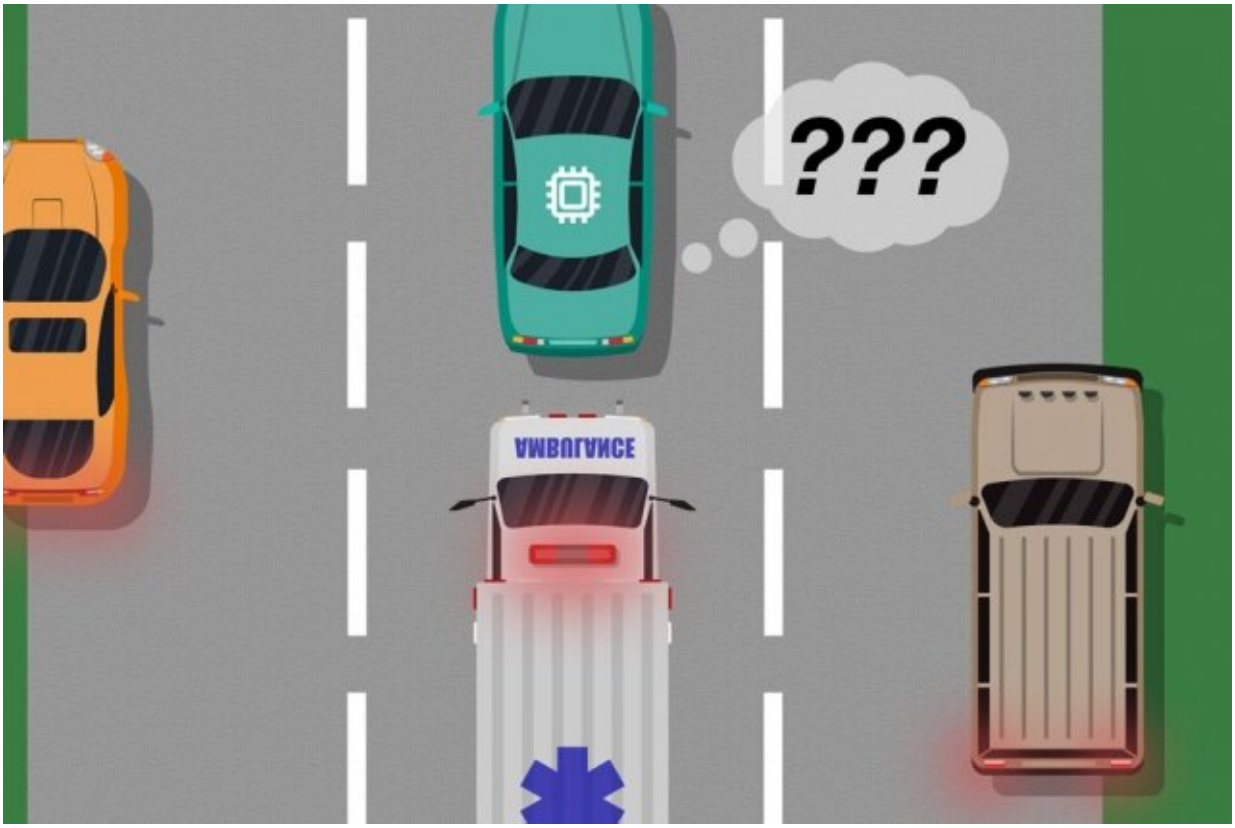
Identifying artificial intelligence 'blind spots'

January 25 2019, by Rob Matheson



A model by MIT and Microsoft researchers identifies instances where autonomous cars have “learned” from training examples that don’t match what’s actually happening on the road, which can be used to identify which learned actions could cause real-world errors. Credit: MIT News

A novel model developed by MIT and Microsoft researchers identifies instances in which autonomous systems have "learned" from training

examples that don't match what's actually happening in the real world. Engineers could use this model to improve the safety of artificial intelligence systems, such as driverless vehicles and autonomous robots.

The AI systems powering driverless cars, for example, are trained extensively in virtual simulations to prepare the vehicle for nearly every event on the road. But sometimes the car makes an unexpected error in the real world because an event occurs that should, but doesn't, alter the car's behavior.

Consider a driverless car that wasn't trained, and more importantly doesn't have the sensors necessary, to differentiate between distinctly different scenarios, such as large, white cars and ambulances with red, flashing lights on the road. If the car is cruising down the highway and an ambulance flicks on its sirens, the car may not know to slow down and pull over, because it does not perceive the ambulance as different from a big white car.

In a pair of papers—presented at last year's Autonomous Agents and Multiagent Systems conference and the upcoming Association for the Advancement of Artificial Intelligence conference—the researchers describe a model that uses human input to uncover these training "blind spots."

As with traditional approaches, the researchers put an AI system through simulation training. But then, a human closely monitors the system's actions as it acts in the real world, providing feedback when the system made, or was about to make, any mistakes. The researchers then combine the training data with the human feedback data, and use machine-learning techniques to produce a model that pinpoints situations where the system most likely needs more information about how to act correctly.

The researchers validated their method using video games, with a simulated human correcting the learned path of an on-screen character. But the next step is to incorporate the model with traditional training and testing approaches for autonomous cars and robots with human feedback.

"The model helps autonomous systems better know what they don't know," says first author Ramya Ramakrishnan, a graduate student in the Computer Science and Artificial Intelligence Laboratory. "Many times, when these systems are deployed, their trained simulations don't match the real-world setting [and] they could make mistakes, such as getting into accidents. The idea is to use humans to bridge that gap between simulation and the real world, in a safe way, so we can reduce some of those errors."

Co-authors on both papers are: Julie Shah, an associate professor in the Department of Aeronautics and Astronautics and head of the CSAIL's Interactive Robotics Group; and Ece Kamar, Debadeepta Dey, and Eric Horvitz, all from Microsoft Research. Besmira Nushi is an additional co-author on the upcoming paper.

Taking feedback

Some traditional training methods do provide human feedback during real-world test runs, but only to update the system's actions. These approaches don't identify blind spots, which could be useful for safer execution in the real world.

The researchers' approach first puts an AI system through simulation training, where it will produce a "policy" that essentially maps every situation to the best action it can take in the simulations. Then, the system will be deployed in the real-world, where humans provide error signals in regions where the system's actions are unacceptable.

Humans can provide data in multiple ways, such as through "demonstrations" and "corrections." In demonstrations, the human acts in the real world, while the system observes and compares the human's actions to what it would have done in that situation. For driverless cars, for instance, a human would manually control the car while the system produces a signal if its planned behavior deviates from the human's behavior. Matches and mismatches with the human's actions provide noisy indications of where the system might be acting acceptably or unacceptably.

Alternatively, the human can provide corrections, with the human monitoring the system as it acts in the real world. A human could sit in the driver's seat while the autonomous car drives itself along its planned route. If the car's actions are correct, the human does nothing. If the car's actions are incorrect, however, the human may take the wheel, which sends a signal that the system was not acting unacceptably in that specific situation.

Once the feedback data from the human is compiled, the system essentially has a list of situations and, for each situation, multiple labels saying its actions were acceptable or unacceptable. A single situation can receive many different signals, because the system perceives many situations as identical. For example, an autonomous car may have cruised alongside a large car many times without slowing down and pulling over. But, in only one instance, an ambulance, which appears exactly the same to the system, cruises by. The autonomous car doesn't pull over and receives a feedback signal that the system took an unacceptable action.

"At that point, the system has been given multiple contradictory signals from a human: some with a large car beside it, and it was doing fine, and one where there was an ambulance in the same exact location, but that wasn't fine. The system makes a little note that it did something wrong,

but it doesn't know why," Ramakrishnan says. "Because the agent is getting all these contradictory signals, the next step is compiling the information to ask, 'How likely am I to make a mistake in this situation where I received these mixed signals?'"

Intelligent aggregation

The end goal is to have these ambiguous situations labeled as blind spots. But that goes beyond simply tallying the acceptable and unacceptable actions for each situation. If the system performed correct actions nine times out of 10 in the ambulance situation, for instance, a simple majority vote would label that situation as safe.

"But because unacceptable actions are far rarer than acceptable actions, the system will eventually learn to predict all situations as safe, which can be extremely dangerous," Ramakrishnan says.

To that end, the researchers used the Dawid-Skene algorithm, a machine-learning method used commonly for crowdsourcing to handle label noise. The algorithm takes as input a list of situations, each having a set of noisy "acceptable" and "unacceptable" labels. Then it aggregates all the data and uses some probability calculations to identify patterns in the labels of predicted blind spots and patterns for predicted safe situations. Using that information, it outputs a single aggregated "safe" or "blind spot" label for each situation along with a its confidence level in that label. Notably, the algorithm can learn in a situation where it may have, for instance, performed acceptably 90 percent of the time, the situation is still ambiguous enough to merit a "blind spot."

In the end, the algorithm produces a type of "heat map," where each situation from the system's original training is assigned low-to-high probability of being a blind spot for the system.

"When the system is deployed into the real world, it can use this learned model to act more cautiously and intelligently. If the learned model predicts a state to be a blind spot with high probability, the system can query a human for the acceptable action, allowing for safer execution," Ramakrishnan says.

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology