Chris Jermaine — Research Statement

The world is awash in data. In medicine, every visit to a doctor, every medical test performed, every device connected to a patient—all these produce data, and increasingly, those data are archived. In medical research, PubMed contains references to 25 million research articles, many of which can be found online and store nearly the totality of human medical knowledge. Commercial concerns generate huge repositories of data; WalMart sees around 140 billion customer visits per year, all of which are logged. 300 hours of video are uploaded to YouTube per minute. The ubiquity of such electronic information has ushered in the Big Data era.

Big Data has the potential to change the world. Data-driven decision making can radically improve patient care. Scientific hypotheses can be automatically generated from a corpus of research papers, pushing science forward. Companies can use data to better serve customers and increase profits.

Unfortunately, realizing value from Big Data is not easy. Analysts struggle with the *size* of the data, the *variety* and *heterogeneity* of the data, and the *accuracy* of the data.

I study *data analytics*: how to build software to analyze, store, retrieve, and manipulate huge and heterogeneous data sets. Among those who work on such problems, my skill set is unique in that I combine expertise in the statistics that underlie modern data analytics and machine learning with expertise in building software systems. My work focuses on:

1 The systems-oriented problems that arise when building software to manage large and diverse data sets; and

2 The foundational, mathematical questions that arise when statistical methods are used to analyze such data sets.

The research questions that excite me most span *both* systems-building *and* applied statistics: How does one build software systems that allow the application of statistical methods to Big Data?

Since my arrival at Rice in January 2009, my research projects have included:

Systems for Managing Uncertainty (supported by the National Science Foundation and the Department of Energy; in collaboration with IBM Almaden Research Center). Sources of uncertainty in data abound: measurement errors, missing values that must be imputed, uncertain predictions about the future, entity resolution, and so on. It is very natural to characterize uncertainty about data using a probability distribution. This introduces randomness into data analysis, giving rise to the field of *stochastic analytics*. A long-term project has been the design and implementation of the *Monte Carlo Database System* (or MCDB), which allows a user to specify the uncertainty present in his or her data; the system then automatically takes those uncertainties into account when answering questions over the data. For example, after specifying an appropriate statistical model, the user can ask, “What would my profits have been in 2014 had I raised my margins by 5%?” and the system will use the model to compute a *distribution* of possible results, taking into account the uncertainty in the modeling process. Because the system itself computes the model and performs the calculation on a distributed compute cluster, MCDB scales to very large data sets (terabytes of data) and very complex models.

Systems for Supporting Large-Scale Machine Learning (supported by the National Science Foundation and the Department of Defense; in collaboration with IBM Almaden Research Center). Applying statistical machine learning to the largest data sets is difficult. The process involves three steps: (1) designing a new statistical model or doing the math necessary to apply an existing one to a particular analysis task; (2) building and evaluating a prototype of the model using a mathematical programming tool such as R or MATLAB; and (3) implementing a version of the model that can run on a cluster of many computers, often in the "cloud" (hosted by Amazon or Microsoft). Step (3) in particular is difficult and error-prone, often taking a team of people with specialized skills months of trial and error. We focus on making this step much less painful, by developing systems that take code that looks a lot like what was prototyped in step (2), and *automatically* figuring out how best to run it in the cloud. One such project is SimSQL, a system that uses techniques from parallel database systems to take a high-level specification of a learning algorithm and run it in parallel on hundreds or even thousands of machines. A public release of SimSQL is available (`http://cmj4.web.rice.edu/SimSQL/SimSQL.html`).

Analysis of Large Biomedical Data Sets (supported by the National Science Foundation; in collaboration with researchers from Baylor College of Medicine, MD Anderson Cancer Center, and Texas Children's Hospital). Increasingly, patient data is collected and stored electronically. This includes structured data (demographics, billing codes, procedures performed), vital sign time series, textual notes and emails, images, and even genetic data. Analyzing these data has the potential to improve medical care and reduce costs. I am particularly interested in developing models to predict long-term outcomes. For example, we recently designed a new statistical model that can be used to analyze surgical vital sign data to predict outcomes such as mortality. The model has several uses, including giving anesthesiologists feedback on performance, and in optimizing post-operative care. We are now working on an outcome-prediction model that ingests data collected as a patient makes his/her way through the surgical process (initial admission, pre-operative assessment, surgery, etc.). As time passes, the model's accuracy increases, to such an extent that a few days after surgery, we can predict outcomes with almost total certainty.

Systems for Management of Complex Data (supported by the Department of Defense). Classically, database systems were not designed to store complex data structures: graphs, trees, maps, etc. But increasingly, modern data analytics tasks feature exactly such types of data. For just one example, consider one of the Big Data tasks we are working on at Rice: mining "Big Code"—the hundreds of billions of lines of program source code produced by millions of programmers over the last few decades. Data extracted from these codes are naturally represented as trees (abstract syntax trees) and graphs (dependencies among program modules and data). We are building a system at Rice called the "Pliny Database," or PDB for short, that is meant for storing just such complex data types on large compute clusters consisting of hundreds or thousands of machines. PDB combines super high-performance with ease of use.