Collaborative learning for robots

June 25 2014

Credit: ThinkStock

Machine learning, in which computers learn new skills by looking for patterns in training data, is the basis of most recent advances in artificial intelligence, from voice-recognition systems to self-parking cars. It's also the technique that autonomous robots typically use to build models of their environments.

That type of model-building gets complicated, however, in cases in which clusters of robots work as teams. The robots may have gathered information that, collectively, would produce a good model but which, individually, is almost useless. If constraints on power, communication, or computation mean that the robots can't pool their data at one location, how can they collectively build a model?

At the Uncertainty in Artificial Intelligence conference in July, researchers from MIT's Laboratory for Information and Decision Systems will answer that question. They present an algorithm in which distributed agents—such as robots exploring a building—collect data and analyze it independently. Pairs of agents, such as robots passing each other in the hall, then exchange analyses.

In experiments involving several different data sets, the researchers' distributed algorithm actually outperformed a standard algorithm that works on data aggregated at a single location.

"A single computer has a very difficult optimization problem to solve in order to learn a model from a single giant batch of data, and it can get stuck at bad solutions," says Trevor Campbell, a graduate student in aeronautics and astronautics at MIT, who wrote the new paper with his advisor, Jonathan How, the Richard Cockburn Maclaurin Professor of Aeronautics and Astronautics. "If smaller chunks of data are first processed by individual robots and then combined, the final model is less likely to get stuck at a bad solution."

Campbell says that the work was motivated by questions about robot collaboration. But it could also have implications for big data, since it would allow distributed servers to combine the results of their data analyses without aggregating the data at a central location.

"This procedure is completely robust to pretty much any network you can think of," Campbell says. "It's very much a flexible learning algorithm for decentralized networks."

Matching problem

To get a sense of the problem Campbell and How solved, imagine a team of robots exploring an unfamiliar office building. If their learning algorithm is general enough, they won't have any prior notion of what a chair is, or a table, let alone a conference room or an office. But they could determine, for instance, that some rooms contain a small number of chair-shaped objects together with roughly the same number of table-shaped objects, while other rooms contain a large number of chair-shaped objects together with a single table-shaped object.

Over time, each robot will build up its own catalogue of types of rooms and their contents. But inaccuracies are likely to creep in: One robot, for instance, might happen to encounter a conference room in which some traveler has left a suitcase and conclude that suitcases are regular features of conference rooms. Another might enter a kitchen while the coffeemaker is obscured by the open refrigerator door and leave coffeemakers off its inventory of kitchen items.

Ideally, when two robots encountered each other, they would compare their catalogues, reinforcing mutual observations and correcting omissions or overgeneralizations. The problem is that they don't know how to match categories. Neither knows the label "kitchen" or "conference room"; they just have labels like "room 1" and "room 3," each associated with different lists of distinguishing features. But one robot's room 1 could be another robot's room 3.

With Campbell and How's algorithm, the robots try to match categories on the basis of shared list items. This is bound to lead to errors: One robot, for instance, may have inferred that sinks and pedal-operated trashcans are distinguishing features of bathrooms, another that they're distinguishing features of kitchens. But they do their best, combining the lists that they think correspond.

When either of those robots meets another robot, it performs the same procedure, matching lists as best it can. But here's the crucial step: It then pulls out each of the source lists independently and rematches it to the others, repeating this process until no reordering results. It does this again with every new robot it encounters, gradually building more and more accurate models.

Imposing order

This relatively straightforward procedure results from some pretty sophisticated mathematical analysis, which the researchers present in their paper. "The way that computer systems learn these complex models these days is that you postulate a simpler model and then use it to approximate what you would get if you were able to deal with all the crazy nuances and complexities," Campbell says. "What our algorithm does is sort of artificially reintroduce structure, after you've solved that easier problem, and then use that artificial structure to combine the models properly."

In a real application, the robots probably wouldn't just be classifying rooms according to the objects they contain: They'd also be classifying the objects themselves, and probably their uses. But Campbell and How's procedure generalizes to other learning problems just as well.

The example of classifying rooms according to content, moreover, is similar in structure to a classic problem in natural language processing called topic modeling, in which a computer attempts to use the relative frequency of words to classify documents according to topic. It would be wildly impractical to store all the documents on the Web in a single location, so that a traditional machine-learning algorithm could provide a consistent classification scheme for all of them. But Campbell and How's algorithm means that scattered servers could churn away on the documents in their own corners of the Web and still produce a collective topic model.

"Distributed computing will play a critical role in the deployment of multiple autonomous agents, such as multiple autonomous land and airborne vehicles," says Lawrence Carin, a professor of electrical and computer engineering and vice provost for research at Duke University. "The distributed variational method proposed in this paper is computationally efficient and practical. One of the keys to it is a technique for handling the breaking of symmetries manifested in Bayesian inference. The solution to this problem is very novel and is likely to be leveraged in the future by other researchers."

More information: "Approximate Decentralized Bayesian Inference." Trevor Campbell, Jonathan P. How. *arXiv*:1403.7471 [cs.LG]. Submitted on 28 Mar 2014 (v1), last revised 12 Jun 2014 arxiv.org/abs/1403.7471

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

Citation: Collaborative learning for robots (2014, June 25) retrieved 12 September 2024 from https://phys.org/news/2014-06-collaborative-robots.html