand wherever they can make different trade offs between say, fairness, efficiency, and revenue in the design of these multi-item slot style auctions.

Crowd questions:

SO there's a way from-

David Parkes:

Yeah, all the different types of optimizations they can make in that general space.

Crowd questions:

Is it just another example that Google could go away from having one bid that you enter, that you could bid separately for each slot?

David Parkes:

Yes, maybe there would be interest in that. That's right. That's right. And obviously there were market design questions everywhere, and I think often times at the moment we solve them somewhat heuristically. Because we cannot handle the multi-dimensional aspect of them.

David Parkes:

Okay, this is a talk about using deep learning to try to attack these problems. So here's my one fight on deep learning. [crosstalk 00:17:56]

Crowd questions:

So, before you get there. Can you speak to how... the models for the allocation of spectrum...

David Parkes:

Yes.

Crowd questions:

Has sort of helped moved these theories along, if in any way?

David Parkes:

Yes, absolutely. The frontier of implied market design in recent years has been advanced quite a lot selling wireless spectrum. Including by the FCC but around the world as well. And those are... very interesting problems, typically because you have to worry about the packaging [crosstalk 00:18:38] of the items, the complementarities. By one New York, and Philadelphia, and D.C. And so it's the package they care about. It was actually those problems that first pulled me into thinking about combinatorial auction design. The advances that have been made there have been essentially the following, first of all, not revenue optimal auction design. That's out of scope for those problems. But what they have been doing is they've recognized that the generalization of Vickrey's ideas often do not work correctly for these settings with complementarities. And they're been developing new pricing rules, but it's somewhat, I think, even the people who have designed these auctions, If they're in the room, they would agree that the design approaches are motivated by theory but somewhat heuristic still.

David Parkes:

But, there's lots of good advances being made. But I don't think it directly plays into what I'm talking about today. And I also, I'll show you some of those things we can scale to. But we, I don't think we can currently scare to combinatorial problems either. Yes.

Crowd questions:

And just in the one dimensional case, like in the add auction case, of thinking about... I'm trying to get a sense of the landscape of how far theory has come in helping to improve practice and kind of whether we should be thinking about other approaches even in that simplified context? I guess I've been surprised by how much gain has come from some of the more straightforward theory like Garbarian, Athy, [inaudible 00:20:21]

David Parkes:

Yeah.

Crowd questions:

And then, I'm thinking about the Yahoo experiment from a few years ago.

David Parkes:

Yeah. Yeah.

Crowd questions:

Where even using basic reserved pricing stuff that they saw big gains.

David Parkes:

Yeah. Yeah, I think that... most of the theory we have is for the one dimensional setting. That's probably the most I want to say in terms of what we can add- there's that and we know how to generalize Vickrey. And we don't- which can handle multidimensional. We don't really have good theory- I'm overstating and I know there are theorists in the room and my friends from the particular Ph. D program against the wall. So I'm overstating things a little big but, okay.

David Parkes:

All right, let me talk about deep learning. There is only one slide. I'll talk about what we do in a bit. You all know that a lot of progress has been made in particular parts of the AI research agenda in recent years. In particular around perception type challenges and also natural language type challenges. And one of the big advances that's been made has been through these more elaborate exotic architectures sometimes for example, convolutional neural nets, this is a cartoon of that, not going to get into the details on that. That are richer, higher parametric than we would have dared to work with in previous years. What changed? Improved tool chains, improved access to computation including GPUs, much larger data sets, and ability through open source software to very rapidly iterate on designs. And these are the various things that have changed to be able to lead to progress, and the cartoon people often draw older algorithms, this sad red curve, deep learning algorithms look more like the blue curve more data and more performance. So they have the capacity to absorb more data and improve their performance.

David Parkes:

And what we wanted to do in this work was we wanted to ask whether we could use deep learning like ideas for optimal enter and economic design. So for the purpose of this work, we think about the deep learning framework. And honestly I'm real just going to be talking about multi-layer neural networks at the moment. We're still fairly early in our research agenda. So if you want, just think about neural networks.

David Parkes:

We are going to think about these as non-linear function approximators. Remember that I said that the optimal design problem is to find functions with good properties? So we're going to think about using the machine learning framework, the technology to pull out functions with good properties. If you'd like its using the machine learning pipeline as a way to optimize. As a way to solve the design problem. We could either generate the data that we use by sampling by distributions, that's what would happen if you were thinking of an auction theorist, you normally assume that you know what the distribution is. You try to find the optimal design. Its quite reasonable to say okay for a distribution like this, let me go ahead and generate lots of data, or you could say this is an approach to data driven economic design where the data that you use is data that you're pulling from some existing platform. Maybe with appropriate work to try to get to values of the underlying participants in the market, not just the bids that they place. You work to do that. And that's how were going to think about machine learning in this context.

David Parkes:

Now, let me get a little bit into the details.

Crowd questions:

So how many, how are you thinking about this, in what... there's a lot of work on machine learning and it is a universal process of functioning, that seems great. But it's this idea of an example seems like a challenge for you. Because lots of the strategies that you want to pull out are things that you've probably never seen with any empirical data. So how do you think about- clearly I think we should be able to do a good job on this as you can see, but you simply...

David Parkes:

Yeah. So the way we're going to think about it is not so much that we're going to be worrying about strategies. What we're going to be doing is we're going to be imposing a sense of compatibility on the design. So we're actually going to, I'm going to talk about that now. We're not going to talk- we're not going to have to worry once we got all of the relevant strategies.

Crowd questions:

Okay.

Crowd questions:

This kind of conceptual analysis. Do you think it will be great if you have a hard prediction task? A hard optimization problem the problem a person can't solve another person that comes to mind.

David Parkes:

[crosstalk 00:25:42]What are we doing?

Crowd questions:

Yeah.

David Parkes:

Yeah so let me give you that eye level. SO, something earlier in literature on trying to use computational techniques, optimizational techniques to solve these problems. As I said earlier, kind of discreetized the input space and had this more explicit representation of the functions if the input is in this voxle then should my out- then who should I give the item to? That kind of explicit representation.

David Parkes:

So one thing I want here is I don't want that. I want to have a parametric... I want to have a parametrized family of functions. Where if were parametrization, I want to have a representation of a continuous mapping from all possibly inputs to the market. At all point- and for each one, an allocation and payments. And, then what are we going to do? We are going to appeal to these nice frameworks to be able to quite quickly prototype rich families of functions that are parametrized. And then what else are we going to do? We're going to do something that shouldn't work, but it is working for quite a few machine learning problems. And for which there is some theory beginning to be developed. We're going to use acatastatic gradient defense for a problem that's very non-complex. And the other thing that we're going to do is we're going to have to handle incentive compatibility that's technical innovation. And I'll explain how we do that.

David Parkes:

There will be some theory by the way, but the theory that I show you, most of the results that I'll show you will be computational results. We do have some theory as well, but not on the ability to optimally solve that non-convex problem. For that we're relying on sharing your computational results. OK?

David Parkes:

All right. Now I have to show you a little about our neural network architectures. This is the architecture for an allocation rule. It looks complicated, it's actually not especially complicated. Let me explain this. So, this is just an example, we have other things going on as well. But in this one example, suppose there are N bidders and M items. Okay? And supposed that the valuation functions are additive. So if I give you two items, you value is the sum of your value for the two items. So what I need in my neural network is I need inputs and outputs and then there will be hidden layers, which we'll be doing computation and we'll be transforming the input into the output. So once the input, the input is M number for bidder 1, the value of the bidder for the first item, the value of the bidder for the M of the item. And then repeat for the other bidders, so that's the input. The input is exactly what you would think it is. The output is going to specify the distribution for a given input. It needs to specify a distribution over allocations.

David Parkes:

So for item 1, this is the probability that item 1 will be assigned to bidder one. This is the probability that item 1 is assigned to bidder N, and so forth, this is for item M. And I can make sure I have a well defined probability distribution on my output layer by using what's called a soft max to make sure everything adds up to 1 in the appropriate way. And then, what's special about a neural network is that these little computers in the middle of the network are doing nonlinear- they're doing non-linear calculations. THey're taking a linear combination of the input they're pushing that linear sum through a non-linear function, in this case we're using Sigmoids. And it kind of looks like this. And then those form the inputs to the next layer and in the networks that we're currently working with, this was the very first architecture we've used in our work. We're doing something very simple. We're just using fully connected hidden layers. So it's not cleverly architected in the form of the convolutional neural networks.

David Parkes:

We're working on new variations now, I have time at the end where I'll explain why we're doing that. SO let me pause. Does anyone have any questions about the basic setup of the, the thing I think I didn't say is where are the parameters? So the parameters live inside each one of these computational units. So this little computational unit has a weight for each one of the inputs. And there's another parameter which is the by bias linet unit. SO think about the parameters of living in the hidden notes. SO I have a parametric representation of a flexible non-linear function approximator and the important property about it is that it's differentiable.

David Parkes:

For a given input, I can computer a gradient for any parameter I care about and computer the parameter of the gradient with respect to the output that I care about. SO if I push a parameter a little bit, how will distribution eradication change? That's what I'm going to care about. So we have an allocation network. We also have, okay, I thought I had one more picture. Yeah, okay this is good. So, forget about the gray arrows. Just look at this bit, I'll explain the gray arrows in a second. We also have a payment network. They're actually components of the same big network. The payments work in an even simpler way. You have the same inputs, you have hidden layers at the end of this, you're just going to compute this thing that we denote P total 1. That's just a number between 0 and 1 that will represent the fraction of your bid that we're going to charge you. SO that's how we'll represent how much you pay. So this is the fraction of the bid on the items that I allocate to you. This is how much you'll pay. You see that here.

David Parkes:

The amount that bidder one pays is this fraction multiplied by the total expected value the bidder has for the items that we allocate. So now I've told you how we can, what we can represent payment rules. And I've told you how we can allocate an allocation rule. And I think I may even have one more picture I'll show you the whole- no I don't have the picture where you see really how these things glue together. What I do want to say is that the objective when you train machine learning models you minimize loss. Loss is typically predictive accuracy what we do here is we minimize negated expected revenue.

David Parkes:

So what's minimizing negated expected revenue? It's maximizing expected revenue. This will play the role of my loss function. The difficult thing, or the technically interesting thins is that if I just told a machine learning engine, okay, find parameters in your allocation net, in your payment net, that do a good job with regard to this loss function, then they're going to learn essentially to charge everybody as much money as they can, and it will be very non-incentivized. Remember Vickrey's auction was strategy proof. Myerson's auction was strategy proof. Strategy proofness is a very useful property because it means that we can say something about what we would expect to happen in the equilibrium of the design.

David Parkes:

The challenging this is we need to find a way to impose incentive compatibility. I'm going to explain how to do that in the next part of my talk. One thing I want to emphasize because this sometimes leads to some confusions. I don't have any training data here. I don't have any labels. I have data but I don't have any labeled examples. Nobody has to come along and tell me how to allocate items or how to charge. All that I need is I need to be able to generate data from a distribution. THat's all I need. And then I'm going to just try to use the framework on machine learning to solve an optimization problem.

David Parkes:

Questions about this before I move on? Yeah?

Crowd questions:

I'm out of my concept. [inaudible 00:34:31] I assume currently that there's value for different items. Like my value for item one is independent from what I get.

David Parkes:

Correct, so this is an additive valuation model. Meaning that if you have value one dollar for wall and value two dollars for the sandwich if they give you both, your value is three.

Crowd questions:

Is it, I assume... generalizing to your earlier call that [inaudible 00:35:07] Is it easy to generate-

David Parkes:

Yeah

Crowd questions:

The algorithm or is it...

David Parkes:

Right. We do have some results in the paper for handling some combinatorial auction examples where you have values on packages. In principle, things generalize, because you have to be able to represent the valuation. So now you have... one part of your input might by the synergy effect if I give you A and B or the substitutes effect. It could be a negative number. Then I just have to have a right way... I have to have a nice way to evaluate your value for a given allocation that I calculate. The nice thing is that I can represent the allocation in the same way. I can still have a distribution over item one, a distribution over item two. I don't need to start representing distributions over packages because I can capture the correlation through what happens inside the network.

David Parkes:

So we do have some results on that, but I wanted to say that the machinery we have is not ready for primetime for very big revenue optimal combinatorial auctions. Yeah?

Crowd questions:

So, I'm sure I'm asking the question not in the right terminology, but how much does the solution, the decisions, the truth of the depend on the number of figures, the number of items and the distribution of the valuations? Cause those are things that you need to know to design it.

David Parkes:

Yes. Great question. The answer is at the moment it is all completely dependent on that. At the moment we're doing the simplest possible thing, which is that we would train a different network for each number of items at each number of bidders. We have generalizations we're currently working on but these results, that's what they do.

Crowd questions:

Okay.

David Parkes:

And they're distribution dependent, and I cannot tell you how robust they have because I haven't tried to do that yet, how robust they are if we change aspects. What I will say is that these questions around robustness, just to take that as an example, these are things that are getting a lot of attention in general in machine learning and the hope is that as there are new improvements made generally, we can bring them into this research agenda. In regard to numbers of bidders, and numbers of items, we're looking to exploit symmetries in the right way, so that we can have one network that can handle multiple bidders and multiple items. And we have some provisional results on that. Yeah. Yes?

Crowd questions:

I'm still a little stuck on the no training labels.

David Parkes:

Yes. Okay.

Crowd questions:

Can you explain that a little more

David Parkes:

Yes.

Crowd questions:

And thinking if I want to train and find the right parameters

David Parkes:

Yes. Yes.

Crowd questions:

For the work I need, it's supervised learning problem, so what am I missing?

David Parkes:

Yes. Right. So normally when we have a loss function, it might represent some approximation of our zero one error with regard to labels. We say dog, it's a cat. That gives you a well defined loss function. But all that I need- and what we do if we're using a deep learning framework, is I use gradient descent. Some flighty, fancy version of gradient defense. To try to move through my parametrized function space to find on with low loss. SO all that we've done here that's different is we've defined a different kind of loss function. And here the loss function- is to be well defined, it needs to be something well defined for every input, at every set of parameters in my network. And what is it here? It's adds up how much money I collected from people on this input for this parametrization and negated. And that's my loss function.

David Parkes:

So I think that one way to think about it is that I have an optimization problem to solve. It's find the parameters and the maximized expected revenue subject to incentive compatibility and the insight is that we can push- we can try to solve that using the standard frameworks from the recent machine learning literature.

Crowd questions:

And you don't need the grant to prove it?

David Parkes:

I don't need, nobody needs to- I don't need Roger Myerson.

Crowd questions:

Right.

Crowd questions:

So with this, just to push on it a little bit, so this is. The contribution in that respect goes far beyond any type of optimal economic design right? You're saying for any optimization problem that we might have.

David Parkes:

Well it's interesting. I don't want to take credit for that because there is now a recent literature in machine learning that- again asking kind of silly things, but it's getting interesting results. Can I use deep learning to solve traveling salesperson problems?

Crowd questions:

Yeah right, right.

David Parkes:

People are beginning to ask that question right now.

Crowd questions:

Right

David Parkes:

Maybe I can model it is a reinforcement learning problem where each action that I take is a new decision by my algorithm.

Crowd questions:

I see.

David Parkes:

But again. WE're paying attention and we want to bring those ideas into this agenda.

Crowd questions:

Cool.

David Parkes:

Yeah.

Crowd questions:

Question, you said you're using these gradient descent methods and yet you don't have convexity. So do you have to have something like a simulated rolling type or thing to allow you to get out of the local minimum or maximum?

David Parkes:

No, but we are going to use randomization because we're going to use acatastatic gradient descent. What that means is that we're going to pick a random subset of our data and calculate the data on the random subset. And then pick another random subset and calculate a gradient.

Crowd questions:

So that does the same thing?

David Parkes:

THere's no theory. I cannot tell you.

Crowd questions:

Sounds good.

David Parkes:

There is some theory now actually, but it isn't my theory and it doesn't apply to something that looks like this. Yes and then I think I'm going to move on because I do want to show you some results. [crosstalk 00:41:01]

Crowd questions:

I just want to pick up a little- If I understand correctly, what you're basically done here is that you're basically built- there's a singultation network that says I have these assumptions about what the bidding functions are going to be therefor, I can create sets of those, I can sort as many as I like, and I know the valuation function at the end because that's the short amount. How much revenue I get. S that's actually just a calculation I can hear. So what you end up having is rather than having- sort of training you for the real world, what you have is a principles based version of creating your features. And then a calculation that gives you your outputs. Is that the right way to think about it?

David Parkes:

Yeah I think you could think about it like that. One thing I think I haven't said is, I'm going to do the following. I'm going to generate my date, assuming that my bidders will be truthful, and then I'm going to make sure that the networks I work will align incentives. So I'm not going to have to speculate about what strategic behavior will look like.

David Parkes:

Okay, good. SO let me move on. Going to skip through fairly quickly, for those of you who are familiar will probably get something out of it. If not, the main point I want to get to show you some of the things this can do. So excuse me for going quickly. So there's a concept of strategy proofness in economic theory that says that you want every bidder for all possible inputs to get more utility by being truthful, not by making a misreport. So the idea of strategy proofness. We can also define the idea of regret. This is how much do you regret having been truthful? SO my regret at bidder I at input V, V is a vector here, it's a value for every bidder. My regret is the maximum utility I could have got for any possible misreport minus the utility I get when I'm truthful. So notice regret is always non-negative, and notice that if my auction is strategy proof, my regret will be 0 on every input. In particular, ignoring measure 0 events, things with no math in regard to the distribution, then I can equivalently write strategy proofness as saying that the expected regret is 0.

David Parkes:

So, you might think that anything that we do, the regret should be 0 everywhere. I have a distribution here, so instead I say the expected regret should be 0. And remember, regret is non-negative. So this is up to measure 0, going to give me strategy proofness if I can do it. So, how to achieve incentive compatibility, what we are going to do is, see I think I'm missing a slide here, apologies. What we're going to do is we're going to solve a constrained optimization problem where we are going to ask that the expected regret be 0. And the way we're going to go about solving that is we're going to use an approach called Augmented Lagrangian Optimization. It's... if you've heard of Lagrangian Optimization, it's a variation on that that does have some nice theoretical results if the problem is convex. What you should think about essentially doing is that, I now don't just care about minimizing my loss. Remember my loss was my negated revenue. I'm also going to penalize my objective if there is regret. I don't like regret. Regret means I am not incentivized. And here's a Lagrangian on that regret term.

David Parkes:

And because we're using what called Augmented Lagrangian, we also have these quadratic terms as well. Essentially you'll get a fixed row and then it makes the Lagrangian recipe will tell you how to update lander.

David Parkes:

Now, we can now use acatastatic gradient descent now jus ton the loss function, but also on these regret terms. We also need to think about how to calculate a gradient on this regret term and we do that in two ways in out work. One way I that we can sample possible misreports. So you should already have in mind that I'm sampling from my sub distribution to generate my training data. And in addition to that, for ever value profile for every bidder in that value profile, I'm going to generate a bunch of additional samples which are the possible strategic manipulations you could use. By picking a bunch of those, I'm going to be able to estimate the regret that that bidder has for being truthful at the current bid price. It's an approximation, I haven't sampled a complete space of misreports. We have another thing that were doing in our recent work and all the results ill show you will take this approach.

David Parkes:

Rather than sampling misreports, think about, as a inner kind of gradient step, I drop a candidate misreport and I'm going to push that candidate misreport through report space as a way to find out your optimal misreport given current parametrization. Maybe ill just show you that, I have an animation of that. I think it might be fun to see, let me show you that right now. I think if I just minimize this, and let's just see if this will play again. Okay great. This is kind of fun. Should be a second... but that's okay. So its going to restart in a second, when it restarts, there will be red dots which will represent the possible misreports. Because this design is truthful, the red dots will all converge to the truth. You see how they start spread out and they follow gradience and they end up all realizing that the right thing to do is to make the truthful misreport. And this is a picture that should be familiar from the one I showed you earlier in my talk.

David Parkes:

So think about this as just being part of the computations that we're doing, as an inner loop to find the right misreport that people should use given the current parametrization as a way to get to the amount that were not strategy-proof. Because I need to be improving rove revenue and failure of strategy proofness as I design. Question there.

Crowd questions:

I guess, I was actually waiting to ask but,

David Parkes:

Yeah

Crowd questions:

so it seems like there's less that you'd actually have to program into it because of adversarial networks vs. that method that you proposed earlier. And so... and there was a well developed set of tools of belief rather than networks used for other problems. Is there a reason you don't just rely on adversarial networks?

David Parkes:

Good. Let's just spend a minute on this because I think not everybody's in the adversarial space but lets just spend a minute on it. Great time to ask a question, this has an adversarial spirit to it and that's maybe why you asked it. I think this inner gradient step is a little bit like there a little learner trying to defeat the main learner. SO actually I think this is adversarial in spirit, what's happening here behind the scenes. But I think I'll maybe take it offline, but we have thought quite a lot about whether we can use generalized adversarial networks as a main design paradigm and I think the main thing we have currently concluded is it's not going to lead to incentive compatibility. And I really want to mimic compatibility I think it might lead to an equilibrium analysis but I'm actually looking for strategy proof designs, and I haven't figured out how to do that. Okay?

David Parkes:

Okay. Let me see if I can get this back up. Full screen. Now I know I'm going to lose some of you at 1 so I'm going to flash some of these results and then well slow down again. So just as a recap. Regret network has an allocation and a payment component. Each have fully connected hidden layers. The training data comes from sampling and known distribution on later values, or from a current equate platform. We make use of Augmented Lagrangian combined with acatastasian gradient descent to solve an incentive constraint training problem. What we're doing is optimizing expected revenue, penalize for non 0 regret, adjusting Lagrangian multiplier during training. We are using the Tensorflow library, we have this coded in Pytorch, we're about to open source everything. We're using adamsovle and running on 1 and video and GPU core. Detest hero learning rate, we need the exercise, etc. etc.

David Parkes:

Data for us is free, so it's not problem generating as many as 640,000 inputs. It's free, we have the distributions. Just generate and at the moment we're using that adversarial grievance approach to compute the regret. We use 1 misreport, not all of the red dots just one red dot, and we push it 25 steps in each of these.

David Parkes:

First of all, can we recover existing results? Let me remind you, Minnelli Vincennes solved in 2006 the one bidder, two item problem. What I've done here is, this is their solution, this is another visualization of their solution. This is how you should allocate item one, this is how you should allocate item two. We train our network, and that's what it learns. [crosstalk 00:50:49] We didn't tell the network any economic theory, other then you should care about incentive compatibility.

Crowd questions:

I just have one question. When you're training this network, you basically hit a button once it's programed in and let it run-

David Parkes:

Yes.

Crowd questions:

Or are you... does it take a lot of... you don't need to do a lot of changes to converge to that? You just, hit a button and you're fine?

David Parkes:

All of the results on the paper are for the same setup. We have had to do some work to get a setup that's working, but we're not reaching yet for the different settings. Yes.

Crowd questions:

You're getting these settings so that causing a constraint becomes a challenge.

David Parkes:

Yes.

Crowd questions:

I'm assuming you do observe non-zero regret.

David Parkes:

Yeah, I'll show you that in a second. This is the first result, this is a recent analytical result by Dashikis at tau for different uniform distributions, this is what the network learns. Notice that it's not exactly the same. I'll show you the revenue of results in a second. It turns out that it looks like the idea that this should be .5 and the network has learned more of a blend. It turns out that it doesn't look important from a revenue perspective. These are low probability parts of the design space, so the network doesn't care very much about getting it exactly right in that region. Here's the Pavlov result I told you about with unit demands. I started seeing these pictures and I'm like okay this is interesting. Here's another Pavlov result uniform to three unit demands. We were very happy when we learned this 0 triangle down here. That was very hard to learn because it's not a very important revenue part of the space

David Parkes:

Just to summarize what I've shown you so far, and on your point. Excuse me. Here are those settings. Here's the optimal revenue, and here's the revenue we learn. And here's our single agent expected expose regret. Very small, one thing that you'll notice in this setting is that we're getting slightly more expected revenue than optimal revenue. Why? Because we're not exactly imposing the incentive compatibility constraints. Now, I don't mean to rush past this. I'll be giving this talk in a slightly different flavor to the MIT, Harvard, micro econ seminar in a few weeks. I can guarantee that the micro economists will give me a hard time about not having exact strategy proofness. And I don't want to hide that from you. I think that's a good debate to have and my personal opinion is this is probably good enough but it's kind of the way of how do I know. So, there is a debate to have here. Yes.

Crowd questions:

Could it be used as congromity to sacrifice. I guess the question is what is more important [crosstalk 00:53:54]

David Parkes:

Right. Right. We don't currently know how to patch this up to get exact strategy proofness. We've thought quite a lot about it and it seems hard.

David Parkes:

Let me keep going, so then the question is can we discover new designs? We've benchmarked things like can we discover the state of economic theory results mind you now not being quite strategy-proof. Can we do new things? So one thing we said is well you know that Yaw result that had two supports. Let's have three supports. We have a new problem now, which is that we don't have the optimal design to compare to because there is no analytical solution to this problem. So what we do instead is we do things like, we benchmark against item-wise Myerson, and bundle-wise Myerson. So these two lines... this our test revenue as we train. This is our test regret as we train. So why is... what, notice the revenue settles down higher than our two baselines, that's good news. Why is revenue falling? It's falling because as we train we tighten its sense of compatibility. That's why revenue falls as we train. Regret falls. This is a generalization of Minnelli Vincennes to two bidders. Remember there were no two bidder designs? There are now two bidder, two item, uniform. Again we end up with higher revenue than the baseline. Regret is essentially 0.

David Parkes:

Okay, what about scaling up. What if we went to three bidders, ten items or five bidders? These are several orders of magnitude more complex than the existing analytical and computational results. Yes.

Crowd questions:

The other people left, so we have less-

David Parkes:

Yes that's true.

Crowd questions:

How do you explain the rules of your allocation option to the participants.

David Parkes:

How, how does. Okay, great question. One thing I will point out before I try to answer is that Google is using GSP, but, which is the generalized second price rule for selling habits. But there's a lot of special magic things happening there that they don't explain. Like a lot of what's happening is they're predicting quality scores. They're predicting reserve prices. There is already, not an excuse for what I'm doing, but there is already, or maybe I'm using it as one, there's already a lot of black box stuff happening. They say it's GSP, no vellicating. But there's a lot of stuff they're not describing because they have all the secret sources around the machine learning techniques. So that's one response. Multi-billion dollar markets that I would argue are pretty opaque.

Crowd questions:

Well they give a lot of simulation. So they give their bidder-

David Parkes:

Yes.

Crowd questions:

[crosstalk 00:56:46]The opportunity to simulate a ton [crosstalk 00:56:48]

David Parkes:

Yes, that's true, that's true.

Crowd questions:

And learn.

David Parkes:

True.

Crowd questions:

What the patterns are.

David Parkes:

Yep. Second response would be that I think we want to be thinking about moving to a world where a lot of the bidding and participating is automated. Not involving humans. My second response, again that doesn't remove the need for some kind of explanation, I think. But I think it helps a lit. And then the third response is I want to agree with you and I want to simply say that there is a rapidly developing interesting on interpretable models within machine learning and the point I made earlier about wanting to import progress if it's made, I just want to say that I hope the extent that it is important, we can do that here. One thing that we are doing is that we have other architectures that will end up generating pictures with really exact lines. I haven't shown them to you here. And we can actually use those to help theorists to refute or confirm conjectures they have about the topology about some of the designs. SO I don't think this thing needs to be completely black box. I think for simple settings, I showed you some visualizations, notice that I'm not showing you visualizations now because obviously I haven't tried very hard but it will get harder and harder to visualize.

David Parkes:

It's a great question, the answer is I think its going to be an important part of the agenda, but I do think there will be settings where we may be willing to trade off much better performance and simplicity for participants because I'm getting strategy proofness. You have to trust me.

Crowd questions:

Trust. Yeah. [crosstalk 00:58:28]

David Parkes:

You have to believe me. [crosstalk 00:58:31] You have to believe me

Crowd questions:

Yes.

David Parkes:

And that's where there could be good synergies with computer science techniques. To kind of establish through possibly crypto style techniques that I am running the function that I claim to be running, so that if I can combine trustworthiness with strategy proofness that could be very powerful. [crosstalk 00:58:54]

Crowd questions:

So you could say that, I'll let you play if- [crosstalk 00:58:57]

David Parkes:

That's actually an interesting point. Of course, we can publish the design and let people play with it but you still have to believe that I'm using that design, but I think we can think about ways to get that to work. Yeah.

Crowd questions:

My question is related to Chiara's and to your point about automated bidders. What do you have to know to even have a breakdown?

David Parkes:

Say it again?

Crowd questions:

What do you have to know to be able to compute your regret? Because it seems to me that-

David Parkes:

As a bidder?

Crowd questions:

As a bidder.

David Parkes:

Yeah, so this is a very strong notion. You have to know everybody else's bits. It's a very strong notion, it's often... I like it very much because it's that strong.

Crowd questions:

It's very strong, [crosstalk 00:59:45]

David Parkes:

But you're right that it's

Crowd questions:

Yeah, it's thinking strategically as a bidder whether to be truthful or not if I know I'm not going to know everyone else's bids.

David Parkes:

Yup

Crowd questions:

Then how would I ever know on regret if I

David Parkes:

Yep, that's right.

Crowd questions:

Is it fair?

David Parkes:

No, that's fair. If you can learn them after the fact, then-

Crowd questions:

So some government auctions and things you could...

David Parkes:

Right.

Crowd questions:

Test the- [crosstalk 01:00:16]

David Parkes:

Yeah. Well the government-

Crowd questions:

Function.

David Parkes:

Well, I should say Spectrum materialization that we don't know the bids. It's literal thinking

Crowd questions:

It's giving me kind of a piece of the design. And I understand you want to have a strong regret function-

David Parkes:

Yes.

Crowd questions:

and then you want to claim the true strategy.

David Parkes:

Yes. Yes.

Crowd questions:

But unless you reveal to me all the other bids-

David Parkes:

You won't know.

Crowd questions:

I can't check that.

David Parkes:

That's true. I think the basic tenants here that we're leveraging within economic theory is that people should know the rules of the game that they're playing. I don't want to move completely away from the standard framing, and here the rules of the game are a little bit complicated to describe.

David Parkes:

If it's okay, lets move on. Let me talk a little bit about scaling up. So what about scaling to three bidders, 10 items, or five bidders. One thing we needed to do here is we needed to do what's called cross validation to see what kind of complexity of network we need. So this is just showing you on some validation data that we don't use for training. One hidden layer, 100 units per layer up to seven hidden layers, 100 units per layer. The minimum regret here was 5 by 100 and you can also see the revenue up here as well. So, I'm going to show you 5 by 100s architectures for the scaled up results. We were able to get higher revenue for these designs, 3 by 10 5 by 10 than again, the relevant either item-wise or the bundles Myerson designs, with low regret. Regret is ticked up a little bit. But also the value on the table is ticked up, so the regret is relative to the values on the table it's still small. I think it's a little bit hard to convey how they... how we think this result is interesting cause I'm just showing you a table of results.

David Parkes:

What I want to say is that not only have the analytical results been very limited but the previous computational results have not gone beyond around 2 bidders and 3 items. So this is really considerably larger than things other people have solved before.

Crowd questions:

How long does it take it to solve?

David Parkes:

So this one takes 13 hours to train on one GPU. We're not using multiple GPUs, we're using multiple GPUs. What I want to say is that in contrast, a linear program takes more than a week for even the two bidder three item setting.

David Parkes:

When we send this paper to places, one reviewed keeps asking us well what about linear programs, what about integer programs? So we bit the bullet and we did some calculations and did some work...

Crowd questions:

It's still out there.

David Parkes:

It's a lot more cumbersome because these things grow exponentially in the number of bidders and potentially the number of items. So I think this is reasonable. I think the way to frame this is you only have to solve your design problem once and then you reap the benefit lots of times.

David Parkes:

Okay, what else can we do- let me briefly tell you some other things we can do. This is a very flexible framework so we can also ask what about budget constrained bidders? There is a literature on this is economic theory. If you're budget constrained then the standard model says that you have utility if you pay the rest of your budget, if I charge you more than your budget your utilities minus incentives. Not good, you cannot find the money, you're very unhappy and what was known about this... it was not known analytically how to optimally sell even a single item to two or more bidders. That's the state of the theoretical literature on revenue optimal budget constrained auction design. One item to two bidders. Why so hard? Because there are two pieces of private information, V and B. Mathematics, thanks, most mathematicians and I [inaudible 01:04:36] so what we can do is we can generalize regret-ness handle this case. All that we do is we redefine regret to amid deviations above budget. That's what we... that's essentially what we need to do.

David Parkes:

So here's a slightly different picture than the one I've been showing you so far. This is pretty much one of the only optimal results in the literature where there's one bidder, one item, values will be distributed and budget uniformly distributed. And this is the density plot showing the probability of allocating that one item to that one bidder and when we trained our network, this was the picture we got. You can see it's not visually exactly the same. You can see that the network has learned, it looks like a reasonable character of the structure, and I believe when we look at the revenue, the revenue will be very competitive to the optimal design. So I saw this as encouraging and actually here's the revenue. This is test revenue as we train, this is the chair Gayle optimal result, and this is our regret by the time, as we train.

David Parkes:

And now we can say okay what about two budget constrained bidders, four items and kind of know where this is going so we can test against the posted price benchmark or an algorithm by Jennifer Chayes and Christian Borgs and others, and we can see we're getting better revenue than other things we can compare to. Yeah, yeah?

Crowd questions:

Is there any, have you considered using other machine learning tools so, you're using the deep learning kind of norm that kind of structured, would there be any benefits from using something different, like some kind of tree or... something different than combining?

David Parkes:

I think it's definitely plausible. We want something that can give me highly configurable... what's beautiful about the neural network structure these days is that because the Tensorflows and the Torches exist, there is just a few lines of codes to capture that feasibility structure. The soft max across, so I don't want to rule out other approaches, but I think that neural network frameworks have an advantage at the moment. If for no other reason then they're very flexible tool kits to code the models in. And also just to remind you that we're not looking for a classifier or something like that. We're looking for a mapping from a continuous space to a continuous space, so it's like a little bit of a strange animal that we want to be able to capture through our architecture.

Crowd questions:

So your goal on this paper would be to find the strategy-proof design, but I'm wondering if that would really be the goal of lets say Google. Bidders are not completely rational, they have cognitive limitations and they might not necessarily figure out the optimal bidding strategies especially if it's fairly complex. It would seem like what you're trying to do here, you could build in some amount of irrationality on the bidders side and that might actually be a huge gain for Google. Might have to take advantage of that. [crosstalk 01:08:07]

David Parkes:

I kind of want to agree and not agree. I agree that one interesting generalization of this work is to build other models of rationality into the framework. At the moment, you're exactly right that through the regret model we're driving towards strategy proofness we're driving towards something that looks like a dominant strategy equilibrium. If you believe the people are doing that ina different way, I believe you can see it pretty explicitly through the adversarial way that I set things up. Yes you can drop in other decision rules there.

David Parkes:

Would Google want to do something like that? Let me not talk to Google specifically but the way I [crosstalk 01:08:56] often think about these things is that... I do a lot of work with equilibrium assumptions but I don't typically want to assume that people will play the equilibrium at least initially, and I don't typically want to assume that people will go ahead and figure out a look out for speculation and figure out what they should do. In some other work, for example I've used replicator dynamics as a way to model the adjustment learning dynamics of a system to see if the population learning dynamics will move toward the truthful points. I tend to think about the world more like that. That is, you design for the fixed points but you don't necessarily assume people will snap into the fixed points and you do assume over time, particularly within the sponsored search engine world there's quite a lot of sophistication around the edges.

David Parkes:

A lot of our algorithmic engines there on behalf... you want to assume that they'll maybe find the equilibrium every time and maybe what you care about, we haven't tried to do this here yet, is designing systems that are learnable. Such that somehow the population will find the point that you want. It is a great question. I try to be very open minded about these things.

Crowd questions:

This is an area where humans shouldn't be allowed to be. There's clearly something that you... there's just no point. And then, the machines can figure out and probably the world will come to your view which is that regret should be small enough so that it's not worthwhile given the computation on the other side to second guess the algorithm. You don't need perfection, you just need a cost function on discovering.

David Parkes:

That's right, and some people have actually tried to model that... this exactly that way, by adding a cost piece and then arguing that now... and since you recover the true equilibrium I'd say that the cost of thinking about manipulation. I think there were some other questions, yeah.

Crowd questions:

You said that a training does not require values in one of the [crosstalk 01:11:24]

David Parkes:

Yes. One per agent per profile.

Crowd questions:

Do you have an idea of is that good enough in general? Or are there cases in which there could be an outcome strategy

David Parkes:

Yeah. This is the place in which we've done most of our computational work, most of our experiments have been to look and see if we're doing the right thing there. There are a lot of observations you can make. You can have 10 particles each moving in five steps or one particle moving 25 steps, you would warm start the particle from round to round, there are a lot of different things that you can do. We've experimented fairly extensively which seems to be a sweet spot I meant when I said we've done some work to tune this thing up and now we're not tuning it for each new result. So, yes I think there are lots of things one can do there to try to improve performance. It's very important to get that right, to be able to robustly drive regret down. Please, yeah.

Crowd questions:

How sensitive are the parameters of the model to the different bidder evaluations you're pulling?

David Parkes:

Yes.

Crowd questions:

Because it seems to me like if you report here on what the actual regret is from the 10,000 samples you trained.

David Parkes:

Yes, yes.

Crowd questions:

What if you take 10,000 samples outside of the distribution, ran it through the same program model-

David Parkes:

Yes.

Crowd questions:

Then what is the regret?

David Parkes:

Yes. I love the question and we haven't looked yet, but we should. These robustness questions are what we need to look at next.

Crowd questions:

Cause I'm curious on... overtime a second price auction that's clear to everyone what it is.

David Parkes:

Yep.

Crowd questions:

The strategy proofness makes sense. But if it's only strategy proof for a fixed set of-

David Parkes:

Yep. One other thing that I will say on that is that we're currently working some theory back into the networks. We can import for example, agent independence. Remember I said that Vickreys design is strategy-proof because your bid doesn't control... because conditioned on you winning, the price you pay doesn't depend on your bid.

Crowd questions:

Yeah.

David Parkes:

So that's very easy architecturally for me to build in here. I just need to drop an edge so that your price doesn't depend on your own report. So we haven't...

Crowd questions:

Oh, so

David Parkes:

We're working on these things right now.

Crowd questions:

Yeah.

David Parkes:

We can actually begin to drop some simple ideas from economic theory back in-

Crowd questions:

Cool.

David Parkes:

As a way to make things more robust. Yeah?

Crowd questions:

So I think you would also modify that network in order to explore theory more. I'm thinking about... I can imagine you adding edges into the graph that are known. You're like this edge is always going to be a 1 or everything is always gone represent the value one your behavior later on in... and you can look and see doesn't the work to converge to make it that known really important or having a particular value around in connection to that note. Like half of your bid or something like that where you would actually read back from the network almost a theory equation for that after you've imprinted.

David Parkes:

Yeah. I think that's a very interesting direction. One of the things we've played a little bit with but haven't had great success yet is pretraining, so you can imagine trying to pretrain with label data. Where the label data maybe comes from Vickrey or Vickrey Clarke Groves or first price or, kind of pretraining is a way to accelerate training. We haven't actually, I don't think we've tried super hard on that yet. But it's a little bit of your flavor, we don't lock things down, but you pretrain them in, start from there. It's something that's quite popular in other types of ML at the moment?

Crowd questions:

Would it be easier to convert?

David Parkes:

Yeah that's the question. I think right now we have a, I don't think we've played with it very much yet.

David Parkes:

Let me just broaden out. I did just want to give you a sense of the broader things that we're not thinking about, about other things that could be in reach. Of course there are lots of optimal economic design questions. I think in fact the title of my talk didn't say revenue optimal auctions, because I think about this more broadly than revenue optimal auctions. Matching problems, we're working with I'm sure some of you know Scott, Scott is an expert on matching. There are lots of things that we don't really know. We know for example that there's an impossibility result that says that you cannot have stability and strategy proofness. Then if I want stability, what's the most incentive compatibility I can get, or if I want strategy proofness what's the most stability I can get? These types of trade offs are things that we just don't understand in the current theoretical literature. Assignment problems without money are also... the way I think about this is that we often don't have grand theories to try to solve these problems, we have quaint solutions.

David Parkes:

Lots of people talk about the way the Harvard Business School lets students access causes and the snake mechanism and there's some theory on that and it's a nice mechanism but I don't know that there's an optimality result per se. Social choice questions, contract design I think might be some of the most interesting directions for this work. Beautiful literature, and I don't know it very well, but the character that I think is true is that most of the results are for one-dimensional problems. Can you use this framework to design optimal contracts for problems that have been out of the reach of the theoretical literature for a long time. I don't know, we haven't tried to work on it, but I think there are lots of things you can try to do with this now.

David Parkes:

We have worked on a facility location problem along a multi-facility location problem. I'm not going to spend any time on it other than to tell you that we do have a paper on this it's a even simpler network, the inputs are the location you want the facility, the outputs are the locations of each of the facilities, we penalize regret, there's no money now. This is a problem that doesn't have good theory for and we learn generalized median style mechanisms. If you know anything about this literature, we're learning to place the facilities, the 25th percentile, the 50th percentile the 75th, the things you would expect to happen are happening.

David Parkes:

So a little bit of discussion, we're already talked about quite a few of these things as being great. I think interpretibility is going to be important in some contexts. I certainly would not advocate at least today that something that looks like this would work for a large, very public project like selling wireless spectrum. I do think that this could be interesting to digital platforms like the Googles, the Facebooks of the world where I think things are already in a way we don't like, but I think they are somewhat opaque. I think this could be an interest there but I think robustness will be very important to drive adoption by those platforms.

David Parkes:

One thing that I should be clear about is that we have approximating sense of compatibility, we have... our notion is minimized expected exposed regret. Exposed regret means you know what other people are doing. And then we take the average of that. What we do not say and what we would love to be able to say, but we have not found a way to do it, we do not say that for any one input, the regret will be smaller than some else. That's what we're unable to say. We're only able to say that in expectation you will leave only a very small amount of money on the table. But it could be that there's one particular input that you find out about, where because our thing somehow was not smooth for some reason, that there was a lot of money on the table. I don't know that that's a problem, but I don't know that it's not a problem.

Crowd questions:

Couldn't you fine it out though? Someplace in your model, points are being calculated.

David Parkes:

The problem is that we only sample. So we don't look everywhere and it's a continuous input space. That's the problem that's why we have a hard time. It's hard even to provide generalization guarantees to say something about this.

Crowd questions:

Som you've got not the perfect...

David Parkes:

I see, can we at least measure this? That's a good point.

Crowd questions:

With enough... with closer, tighter sampling.

David Parkes:

I think we've looked at that, and I think it looks okay, but I'd have to double check.

Crowd questions:

You would get a sense of-

David Parkes:

Yeah, I think it's okay actually, when we have looked.

Crowd questions:

And there might be some nice popular...

David Parkes:

Yeah, so we agree. We're interested in scaling this up. WE're interested in Chiara's question of having one network for generalizers, we have some theory, we have a generalization balance that implies low regret in test data and high probability. We're interested in exploiting symmetry. I think this can handle things like stability matching, envy freeness if you care about that, group strategy proof type properties, I think it can provide theory, we've been trying to use it that way. So to conclude, a new use a deep neural nets for optimal enter ane economic design, it's a data driven approach, we can solve problems currently out of reach of economic theory, it's very important that we're adopting in-expectation objectives. Those are the ones that fit well with learning, and it's flexible. The tagline I've been using for this is differentiable economics. What I mean by that is that you want your economic walls to be things with respect to which you can take a gradience. Diffirentiable economics. If I have differentiable economics, I can use acatastatic rating descents, which I want to use right now. So those are the papers we had on it, thanks for your attention.