Hey, everyone. Thank you for joining us today. My name is Sarah Druss and I'm part of the MIT CSAIL Alliances team. I'd like to thank you for joining us today for the first ever machine learning applications at CSAIL initiative lecture series. I'm pleased today to introduce Dr. Peter Cotton, who is currently Senior Vice President and Chief Data Scientist at Intech Investments.

Dr. Cotton heads Intech's data science efforts in collaboration with their investment team and is also the primary developer of open source software supporting a prediction network. So Dr. Cotton is here today to speak with us about constructing an open prediction network powered by a lottery paradox. If you have any questions throughout his presentation, please feel free to ask them in the chat function and we will present them to Dr. Cotton after his presentation. So please, take it away Dr. Peter.

Thank you very much, Sarah, and thank you for inviting me. It's great to be virtually back in it with you. So I'm going to talk today about a little paradox that always amused me when I was a young child, actually. And I posted it on LinkedIn recently and predictably, someone responded with incredulity.

So, hello. I work for Intech Investments, as Sarah mentioned. And I hope you perhaps contemplate some of the connections between investment and what I'm going to talking about today. But don't worry, I'm not selling you on anything. I refer to myself tongue in cheek as asymptotically the world's most productive data scientist, at least that's what I tell my boss, fellow mathematician Adrian Banner, who runs that company. To illustrate, if there's a quantity of interest that needs to be predicted, for example, here's two lines of Python that calls real time water height from somewhere on planet Earth and I think you know it.

Then we simply take that data, we publish it to microprediction.org, and then before too long, if we just keep doing that, hundreds of algorithms swarm around it and start predicting it. So in this sense, I'm able to start and finish a data science project in 10 minutes, making me arguably 15,000 times more productive than your typical data scientist in a large organization. Anyway, today I'm going to talk about some of the games that occur inside of this contest between algorithms.

And let's start with the lottery paradox. So I know this is elementary, but it's a somewhat acute observation, I think. A town runs a lottery. Here are the rules. You buy a ticket, you write a number between 1 and 10,000. We're going to imagine that perhaps 15,000 or 20,000 people all enter this lottery and they all choose their numbers randomly, showing no preference of one number over another. We're going to assume the town is going to keep 10% rake making it, in theory, a negative expectation game.

But then along comes Mary. Mary says I'm just going to write down 1 to 10,000. I'm going to buy 10,000 tickets, cost of $10,000, perhaps, and ensure that I will win the lottery. Of course, Mary might get unlucky and share the prize with a large number of people. But nonetheless, on average, she's going to make a positive return on this investment. And the ability to reap a positive return in what would seemingly be a negative expectation game is what I will call the lottery paradox.

I call it the lowercase lottery paradox because there is a different uppercase lottery paradox which is quite different. You can look that up on Wikipedia. So how is it that Mary is able to profit? There are a couple of different ways to resolve this paradox, which I think gives us different interesting angles on the problem and I'll run through them.

We can start by just simplifying the problem. Always a good technique in math. Let's pretend the lottery is just heads or tails. You have to write heads or tails and Alice and Bob do that randomly and Mary does it systematically, always writing heads and always one heads and one tail. It's pretty clear that she can do no worse than the combined efforts of Alice and Bob when it comes to maximizing their return.

In the case where Alice and Bob trip on each other's toes and both choose the same outcome, Mary will split the pot with three people. If she wins, which she will, but there's a good chance she'll split the pot with nobody else. And due to the convexity of this game, on average she's going to win.

Now, a different way to look at this is perhaps a different spin on the same lottery paradox. And that is that if w is the average number of people who share the prize and Alice is a random ticket buyer, then Alice is going to share the prize with approximately w other people. Not w minus 1, as is the case for Mary. So why is she unlucky and how can Alice, who is an average random lottery ticket buyer, share the pot with more people than the average?

Again, we can enumerate in the case of a simplified lottery and it's not quite one less because here in the small, it turns out that Alice shares with w minus 1/2, in this case. And you can think of that because Alice is sort of halfway between Mary and a very bad Mary who deliberately picks the same number twice. There's a different sort of paradox lurking here, relating the difference between the population average and the average when we look across different tickets, which is the same as the probabilistic average.

You think about the times when Alice, Bob, and Joe share the prize. let's suppose they're all random lottery ticket buyers. And that's going to count three times in a population average, which is the relevant calculation if you care about what's happening with Alice and Bob. And so it's different from an average over the number of tickets or the number of outcomes. And that's what allows one average to differ from the other.

I think a better resolution of this is as follows. Imagine that Alice wins with ticket number 137. Let's introduce that information. Conditioned on that information, we now know that the mean number of people choosing 137 has gone up by almost one, not quite, but that's an approximate base calculation. In contrast, if I told you that Mary has won [INAUDIBLE] lottery, that conveys, obviously, absolutely nothing. So she doesn't get this plus 1 penalty. And that's why the systematic lottery via Mary is better than Alice.

By the way, there's a different resolution that is perhaps similar. Consider Mary's last ticket. When Mary buys her last ticket, she knows that she's already bought every other ticket. And so the last ticket she buys is kind of a plus 1 when it comes to the number of people who are going to share the prize. Of course, all the tickets are the same, so by symmetry, she's got this kind of plus 1 edge over the random lottery buyer. It's kind of interesting that her first ticket is the same as Alice's ticket, because it's random. But her first ticket subsequently gets better because Alice-- Mary is buying up the other tickets and deliberately not buying the same ticket again.

Anyway, these are just different ways to look at this sort of cute little lottery paradox. If you want to run out and start exploiting lotteries then there's a whole area of mathematics that might interest you, and some other aspects of lotteries, of course. But what we really wanted to talk about is a more surprising paradox, which we'll call indifference.

To see this one, let's suppose now that the town doesn't take a rake from the lottery. It returns all the money. So this is a fair game, if you will. Now, we'll also assume that Mary's investment is very small. As at least one person on this call knows, there are some places in the world where you can buy the equivalent to infinitely divisible lottery tickets of a sort, if you look at combinatorial bets on racetracks.

So we'll imagine that she could buy a hundredth of a lottery ticket or a thousandth, if she wants to, and that her investment doesn't contribute it, materially. We'll also assume that Mary wants to optimize her wealth in the long run and she's allowed to play this game many, many times. So you could imagine she has a logarithmic utility, if you prefer. And finally, and this is the big difference to the previous game, we're going to assume that she can see everyone else's ticket.

So for instance, she knows the ticket number 172 has been chosen five times by other people, whereas ticket 182 has-- number 182 has not been chosen by anyone. You would think that Mary would change her investment strategy in order to spend more money on the underbet lottery tickets and less on those for which the prize is going to be split many ways.

You would think that, but the math says otherwise. It turns out that Mary still buys one of each ticket. Somehow, the risk return exactly balances out, the change in the value of another ticket. And Mary, even though she has access to everyone else's ticket choices, actually doesn't care. Well, when I first came across this I didn't believe it.

The analogy of the racetrack would be that Mary, if we assume she knows with certainty the probability of every horse winning the race, doesn't need to look at the odds. Now, there are some practical things that occur at a racetrack, which sort of spoil this beautiful observation. You have to assume there's no rake, you have to assume that Mary invests the entirety of her wealth every race. Although if you think about it, there's a risk free combination of horses, so that's not really a constraint.

But it is surprising that Mary doesn't take the odds into account when she's deciding how to bet in an optimal fashion. If you don't believe me, let's just do the math. Let's suppose that pi i represents how Mary spreads her bets across different horses. That's her strategy. Let's assume that p i is the true probability of each outcome and that q i is proportional to the investment by everyone else.

So you're playing this game where you're trying to maximize the log of your return, which is kind of ratio of these two numbers. And you have one constraint, that the sum of your investments is presumably some fixed sum number. This kind of problem we typically approach through the first order Lagrange condition. And we take the objective function and subtract some Lagrange multiply, multiply by the constraint, take derivatives.

And what comes out of this is that this quantity, p i over pi i, is independent of i. And it's just proportional to the Lagrange multiplier. So that tells you that Mary is going to invest in proportion to the true odds, that she apparently knows, and doesn't care about q. The log q here is sort of falling out.

If you don't believe me, here is an even more elementary proof. Imagine that she's invested pi i on horse number one and pi 2 and horse number two and that we transfer a tiny amount of her investment from one horse to the other. Set this derivative equal to 0, which it must be if she's doing the optimal thing, and it follows that they are proportional.

There's some interesting connections to measures of distance. Notice that there's a term here that looks like entropy. And Mary's return looks like cross-entropy. I've never personally been able to understand what cross-entropy is and so I like this interpretation because it tells me that cross-entropy is exactly the degree to which you can exploit a lottery if you know the truth. The distance of everyone else's spray of lottery tickets to the truth, in this case uniform, is this [INAUDIBLE] cross-entropy distance.

OK, so much for lotteries. But now what I want to get you thinking about is the fact that the lottery problem is actually far more general than it might first appear. In fact, I suggest to you that most of investment theory is kind of a lottery problem in disguise. And we are distracted by the fact that we tend to invest in linear combinations of different outcomes, called stocks and options and so forth. I won't go into the finance side of things today, though.

But what I will do is delve a little more deeply into a slightly different game. I'm going to call this the continuous lottery game. It's the same as before except this time the town mayor is going to not draw a ticket from 1 to 10,000 integer, he's going to draw a real number. And it's going to be drawn from this distribution. This distribution doesn't have a name yet. I'm going to call it the normalish distribution. It's not the normal distribution, if you look very carefully. And it will come clear why I've chosen it in a minute.

Participants write a number down as before, the mayor's going to pull the real number out of a hat, as it were, and see who's close, see who's within epsilon. And they are the winners. Now I ask you, is this game any different to the original lottery game? I would suggest, in some sense, absolutely not. In fact, here is a function which you can apply to each lottery ticket labeled from 1 to 10,000. And it would take your lottery ticket, and then we'll stick it somewhere on this histogram, which by definition, is the normalish distribution.

By the way, you'll notice this transformation is pretty straightforward. I'm just creating a uniform random number roughly here. And the log 1 minus this guy is a standard way of creating an exponentially random variable from a uniform. What's less appreciated, perhaps, is that if you take an exponentially random variable and take the fourth root, it's approximately normal. So that's just an illustration of going from uniform to this guy.

Are they the same game? Well, they're awfully close, aren't they? Because if you're choosing a point on this curve, you might as well be choosing from here. And if you're the mayor drawing the prize with this distribution, you might as well be drawing a lottery ticket and then applying GI to it. What's nice about this setup is now that we understand this sort of equivalence, we can think about the accuracy.

If we change the game now a bit so that nobody knows the true distribution-- although maybe Mary does-- and the market gets it a bit wrong, the other lottery buyers get it a bit wrong, then we can calculate her return, approximately. It's easier, actually, in the exponential case. But as I've attempted to justify to you, all lotteries are the same anyways, who cares. Let's assume that this is the market's sort of lottery buying and Mary does it exactly right, then we get this term.

And we notice that her return goes as the square, roughly, of how far other people get it wrong. That's not surprising. If you remember the Kelly criterion, it's going to tell you the profit goes is the square of your edge with similar results. Because, of course, and what you win is proportional to how close you are to get it right. But you can also bet more.

For example, if we now relate the exponential to the normal distribution, then you can ask-- if you're playing this normalish game and the market gets it wrong, let's say the market thinks the normal distribution is shifted 10% to the right, 10% of standard deviation. Then Mary's return is going to be 20 points. If it's 1%, it's going to be 20 basis points or 0.2%. So, very interesting connection between her accuracy and profits.

OK, so it may surprise you to know that real time distributional prediction occurs all day every day and it's performed by algorithms. And the algorithms are playing this game. And what we do is we invite them to play a game where they submit distributional predictions of a future data point of live data.

For example, traffic speed in the Bronx. What's the traffic speed in the Bronx going to be 15 minutes from now? Please write down 225 exhibits of what you think it's going to be. And that's your distributional submission. Algorithms do this, as I mentioned, every minute of every day. Hundreds of different data streams. And you can do it too, if you click on the link I posted in the chat.

Why not point estimates? Well, here's a proof without words that point estimates are utterly useless. What happens is, when it comes to rewarding, well, it's just like the lottery. Data point arrives at 12:37, we look back 70 seconds, for example, and we say, who made a prediction at least 70 seconds in the past? And then we reward them.

That's because the community is creating a cumulative distribution. And the cumulative distribution function can be used-- can be provided for anything. Such as the electricity production for New York, generated by wind, one hour ahead from now. And this normalization of data that occurs automatically makes it really easy to see what's sticking out.

So for instance, there's a real time feed of emoji use from the emojitracker project. I put this through the system and out popped all of the emojis that people were using during the first presidential debate. I probably don't have to tell you, the internet reacted with immediate disgust and all sorts of morbid use of emojis vastly exceeded their usual use.

We can also stack these lotteries together now. Now that every incoming data point gets normalized, let's say into a number between 0 and 1, or to a normal looking thing, if you prefer. The algorithms predict those as well. So this becomes kind of a stacking or residual analysis, if you like. You can take it two of those [INAUDIBLE].

--available helicopter challenge data set, pitch and yaw of a helicopter measured in a laboratory. We put it through, the algorithms [INAUDIBLE] percent of yaw, two numbers between 0 and 1. [INAUDIBLE] and we shove them into this time series, these cumulative distributions. You're taking data sets, like electricity, you're passing it through these transformations and you can keep repeating this game.

So you could think of this as sort of like lensing. Composing these monotone transformations until you get to uniform. But doing it in a kind of competitive, collective way. I like to think of this as almost like growing pathways in a sort of probability brain.

Suppose, for example, that somebody submits a distribution of a future data point. Then I could submit a prediction of the transformed version of the original data implied by their contribution. After all, their contribution is a monotone transformation after you do some piecewise linear interpolation or something. And this game can be repeated.

And actually, you can see that down the bottom here, you might have a time series that looks really, really close to uniform. Or if you want to transform it, close to N01. And a prediction-- 225 predictions of the future value of something it's almost uniform, you could take those predictions and you could flip them back up through these monotone transformations all the way to the top level lottery. And so in this manner, you get a composition of intelligence.

And one way to think about this is the law of it interated expectations. We're trying to predict is some quantity, y, extant quantity in the world. And perhaps someone has access to some important piece of data, x, which materially changes the conditional probability of y. Perhaps x is humidity. As in fact, one of our contributors, Rusty Conover, pointed out, humidity is very important in predicting wind generation. Things get heavier and so it goes. But down here, someone might have a different source of data or just a different statistical technique.

And over time these sort of pathways and one of 10 transformations can grow in this collective manner. Now if you want to play, as I mentioned, it's incredibly easy. Just [INAUDIBLE] install microprediction, everything is open source-- not everything. Created a full crawler and run it. It'll generate you an identity, you can cut and pasted into your dashboard. And you can see if it's winning or losing. You can modify it. On the other hand, if you want anything predicted, the same is true.

My hope, of course, is that at some point I'm not the only one standing up a version of it and there are many of them in different parts of the world. And we can all collectively create free bespoke prediction for anything. Why would we want to do that? Well, of course as a hedge fund we have our own reasons. The stock market already is a competitive source of prediction, obviously. The mean of a stock is extremely well predicted.

Everything else is not. Everything else is terrible. Everything else in finances is created by mediocre in-house models or whatever. Whether you're looking at liquidity costs or trading costs, holding periods, client flows, correlations, response to inquiry, cover prices, et cetera, et cetera, et cetera. Everything actually can dictate your operational concerns. It's typically not predicted in this kind of competitive fashion, so it's probably bad.

Another example is prioritizing human work. People think of data cleaning as something you do but before the machine learning. That isn't true. Data cleaning is an inference problem and a prediction problem. You can predict which data points are going to be fixed by humans, you can prioritize their work. It's a great use of repeated prediction.

There are all sorts of different ways to use this kind of setup for enhancing live data feeds, whether it's making sporadic updates of data continuous or discovering exogenous data that's relevant or finding the best techniques or locating the best open source packages for a given purpose. And if you kind of chum the water, as it were, by just throwing out data feeds that are either exactly what you want predicted or maybe just correlated with what you want predicted, then you might find that over time, the algorithms and the people who author them might find relevant sources of data would help you predict something better.

And prediction is an inherently collective activity. A lot of firms spend a lot of money creating dashboards, for example. And-- I think just lost the screenshare, no. And it costs very little extra to create a prediction of the same thing so a company can see forward into the future. Fairness and explanation is also important machine learning these days. But if you can't find the relevant data, how do you know that a particular source of data isn't, in fact, relevant?

Another use of surrogate models. For example, if you have a complex simulation you might want to try to find algorithms that predict it in shorter time. And there are lots of other things that you can do with this if you let your imagination run wild. So I'll leave you with this thought and I'll just briefly give you a so slightly tongue in cheek existence proof for a machine learning network that replaces artisan data science based on these sort of considerations. I hope it's clear.

Maybe it doesn't need to be said that quantitative business optimization is eventually going to be a survival requirement for companies. But most companies can't afford teams of data scientists, so something's going to happen. It's perhaps slightly more controversial that AI, and what we call AI or ML or quantitative business optimization, is essentially frequently repeated prediction.

I don't have time to go into that but once you realize the value functions can be predicted the same way that extant measured quantities can be predicted, you realize that there really isn't anything kind of do with the same setup. And in some sense, frequently repeated prediction is kind of like the electricity itself that drives our business applications. And you want to be able to divorce the intelligence of any application from a source of that predictive power.

With a simple API, you can create an intelligent application and not worry about which models to use or whether to use TensorFlow or forecast [INAUDIBLE] or whatever the latest automated machine learning library is that's come through. Whether you use Flux or so forth. That can all be separated out. You shouldn't have to think about building a power station when you design a toaster.

So if everything is really microprediction, I'll point out that strangers can do the microprediction for you. And I think this is one venue where I don't have to explain that because privacy preserving machine learning is an important part of the research agenda here. However, there are also really simple ways. So if you're interested in having a talk offline about a dozen ways that we use public prediction for private purposes, it's not that hard.

And so whilst we might think about this luxury game I've presented you as just sort of a game, you could come at it from a different angle as well. And you could say, look, this is just an example of algorithms managing other algorithms. And I put it to you that the production of frequently repeated prediction, and therefore the production of artificial intelligence, is something whose fundamental organizing structure is going to undergo a kind of phase change in the future. It has to. Right now it's kind of organized by humans.

But there's no reason that algorithms can't organize it. And there's certainly no reason why, if you setup repeated statistical games such as this and if there is the right incentives in place and if the frictional costs are driven to zero, that more of a market based orchestration of the supply of microprediction is going to dominate. Because the problem isn't a shortage of clever models or data, it's very much a last mile problem. It's a problem of sort of fleeting local knowledge, which is discussed back in the '40s by high ec and other economists.

And I think once we accept that humans are going to step away from this kind of locking role in the production of prediction and the organization of algorithms, then it is perhaps a logical implication of this that it will be orchestrated instead by hierarchies of real time generalized contests. And the reason is quite simple. It's that, if nothing else, you have to compare it to something, it has to be open. You can't claim that it's going to be the best thing or even the canonical thing in the future if someone else isn't free to improve it the same way they can currently improve the price of Apple stock, if you will.

So again, thanks for listening. I hope if this is interesting to you and you join us, here is where you can find us. Thanks to a couple of our contributors, Eric Lou, Rusty Conover. I hope you join us on Fridays at noon for the informal chat. And I'll open it up to questions.

We don't currently have any questions in the chat. If anyone has any questions, you can raise your hand and--

I see one. Oh, one question was-- I see a question. So to clarify what I meant by lensing. Yeah, so perhaps that's an unnecessary Australian analogy. I simply mean taking an incoming data point and applying a distributional function to it. In this case, the distribution function is implied by everyone's predictions.

Ordinarily, we might do something like apply some standard statistical function instead, like the normal distribution, and convert something from approximately normal into approximately uniform, in that fashion. Instead, I'm suggesting that you use the contributed predictions of all of these algorithms to do the same thing.

I just got one, Peter. Do you see intelligence moving to the edge versus the cloud?

Well, I've been pretty neutral on the question of architecture or the practicalities of launching this. It's an interesting question, you know. There's this terminology used currently with-- usually accompanied by a certain set of ideas about how data science should be organized. I mean even what you're saying kind of implies perhaps, what's the setup going to look like inside a large organization where you have a bunch of data scientists producing models?

So perhaps I would answer the question this way in saying that I think it's going to move outside the edge. It's going to fall off the edge. Because anyone can stand up a model right now that just runs on their home computer, anyway, on the cloud, wherever you want to do it. You can run it on GitHub Actions if you want to have Microsoft pay for it. And it can do useful computations and it can monitor data feeds which are sent through the system.

And so, in this brave new world which I'm suggesting, the computation takes place everywhere. The same way that anyone can, if they want to, stand up a web server and participate in the internet, itself. There's actually no reason why anyone couldn't stand up algorithms that participate in a kind of prediction web that powers all sorts of uses inside private organizations or outside them.

So we have one more question. Do you think that relying on microprediction is analogous to news media getting rid of investigative journalists?

That's a great question. That's funny. That's an interesting one. I think there's an element in which that is true and sad, yeah. I think the difference would be that-- I think it's a little different.

I think a better analogy would perhaps be that at some point the distribution of literature to humanity has relied on the monastic calligraphy, or something, and then we found cheaper ways of doing this, which you could argue aren't quite as good. Or don't start off being as good. But at the end of the day, you get a lot more people a lot more benefit.

So yeah, this is certainly an open free attack from below on the analytics industry. But I would point out that right now the vast numerical majority of businesses can't afford that particular thing anyway. Most businesses can't afford to hire a third of a data scientist, never mind a team of multi-disciplinary people to build models for them on uncertain time frames.