Scientist spans disciplines by looking at networks probabilistically and probabilities as networks

February 8 2013, by Larry Hardesty

Devavrat Shah is an associate professor of electrical engineering and computer science at MIT. PHOTO: ALLEGRA BOVERMAN

Devavrat Shah arrived at Stanford University as a graduate student in computer science in 1999, just a few months after a couple of other students in the department, Sergey Brin and Larry Page, received $25 million in financing for a company that they'd started in a friend's

garage, which they called Google. "The first time I met someone from outside Stanford," Shah says, "the guy said, 'Oh, so you're a PhD in the computer science department? What's the name of your startup?'"

Shah, now an associate professor of electrical engineering and computer science at MIT, had grown up in Vadodara, a city that he describes as "small by Indian standards—only two million people." While his father and grandfather had been professors of mechanical engineering and chemistry, respectively, most of his relatives had gone into business or finance, with considerable success. "I always thought that, since I come from, basically, a business family, I should be doing business," Shah says.

Indeed, only a year into his graduate career, Shah took a leave of absence to work for a networking startup, developing a cheap, high-speed memory circuit for the company's chips. "Once in a while, you get a problem where you can write down a perfect mathematical model for it, and you get an absolutely pretty, elegant solution, which actually becomes highly implementable," Shah says. This was one of those rare instances. Once he had extracted his elegant solution, however, "I realized that the innovative phase is over," Shah says. "The question was, would I continue in the mundane day-to-day job of making sure execution was happening, or look for the next interesting idea?"

Unlike Page and Brin, he decided to go back to Stanford. "Any such industrial job will be 1 percent idea and 99 percent execution," Shah says. In the academy, he says, the ratio is more like 15 percent idea to 85 percent execution. "And that 15 percent somehow appealed to me more," he says.

Taking turns

Shah says that his academic work falls into two main categories:

communication networks and statistical inference. But even that distinction is a little misleading, because he brings statistical analysis to bear on communication protocols and network models to bear on inference problems.

One of his major results in communication, for instance, was an algorithm that allows wireless devices to share access to a wireless router. If two devices try to access the router at the same time, their requests "collide," and neither is able to establish a connection. The ideal communications protocol would ensure that only one device sends data to the router at a time, but that over time, all the devices get equal access.

In a setting like a coffee shop with a Wi-Fi router, where devices are constantly joining and leaving the network and no one device knows what any of the others are doing, that's a demanding requirement. But Shah met it by adopting a statistical approach.

With his algorithm, data transmission proceeds in rounds, and each device starts off with a certain probability that, in the next round, it will attempt to access the router. Every round that it doesn't attempt a connection, its probability goes up, and every time it does establish a connection, its probability drops. Initially, there may be some collisions, but Shah was able to show that the devices will sort themselves into a pattern that, over time, converges on the optimal allocation of airtime.

Graphic design

Conversely, when Shah tackles problems of statistical inference, he generally treats them like networking problems. He represents the problems as graphs, data structures that consist of "nodes"—usually depicted as circles—and "edges"—line segments that connect the nodes. A network diagram is the most familiar example of a graph, but in

Shah's case, the nodes represent data points, and the edges represent correlations between them. The algorithms that establish the strength of those correlations are in fact variations on algorithms used to disseminate messages across networks.

Shah and his group have applied their graphical approach to a wide range of inference problems. One is the prediction of consumers' product preferences on the basis of their buying histories: In tests with a major automaker, Shah's algorithm predicted car buyers' preferences with 20 percent greater accuracy than existing algorithms. Another is predicting what topics will trend on Twitter. With 95 percent accuracy, Shah's algorithm predicted trending topics an average of an hour and a half ahead of time and sometimes as much as four or five hours ahead.

In more recent work, Shah says, his group has addressed problems with crowdsourcing, a technique for breaking up information-processing tasks that are difficult for computers but easy for humans and parceling them out to legions of workers recruited online. Typically, workers are paid a small fee—often just a few cents—for each portion of a task that they complete. Inevitably, some workers will pocket the fee and submit sloppy or outright fraudulent answers. Using graphical techniques, Shah's group was able to calculate the most efficient scheme for introducing redundancy into the distribution of tasks, so that every task elicits at least one correct answer.

As networked computing devices continue to generate data at an exponential rate, Shah says, statistical-inference techniques like the ones he studies will become even more important. "In principle, we can use the information extracted from this data for efficient business operations, improved social living or even winning elections," he says. "I am truly excited about the opportunities that we can realize at MIT in coming years."