Reminiscences on Influential Papers

Kenneth A. Ross, editor

In a recent New York Times article, Roxana Robinson writes that it is great to be a writer, because "If you invent the story, you're the first to see how it ends." I feel similarly about this column (even though I'm merely the editor). The contributions all tell a story, and I feel privileged to be the first to read them as a collection.

I continue to invite unsolicited contributions. See `http://www.acm.org/sigmod/record/author.html` for submission guidelines.

Charu C. Aggarwal, IBM T. J. Watson Research Center, `charu@watson.ibm.com`.

[K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft. When is nearest neighbor meaningful? International Conference on Database Theory, 1999: 217–235.]

This is truly an interesting and compelling piece of work; it raises several issues for high dimensional data in a simple, and succint way so as to provide considerable intuitive insight into the nature of high dimensional sparsity. The paper sets certain pre-conditions on data distributions and distance functions under which the nearest neighbor problems may not be very meaningful for high dimensional problems. Most researchers have usually been more focussed on issues such as *performance* rather than meaningfulness for high dimensional indexing and clustering problems. For example, almost every well known high-dimensional index structure such as the X-Tree, VA-File, SS-Tree, or TV-Tree has been built with a focus on performance; it is assumed by these papers that certain distance functions such as the L_p-norm are relevant for arbitrary data distributions in high dimensions. In my estimation, *almost all* of the multi-dimensional indexing/clustering papers for high dimensional quantitative data in top database conferences over the past 15 years have similar issues in assuming too much about methods for proximity measurements which are relevant for low dimensional applications but start having questionable meaningfulness with increasing dimensionality.

This paper has influenced my work profoundly in two ways. One is to re-define problems such as clustering for high dimensional data in order to make them more meaningful. For example, clustering is re-defined in the context of *projected clustering*; a set of points form a cluster if they are close in some low dimensional projection which is specific to that set of points (or locality) [Aggarwal et. al. Sigmod'99; Sigmod'00]. Similarly, nearest neighbor search problems are redefined as *projected nearest neighbor search methods* [Hinneburg et. al. VLDB'00], where query-specific projections are used in order to identify the most similar objects. These methods have the added advantage of providing greater understanding; similar clusters or query results are understood in the context of locality specific discriminatory projections — these provide additional information about the data.

A second line of attack is to redefine distance functions for high dimensional data in order to make them qualitatively effective; this is easier said than done — for certain data domains such as Information Retrieval, the problem of designing distance functions has intrigued researchers over three decades. Unlike IR applications, we cannot use specific information about the "typical" nature of the data for arbitrary applications; designing distance functions in these cases is an even more challenging task. The key is to be able to design *dimensionality sensitive*, *meaningful*, and *index-friendly* distance functions. Our first approach in this direction is the IGrid index [Aggarwal and Yu, SIGKDD'00], which is meaningful *both* from a qualitative and performance perspective.

Alfons Kemper, University of Passau, Germany kemper@db.fmi.uni-passau.de.

[T. Härder. Implementing a generalized access path structure for a relational database system. ACM Transactions on Database Systems, 3(3):285–298, September 1978.]

Out of the set of important papers that have influenced my own work I would like to single out this rather early paper on generalized indexing by Theo Härder. This work was done in the context of the System R development and adapted concepts from the then popular CODASYL network data model to devise a (transparent) access structure for relational database systems. Rather than restricting an access structure (index) to a single relation, so-called *link* access structures were devised to support the retrieval of sets of related tuples belonging to the same hierarchical structure. It turns out, that the paper is the "grandfather" of quite a few more recent works. The most direct successor appears to be Patrick Valduriez' paper on join indexes which are a generalization of Theo Härder's binary links. By way of the join index paper, Härder's paper also influenced the design of access structures for other data models. Among them are the path-oriented access structures for object-oriented and object-relational database systems (e.g., the path indexes or the access support relations Guido Moerkotte and I designed). In effect, the influence of Härder's paper could be traced all the way to the most recent work on indexing XML data (e.g., the data guides). The paper even contains some ideas underlying recent algorithmic work in query evaluation: The algorithms for join evaluation via join indexes, the pointer join and the functional join evaluation techniques can partially be traced back to the generalized access path structures — which were published 22 years ago!

Sunita Sarawagi, Indian Institute of Technology, Bombay, sunita@it.iitb.ernet.in.

[B. S. Everitt. The Analysis of Contingency Tables. Chapman and Hall, 1992.]

Mapping densities of discrete multidimensional spaces is a core problem in several database topics including approximation, selectivity estimation, association rule mining and deviation detection. This book (or any of the several related books on this topic) is a must read for anyone doing research on any of these topics. The most important concept I learnt from the book is how to estimate the frequency of a k-dimensional space given frequencies of a partial subset of the 2^k possible marginals of this space using a iterative procedure.

For association rules this gives a robust notion of when an itemset is surprising. A k-itemset should be called surprising only when its support is significantly different from what is expected given marginal supports of all proper subsets. The expected values are calculated using the iterative procedure referred to above. We used this same notion in finding interesting temporal patterns in our VLDB-98 paper. Brin et. al. in "Beyond market basket analysis" (SIGMOD-97) propose a special case of this method. They make the simplifying assumption that k-way itemsets are interesting only when none of their subsets are interesting. Such an assumption was necessary to get a closed form formula for finding expected values. The iterative process makes this assumption unnecessary. A slight twist of the above idea can be used to do good multidimensional selectivity estimations as proposed by Mannila et. al. in "Prediction with Local Patterns using Cross-Entropy" (KDD-99).

Ever since early 1997, when I first came across this body of work, all my OLAP and mining related work has been strongly influenced by it. OLAP datasets because of their multidimensional structure and categorical dimensions can be thought of as contingency tables. This analogy forms the basis of our work on finding exceptions in OLAP data as reported in EDBT-98. This method provided us a way to *combine* user's prior expectation obtained from multiple different aggregates — a problem that I was struggling with quite a bit before chancing upon this body of work.

S. Sudarshan, Indian Institute of Technology, Bombay, sudarsha@cse.iitb.ernet.in.

[Goetz Graefe and William McKenna. Extensibility and Search Efficiency in the Volcano Optimizer Generator. ICDE, 1993: 209–218.]

The work that has influenced my research the most in recent times is undoubtedly the Volcano Optimizer Generator. What I like about this work is that it showed how to elegantly apply dynamic programming to handle query optimization for a very general class of algebraic expressions and transformation rules. The hashing technique used for detecting duplicate expressions is also rather neat. It was a revelation to me that you can eat your cake (of extendibility) and have it too (good performance due to dynamic programming coupled with neat optimizations to detect duplicate expressions). This lead to my using it as the framework for research on multiquery optimization, materialized view maintenance and caching that I (along with students and colleagues at IIT Bombay) have been doing in the last few years.

This work has attracted less attention in the literature than it deserves, especially considering that at least two commercial optimizers (Microsoft SQL Server and Tandem) are based on the Volcano optimizer. Perhaps this is because the paper is hard to understand — too much technical detail has been compressed into a page or two. To be honest, discussions with Bill McKenna played a significant role in helping me understand how the Volcano optimizer actually worked. But make the effort to read the hard parts of the paper carefully (and perhaps bug Bill McKenna to understand the hard parts, as I did!) and the beauty of the scheme emerges! And don't forget to check out subsequent work on improving efficiency for join rewriting, by Pellenkoft et. al. in VLDB 97.

Mihalis Yannakakis, Lucent Bell Laboratories, mihalis@research.bell-labs.com.

[K. P. Eswaran, J. N. Gray, R. A. Lorie, and I. L. Traiger. The Notions of Consistency and Predicate Locks in a Database System. Communications of the ACM, 19(11): 624–633, 1976.]

This paper laid the foundations of concurrency control, introducing a simple model and defining the basic concepts of transaction, schedule, consistency (serializability), and the two-phase locking policy. The paper contains a wealth of other fundamental ideas. I still have my early copy of this paper, which I read at the time word for word, and filled it with comments on the margins. Besides its significance specifically to databases, I think the paper is more generally valuable (at least, certainly to me) as an illustration of how to go about modeling a problem or area and building a theory for it. For example, exercising taste and good judgment in deciding what to include in the model and what to abstract from is critical in getting to the heart of the problem. It is often tempting, when first modeling a problem, to include all the aspects and peculiarities of the situation, yielding complex models with all kinds of parameters, that are hard to understand and analyze, and often obscure the central issues. In contrast, this paper has a strikingly simple model of a database: it is just a set of entities. One can imagine having more elaborate models with more detailed information about the nature and the organization of the database; however, for the purposes of isolating the problem and identifying the basic concepts, the simple model is quite sufficient and brings the issues to the foreground. Of course, once the foundation has been laid, one can go on to extend the model, bring in more components of the problem, and study the issues in more depth. And this is what happened in this area. Many other papers followed, building on the foundation of this paper, and concurrency control blossomed into a rich area of research.