Research Statement and Agenda

Eugene Wu
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
eugenewu@mit.edu

"Big Data" is a very real problem, and as organizations embrace data-driven decision making, tools that organize, simplify, and automate the process will be critical. With such widespread adoption, database users cannot be expected to have broad or deep understandings of data management techniques. Instead, they will be domain experts that need simple ways to express and manage complex analysis processes. This fundamental shift highlights the need to address challenges at every step in the analysis pipeline — from automating the process of loading raw data into management tools [4], improving core query performance [**?**, 5, 8], extending databases to support new forms of data [6, 2, 10], helping the user understand anomalies in their query results [9], and extending database techniques to support visual interaction [7].

My research approach emphasizes end-to-end systems building that range from developing efficient indexes and storage layouts that improve core database query processing, to leveraging machine learning that automate user analysis, to designing intuitive query and visual interfaces. I believe that no single technique is sufficient for addressing all data analysis problems and taking a comprehensive view is necessary to identify and develop real data management solutions.

Research Projects

In my thesis, I developed a primitive in data management systems that can track the records that contributed to a given output record. This is particularly important in exploratory settings where data scientists run a complex array of transformations that reformat and clean input data, rescale dimensions, compute summaries using different algorithms, and combine outputs. When the analyst identifies trends or outliers in the result, it is crucial be able to ask how those results were computed. Tracking such provenance information is expensive, as many common statistical summaries depend on a large number of input data elements and if implemented naively, the overhead of storing and querying the provenance is impractical. Similarly, provenance results can easily innundate the analyst with non-informative results (e.g., the sum of all values in a table depends on the entire table). Thus, although the concept of provenance has been explored in database systems for several years, general purpose provenance systems remain widely unused in modern applications

To address these limitations, I have explored the three key dimensions of the provenance problem – overhead, latency, and query result quality – that must be tackled to make provenance systems practical. Specifically, I have developed three research systems that address these dimensions. Subzero [10] is a data management system reduces the overhead by tuning the system's indexing, materialization, and encoding behavior based on user-specified runtime and storage constraints. Scorpion [9] is an outlier explanation framework that summarizes the most influential inputs that generated an aggregate outlier. Finally, Smoke [7] is a provenance system for interactive scenarios that can execute provenance queries with low-latency.

SubZero: A Low Overhead Provenance System

Scientific applications such as the Large Synoptic Survey Telescope (LSST) demand provenance at the record or pixel level (e.g., "what pixels generated this star?") for debugging and data validation purposes. Such workflows truly fall into the category of "big data" (LSST processes 2GB/sec each night), but also have tight constraints on the amount of storage and runtime overhead they are willing to incur (LSST can allocate <20% of storage for provenance information and must process each image within 15 seconds). Furthermore, many operators are vectorized, so even the act of generating provenance information can cause operators to slow down by orders of magnitude. Finally, science applications regularly use custom operators, and the provenance system must provide an efficient means to expose an operator's provenance information.

I built SubZero [10], a provenance system that separates the mechanisms of *how* provenance is specified and stored from decisions of *what* provenance to generate. To efficiently expose operator provenance, we differentiate classes of operators by the amount of provenance they need to store (constant, linear, or polynomial with the output dataset size) and develop efficient APIs for each class. This allows operators to incur the cost of generating and writing provenance in proportion to their complexity. When SubZero executes a provenance query, it can make use of previously stored provenance information, or re-run previous operators to generate provenance. This lets the optimizer make policy decisions that trade off between provenance query performance and the amount of runtime and storage overhead necessary to achieve such query performance by deciding which operators should generate provenance and what their encoding and indexing strategies should be. In our experiments, SubZero reduced storage overhead by nearly 70× and speeds query performance by almost 255× as compared to existing provenance storage models, which is a huge leap towards making provenance systems feasible for use in high-throughput applications like LSST.

Scorpion: Using Provenance to Explain Outliers

As datasets become larger and analysis pipelines grow in complexity, even making sense of query results becomes very difficult. Consider a simple analytic query that computes a hospital's total expenses by disease and shows that lung cancer cases disproportionately account for millions of dollars. An analyst will naturally want to understand why – is it because lung cancer patients tend to need a very expensive treatment, or some other factor? A typical provenance system can automatically identify all of the lung cancer patients, however an analyst must still manually split the data along different dimensions (e.g., treatment, age) and hope one of her hunches point to the cause when re-running the query on the subset. If there is more than one outlier, or many dimensions in the dataset, this ad-hoc process quickly becomes untenable. In fact, our Harvard Medical School collaborator spent six months manually analyzing this problem. With the Scorpion system, we can identify the two doctors who over-treated their patients and are responsible for a significant amount of the costs within a few minutes of visual interaction and computation.

Scorpion is an example of an interactive system that uses and simplifies provenance information to answer these "why" questions. Scorpion provides a novel user interface that lets an analyst simply select outlier and normal results and the system then constructs predicates that most influenced the outliers. Our work formalized this notion of predicate influence in terms of sensitivity analysis, provided a framework to search for influential predicates, and identified aggregation operator properties that help reduce search times by orders of magnitude as compared to a naive exhaustive algorithm.

Smoke: Low Latency Provenance for Interactive Visualizations

Increasingly, people are publishing data as interactive visualizations, and while tools for creating static visualizations and animations are prevalent, there are limited tools for creating rich interactions – many visualization authors manually implement and optimize interactions such as brushing and linking [1].

Smoke establishes the parallel between common forms of visual interaction and data provenance by modeling visualizations as workflows from raw data to visual elements and visual interactions as provenance queries. This allows visualizations to leverage performance optimizations in provenance systems and scale interactions to larger datasets. I am currently building Smoke, which uses two key insights to to execute provenance queries at interactive (<100ms) speeds. First, when designing a publishable visualization, the author can specify the exact set of provenance queries that the system will need to run. This allows Smoke to optimize its storage representation to make these queries fast. Existing systems are designed for ad-hoc queries so the benefits of optimization opportunities cannot be evaluated. Second, many interactions do not need all of the source data specified by a provenance query, and a sample of the data, their identifiers, or a summary statistic is often sufficient. Thus we are defining classes of provenance queries with reduced expressivity that can be used for more optimizations.

Improving Data Analysis Along Other Dimensions

The above three projects are designed to move the state of data analysis to one where an end-user can effectively use the expressive power of data provenance in a visual environment. In addition to provenance-related projects, I have been broadly interested in, and worked on other projects that simplify the analysis process.

Complex Event Processing

Sensor devices generate streams of unreliable, raw sensor readings (e.g., "RFID tag 123 was scanned"). On the other hand, sensor-based applications expect high level event notifications (e.g., "shoplifting at register 3"). Along with Yanlei Diao, I built one of the first high-performance complex-event detection systems that lets users declaratively specify high level events from patterns of raw or other high level events. Our main insight was to compile sequence patterns into a nondeterministic finite automata and introduce relational optimizations such as filter push-down and early pruning.

[1] A common form of brushing and linking [1] is when a user selects (brushes) a set of points in one view, and the corresponding (linked) objects in other views are also selected

Human Computation

Together with Adam Marcus, I developed one of the earliest data management systems to introduce large-scale human computation (e.g., crowd-sourcing) as a new class of query operators. Our work explored the trade-offs between query latency, cost, and result quality, and implemented an asynchronous query engine that takes into account wildly unpredictable response times and the three trade-offs. Our results identified the tight coupling between the user interface the crowd workers are presented with and the result latency and quality. We recently proposed [3] a system that uses a synergistic relationship between data integration in dataspaces and news consumers that want more context about data in the news articles they read.

Future Research

I intend to expand the intersection between databases, usability and interaction. Data analysis is ultimately driven by a human being, and while reducing the cost of query execution is certainly important, the bottleneck is more often centered around the analyst. How can analyst tasks such as picking the questions to ask, understanding query results, and testing hypotheses be offloaded to the system so the analyst can fully utilize her domain expertise and shift her role from *implementer* to *decision maker*? Answering these questions will be central in the upcoming decade of data management.

My thesis pursued a narrow version of this problem in the form of Scorpion and Smoke. Scorpion scratched the surface of explanatory analysis by focusing on simple statistical functions, and the general framework can be extended to explain more complex aggregations as well as data constraint violations (e.g., functional dependency violations), which are crucial features when cleaning and integrating datasets. Smoke is built into a full-featured visualization system, which I will release to gather usage information about how the visualization and provenance systems are used. This information will be developed into a fine-grained provenance and interaction benchmark that is currently missing in the field.

In the longer term, I am excited about the prospect of a "visualization and analysis assistant" that observes how the user interacts with a dataset and quietly improves her experience in the background. This can include automatic anomaly detection and explanation using Scorpion-like facilities, highlighting and summarizing related data and views by tracking data provenance and finding similar or contrary trends and outliers in other subsets of the data. Such an assistant would not be useful unless it operates at interactive speeds. I would like to explore how database optimization techniques such as shared query execution and materialization can be augmented with interaction-level cues to pick the most effective optimizations. Take brushing and linking for example: the time it takes for the user to center her pointer on the edge of a selection box and resize it can be used to detect her intent and pre-compute the linked data that needs to be highlighted when the box is resized. Similar cues that take advantage of cursor movement, user perceptual inaccuracies, output resolution, and interaction history provide a rich area for optimization.

References

[1] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[2] A. Marcus, E. Wu, D. R. Karger, S. R. Madden, R. C. Miller, et al. Crowdsourced databases: Query processing with people. In *CIDR*, 2011.

[3] A. Marcus, E. Wu, and S. Madden. Data in context: Aiding news consumers while taming dataspaces. *DBCrowd*, 2013.

[4] E. Wu. Dbtruck: Humane data import, October 2012.

[5] E. Wu, C. A. Curino, S. R. Madden, et al. No bits left behind. In *CIDR*, 2011.

[6] E. Wu, Y. Diao, and S. Rizvi. High-performance complex event processing over streams. In *SIGMOD*. ACM, 2006.

[7] E. Wu and S. Madden. Smoke: Visualization interactions using provenance (in preparation).

[8] E. Wu and S. Madden. Partitioning techniques for fine-grained indexing. In *ICDE*, 2011.

[9] E. Wu and S. Madden. Scorpion: Explaining away outliers in aggregate queries. In *VLDB*, 2013.

[10] E. Wu, S. Madden, and M. Stonebraker. Subzero: a fine-grained lineage system for scientific databases. *ICDE*, 2013.