Reminiscences on Influential Papers

Kenneth A. Ross, editor

See `http://www.acm.org/sigmod/record/author.html` for submission guidelines.

Graham Cormode, Lucent Bell Laboratories, `graham@dimacs.rutgers.edu`.

[Noga Alon, Yossi Matias and Mario Szegedy: The Space Complexity of Approximating the Frequency Moments. Journal of Computer and System Sciences 58(1), 137–147, 1999. Conference version appeared in STOC 1996.]

This paper recently won the theory community's prestigious Goedel award, yet it has also been tremendously influential in the the database community in general and at a personal level on my research. In one short paper, the authors established many founding principles of the data stream model, gave ingenious algorithms for computing the second frequency moment, a general method for finding all frequency moments, and lower bounds on the space needed. While these problems initially appear abstract, the paper has had wide influence on algorithms, database and network research.

Reading this paper as a graduate student was a revelation: having been thinking about some related questions before reading the paper, I found some of the results almost unbelievable. The crisp presentation inspired me to learn the relevant tools used to give such initially surprising proofs. For the database community, the best known result from this paper is a simple randomized "sketch" data structure which neatly and simply computes F_2 (sum of squares) of a stream of values that can be arrive in a very general update model. Subsequently, it has been shown that this summary can also be used to compute point estimates, range sums, inner products (join size estimates) and more. All this from just one page of the original conference paper! Indeed, one of the inspirational aspects of the paper is that it contains not just one excellent idea but many excellent ideas. Further, the fact that the paper appeared in a theory conference reminds us that researchers are not confined to a single area, but can work across boundaries, and read and publish wherever seems most appropriate.

Amol Deshpande, University of Maryland, `amol@cs.umd.edu`.

[Ron Avnur, Joseph M. Hellerstein. Eddies: Continuously Adaptive Query Processing. Proceedings of the 2000 ACM SIGMOD Conference, May 16-18, 2000, 261–272.]

I was initially hesitant to write about a relatively new paper written by my graduate advisor, but no other paper has quite influenced my research career as much as this original *eddies* paper. This paper was published when there was an increasing concern about the applicability of traditional query optimization techniques in domains such as static web sites, dynamic web services, and data streams, and various *adaptive query processing* techniques were being introduced. This paper presented probably the most radical overhaul of the query processing architecture, proposing use of a new operator called an *eddy* (the work was done in the context of the *River* data flow system, hence the name), through which all tuples processed by the system are routed; the eddy reacts to observed data characteristics by changing the order in which tuples are routed through the remaining operators, thus effectively changing the query plan used to execute the query.

This paper was fairly controversial when it first appeared, with many concerns being raised about its efficiency. The technique did form the basis of the Telegraph system, and its impact can also be seen in other follow-on work in adaptive query processing. I am personally biased, and won't discuss the merits of the idea itself in this note.

Although the jury may still be out on the long-term impact of this paper, there is no doubt that it significantly influenced my own research in many ways. To be honest, even though there was much work going on in our group on eddies, it wasn't until much later that I became interested in it. Because of its lack of rigorous analysis, this paper confuses the relevant, impactful concepts presented from those that were ad-hoc or had no merit. For example, it never proves under what conditions an eddy will produce correct results (this is not obvious). But once I started analyzing the technique and tried to put it in perspective, I was amazed by its simplicity and beauty, and very surprised that something so innovative could be discovered in a well-established area. Other than its direct influence on my research, this paper also changed the way I think about query processing, and taught me to think twice about accepting even well-established doctrines. In retrospect, it was the perfect paper for me, then a relatively immature graduate student looking for research ideas, to have run into.

Panagiotis Ipeirotis, New York University, panos@stern.nyu.edu.

[David R. Cox: Regression models and life-tables. Journal of the Royal Statistical Society, Series B 1972; 34:187-220.]

I read this paper while working on the problem of updating word- frequency histograms that characterize a web database. Since databases change over time, the associated statistics should also be updated to reflect content changes. The fundamental challenge in this problem is to learn how long it takes for a database to "change." Knowing this time, it is then possible to schedule updates appropriately. Most of the solutions for related problems in the database field either assumed that this time is given, or that we can learn it by observing a database for long periods of time: a costly solution when dealing with web databases. Then I realized that actuaries face a similar problem all the time: predicting the lifetime of insured clients. I started checking the literature, and this paper by Cox was by far the most frequently cited reference. I started reading the paper, realizing that statisticians have been developing solutions for the problems that I faced, for more than three decades. While it took me some time to read and digest the proposed methods, at the end I was impressed. The proposed regression model was conceptually simple, theoretically sound, and made a very limited number of assumptions about the data. The model could effectively use incomplete ("censored") data and could deal with non-linear effects. Furthermore, the applications seemed (and are) endless. However, even after reading the paper, I was afraid that such a "traditional" method from statistics could only be applied to relatively small number of data points, and that it would not scale for the large data sets. I could not be more wrong. The regression ran quickly, returning simple and elegant formulas that I could subsequently use for scheduling updates.

This paper, and subsequent work in the survival analysis field made me realize that "no man is an island." Research from other fields can be easily applied to traditional database problems. I am still working on similar problems, and the work by Cox and other statisticians is a very fertile source of methods for my research. I strongly believe that anyone who works on similar problems (view maintenance, histogram refreshing, updating cached data, publish/subscription techniques, and so on) should read this paper and be familiar with the general field of survival analysis. The 8,000 citations returned by Google Scholar for this paper just prove its importance.

Donald Kossmann, ETH Zurich, kossmann@inf.ethz.ch.

[Michael J. Carey, David J. DeWitt, Jeffrey F. Naughton: The OO7 Benchmark. SIGMOD Conference 1993: 12–21.]

I have a personal and more general perspective on this paper. From my personal perspective, this paper saved my PhD thesis and so it had a huge impact on my career. In my thesis, I was working on "pointer swizzling" and "dual buffering". I strongly believed in my ideas, but I had no way to show that they were

worth-while using the existing benchmarks. It turned out that my ideas worked extremely well on the OO7 Benchmark. I was lucky.

In general, benchmarks shape a field and strongly impact the research directions pursued. This was definitely true for OO7 because in the OODB research community everybody tried to win the OO7 battle. Today, we know that OODBs were a (commercial) failure and research on OODBs has stopped. Now, what does that mean for OO7? There are two standpoints: (a) OO7 was harmful because it made us work on the wrong problems and, as a result, we could never get the glorious ideas of OODBs to work; (b) OO7 was very helpful because it made us realize early what the potential killer-applications of OODB technology would be and that OODBs were doomed to fail. No matter what standpoint you take, OO7 was clearly extremely influential: to the better or to the worse — I cannot judge.

Lucian Popa, IBM Almaden Research Center, `lucian2@us.ibm.com`.

[Catriel Beeri and Moshe Y. Vardi: A Proof Procedure for Data Dependencies. JACM 31(4), October 1984, 718–741.]

This is one of the classical papers of relational database theory. I first read the paper as a graduate student at the University of Pennsylvania. It was my advisor, Val Tannen, who realized that our work on equational rewriting of queries under constraints had something to do with the chase. He asked me to read several papers on the subject, including the Beeri and Vardi paper. I took several weeks to read the paper. The notation is a bit outdated and it may take a while to get used to it, but underneath, the paper is a gem. It is probably the single paper that influenced my research the most, as I learned two fundamental concepts that afterwards I used in many occasions and I still use: 1) the class of tgds and egds to express data dependencies, and 2) the chase with tgds and egds.

The paper introduces tgds (standing for tuple-generating dependencies) and egds (standing for equality-generating dependencies) as a general formalism to include most prior forms of database dependencies: inclusion dependencies, foreign key constraints, multivalued and join dependencies, functional dependencies, key constraints, and more. There were two other independently developed formalisms that turned out to be equivalent to tgds and egds (the embedded implicational dependencies of Fagin and the algebraic dependencies of Yannakakis and Papadimitriou). However, Beeri and Vardi are the first to define the chase at this level of generality. The paper goes into great depth to prove the important properties of the chase, such as completeness. In addition to the main results, many of the technical lemmas, the proof techniques as well as the various subclasses of tgds and egds that appear in the paper are still relevant and instructive today.

Although the formalism of tgds and egds did not quite make it as a practical framework for database dependencies (the more common foreign key and key constraints are simpler to handle and maintain), today such dependencies are striking back and not necessarily as database constraints. Tgds and egds are at their best use when describing relationships between schemas or between components of schemas. They can describe relationships between physical database structures and logical schemas. Such description can then be used together with the chase and other techniques to perform query reformulation. Tgds and egds can also be used to describe data integration style of schema mappings, in which case the chase can be used to give an elegant semantics for the process of transforming data from one schema into another. This is in fact the formal basis for the schema mapping language developed at IBM for the Clio data exchange system. So, the language of tgds and egds and their chase are still very relevant today, after more than two decades.
