Research Statement and Agenda

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

In my research, I have implemented and evaluated high performance workflow provenance systems that efficiently track the relationships between each operator's input and output data at the tuple granularity and investigated the research challenges, such as designing the appropriate provenance APIs for user defined operators to efficiently expose their provenance information, optimization frameworks that control the provenance storage layout, and the applications of provenance systems for data analysis. For instance, Scorpion [**?**] is an analysis system I have built that lets analysts specify outliers in aggregation queries (e.g., last month's sales were too high) and generates predicates that explain away these outliers. Scorpion can reduce analysis times from hours of manual filtering to minutes, but depends on an efficient provenance system to function.

Background and Research Approach

My previous research projects fit into my over-arching interest in systems and algorithms that simplify the way end-users interact with and express complex analyses over data. This is critical because a broad audience is quickly becoming attuned to the importance of data-driven decision making. Thus, tools cannot assume the users are experts with deep understanding of data management techniques, but instead are domain experts that need simple ways to express and manage complex analyses. Thus a full solution must tackle challenges at each step of the data analysis pipeline – from simplifying the process of loading data into management tools [**?**], improving core query performance through relational optimization principles such as partitioning and indexing [**?**, **?**], extending database expressiveness to new forms of data [**?**, **?**], helping the user understand outliers in their query results [**?**], and extending database optimization techniques to visualization interaction [**?**].

My research approach emphasizes builting end-to-end systems that are workload driven. In each research project, I strive to identify unique properties of the workload that can be leveraged to both improve query performance and directy how the queries can be expressed through language extensions or visual interfaces.

Projects

Data analysis extends beyond simply manipulating and extracting patterns from our data, but also the ability to reason and explain the consclusions that we reach. While (existing techniques) can make analyses faster, data provenance is a key component of the latter.

The bulk of my thesis work bas focused on practical data provenance systems and their application to data analysis. An ideal data provenance system that automatically and cheaply tracks the relationships of data as it is transformed within a workflow and provides an efficient query interface over the provenance information has broad applications in many domains (e.g., scientific computing, data analysis, entity resolution, security). However, tracking these input-output relationships in a data-intensive environment at a fine-grained (per record) level is non-trivial. Even a single transformation can generate $O(N^2)$ pair-wise relationships for a dataset of size N. In addition, it is unclear how data analysts and systems can effectively *use* these provenance capabilities to enhance their analyses. The SubZero, Scorpion, and Smoke projects investigate the performance, applications and usability of fine-grained provenance systems.

A Low Overhead Provenance System

Scientific applications such as LSST are now demanding provenance at the record or pixel level (e.g., "what pixels generated this star?"). These is important for debugging and data validation purposes but is particularly difficult to solve because scientific workflows have tight storage and runtime overhead constraints and process large amounts of dense matrix data. For example, LSST processes 2GB/sec each night, but has a space budget of ¡20provenance. Many operators are vectorized, so even the act of generating provenance information can cause operators to slow down by orders of magnitude. In addition, science applications regularly use custom operators so the provenance system must provide an efficient means to expose the operator's provenance information.

I proposed and built *SubZero* [**?**], a provenance system that separates the mechanisms of *how* provenance is specified and stored from decisions of *what* provenance to generate. Specifically, 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. The optimization framework can trade off between provenance query performance and the amount of runtime and storage overhead by deciding which operators generate provenance and the encoding and indexing strategies on a per-operator basis.

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 company's total expenses by month, and shows that last month's expenses was unexpectedly high. The analyst will naturally want to understand why – perhaps the company has put more resources into a new customer demographic, or a department is overspending. Currently, the analyst must manually split the input data along different dimensions (e.g., dept, customer age), and hope that re-running the query will cause changes. If there is more than one outlier, or many dimensions in the dataset, this ad-hoc process quickly becomes untenable. As a real-life example, a local medical researcher spent six months manually performing a similar process to investigate why lung cancer patients cost XXX! (It turns out that two doctors that over-perscribed chemo treatment were responsible for XXcosts). Scorpion (Figure X) is an example of an interactive system that uses provenance capabilities to answer these "why" questions. The analysts simply selects outlier and normal results and the system constructs predicates that potentially explain the outliers.

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. Our main contribution is to establish the parallel between many forms of visual interaction and data provenance. By modeling a visualization as a workflow from raw data to visual elements, many user interactions can be expressed as provenance queries and potentially leverage an expressive query interface and performance optimizations. I proposed and am building a provenance system that can execute provenance queries at interactive (¡100ms) speeds based on two key insights. The first is that the visualization author knows the exact query workload apriori and the system can optimize for just those queries. This is impossible to do in existing systems designed for ad-hoc queries. The second is that the expressiveness of the queries can be restricted while supporting many interactions, which enables more optimization opportunities.

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.

Human Computation

Marcus and I developed one of the earliest data management systems that introduces large scale human computation as a new class of query operators. Our work focused on designing an asynchronous query engine that takes into account wildly unpredictable response times and the trade-offs between query latency, cost, and quality. Furthermore, we identified the tight coupling between task interface design, latency, and result quality. ple working on this now. Moving towards games with a purpose where the useful service with side effect that is useful. We recently proposed a system that creates a synergistic relationship between news consumers and data spaces.

SASE

Is this worth discussing at this point?

DBTruck

Same here

Future Research

Large scale interactive data analysis is still a burgeoning field and there are a huge number of fascinating research directions in this space.

In the short term, there are many valuable extensions to my thesis. Scorpion only scratched the surface of explanatory analysis by focusing on simple statistical functions. However, the general framework could be extended to explain more complex aggregations, functional dependency or other data constraint violations, which are crucial features when cleaning and integrating datasets. Smoke is built into a full-featured visualization system. I plan to release the system and gather information about how the system and its provenance workload, which can be used to develop a valuable fine-grained provenance benchmark that is missing in the field.

In the longer term, there are opportunities to apply database optimization and recommendation techniques by leveraging cues as the user interacts with a visualization. For example, a selection box has a limited number of directions it an be extended and moved, which can be used as prefetching hints.

Individual analyses, consumer analyses. Scaling to individual users each running powerful analytics non experts so need expressive but simple interfaces.

In general, the intersection of databases, usability and interaction is an area ripe for future research. 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 aronud 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*?