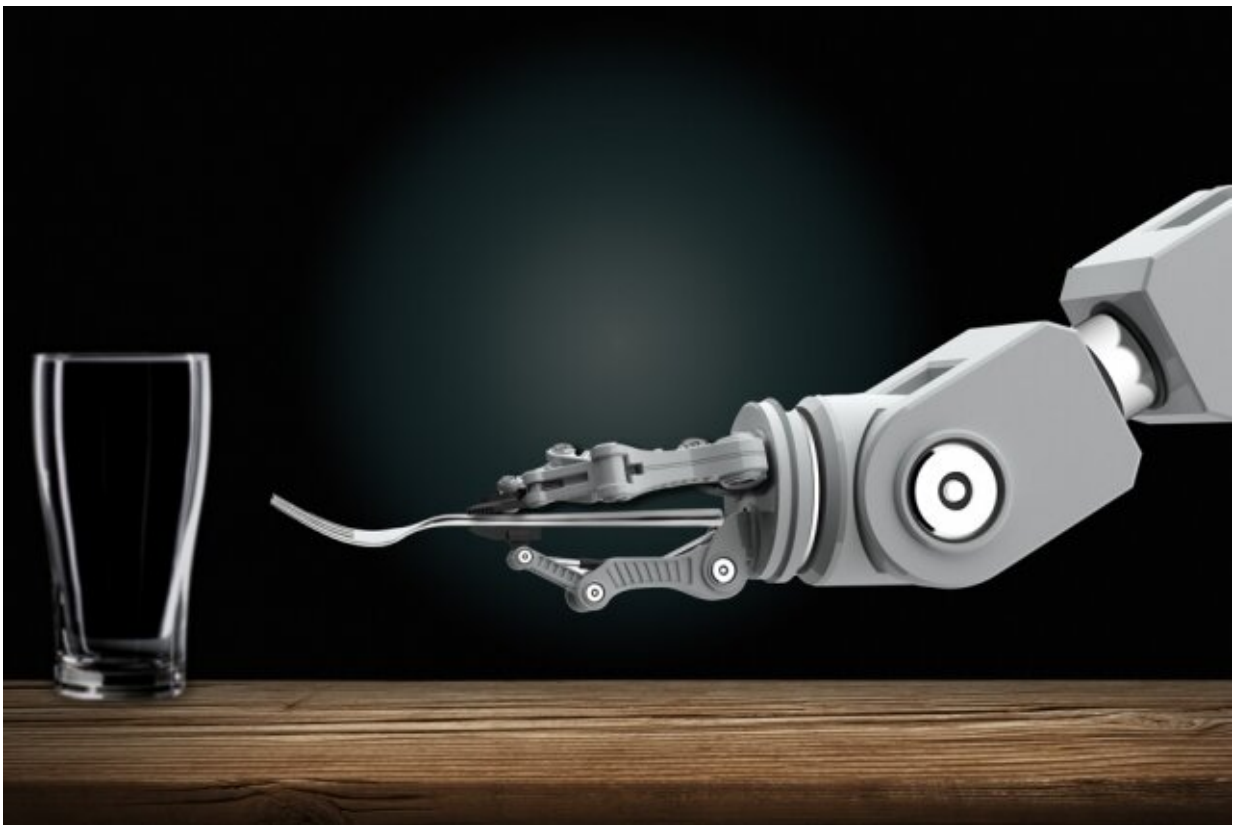# By observing humans, robots learn to perform complex tasks, such as setting a table

March 6 2020, by Rob Matheson

Roboticists are developing automated robots that can learn new tasks solely by observing humans. At home, you might someday show a domestic robot how to do routine chores. Credit: Christine Daniloff, MIT

Training interactive robots may one day be an easy job for everyone, even those without programming expertise. Roboticists are developing automated robots that can learn new tasks solely by observing humans. At home, you might someday show a domestic robot how to do routine chores. In the workplace, you could train robots like new employees, showing them how to perform many duties.

Making progress on that vision, MIT researchers have designed a system that lets these types of robots learn complicated tasks that would otherwise stymie them with too many confusing rules. One such task is setting a dinner table under certain conditions.

At its core, the researchers' "Planning with Uncertain Specifications" (PUnS) system gives robots the humanlike planning ability to simultaneously weigh many ambiguous—and potentially contradictory—requirements to reach an end goal. In doing so, the system always chooses the most likely action to take, based on a "belief" about some probable specifications for the task it is supposed to perform.

In their work, the researchers compiled a dataset with information about how eight objects—a mug, glass, spoon, fork, knife, dinner plate, small plate, and bowl—could be placed on a table in various configurations. A robotic arm first observed randomly selected human demonstrations of setting the table with the objects. Then, the researchers tasked the arm with automatically setting a table in a specific configuration, in real-world experiments and in simulation, based on what it had seen.

To succeed, the robot had to weigh many possible placement orderings, even when items were purposely removed, stacked, or hidden. Normally, all that would confuse robots too much. But the researchers' robot made no mistakes over several real-world experiments, and only a handful of mistakes over tens of thousands of simulated test runs.

"The vision is to put programming in the hands of domain experts, who can program robots through intuitive ways, rather than describing orders to an engineer to add to their code," says first author Ankit Shah, a graduate student in the Department of Aeronautics and Astronautics (AeroAstro) and the Interactive Robotics Group, who emphasizes that their work is just one step in fulfilling that vision. "That way, robots won't have to perform preprogrammed tasks anymore. Factory workers can teach a robot to do multiple complex assembly tasks. Domestic robots can learn how to stack cabinets, load the dishwasher, or set the table from people at home."

Joining Shah on the paper are AeroAstro and Interactive Robotics Group graduate student Shen Li and Interactive Robotics Group leader Julie Shah, an associate professor in AeroAstro and the Computer Science and Artificial Intelligence Laboratory.

Bots hedging bets

Robots are fine planners in tasks with clear "specifications," which help describe the task the robot needs to fulfill, considering its actions, environment, and end goal. Learning to set a table by observing demonstrations, is full of uncertain specifications. Items must be placed in certain spots, depending on the menu and where guests are seated, and in certain orders, depending on an item's immediate availability or social conventions. Present approaches to planning are not capable of dealing with such uncertain specifications.

A popular approach to planning is "reinforcement learning," a trial-and-error machine-learning technique that rewards and penalizes them for actions as they work to complete a task. But for tasks with uncertain specifications, it's difficult to define clear rewards and penalties. In short, robots never fully learn right from wrong.

The researchers' system, called PUnS (for Planning with Uncertain Specifications), enables a robot to hold a "belief" over a range of possible specifications. The belief itself can then be used to dish out rewards and penalties. "The robot is essentially hedging its bets in terms of what's intended in a task, and takes actions that satisfy its belief, instead of us giving it a clear specification," Ankit Shah says.

The system is built on "linear temporal logic" (LTL), an expressive language that enables robotic reasoning about current and future outcomes. The researchers defined templates in LTL that model various time-based conditions, such as what must happen now, must eventually happen, and must happen until something else occurs. The robot's observations of 30 human demonstrations for setting the table yielded a [probability distribution](#) over 25 different LTL formulas. Each formula encoded a slightly different preference—or specification—for setting the table. That probability distribution becomes its belief.

"Each formula encodes something different, but when the robot considers various combinations of all the templates, and tries to satisfy everything together, it ends up doing the right thing eventually," Ankit Shah says.

Following criteria

The researchers also developed several criteria that guide the robot toward satisfying the entire belief over those candidate formulas. One, for instance, satisfies the most likely formula, which discards everything else apart from the template with the highest probability. Others satisfy the largest number of unique formulas, without considering their overall probability, or they satisfy several formulas that represent highest total probability. Another simply minimizes error, so the system ignores formulas with high probability of failure.

Designers can choose any one of the four criteria to preset before training and testing. Each has its own tradeoff between flexibility and risk aversion. The choice of criteria depends entirely on the task. In safety critical situations, for instance, a designer may choose to limit possibility of failure. But where consequences of failure are not as severe, designers can choose to give robots greater flexibility to try different approaches.

With the criteria in place, the researchers developed an algorithm to convert the robot's belief—the probability distribution pointing to the desired formula—into an equivalent reinforcement learning problem. This model will ping the robot with a reward or penalty for an action it takes, based on the specification it's decided to follow.

In simulations asking the robot to set the table in different configurations, it only made six mistakes out of 20,000 tries. In real-world demonstrations, it showed behavior similar to how a human would perform the task. If an item wasn't initially visible, for instance, the robot would finish setting the rest of the table without the item. Then, when the fork was revealed, it would set the fork in the proper place. "That's where flexibility is very important," Shah says. "Otherwise it would get stuck when it expects to place a fork and not finish the rest of table setup."

Next, the researchers hope to modify the system to help robots change their behavior based on verbal instructions, corrections, or a user's assessment of the robot's performance. "Say a person demonstrates to a robot how to set a table at only one spot. The person may say, 'do the same thing for all other spots,' or, 'place the knife before the fork here instead,'" Shah says. "We want to develop methods for the system to naturally adapt to handle those verbal commands, without needing additional demonstrations."