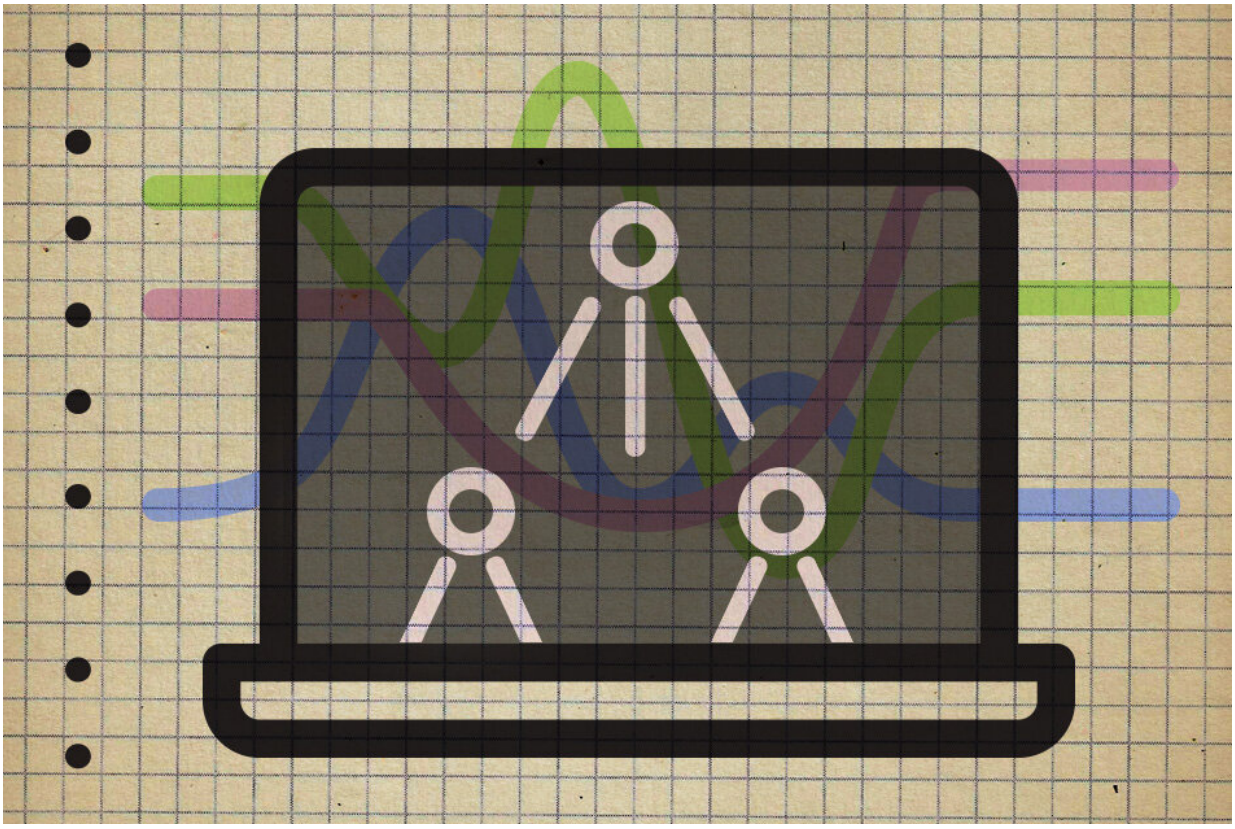
Tool for nonstatisticians automatically generates models that glean insights from complex datasets

January 16 2019, by Rob Matheson



MIT researchers are hoping to advance the democratization of data science with a new tool for nonprogrammers that automatically generates models for analyzing raw data. Credit: Christine Daniloff, MIT

MIT researchers are hoping to advance the democratization of data science with a new tool for nonstatisticians that automatically generates models for analyzing raw data.

Democratizing data science is the notion that anyone, with little to no expertise, can do data science if provided ample data and user-friendly analytics tools. Supporting that idea, the new tool ingests datasets and generates sophisticated statistical models typically used by experts to analyze, interpret, and predict underlying patterns in data.

The tool currently lives on Jupyter Notebook, an open-source web framework that allows users to run programs interactively in their browsers. Users need only write a few lines of code to uncover insights into, for instance, financial trends, air travel, voting patterns, the spread of disease, and other trends.

In a paper presented at this week's ACM SIGPLAN Symposium on Principles of Programming Languages, the researchers show their tool can accurately extract patterns and make predictions from real-world datasets, and even outperform manually constructed models in certain data-analytics tasks.

"The high-level goal is making data science accessible to people who are not experts in statistics," says first author Feras Saad '15, MEng '16, a Ph.D. student in the Department of Electrical Engineering and Computer Science (EECS). "People have a lot of datasets that are sitting around, and our goal is to build systems that let people automatically get models they can use to ask questions about that data."

Ultimately, the tool addresses a bottleneck in the data science field, says co-author Vikash Mansinghka '05, MEng '09, Ph.D. '09, a researcher in the Department of Brain and Cognitive Sciences (BCS) who runs the Probabilistic Computing Project. "There is a widely recognized shortage

of people who understand how to model data well," he says. "This is a problem in governments, the nonprofit sector, and places where people can't afford data scientists."

The paper's other co-authors are Marco Cusumano-Towner, an EECS Ph.D. student; Ulrich Schaechtle, a BCS postdoc with the Probabilistic Computing Project; and Martin Rinard, an EECS professor and researcher in the Computer Science and Artificial Intelligence Laboratory.

Bayesian modeling

The work uses Bayesian modeling, a statistics method that continuously updates the probability of a variable as more information about that variable becomes available. For instance, statistician and writer Nate Silver uses Bayesian-based models for his popular website FiveThirtyEight. Leading up to a presidential election, the site's models make an initial prediction that one of the candidates will win, based on various polls and other economic and demographic data. This prediction is the variable. On Election Day, the model uses that information, and weighs incoming votes and other data, to continuously update that probability of a candidate's potential of winning.

More generally, Bayesian models can be used to "forecast"—predict an unknown value in the dataset—and to uncover patterns in data and relationships between variables. In their work, the researchers focused on two types of datasets: time-series, a sequence of data points in chronological order; and tabular data, where each row represents an entity of interest and each column represents an attribute.

Time-series datasets can be used to predict, say, airline traffic in the coming months or years. A probabilistic model crunches scores of historical traffic data and produces a time-series chart with future traffic

patterns plotted along the line. The model may also uncover periodic fluctuations correlated with other variables, such as time of year.

On the other hand, a tabular dataset used for, say, sociological research, may contain hundreds to millions of rows, each representing an individual person, with variables characterizing occupation, salary, home location, and answers to survey questions. Probabilistic models could be used to fill in missing variables, such as predicting someone's salary based on occupation and location, or to identify variables that inform one another, such as finding that a person's age and occupation are predictive of their salary.

Statisticians view Bayesian modeling as a gold standard for constructing models from data. But Bayesian modeling is notoriously time-consuming and challenging. Statisticians first take an educated guess at the necessary model structure and parameters, relying on their general knowledge of the problem and the data. Using a statistical programming environment, such as R, a statistician then builds models, fits parameters, checks results, and repeats the process until they strike an appropriate performance tradeoff that weighs the model's complexity and model quality.

The researchers' tool automates a key part of this process. "We're giving a software system a job you'd have a junior statistician or data scientist do," Mansinghka says. "The software can answer questions automatically from the data—forecasting predictions or telling you what the structure is—and it can do so rigorously, reporting quantitative measures of uncertainty. This level of automation and rigor is important if we're trying to make data science more accessible."

Bayesian synthesis

With the new approach, users write a line of code detailing the raw data's

location. The tool loads that data and creates multiple probabilistic programs that each represent a Bayesian model of the data. All these automatically generated models are written in domain-specific probabilistic programming languages—coding languages developed for specific applications—that are optimized for representing Bayesian models for a specific type of data.

The tool works using a modified version of a technique called "program synthesis," which automatically creates computer programs given data and a language to work within. The technique is basically computer programming in reverse: Given a set of input-output examples, program synthesis works its way backward, filling in the blanks to construct an algorithm that produces the example outputs based on the example inputs.

The approach is different from ordinary program synthesis in two ways. First, the tool synthesizes probabilistic programs that represent Bayesian models for data, whereas traditional methods produce programs that do not model data at all. Second, the tool synthesizes multiple programs simultaneously, while traditional methods produce only one at a time. Users can pick and choose which models best fit their application.

"When the system makes a model, it spits out a piece of code written in one of these domain-specific probabilistic programming languages ... that people can understand and interpret," Mansinghka says. "For example, users can check if a time series dataset like airline traffic volume has seasonal variation just by reading the code—unlike with black-box machine learning and statistics methods, where users have to trust a model's predictions but can't read it to understand its structure."

Probabilistic programming is an emerging field at the intersection of programming languages, artificial intelligence, and statistics. This year, MIT hosted the first International Conference on Probabilistic

Programming, which had more than 200 attendees, including leading industry players in probabilistic programming such as Microsoft, Uber, and Google.

More information: Feras A. Saad et al. Bayesian synthesis of probabilistic programs for automatic data modeling, *Proceedings of the ACM on Programming Languages* (2019). DOI: 10.1145/3290350