Verdict: A System for Stochastic Query Planning

Barzan Mozafari
University of Michigan, Ann Arbor
mozafari@umich.edu

Online services, wireless devices, and scientific simulations are all creating massive volumes of data at unprecedented rates. This abundance of rich datasets has made *data-driven discovery* the predominant approach across biology, medicine, physics, economics and even social sciences. Ironically, existing data processing tools have now become the bottleneck in data-driven activities. When faced with large-enough datasets (say, a few terabytes), even the fastest database systems can take hours or days to answer the simplest queries (see [1]). This response time is simply unacceptable to many users and applications. The data-driven discovery is often an *interactive* and *iterative* process: data scientists form a hypothesis, consult the data, adjust their hypothesis accordingly, and repeat this process until a satisfactory answer is discovered. Thus, slow and costly interactions with data can severely inhibit the data scientists' productivity, engagement, and even creativity.

Driven by the growing market for interactive analytics, both commercial and open source data warehouses continuously strive to provide interactive response times through various optimizations, such as parallelism, indexing, materialization, better query plans, data compression, columnar formats, and even in-memory processing. These optimizations are in essence no different than the mainstream database research on query processing, where the goals for the past four decades have simply been to:

1. efficiently access *all* the tuples that are relevant to the query while avoiding those that are not (e.g., indexing, materialization, compression, caching); and

2. choose a query plan with the least cost among a set of alternative plans that are all *equivalent* (i.e., all plans are correct).

As long as we pursue these utopian goals, all our query optimization techniques are eventually destined to fail. This is because the data growth rate has already surpassed Moore's law[1]. This means that even if we know which exact tuples are accessed by a query in advance, at some point the number of those tuples will be large enough that they will no longer fit in memory (even when compressed). In other words, since memory and CPU improvements are lagging exponentially behind the rate of data growth, our inability to provide interactive response times will only increase. Eventually, when the datasets become large enough, no optimization technique will be able to provide interactive response times unless we relax our goal of processing all relevant tuples.

Therefore, we propose a new generation of approximate query processing systems, called `Verdict`, that departs from the traditional goals of databases in order to exploit the new data deluge opportunities. Like other approximate databases [1], `Verdict` uses carefully selected samples of data (instead of the entire data) to provide fast approximate answers with statistical accuracy guarantees. However, `Verdict` differs from its predecessors by introducing a novel query planning strategy, called *stochastic query planning*, whose goal is to guarantee an interactive and instantaneous interface to data. In other words, data scientists receive immediate answers to their queries, no matter how large their dataset. Stochastic query planning is based on the following key ideas.

1. Pursuing different approximations. Unlike traditional databases that choose the best query plan from a set of equivalent alternatives, `Verdict` *deliberately* pursues multiple query plans that are *not* necessarily equivalent. In addition to executing the original query on a small sample, `Verdict` concurrently obtains several other approximations using column-wise, tuple-wise, and temporal correlations in the data. When available, `Verdict` also invokes existing regression models to estimate a specific column's marginal or conditional distribution, given a combination of other columns. These various estimates are carefully combined to calibrate the original approximation and produce a more accurate answer.

2. Learning from past queries. In traditional databases, past queries are rarely beneficial to future ones. For instance, unless a query accesses the same tuples that were accessed (and hence, cached) by a previous query, past queries do not significantly improve future performance. In contrast, `Verdict` has ample opportunities to re-use computations performed by previous queries. Even when queries use different filtering criteria, `Verdict` can still exploit the correlations and partial overlaps between the current and previous queries to continuously improve its approximate answers.

We argue that `Verdict`'s statistical query planning is not only practical, but also more suitable for many real-world applications:

(i) Since many datasets are inherently statistical or noisy, statistical query planning will serve as a sensitivity analysis step, yielding answers that are more robust against noise.

(ii) By exploiting the correlations and associations between different columns and tuples, the negative effects of missing values—which are common in real-world—can be hidden from the end user.

(iii) By concurrently pursuing different approximations, the same level of accuracy can be achieved using significantly smaller samples, leading to faster response times.

(iv) As the number of analysts querying a shared database increases, statistical query planning allows the database to continuously *learn and improve*; the more queries issued in the past, the faster and more accurate answers in the future.

[1] `http://tinyurl.com/lvup9e4`

7th Biennial Conference on Innovative Data Systems Research (CIDR âĂŹ15) January 4-7, 2015, Asilomar, California, USA..

1. REFERENCES

[1] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb: queries with bounded errors and bounded response times on very large data. In *EuroSys*, 2013.