EUROPEAN UNIVERSITY INSTITUTE

15 March 2021

Frontier Talk

Memo by Philip Hanspach

Cooperating with Machines

Speaker: Jacob Candrall (Brigham Young University)

This event has been organised by the Technological Change and Society Interdisciplinary Research Cluster

This memo summarizes the first in the Cluster's series of "Frontier Talks", which are meant to serve two goals: (1) inviting people to lecture about frontier technologies that are not yet deployed at scale in society; (2) learning from speakers beyond the EUI's social science faculties. The first speaker at a Frontier Talk, Jacob Crandall, from the Computer Science department at Brigham Young University, previously Masdar Institute Of Science and Technology in Abu Dhabi, is an expert in robotics, machine learning and game theory. The topic of his talk is cooperation between humans and machines.

The guiding question of his talk is: How do we create intelligent machines that cooperate with people? In the discussion, we also delved into the questions of how to better understand intelligent machines and how to regulate intelligent machines.

Interacting with humans is important for everyone, including machines.

The need for studying human-machine interaction stems from an insight that has been popularized at least since Dale Carnegie's How to Win Friends and Influence People (1936). This foundational self- help book stresses the importance of the ability to deal with people, even in technical professions.
Dealing with people is also perhaps the biggest problem machines face. It follows that the success of machines will be greatly influenced by their ability to interact with humans.

"Humans will bully mild-mannered autonomous cars."

However, Crandall cautions, there is a catch that becomes apparent, for example, when testing self- driving cars. If a human driver realizes that they are facing a very safety-focused AI-driven car, the human will be tempted to "bully" the AI-driven car by taking its right of way, expecting it to stop.
Another example: Suppose your grandparents live far away and you can only provide them a robot to help them out. The robot needs to "teamplay" with your grandmother, but there may be conflict.

Maybe she doesn't want to take her medicine or maybe the robot does things grandmother doesn't want, but we do. According to Crandall, the key to solve this conflict is for the robot to develop a good relationship with the human.

More generally, there might be conflict between the user and creator of a device. Anecdotally, Crandall recalls his difficulties when first stepping onto a hoverboard: although very much a cooperative problem between human and machine, conflict might arise from the different information available to human and machine. The importance of navigating situations of cooperation and conflict is salient in the shift of international relations that resulted in the system of alliances fighting World War One. An interesting thought-experiment that could be answered by cooperative AI would be counterfactual analysis of alliances and constellations that could have arisen.

Adaptive behavior is key to successful relationships.

As another personal anecdote, Crandall recalls that as his family grew, his oldest child experienced a changing role as two siblings were born. His oldest child had to adapt her behavior to build a good relationship with two siblings. Crandall's eldest daughter took a more assertive role towards the middle sibling, but met with resistance from the youngest sibling using similar behavior. She had to adapt to a more cooperative strategy towards the youngest sibling in order to have a successful relationship (difficult as that may be to measure) with both. What if we replaced the oldest child with a robot? Could a machine have developed different kinds of behavior to develop successful relationships? Moreover, due to the importance of first impressions in human interaction, an emphasis is also to establish good relationships fast.

Game theory meets human psychology.

Crandall analyzes these interactions through game theory. However, human-machine interactions are typically repeated. We know from the Folk Theorem that this can result in many potential equilibria. As humans adapt their strategies and behavior, non-stationarity may complicate the problem. Human psychology also changes around human-machine interaction. Humans might not react well to random exploration of actions and slowly learning algorithms.

A famous early example of human-machine interaction was the pair of chess matches between world chess champion Gary Kasparov and an IBM supercomputer Deep Blue. After winning the first match in 1996, Kasparov was initially in disbelief over his loss in 1997 and initially suspected a human master at play.

Algorithmic strategies in a prisoner's dilemma tournament.

Robert Axelrod in "The Evolution of Cooperation" describes some human interactions as the "Prisoner's Dilemma" (PD), a well-known game in game theory that allows players to cooperate or defect. Importantly, defection yields a higher payoff no matter the other player's choice. Each player could achieve a higher outcome from mutual cooperation than mutual defection. Nonetheless, the Nash Equilibrium, a solution-concept for games studied by game theorists, of a single round of PD is mutual defection.

Observing that humans sometimes cooperate successfully to achieve better outcomes, Axelrod asks what strategy could work to induce cooperation. He finds that "tit-for-tat", imitating the previous move of the opponent, is a robust strategy, winning "tournaments" of repeated PD games against more sophisticated strategies. "Tit-for-tat" can be summarized as "reciprocate cooperation, reciprocate defection, but forgive quickly". However, this strategy falls short when we look at other strategic environments, such as the "Battle of the Sexes" game. Changing the payoffs that arise in the game changes the strategic environment. In fact, there is a whole taxonomy of related games. How do we design algorithms that can deal with these different environments?

Crandall discusses a ranking of many algorithms that face off against each other (plus clones of themselves) in a variety of games. While algorithms manage to beat random play almost all the time, substantial variation emerges between different classes of algorithms.

Reinforcement learning: sometimes powerful, but trending towards myopic decisions

One such class of algorithms are reinforcement-learning (RL) algorithms, a variety that has been popularized by researchers at DeepMind, a sister company of Google. For example, RL algorithms manage to play Atari games at human expert levels and beat skilled Go players. Such problems require large amounts of computational resources and data, or "experience", in the jargon. However, training an RL algorithm is not a magic process where data is fed into an AI to solve any arbitrary problem. What happens behind the scenes requires what Crandall calls "hackery". This means that the path towards a successful RL algorithm requires a lot of tweaking. In addition, the math of these games is clean because of the adversarial nature of the interaction. What happens in self-play with deep-learning algorithms in a PD? The algorithm optimizes towards static optimization and average payoff falls over time. That's because these RL algorithms have a tendency to move towards myopically optimal solutions. In the overall tournament between different algorithms, RL algorithms end up in the middle of the ranking.

Tree-search algorithms: Good at exploiting, bad at cooperating.

There are also new algorithms built around “regret minimization”, the which have found applications, for example, in poker. One researcher in this field is Michael Bowling (University of Alberta). These algorithms sometimes beat experts and are based on tree-search. A problem: poker games are fast and small complications can make the computations very costly quickly. These algorithms end up at the bottom of the tournament scale. They are very good at exploiting optimal actions but find it difficult to establish cooperative equilibria.

Tit-for-tat or bullying?

One approach to build more effective algorithms is to build leaders, which, instead of trying to figure out the other player’s strategy, try to dictate it. This leader-follower model was popularized by Michael Littman and Peter Stone. These researchers moved away from the previous paradigm of asking: What is my associate’s (the other player’s) strategy? What is my best strategy given my model of my associate’s strategy? Instead, they ask: What desirable outcome is likely to be acceptable to my associate? Then the algorithm’s goal is to establish a process that enforces this outcome. Through punishment, the machine can ensure cooperation.

“Generalized tit-for-tat” is only middle of the pack in terms of tournament strategies. Although it does well in the prisoner’s dilemma, it doesn’t do well in other strategic environments. What is better? A “bully” algorithm scores very highly. The “bully” algorithms answer the question what is acceptable to himself and the associate. In result, it offers a strategy to the associate that is better to them than chaos. The same strategy that is effective to manipulate machines may not work against humans, given human psychology, rather than pure rationality.

Builder algorithms: The king and his court carry the day.

The algorithms have to do well against a lot of algorithms. While bully does well against many other algorithms, it does badly when playing against itself. Another class of algorithm succeeds here: “Builder” algorithms. They ask first, what desirable outcome is likely to be acceptable to my associate? Does my associate agree to the outcome I propose? Thus, it has more introspection. If the second question is answered negatively, the builder proposes a new outcome. “Builders” do a lot better, representing 4 out of 5 top ranked algorithms in the tournament example Crandall presents.

Crandall goes in-depth into the best-performing algorithm of the tournament, “S++”. It is

top-ranked in almost all circumstances. One way to describe it is an expert algorithm. Crandall uses the metaphor of a king. The king needs to interact with the queen when both don't necessarily understand each other. The king has a set of experts in his court to give advice. "S++" is a king who picks a set of experts to fill his court. The two problems are then to identify a good set of experts and then to find the best one to listen to. The set of expert strategies represent ideas that the algorithm would want to follow. Such ideas could be safety, being a bully, to accept bullying, modeling the best response to the other person, recommending cooperation, or a mix of cooperation and punishment. Expert strategies are defined by the algorithm designer, but their value for a specific game is computed by an automatic process. Through Markov-processes, we can calculate the value of different strategies. How does the king learn given that he doesn't know the queen's strategies?

The king needs to learn to interact quickly with the queen. For the algorithm, this is ensured through a pruning process: first, we estimate the potential value of each expert's strategy. The algorithm only considers strategies above a certain threshold of predicted value. He will not try out strategies that are initially estimated to deliver low outcomes. This estimate is based on the best outcome that a strategy could deliver, e.g. the payoff of a "bully" strategy if the other player allows to be "bullied".

The real outcome might be lower, depending on the reaction of the other player. The optimal level of pruning also needs to be learned over time and is adapted. For example, if an initially promising "bullying" strategy turns out ineffective other strategies/experts will be considered.

Do humans play well with machines?

In self-learning, the "S++" algorithm also outperforms humans playing against each other over the long term. What happens if we combine them? In self play, "S++" cooperates often, while humans and RL algos cooperate rarely. When pairing humans with "S++", we do not obtain cooperative outcomes nearly as often as in "S++" self-play, and just barely higher than in human self-play. What is the problem? Perhaps the game is not realistic enough to mimic strategic decision problems that humans face. What if we add "cheap talk", i.e. costless communication that carries no commitment? This is known as an effective coordination tool which succeeds in increasing human cooperation. It is difficult do get black box algorithms to speak persuasively, but "S++" has high-level interpretations.

This allows programmers to make it talk, but it is also allowed to stop talking if this is

considered beneficial by the algorithm. A hard-coded process allows for speech. If we let the algorithm talk to the human, do we observe more cooperation? Indeed, the introduction of “cheap talk” allows to double human-machine cooperation. In the speech used towards humans, the machine is often abrasive, using curses and threats. Humans react with praise and explanations. The machines also benefit from communication in self-play. Why do two humans obtain worse outcomes than machines? It turns out that humans lied a lot to each other, hurting relationships long-term for short- term gain. If humans were as loyal as machines, they would have obtained similarly good outcomes.

Some humans consider it unethical to talk to a machine without knowing it. One such example was Google’s chatbot Duplex. When it was first deployed, Google was not upfront revealing that it was a machine. The reason: There is a tradeoff in transparency vs. efficiency uncovered in experiments.

When humans are being told they are interacting with a machine, they are less cooperative. However, when their partner is a machine, but they are being told that they are playing against a human, this results in the highest amount of cooperation. The scenario that results in the lowest degree of cooperation is when a human interacts with another human but is being told that they are interacting with a machine. The psychology of interacting with a machine apparently reduces the quality of the interaction, which probably motivated Google to hide when it was using a bot. Crandall used a Turing test in this environment to learn if humans could tell whether they played against another human or a machine. Although humans were more likely to suspect human play when this was really the case, the algorithms still did a good job of masking the fact that they are, indeed, machines.

Further discussions: Examples of an algorithm, more than two players, learning states in RL.

Crandall’s talk generated much interaction and interest in the audience. One participant asked for more specific examples for the expert strategies. Crandall illustrated some of them using the example of the self-driving car at an intersection. While a “bully” strategy would prescribe to go, even when the other car takes your right of way, a “safety” strategy would be to stop. A “bullied” strategy would react similarly to “safety”. “Cooperate” might resemble a fair process.

Another question was whether the results extend to greater numbers of players? Crandall clarified that the research presented is focused on two-player interactions. The

theory also changes if there are more than two players.

One question went back to the example of learning Atari games from reinforcement-learning based on the observed state. From the sometimes-difficult task of reading a game-state from a screen, what can we learn about reading states? Crandall notes that algorithms for Atari game have problems learning when the reward is delayed. The problem of interpreting the state is that this is a completely data-driven solution, and the algorithm designer must hack-in the entire architecture.
Crandall summarizes a take expressed by Joshua Tenenbaum from MIT: deep learning misses the fundamental aspects of intelligence. His argument is that there is some structure to the brain, which he calls a game engine. This allows the brain in any situation, to predict and simulate what can happen, even in new environments. Algorithms have no such game engine. Data driven statistics is powerful, but it is lacking this notion of intuition that we associate with intelligence.

Further discussions: Regulating intelligence, transparency and explainable AI, AI apocalypse.

How should we regulate intelligence? To this question, Crandall replies that he imagines what a society of machines would look like. One example is to put algorithms in charge of regulating tolls in a classic routing game. Algorithms, both governing cars and tolls, can learn potentially very sophisticated strategies. Which combination of regulatory power and algorithmic sophistication performs best: limited regulatory power and simple algorithms deliver the most efficient outcomes. The combination of no regulatory power and simple sophistication does worst. Unlimited regulatory power is tied for second-best for both levels of sophistication, and adaptive algos with none or limited power are tied for fourth place. Unlimited regulatory power can be bad because it results in cars doing things they don't need to do and a delay in the algorithm reacting to cars. With simpler rules, the cars can model the actions of the regulator accurately and vice-versa. As a clarification, unlimited power here does not mean dictating each car's actions. Efficient regulation perhaps must be dictatorial to be effective in this case. At the same time, this might not be desirable if we care about human autonomy. Humans might also not react well to dictatorial regulation, especially if it is considered not fair.

The audience also solicited Crandall's thoughts on transparency and explainability as one of the longest-running debates in AI. Crandall pointed to a research project with the Office of Naval Research in the United States related to proficiency assessment. As a metaphor,

we might do a proficiency assessment when we look at a rock wall and decide if we can climb it or not. Self-assessed proficiency requires a kind of pseudo consciousness which would an important step to explainable AI. However, given the strong views against creating pseudo conscious AI among some social scientists, the word “consciousness” may be misleading because of its many connotations. But there is an aspect of introspection that is important to intelligence. This becomes apparent when we remember that algorithms are very “brittle” in the sense that they perform very badly once the environment of the problem is changed.

As a comment from a lawyer in the audience, some years ago there was an explosion of apocalyptic claims what AI and machines can do very soon. This sparked writing in legal theory and practice to push lawmakers to “roll up their sleeves” and adapt appropriate legislation. What is the advice for social scientists when dealing with hard science and not fully understanding the potential of the technology? Crandall recalled how at one of the main AI conferences hosted by AAAI, a panel of elderly scientists commemorated the 50th anniversary of Shakey the Robot. There were many predictions that within 10 years, many robots would be able to do many superhuman things. The same problems the AI community is struggling with now, were the same problems they were struggling with, too. We do, however, observe a great increase in the number of people working on AI. Some of it is “AI snake oil”. I don’t see the AI apocalypse coming any time soon, but a transformation through the large number of people that are incentivized to use and develop AI tools for their own entertainment or financial advancements.

Further discussions: Algorithmic bias, curiosity in learning.

What are your thoughts on algorithmic bias? Bias is a fascinating thing, that attracts a lot of attention when discussing AI, e.g. chatbots that appear sexist or racist due to the data they are fed. One way to think about this is the “no free lunch” theorem. You cannot have a classifier that is good in every domain. Learning means in a sense developing bias. When we learn to do something well, this means developing biases. Of course, biases can be very harmful when it results in making stereotypes about individuals and making conclusion about different characteristics about them. But thinking about algorithmic bias might generate valuable insight about human biases. Can AI explain our human biases when it has them itself? Could an AI have a different bias than we have? This could potentially help better understanding human bias.

The discussion concluded with a question on whether including curiosity in the learning process of algorithms is a good thing? Crandall affirms this, recognizing that the

fundamental driving tenet of RL is exploration, e.g. epsilon-greedy exploration. However, this seems different from human, curiosity- driven learning which doesn't seem to be driven by randomness. Therefore, exploring the role of curiosity is a promising direction for research.