# Report on the Second International Workshop on Exploratory Search in Databases and the Web (ExploreDB 2015)

Georgia Koutrika
HP Labs, Palo Alto
koutrika@hp.com

Laks V.S. Lakshmanan
Department of Computer Science, University of British Columbia
laks@cs.ubc.ca

Mirek Riedewald
College of Computer and Information Science, Northeastern University, Boston
mirek@ccs.neu.edu

Mohamed A. Sharaf
School of ITEE, University of Queensland, Australia
m.sharaf@uq.edu.au

Kostas Stefanidis
ICS-FORTH, Heraklion
kstef@ics.forth.gr

## 1. INTRODUCTION

To make Big Data that is growing in both size and diversity widely accessible, data management and analysis systems have to provide appropriate *exploration services*. An analysis might include structured (relations, tables), semi-structured (XML), and "unstructured" (text) data, linked together through relationships encoded as a graph. Some of the data can be precise, others might be probabilistic [15], e.g., due to measurement error or because it was generated by a statistical model. At the same time, the community of potential users is becoming more diverse as well, ranging from database experts and domain scientists to citizen scientists. These users need system services that help them understand the data and enable them to find relevant information, even if they do not completely comprehend the content and relationships in a complex data collection. This broad goal can be addressed in a variety of ways.

Research in the database community has long been exploring how to simplify the process of composing non-trivial queries, starting with query-by-example [17] in the 1970s. Today many structured data collections can be accessed through Web form interfaces and even keyword search [2, 7], where joins are inferred automatically. Query steering [3, 6, 8] extends the idea of example-based query composition by asking the user to label potential result tuples as (ir-)relevant, a topic covered by one of the keynotes. Then query conditions are automatically derived from the labeled examples. Example-based query composition and modification can be further extended by adding more sophisticated search capabilities that automatically include connected entities and information sources.

Exploration also plays a crucial role when dealing with queries that return too many result tuples, or where expected results are missing—the main topic of the second keynote. For example, why-not [4] and how-to [10] queries are reverse data management approaches that explain or automatically modify a given query if it does not produce the desired outputs. Instead of having the user debug and rewrite a query in a tedious trial-and-error process, the system automatically modifies the query based on examples of missing (or undesirable) query result tuples [16]. Query relaxation techniques have a similar goal for over-constrained queries [9, 12]. An alternative to query relaxation based on examples of missing results is to offer query languages that support imprecise conditions. One option are similarity predicates [11, 13], e.g., searching for cars "like" a given model with a price "near" some value. Another is to allow probabilistic conditions [14], e.g., to express that the user is 80% sure that the entity she is looking for had property X.

For a query returning too many results, *ranking* helps the user explore the most important ones [1, 5]. Its success hinges on the selection or design of an appropriate ranking function. In general, it should capture some natural notion of result relevance, measured based on concepts such as novelty, diversity, and surprise. Ranking functions can be personalized based on historic queries or by requesting user input revealing her preferences. Typically personalization should be achieved with minimal effort required from the user, as discussed below.

In summary, the field of data exploration is diverse in terms of research directions and potential user base. Hence the ExploreDB workshop intends to bring together researchers and practitioners from different fields, ranging from data management and information retrieval to data visualization and human computer interaction. Its goal is to study the emerging needs and objectives for data exploration, as well as the challenges and problems that need to be tackled, and to nourish interdisciplinary synergies. We summarize the outcomes of the second workshop instance held in conjunction with ACM SIGMOD 2015 in Melbourne, Australia.[1]

## 2. WORKSHOP OUTLINE

The workshop program consisted of two keynote talks and six peer-reviewed research papers.

### 2.1 Invited Talks

The first keynote talk titled *"Explore-By-Example: A New Database Service for Interactive Data Exploration"* was given by Prof. Yanlei Diao from the University of Massachusetts at Amherst. Prof. Diao pointed out that while computing power, memory size, and the ability to collect data are growing exponentially, human ability to *understand* data remains practically flat. This "big data, same humans" problem motivates the need for new database services that support automated data exploration. To work effectively with a traditional database management system (DBMS), the user needs to understand the database content well, including structure and meaning of relations, and be able to formally express the exact query to obtain the desired result. For applications and users where this does not apply, a new DBMS service for *interactive data exploration* should have the following features: First, users make sense of the data space via *navigation*, automated by the DBMS. Second, the DBMS interprets user interactions and *learns user interests*, so that it can retrieve all relevant results. Third, both online learning and query processing have *interactive performance.*

Explore-by-example supports this functionality by presenting example tuples to the user in order to obtain feedback about their relevance. Classification models trained based on this feedback drive the process of selecting new samples for additional feedback, as well as the generation of the final SQL query that retrieves a result that includes the relevant samples, but not the irrelevant ones. This approach dramatically changes interaction with the DBMS. The traditional query-cycle consists of query formulation and processing, followed by result review that informs query modification. It is somewhat ad-hoc as the "correct" query predicates are unknown initially, labor-intensive as the user has to review possibly large query results, and resource-intensive as the DBMS executes sequences of queries on big data. With explore-by-example, the traditional query-cycle is replaced with a new cycle that starts with labeling of samples as (ir-)relevant, followed by training of a classification model that informs the choice of another set of samples.

Key research challenges revolve around capturing user interest with high accuracy, minimizing user effort for labeling samples, and keeping user wait time acceptable. A decision-tree based algorithm for identifying hyper-rectangular relevant areas in multi-dimensional space performed well in experiments, requiring a few hundred samples to home in on the target regions. User wait time ranged from 1 to 6 seconds, which Prof. Diao considers acceptable. Interestingly, larger database size did not result in larger required sample size, indicating that the approach scales well to big data. A preliminary user study involving seven CS majors familiar with SQL indicated significant reduction in user effort and exploration time.

While successful for linear predicates (i.e., hyper-rectangular regions), dealing with more general predicates significantly increases complexity. Prof. Diao discussed remaining research challenges related to convergence with a minimum number of labeled samples, DBMS optimizations to minimize user wait time, automatic learning of user profiles, more general queries including join and aggregation, and visualization.

In the second keynote, titled *"Principled Optimization Frameworks for Query Reformulation of Database Queries"*, Prof. Gautam Das from the University of Texas at Arlington focused on solutions for the many-answers and the empty-answers problems. He proposed to address both problems through *ranked retrieval.* In particular, when a query is too selective (empty-answer problem), the user can be steered to "partially matching" tuples. And when a query is not selective enough (many-answers problem), she might be steered to the "top-ranked" tuples. In both scenarios, an appropriate

[1] For a summary of the first instance of ExploreDB, please refer to "Georgia Koutrika, Laks V. S. Lakshmanan, Mirek Riedewald, Kostas Stefanidis: Report on the First International Workshop on Exploratory Search in Databases and the Web (ExploreDB 2014). SIGMOD Record 43(2): 49-52 (2014)."

ranking approach is needed.

For the many-answers problem, Prof. Das discussed dynamic faceted search, which suggests additional constraints by presenting values for attributes from the database schema. By picking a value, the user refines the query. Suggestions are ranked based on the objective of minimizing user effort, which is measured in terms of the number of additional query conditions considered by the user before reaching the entity of interest. This ranking problem can be solved by finding an appropriate fully-grown decision tree with minimum expected height.

Similarly, query relaxation suggestions for the empty-answer problem can be ranked based on the objective of minimizing user effort. Intuitively, the system should suggest relaxations that are likely to be accepted by the user and that will steer her toward minimum effort. Prof. Das presented a probabilistic framework for achieving this goal, which relies on estimates for the probability that the user believes a tuple exists in the database and for the likelihood that the user will prefer a tuple in the answer of a relaxed version of the query. An optimal precise and a faster approximate algorithm find the top-ranked relaxations.

## 2.2 Paper Presentations

The six talks of the technical program covered a variety of issues related to exploratory data analysis, ranging from personalization for query result presentation to complex event processing.

In *"Data Like This: Ranked Search of Genomic Data"*, V.M. Megler, David Maier, Daniel Bottomly, Libbey White, Shannon McWeeney and Beth Wilmot presented their vision "to make searching for data as easy for scientists as searching the Internet." To this end, they proposed ideas for adapting ranked search to big genome data, which contains position-indexed annotations that are a mix of numeric, ordinal, and binary data types. A major challenge is to find and compare different regions based on their similarity of annotations. Indexing and summarization techniques were proposed to achieve acceptable interactive performance.

Query personalization through preferences was explored in *"Unifying Qualitative and Quantitative Database Preferences to Enhance Query Personalization"* by Roxana Gheorghiu, Alexandros Labrinidis and Panos Chrysanthis. A graph-based framework enables the user to specify both qualitative (i.e., which tuple is preferred over the other in a given pair) and quantitative (i.e., a numerical score for each tuple) preferences. These preferences *together* are leveraged for ranking of database tuples, based on a newly introduced notion of preference "intensity."

Xiaoyu Ge, Panos Chrysanthis and Alexandros Labrinidis (*"Preferential Diversity"*) explored how to achieve personalization through preferences on result *diversity.* Since diversity's goal of reducing redundancy can be in conflict with ranking based on relevance, the proposed approach lets the user control the tradeoff between the two. An iterative algorithm then efficiently processes the data, repeatedly selecting the most relevant records and eliminating others similar to them.

Diversity was also the focus in *"Diversifying with Few Regrets, But too Few to Mention"* by Zaeem Hussain, Hina Khan and Mohamed Sharaf. To balance the tradeoff between maximizing diversity and minimizing regret, which measures loss in utility, a hybrid objective function is proposed. The approach distinguishes between *preference* dimensions, for which regret is minimized, and *neutral* dimensions, for which diversity is maximized. The hybrid objective function is a linear weighted combination of the diversity and regret objectives. A greedy heuristic and an algorithm based on local search find solutions efficiently.

Chen Zhang, Rui Meng, Lei Chen and Feida Zhu (*"CrowdLink: An Error-Tolerant Model for Linking Complex Records"*) proposed a new probabilistic model to better leverage crowdsourcing for record linkage, i.e., the task of finding records that refer to the same entity across different data sources. Questions are selected with the goal of minimizing monetary cost. The algorithm is designed for robustness to errors in the workers' answers.

Tatsuki Matsuda, Yuki Uchida and Satoru Fujita (*"Method of Complex Event Processing over XML Streams"*) argued that complex event processing (CEP) can play a major role in exploratory analysis. As events are detected, they can interrupt an exploration process and affect its direction in realtime. To support a wide variety of applications, they focus on streams of XML data. High performance is achieved by optimizing visibly pushdown automata (VPA) used to execute queries.

# 3. WORKSHOP CONCLUSIONS

Several themes emerged in the discussions.

- Dealing with large query results is a promising direction for exploratory search. Many meaningful and natural ranking approaches have been proposed, but they are often in conflict with each other. For example, many highly relevant results might be very similar to each other, resulting in low diversity if they all are presented at the top. More research is needed to be able to combine these ideas into frameworks where the user can customize the ranking function based on desired properties.

- Personalization plays a crucial role for exploration of Big Data. Research challenges revolve around the central issue of user effort, in particular how to learn a personalized ranking function with minimal user input or easy-to-obtain input. For example, it might be easy to label individual records as (ir-)relevant, but it would be practically impossible to expect a user to specify an explicit ranking function.

- The curse of dimensionality is further underscored in Big Data exploration. In particular, guiding users through an uncharted high-dimensionality data space increases the complexity of the data exploration process and challenges its effectiveness. The impact of dimensionality is equally emphasized when ranking a query result, or refining and steering imprecise queries. Hence, it is essential to integrate emerging data exploration techniques with effective methods for handling high-dimensional data.

- System performance, in particular response time experienced by the user, remains a major challenge for exploratory search. Traditional database approaches for indexing, materialization, and data reduction need to be extended and customized for exploratory search on Big Data.

This second instance of ExploreDB made clear that a lot of research work still needs to be done in the general area of exploration for Big Data. Given the growing interest in industry and academia, we are looking forward to the next instance of this workshop.

## 4. REFERENCES

[1] S. Agrawal and S. Chaudhuri. Automated ranking of database query results. In *Proc. CIDR*, pages 888–899, 2003.

[2] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: A system for keyword-based search over relational databases. In *Proc. ICDE*, pages 5–16, 2002.

[3] U. Cetintemel, M. Cherniack, J. DeBrabant, Y. Diao, K. Dimitriadou, A. Kalinin, O. Papaemmanouil, and S. B. Zdonik. Query steering for interactive data exploration. In *Proc. CIDR*, 2013.

[4] A. Chapman and H. V. Jagadish. Why not? In *Proc. ACM SIGMOD*, pages 523–534, 2009.

[5] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. Probabilistic information retrieval approach for ranking of database query results. *ACM Transactions on Database Systems (TODS*, 31(3):1134–1168, 2006.

[6] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. Explore-by-example: An automatic query steering framework for interactive data exploration. In *Proc. ACM SIGMOD*, pages 517–528, 2014.

[7] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *Proc. VLDB*, pages 670–681, 2002.

[8] M. S. Islam, C. Liu, and R. Zhou. A framework for query refinement with user feedback. *J. Syst. Softw.*, 86(6):1580–1595, 2013.

[9] U. Junker. QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems. In *Proc. AAAI*, pages 167–172, 2004.

[10] A. Meliou, W. Gatterbauer, and D. Suciu. Reverse data management. *PVLDB*, 4(12):1490–1493, 2011.

[11] A. Motro. VAGUE: A user interface to relational databases that permits vague queries. *ACM Trans. Inf. Syst.*, 6(3):187–214, 1988.

[12] D. Mottin, A. Marascu, S. B. Roy, G. Das, T. Palpanas, and Y. Velegrakis. A probabilistic optimization framework for the empty-answer problem. *Proc. VLDB Endow.*, 6(14):1762–1773, Sept. 2013.

[13] U. Nambiar and S. Kambhampati. Answering imprecise queries over autonomous web databases. In *Proc. ICDE*, pages 45–54, 2006.

[14] B. Qarabaqi and M. Riedewald. User-driven refinement of imprecise queries. In *Proc. ICDE*, pages 916–927, 2014.

[15] D. Suciu, D. Olteanu, C. Re, and C. Koch. *Probabilistic Databases.* Morgan & Claypool, 2011.

[16] Q. T. Tran and C.-Y. Chan. How to ConQueR why-not questions. In *Proc. ACM SIGMOD*, pages 15–26, 2010.

[17] M. M. Zloof. Query-by-example: The invocation and definition of tables and forms. In *Proc. VLDB*, pages 1–24, 1975.