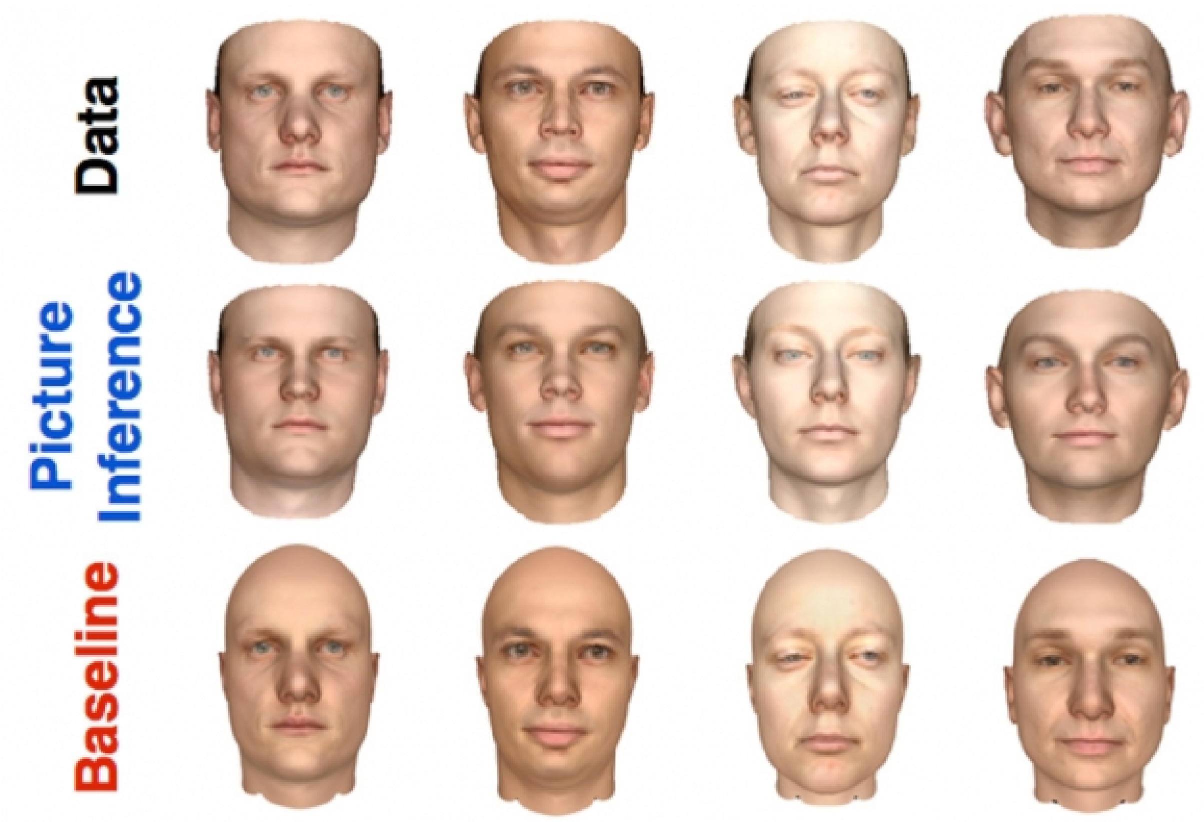
PHYS ORG

Probabilistic programming does in 50 lines of code what used to take thousands

April 13 2015, by Larry Hardesty



Two-dimensional images of human faces (top row) and front views of three-dimensional models of the same faces, produced by both a new MIT system (middle row) and one of its predecessors (bottom row).

Most recent advances in artificial intelligence—such as mobile apps that convert speech to text—are the result of machine learning, in which computers are turned loose on huge data sets to look for patterns.

To make machine-learning applications easier to build, computer scientists have begun developing so-called probabilistic programming languages, which let researchers mix and match machine-learning techniques that have worked well in other contexts. In 2013, the U.S. Defense Advanced Research Projects Agency, an incubator of cutting-edge technology, launched a four-year program to fund probabilistic-programming research.

At the Computer Vision and Pattern Recognition conference in June, MIT researchers will demonstrate that on some standard computer-vision tasks, short programs—less than 50 lines long—written in a probabilistic programming language are competitive with conventional systems with thousands of lines of code.

"This is the first time that we're introducing probabilistic programming in the vision area," says Tejas Kulkarni, an MIT graduate student in brain and cognitive sciences and first author on the new paper. "The whole hope is to write very flexible models, both generative and discriminative models, as short probabilistic code, and then not do anything else. General-purpose inference schemes solve the problems."

By the standards of conventional computer programs, those "models" can seem absurdly vague. One of the tasks that the researchers investigate, for instance, is constructing a 3-D model of a human face from 2-D images. Their program describes the principal features of the face as being two symmetrically distributed objects (eyes) with two more centrally positioned objects beneath them (the nose and mouth). It requires a little work to translate that description into the syntax of the probabilistic programming language, but at that point, the model is complete. Feed the program enough examples of 2-D images and their corresponding 3-D models, and it will figure out the rest for itself.

"When you think about probabilistic programs, you think very intuitively when you're modeling," Kulkarni says. "You don't think mathematically. It's a very different style of modeling."

Joining Kulkarni on the paper are his adviser, professor of brain and cognitive sciences Josh Tenenbaum; Vikash Mansinghka, a research scientist in MIT's Department of Brain and Cognitive Sciences; and Pushmeet Kohli of Microsoft Research Cambridge. For their experiments, they created a probabilistic programming language they call Picture, which is an extension of Julia, another language developed at MIT.

What's old is new

The new work, Kulkarni says, revives an idea known as inverse graphics, which dates from the infancy of artificial-intelligence research. Even though their computers were painfully slow by today's standards, the artificial intelligence pioneers saw that graphics programs would soon be able to synthesize realistic images by calculating the way in which light reflected off of virtual objects. This is, essentially, how Pixar makes movies.

Some researchers, like the MIT graduate student Larry Roberts, argued that deducing objects' three-dimensional shapes from visual information was simply the same problem in reverse. But a given color patch in a visual image can, in principle, be produced by light of any color, coming from any direction, reflecting off of a surface of the right color with the right orientation. Calculating the color value of the pixels in a single frame of "Toy Story" is a huge computation, but it's deterministic: All the variables are known. Inferring shape, on the other hand, is probabilistic: It means canvassing lots of rival possibilities and selecting the one that seems most likely.

That kind of inference is exactly what probabilistic programming languages are designed to do. Kulkarni and his colleagues considered four different problems in computer vision, each of which involves inferring the three-dimensional shape of an object from 2-D information. On some tasks, their simple programs actually outperformed prior systems. The error rate of the program that estimated human poses, for example, was between 50 and 80 percent lower than that of its predecessors.

Learning to learn

In a probabilistic programming language, the heavy lifting is done by the inference algorithm—the algorithm that continuously readjusts probabilities on the basis of new pieces of training data. In that respect, Kulkarni and his colleagues had the advantage of decades of machine-learning research. Built into Picture are several different inference algorithms that have fared well on computer-vision tasks. Time permitting, it can try all of them out on any given problem, to see which works best.

Moreover, Kulkarni says, Picture is designed so that its inference algorithms can themselves benefit from machine learning, modifying themselves as they go to emphasize strategies that seem to lead to good results. "Using learning to improve inference will be task-specific, but probabilistic programming may alleviate re-writing code across different problems," he says. "The code can be generic if the learning machinery is powerful enough to learn different strategies for different tasks."

"Picture provides a general framework that aims to solve nearly all tasks in computer vision," says Jianxiong Xiao, an assistant professor of computer science at Princeton University, who was not involved in the work. "It goes beyond image classification—the most popular task in computer vision—and tries to answer one of the most fundamental questions in computer vision: What is the right representation of visual scenes? It is the beginning of modern revisit for inverse-graphics reasoning."

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: Probabilistic programming does in 50 lines of code what used to take thousands (2015, April 13) retrieved 10 April 2024 from https://phys.org/news/2015-04-probabilistic-lines-code-thousands.html