EPISODE 99

[INTRODUCTION]

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

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

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

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

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

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

Now, a bit about today's show. In this episode, I speak with Tuomas Sandholm, Carnegie Mellon University Professor and Founder and CEO of startups Optimized Markets and Strategic Machine. Tuomas, along with his PhD student Noam Brown won a 2017 NIPS best paper award for their paper Safe and Nested Subgame Solving for Imperfect-Information Games.

Tuomas and I dig into the significance of the paper, including a breakdown of perfect versus imperfect information games, the role of abstractions in game solving and how the concept of safety applies to gameplay.

We discuss how all these elements and techniques are applied to poker and how the algorithm described in this paper was used by Noam and Tuomas to create Libratus, the first AI to beat top human pros in No Limit Texas Hold'Em; a particularly difficult game to beat due to its large state-base.

This was a fascinating interview that I'm really excited to share with you. Enjoy.

[INTERVIEW]

[0:02:54.3] SC: All right, everyone. After a busy week at NIPS, I am back home and on the line with Tuomas Sandholm. Tuomas is a Professor in the Computer Science Department at Carnegie Mellon University, as well as Founder and CEO of two startups, Strategic Machine and Optimized Markets, as well as an author of the best paper award-winning Safe and Nested Subgame Solving for Imperfect-Information Games at this year's NIPS.

Tuomas, welcome to the podcast.

[0:03:28.7] TS: Thank you for having me.

[0:03:29.9] SC: Absolutely. It is a little bit of a tradition here to have our guests get started by telling us a little bit about their backgrounds and how they got involved in machine learning. You sound like quite a busy guy with your two startups and posts at CMU, so feel free to take some time and let us know how all that fits together.

[0:03:50.3] TS: Yeah, happy to. I've been working on AI since around 1989. In my lab at CMU, there are maybe 25 different research trends, maybe six of them active at any one time. We're probably best known for combinatorial auctions and kidney exchange, so we run the nation with kidney exchange for UNOS with our algorithms. Game solving; solving this large imperfect information games, where we actually reach super human level at strategic reasoning this January by beating the top players in Heads Up No Limit Texas Hold'Em.

My two startups, well the first one of my current ones, I'm also a serial entrepreneur with successful startups in the past. The first one, Optimized Market is really like a combinatorial sales support system for advertising campaigns across media.

In TV, both linear TV and nonlinear streaming, both TV and radio and display advertising and so forth. Where we're optimizing the allocation of inventory to campaigns and pricing and scheduling and so forth.

Then newer startup, which is really brand new is called strategic machine, where we have taken this strategic reasoning technology that we have developed in my lab here at Carnegie Mellon over the last 14 years, and we are commercializing it across a wide range of applications ranging from of course recreational games like poker to business applications and automated negotiation, strategic pricing, product portfolio construction, auctions and so on, all the way to military applications like cyber security, physical security, physical military. Even in stirring biological evolution and adaptation for treatment planning.

[0:05:45.1] SC: Well, that's a pretty broad set of applications.

[0:05:47.2] TS: Yeah. When you're doing the early stages of a startup with a platform technology like this, really the first challenge and the first order to do after technology is ready is to really prioritize the markets and pick and choose where it makes the most benefit and where you can penetrate the fastest and doing that prioritization. We're actually in that process right now.

[0:06:08.5] SC: The paper that we want to talk about is Safe and Nested Subgame Solving for Information-imperfect Games. It sounds like the big distinction there between what you're working on and some of the other things we've seen in the machine learning apply to game arena is in that perfect information part, can you tell us a little bit about that and why that's important?

[0:06:29.9] TS: Yeah, that's exactly right. When you talk about games like checkers, chess, or Go, where AI has seen a lot of success, they are what's called perfect information games. When it is a player's turn to move, the player knows the exact state of the world.

In contrast, in most real-world applications where you have more than one party acting, they're what's called imperfect-information games, where when it's your turn to move, you don't really know the state of the world exactly, so there is hidden information. That can come from us not observing what chance has done so far, as in let's say from the piece, but there's an additional element that there are other players, at least one other player. You may not even observe what the other player has done in the game so far.

Also the other player may have observed different things about what chance has done so far than you have. Now you have private information. You know things that your opponent, or opponent don't know and vice-versa they know things that you don't know. Now it becomes much harder to solve, because you have to think about issues not present in perfect information games, like how do my action signal to my opponent about my private information? Conversely, how do my opponent's actions, how should they signal to me about the opponent's private information?

[0:07:58.0] SC: The approaches that you need to take to solve these games are these two different classes of games are very different then?

[0:08:04.3] TS: They are exactly very, very different. One difference at an intuitive level is that in a perfect information game, which is also called a complete information game, you can actually solve a subgame over the game tree information from that subtree only. That is not true in imperfect-information games. In perfect information games, you have to balance your strategies across the different subgames and that ties the whole game together, so it doesn't decompose.

[0:08:34.8] SC: The idea being that you've got a given player only has access to private information, but there is some other broader set of information that the other player has that you can't use in solving the subtree?

[0:08:47.6] TS: Yes, that's right. Conversely there is information you have that the other player or players don't have.

[0:08:53.4] SC: You've applied this with pretty great success to poker. Can you talk a little bit about the poker as a field of application for these types of approaches and what some of the past approaches have been to solving it?

[0:09:08.3] TS: Yeah, happy to. To me, poker is not really an application. To me, poker is the benchmark and in particular Heads Up Limit Texas Hold'Em has become the leading benchmark in the AI community for testing these application-independent algorithms for solving imperfect information games.

Poker actually goes way back in the history of game theory. If you think about early pioneers, like von Neumann, Morgenstern, Nash and so on, they actually were working on poker as perhaps the lead application of game theory. They were working on small – very small variance of poker that could be solved by hand, like what's called Koon Poker. Now we're able to solve these much larger games, where you have a full deck and you have the full complexity of let's say No Limit Texas Hold'Em.

[0:09:59.7] SC: If I can interrupt briefly, one of the distinctions you make in the paper is that with – is between Limit Texas Hold'Em and No Limit Texas Hold'Em. I am not a poker aficionado, but what you specifically call out that the former that Limit Texas Hold'Em has 10 to the 13 decision points, whereas No Limit creates a much bigger game of 10 to a 161 decision points. What does that all that mean and what's different about the games that makes them so different in terms of the number of possibilities?

[0:10:31.7] TS: Yeah, that's right. Those numbers are exactly right. We can talk about the size of the game based on the number of different situations that the player can face. In Limit, it's 10 to the 13. In No Limit it's 10 to the 161, so fundamentally different.

What makes it different is at in Limit Texas Hold'Em, whatever it is you're trying to bet or raise, there is only one size that you can make. The branching factor in your actions is like two or three. In contrast, in No Limit you can bet any number of your chips up to all of your chips. The branching factor is much larger.

[0:11:10.1] SC: Got it. Based on the different sizes and scopes of these games, with the smaller games one way to solve it is to just solve the whole game. Is that right?

[0:11:20.0] TS: Yes. Yes. That's exactly right. In 2005, my student Andrew Gilpin and I, we solved exactly a previous AI challenge problem called Rhode Island Hold'Em. That has 10 to the power of 9 different situations. We solve it as a whole using firs the technique called lossless abstraction. Then the remaining game, which was about 10 to 7, or 10 to the 8, that we could solve holistically.

Similarly, Michael Bowling's research group from Alberta near optimally solved the two-player Limit Texas Hold'Em by first doing the lossless abstraction, and then having a custom algorithm for holistically solving the whole game near optimally.

[0:12:05.5] SC: What does it mean to do abstraction as a solution for a game like this?

[0:12:10.0] TS: Yeah. Abstraction means that you're generating a smaller but strategically similar game, and then you're solving that smaller but similar game using a Nash equilibrium finding algorithm. The first work on abstraction really in games was manual. People did the abstraction phase manually and the equilibrium finding phase computationally.

This started around really over 15 years ago. Nowadays for the last I would say 12 years, or 13 years, both phase have been done computationally, so there are abstraction algorithms that will find just from the rules of the game, find a strategically similar, but small enough game to solve, or almost exactly solve using equilibrium finding algorithms.

[0:13:01.2] SC: Is there a simple example of abstraction applied to poker that you can give?

[0:13:06.3] TS: Yes. Let me give you two kinds of examples. One kind of example is information abstraction, or abstracting their actions of chance. I might say that in poker, you are dealt ace, ace, or you're dealt king, king. Those are different situations, but they're really very similar. I could say that okay, I'm going to treat them as if they were the same.

The other form also known as action abstraction is abstracting out the players' actions. For example, I might say that if you bet 200 chips, it's almost the same as if you bet 201 chips. I'm going to treat them as if they were the same.

[0:13:47.7] SC: Okay. Does that tend to manifest itself like some kind of binning or quantization type of a course?

[0:13:53.6] TS: You could call it that, but the abstraction techniques are really much more sophisticated than just clustering. You have to think about their downstream effects and all of that. For example, if I have the ace of spades and ace of hearts and king of spades and king of hearts, those might seem very similar in the beginning, but depending on whether they end up in a flush later, they might look very different. I mean, depending on whether they end up in a flush or a straight and so on, while the other hand that was originally similar might actually be very different later.

[0:14:26.1] SC: The abstraction is sounds like a key piece in solving these types of games, but not something that scales for the very large games?

[0:14:36.6] TS: No. I would say almost the opposite. For the very large games, you need to do some form of abstraction, because you cannot holistically solve these very large games. This information abstraction, we talked about an action abstraction. Then there's also a third approach, which is – has traditionally been called face-based abstraction, where you solve some head of the game and have some approximation for the rest of it instead of solving the rest of it exactly. Then you keep doing it over and over.

[0:15:06.7] SC: Got it. The approach that you described in the paper that uses abstraction as a way – as a framework for getting to the solution.

[0:15:17.3] TS: It can. It can. Although just to be clear, abstraction algorithms were not the contribution of this particular paper. Abstraction in imperfect-information game solving a poker goes way back, at least to 2001, if not earlier. The new contribution in this paper was a little different.

[0:15:37.6] SC: Why don't you tell us a little bit about what this paper contributes to this broader problem and with the example application in poker?

[0:15:46.0] TS: Okay. Yeah. All of these algorithms are game independent, so they're not specific to poker. We have a lot of experiments in poker in the paper. What the new techniques are in this paper, they are what we call safe and nested subgame solving techniques. Let me try to unpack that a little bit here.

Safe means that we can guarantee that as we refine our solution, so again we start by computing core solution or what we call a blueprint solution for the whole game and then you start refining your strategy to make it better in subgames that you reach during play. Offline you compute the blueprint, then online when you're playing you refine the strategy in those parts of the search space that you actually reach.

You can afford to do that, because there is less game tree left than in the whole game. You can also afford to do it in a finer model, or a finer and finer abstraction, the closer to the end of the game you are. The problem is as we talked about earlier in the podcast, these imperfect information games, the subgames don't separate out. They don't decompose. How I should play in this current subgame depends on how I should play in other subgames, so I can't really reason about them independently.

Here, what we're doing, we're taking the blueprint strategy that gives values for different alternatives that the opponent could've taken to get here. We use that reasoning to enable subgame solving that has guarantees. It guarantees that the solution quality is no worse than that of the blueprint strategy. Even our worst-case nemesis couldn't take us for more than it could take the blueprint.

Similarly, if the values from the blueprint are off by a little bit compared to what ideal play would give for different states, we can guarantee that the answers that come out of the end-game solver, or sub-game solver also are off by only a little bit. That's the safe part. Then there are two other innovations in the paper.

One is that we can do much better than prior safe in-game solving techniques, by taking into account the fact that actually there are several techniques there to do so, but one of them I'd like to highlight here, which is what we call taking into account the gifts that the opponent has given us so far.

If the opponent has made a mistake in the game so far, we can afford to give back to the opponent as much payoff as the expected mistake cost our opponent so far. You might say, why do we want to give anything back? Well, that allows us to have a bigger space of safe strategies to optimize over and we can do better against other hands, or other private information in general that the opponent might have. That gives us more opportunity to worry about hands that the opponent is really actually likely to have and do even better against those hands, when we can for some unlikely hands that the opponent is going to have, which would've been mistakes to play into this situation. We can afford to give some money back.

[0:19:00.2] SC: If I'm understanding this piece, it's a bit counter-intuitive. Can you provide some intuition or example of how this plays out in a game?

[0:19:08.0] TS: Yeah, absolutely. If you're a poker player you know that that 2, 7 offshoot is the worst starting hand in Texas Hold'Em. That should be folded right away. Let's say we get later into the game and there are two 7's and a 2 on the board, now if you're there with a two 7, that's a great hand because you have a full house. If I think about it, I don't know your cards, but I can pretty much figure out that you don't have the full house, because if you had 2, 7 in your hand, you would've already folded.

Let's say the mistake of getting to this current situation where 2, 7 would've been a $100 for you, then I can afford to give you a $100 back just in case in that scenario where you have that full house. In other words, I have to not be as defensive against that 2, 7 hand and because I don't have to be as defensive there, I can use the flexibility of balancing my strategies to play better against you, against other hands you are more likely to have, like for example a pair of aces or a pair of kings and so forth.

[0:20:09.9] SC: Right. Okay. Got it.

[0:20:11.1] TS: Then the final piece in this paper, final conceptual piece is this nested end-game solving, or nested subgame solving, where we are not just taking the traditional approach of solving the end-game and then playing it out, but we are actually resolving it over and over every time the opponent has made a move.

As we get closer and closer to the end of the game, we can actually add the opponent's moves into our model if the opponent has played actions that we're not in a model in the first place. That way, we don't get confused as to what the opponent has exactly done and how much money is in the pot and so forth. We can still do this in a way that is probably safe.

[0:20:52.6] SC: A lot of the paper is concerned with this formal guarantees, how does this method perform?

[0:21:00.7] TS: It performs very well. We actually reach super human level with this technique. Just to be clear, we reach super human level by playing our AI called Libratus against four of the top 10 human players at Heads Up No Limit Texas Hold'Em in January in this huge 120,020 day event. We beat them with high statistical significance and by a big margin. This end-game solving techniques was one of the three new algorithms that we had in the three different modules of Libratus.

[0:21:36.3] SC: Okay. This algorithm, does the paper give someone what they would need to implement the algorithm, or –

[0:21:43.9] TS: Yes, yes. Actually, let me go back on your previous question also and in addition to humans, we have shown that Libratus beats the best prior AI, which was called Baby Tartanian8, which won the annual computer poker competition in 2016. Libratus actually beats Baby Tartanian8 by a huge margin of 63 milli-big blinds per hand. Just to clarify, we're not just able to beat the best humans, we're also able to beat the best prior AI.

[0:22:12.1] SC: Got it. What was that unit? 63 what?

[0:22:14.4] TS: Yeah, good question. 63 milli-big blinds per game. That's a big blind is a measure of the size of the ante in poker and milli-big blind is $1/1000^{th}$ of that. If you're a poker player and you don't like to use milli-big blinds per game as we do in AI, it's called 6.3 big blinds per hundred. That's just another way of saying the same thing. That's a big margin of victory.

Typically when the AIs play each other in the annual computer poker competition, the top two AIs are separated maybe by 10 or 20 milli-big blinds per hand. Libratus beats Baby Tartanian8, which was the best prior AI by 63 milli-big blinds per hand.

[0:23:02.0] SC: You were about to describe how someone might go about implementing this for poker, or another game.

[0:23:08.7] TS: Yeah. We do layout the algorithms in the paper and we prove their safety. Of course, these are fairly complex algorithms to be honest, especially if you run them on a super computer. It's not that easy to implement, but we tried to make it do our best to explain how they're done.

[0:23:26.1] SC: Is that the typical way that you'd run them on a super computer?

[0:23:29.9] TS: Depends on the game. Most games you probably could run on a laptop, but when we get into these very large games, you can still run them on a laptop, but then you're not doing as well as you would on a super computer.

[0:23:43.8] SC: For the competition, were you running them on a super computer?

[0:23:46.8] TS: Yes, that's right. We were running on Bridges, which is the newest super computer at the Pittsburgh Supercomputing Center.

[0:23:54.3] SC: Okay. This is an evolving game with humans, so each of the moves was then input into the super computer and it would run – what's the typical response time between after entering the human's latest move to getting back what the machine's next move should be?

[0:24:14.1] TS: Yeah. Just to be clear, so the humans played through a browser-based UI. There was no human actually entering the moves into the super computer like there was with Deep Blue playing chess. It goes quite quickly. The entry of data, or back and forth goes very quickly because it's all automated.

The thinking time, well in the first two betting rounds the AI doesn't think at all. It has precomputed its strategy, or I should say typically it doesn't think at all. Then it typically starts thinking on the first move of the third betting round. That's where it thinks the longest.

Overall, if we think about the game of Heads Up No Limit Texas Hold'Em, these top humans were playing on average at 20 seconds per game. Not per move. 20 seconds per game. There's a big difference between the human. Some of them may have been going at like 17 seconds, others may have taken 25 seconds on average, but pretty much 20 or 21 seconds on average.

Libratus played at 13 seconds per average per game. We were playing a little bit faster than the top humans, but by in large at the same speed on average. It's interesting, the thinking pattern is very different. Humans think also on the first two betting rounds while we don't typically, or sometimes in certain situations the AI will they're too, but typically not.

The funny part in my opinion is that humans typically think the longest when there's a lot of money in the pot, because that is a big decision. In contrast, the AI does the opposite. The AI thinks longer if there's less money in the pot, because that's when there is more game tree left, so there's more to be thinking about. It can be quite frustrating for humans when they're used to playing other humans who think a lot on the big pots and go fast on the small pots when the AI does the opposite. It makes big decisions instantaneously almost. Then on the tiny decisions, it can take a long time to think.

[0:26:11.9] SC: This was one of three modules that Libratus used in this most recent match. Are the other published, or are there plans to publish them?

[0:26:22.0] TS: We have talked about them. For example, in my keynote talk at the International Joint Conference on Artificial Intelligence in August, I talked about all three modules, so you can find that on YouTube. We are about to publish the whole thing. It's currently still under review.

[0:26:39.0] SC: Okay. Going back to you mentioning this running on a super computer, what language or other tools does it use? Is it written in something like a Python, or something totally different?

[0:26:49.7] TS: Python would be way too slow for this. We wrote it in C++.

[0:26:53.5] SC: Okay. Does it use like an MPI for inter-process communications or something like that?

[0:26:58.1] TS: Yeah, that's right.

[0:26:59.6] SC: Okay. Interesting. Interesting. Are there any plans to publish source code for this and the other algorithms?

[0:27:07.1] TS: No, not really. It wouldn't be usually that helpful. The supercomputing codes, they are very, very complex. We don't really have any plans to do that. With some other students in my lab, we were writing a game-solving framework that's easier to understand. Wouldn't be as scalable as this, but we might open source something like that, so people can use it for teaching and like I use in my AI courses when I teach AI. We do some amount of imperfect-information game solving there as well for small games. There we do provide source code so that the students can start from that.

[0:27:40.0] SC: Yeah. I would think folks would be very interested in that. Where can folks learn more about your lab and your work?

[0:27:46.5] TS: I would say the best spot to start would be my homepage, so www.cs.cmu.edu/~sandholm. S-A-N-D-H-O-L-M. If you go into the section on equilibrium finding and abstraction and games on my homepage, that's where you can find our papers on this topic. There's a section for all the different trends or research going on in my lab, so you can find papers on all of the different topics.

[0:28:18.4] SC: Awesome. Well, Tuomas thank you so much for taking some time to chat with us about your paper and congratulations on the award.

[0:28:25.4] TS: My pleasure. Thank you very much.

[0:28:26.7] SC: Thanks.

[END OF INTERVIEW]

[0:28:31.9] SC: All right everyone, that's our show for today. Thanks so much for listening and for your continued feedback and support. For more information on Tuomas or any or the topics covered in this episode, head on over to twimlai.com/talk/99.

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

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

[END]

