New Trends on Exploratory Methods for Data Analytics

Davide Mottin
Hasso Plattner Institute
davide.mottin@hpi.de

Matteo Lissandrini
University of Trento
ml@disi.unitn.eu

Yannis Velegrakis
University of Trento
velgias@disi.unitn.eu

Themis Palpanas
Paris Descartes University
themis@mi.parisdescartes.fr

ABSTRACT

Data usually comes in a plethora of formats and dimensions, rendering the exploration and information extraction processes cumbersome. Thus, being able to cast exploratory queries in the data with the intent of having an immediate glimpse on some of the data properties is becoming crucial. An exploratory query should be simple enough to avoid complicate declarative languages (such as SQL) and mechanisms, and at the same time retain the flexibility and expressiveness of such languages. Recently, we have witnessed a rediscovery of the so called example-based methods, in which the user, or the analyst circumvent query languages by using examples as input. An example is a representative of the intended results, or in other words, an item from the result set. Example-based methods exploit inherent characteristics of the data to infer the results that the user has in mind, but may not able to (easily) express. They can be useful both in cases where a user is looking for information in an unfamiliar dataset, or simply when she is exploring the data without knowing what to find in there. In this tutorial, we present an excursus over the main methods for exploratory analysis, with a particular focus on example-based methods. We show how different data types require different techniques, and present algorithms that are specifically designed for relational, textual, and graph data.

1. SCOPE OF THE TUTORIAL

Exploratory methods refer to approaches that allow users to understand data without knowing the user's information needs. Traditional exploratory methods include data exploration, data visualization, interactive interfaces, and predictive models. However, the existing body of work assumes the user is willing to pose several queries to the underlying database in order to progressively gather the required information. This assumption stems from the intuition that the user, being accustomed to data analysis, can more intuitively dig into the data. However, this assumption does not always hold, since it requires that users have a minimum level of expertise, which is only true for a very limited number of (potential) users.

Recently, the research community has resorted to the use of *examples* as a proxy for exploratory analysis. One of the earliest attempts to bring examples as a query method is query-by-example [40]. The main idea was to help the user in the query formulation, allowing her to specify the shape of the results in terms of templates for tuples, i.e., examples. Query-by-example has been lately revisited, and the use of examples have found application in several areas across various data types. The definition of example has transformed from a mere template to the representative of the intended results the user would like to have. These example-based approaches are fundamentally different from the initial query-by-example idea, and have been successfully applied to relational [8, 26, 31], textual [5, 38, 39], and graph [10, 13, 19] data.

We note that the flexibility of examples does not compromise the richness of the results, yet, it overcomes the ambiguity of simple keyword searches, which is traditionally studied in information retrieval. On the other hand, while data exploration techniques (Idreos et al.: Overview of data exploration techniques, tutorial in SIGMOD 2015) assume the user is willing to pose several exploratory queries, the use of examples requires almost no supervision from the user perspective, making example-based methods a more palatable choice for novice users, as well as for practitioners. This new functionality can empower existing data exploration methods with a complementary tool: whenever a query is too complex to be expressed with a query language, such as SQL, examples represent a natural alternative. Moreover, the use of examples has been demonstrated to be very effective in data visualization [18, 28].

In this tutorial, we aim at describing the main developments of examples as an expressive and powerful method for exploratory data analysis.

The first part of the tutorial gradually introduces the broad topic of data exploration, highlighting the hardness of query languages for simple users and advocating the need of different query methods. We will introduce the example-based methods as flexible delegates for more complex queries that would otherwise need to be expressed through a very complex traditional query. In this part, we will discuss various cases, where queries cannot be expressed in declarative languages without requiring complex constructs. We will also show novel applications where examples can be easily leveraged.

Proceedings of the VLDB Endowment, Vol. 10, No. 12

The second part of the tutorial discusses the current main techniques for relational, textual, and graph data. In this part, we will present the algorithms, show how they work, and demonstrate their ability to (conceptually) infer very complex queries from simple examples. We will also highlight the differences among data types, focusing on the scalability perspective, presenting the motivations and drawing parallels among methods for different data types.

The third part of the tutorial focuses on the latest developments of machine learning to progressively discover user intention. We will introduce some early methods based on relevance feedback [12], and show some recent applications that include active learning and active search.

Challenges and open research questions. The last part of the tutorial is dedicated to the challenges and open research questions. Exploratory search based on examples is rapidly attracting attention and getting traction, though, the support for such techniques in modern data management systems is lagging behind. Some challenges have already been discussed in recent vision papers [35,37]. We will discuss the following major challenges.

• *Adaptivity*: current data management technologies and systems do not take into account individual user preferences, and tend to optimize certain kind of queries and respond slowly to others.

• *Explanation*: Data management systems usually include little or no explanation for the results of a query. In example-based methods, in which the user query is only implicit, this requirement is even more prominent.

• *Interactivity*: Current prototypes show the advantage of example based methods with regards to visualization techniques. However, in order to achieve the real-time, interactive performance needed by visualization tools, the algorithms should incorporate intelligent and efficient techniques for navigating through the search space.

Finally, we will conclude the tutorial with remarks about the current state of affairs, and engage the audience in a discussion about their experiences with needs, tools, and challenges in this area.

2. TUTORIAL OUTLINE

In this tutorial, we will provide an overview of exploratory methods, especially in the new area of example-based methods, surveying the relevant state-of-the-art techniques. Moreover, we will present future directions discussing various machine learning techniques used to infer user preferences in an online fashion.

Next, we report the summary of the outline. We also provide an extended description of example-based approaches in Section 2.1, and machine learning approaches in Section 2.2.

I. **Introduction and motivation**

- Usefulness of exploratory analysis
- Main characteristics of exploratory analysis
- Example-based methods for exploratory analysis
- Use cases of failing keyword and declarative queries
- Applications in current database systems and data analysis

II. **Example-based approaches**

- Query-by-example: [40]
- Example methods in relational databases:
  - Using tuples as examples [8, 22]
  - Examples for Query suggestion [25]
  - Tuple explanations [6]
  - Reverse engineering of SQL queries [4, 15, 31, 36]
- Example methods in textual data:
  - Web documents as examples [39]
  - Incomplete web tables [38]
  - Entity extraction [11, 27]
  - Related searches based on current visited page [5]
- Example methods in graphs:
  - Similar entity search [17, 28]
  - Structure-based approaches: Exemplar Queries [19], Graph Query By Example [13]
  - Learning Path queries [2]
  - Reverse engineering of SPARQL queries [1, 7]
  - Node-based approaches: to discover communities [14], dense regions in the graph [10,24], and outliers [21].

III. **Learning methods based on examples**

- Relevance feedback learning [8, 12, 32]
- Relevance feedback for graphs [16, 29]

IV. **Challenges and Remarks**

- Can we *interactively* assist the user toward the retrieval of the correct answer?
- Can we provide *explanations* for the query results?
- How can machine learning help in exploratory analysis?
- Can we easily integrate these techniques into existing data management systems?

2.1 Example-based approaches

We survey the main approaches for exploratory queries, highlighting the main differences among data models, and presenting in-depth insights of the current status of research in this area. We first introduce query-by-example [40] as a first attempt to simplify query formulation. In query-by-example the user, instead of explicitly typing a query, specifies the shape of the results in a tabular fashion. We present the main body of work within relational, textual, and graph data, even though examples have been successfully employed also in learning syntactic program transformations [23], time series [9], and image search [30].

For *relational data* the tutorial introduces techniques that exploit the database schema, inferring the results from input tuples [8,22]. We show how the use of examples can substantially help query suggestion [25] and query profiling [6]. Recent advances suggest that SQL queries can be constructed using simple example output tuples [4,15,31,33,34,36]. However, there are limitations in theoretical terms that hinder the efficiency of the final solution. More specifically, exact algorithms when positive and negative tuples are provided mostly rely on satisfiability problems and are therefore hard to solve [36]. The purpose is to find a query that returns all the positive tuples and none of the negative tuples. On the other hand, heuristic or relaxed solutions based on skyline points [15], decision trees [31], or cost models [33] have been proposed. Examples have been successfully employed in interactive schema mapping starting from input and output examples of two different databases [4].

For *textual data* the techniques include search approaches based on documents used as representatives for the set of results [38], and serendipitous search based on the current

visited pages [5]. These approaches focus on documents as examples for retrieving related information. Recently, examples has been successfully employed in entity extraction [11, 27], in which the user provides either mentions of entities in a text [11], or tuples and similarities among attributes [27], and the system automatically returns extraction rules that can be applied to the given dataset.

For *graph data* there are two prominent approaches: the first use subgraphs, or partially specified structures as input examples [1,7,13,19], while the second focuses on the vertices of the graph, which are used for making the selections [2,14, 21]. Structures convey a more precise information and therefore can be used to quickly prune the search space. Among the existing approaches exemplar queries [18,19] and Graph Query by Example (GQBE) [13] use subgraph isomorphism or structural similarities to identify structures related to the one the user provided. A different approach is the reverse engineering of SPARQL queries [1, 7] in which the input is a set of positive and negative entity mentions in a RDF dataset. This approach is similar to those discussed for the relational case, and is related to learning path or join queries given positive and negative nodes [2, 3].

Instead of returning results of interest, examples can also be employed for targeted analysis of networks, in order to discover communities [14], dense regions [10, 24], or subspaces along outliers [21]. Such approaches ask the user to mark nodes that belong to a community and perform an analysis using the information in the nodes and in their connections do discover regions of interest in the graph. These regions can then be used for targeted analyses or advertising campaigns.

2.2 Machine learning with examples

Current techniques use ad-hoc notions of similarity to retrieve results that are likely to be part of the solution of an unknown query. The current development in machine learning and active search [16, 20, 29] present a different perspective: user preferences can be learned from user interactions instead of manually crafted in the system. Current hardware capabilities allow to process large amount of data, and at the same time dynamically change the internal preference model. One of the earliest work in this direction is MindReader [12] in which the user specifies a set of tuples and optional relevance scores and the system infers a distance function on the objects in the database. The exploration of such *relevance learning* or *metric learning* approaches form the basis of interactive exploratory systems. Moreover, the study of Gaussian Processes as a mean of interactively learning any function given a set of points from the user has recently found applications in graphs [16, 20]. Therefore, we will present a body of work that takes the machine learning perspective into account. The research in this area is still at its infancy and forms a fertile ground for a new generation of data management systems.

3. TARGET AUDIENCE

This tutorial is intended for researchers and practitioners interested in big data analytics, graph analytics, and data exploration methods. No prior knowledge is required in order to understand the concepts in the tutorial, but we assume a familiarity with database and graph concepts and basic machine learning terminology.

The tutorial aims at fostering collaborations between several disciplines, including data management, data mining, and machine learning. Researchers and students will find interesting ideas and challenges to start research in exploratory analysis, with a focus on example-based methods. Moreover, they will get an overview of the existing approaches for various data types. Addressed to practitioners, this tutorial will present a new generation of exploratory analysis techniques based on examples, which can be easily applied, and improve on a variety of existing data exploration tools for structured and non-structured data.

4. PRESENTERS

The proponents of the tutorial have several years of expertise in data management and organization of tutorials, workshops, university courses, projects, and conferences. Davide Mottin presented tutorials on graph exploration at CIKM 2016, and at SIGMOD 2017; Themis Palpanas presented tutorials on blocking techniques for entity resolution at ICDE 2016 and WISE 2014, and on event processing architectures at DEBS 2010; Yannis Velegrakis presented a tutorial on goal mining at ICDE 2015, and on data management at ICDE 2012.

Davide Mottin is a postdoctoral researcher at Hasso Plattner Institute. His research interests include graph mining, novel query paradigms, and interactive methods. He presented graph exploration tutorial in CIKM 2016, and in SIGMOD 2017. He also presented exploratory techniques in KDD 2015, VLDB 2014, and SIGMOD 2015 and is actively engaged in teaching database, big data analytics, and graph mining for Bachelor and Master courses. He is the proponent of exemplar queries paradigm for exploratory analysis [19]. He received his PhD in 2015 from the University of Trento.

Matteo Lissandrini is a PhD student and a member of the dbTrento (Data and Information Management) group at the University of Trento, Italy. He received his BSc degree in Computer Science from the University of Verona, Italy, and his MSc in Computer Science from the university of Trento, Italy. He has also spent time as a visitor at HP Labs, Palo Alto, California, and at the Cheriton School of Computer Science at the University of Waterloo, Canada. His scientific interests include novel query languages for large scale data mining and information extraction with a focus on exploratory search on graph data, publishing the first Exemplar Query methods for Knowledge Graphs in VLDB and VLDBJ, and presenting the application of such methods in SIGMOD 2014. He is also active as teacher assistant for the Database Management System, and for the Information Systems courses at the University of Trento.

Yannis Velegrakis is a faculty member at the University of Trento, with expertise in schema mapping, interoperability, heterogeneous information integration, data exchange, view management, and keyword searching. He graduated from the University of Toronto, with a thesis on mapping management. Prior to joining the University of Trento, he held a researcher position at ATT Research Labs (USA). He has also spent time as a visitor at the IBM Almaden Research Center (USA), where he participated in the development of the Clio schema mapping tool, the Center of Advanced Studies of the IBM Toronto Lab (Canada), and the University of California, Santa-Cruz (USA), where he and his collaborators developed the STBenchmark, a generic and

multi-purpose benchmark for schema mapping systems. He has served in program committees of many national and international conferences, has been a reviewer for numerous international journals and was a Marie Curie Reintegration fellow between 2006 and 2008. He has been a general chair for the DESWeb 2010 and 2011 ICDE Workshops and was General Chair for VLDB 2013.

Themis Palpanas is Senior Member of the Institut Universitaire de France (IUF) and professor of computer science at Paris Descartes University, France. Before that he was a professor at the University of Trento, Italy, and he has worked as a researcher at the IBM T.J. Watson Research Center and the University of California at Riverside, as well as Microsoft Research and IBM Almaden Research Center. He is the author of nine US patents, three of which are part of commercial products. He has received three best paper awards, the IBM Shared University Research (SUR) Award, and was General Chair for VLDB 2013. Professor Palpanas has been working on the field of exploratory data analytics for both structured and non-structured data for the last several years, publishing relevant methods to major journals (TKDE, VLDBJ) and conferences (VLDB, SIGMOD).

References

[1] M. Arenas, G. I. Diaz, and E. V. Kostylev. Reverse engineering sparql queries. In *WWW*, pages 239–249, 2016.

[2] A. Bonifati, R. Ciucanu, and A. Lemay. Learning path queries on graph databases. In *EDBT*, 2015.

[3] A. Bonifati, R. Ciucanu, and S. Staworko. Learning join queries from user examples. *TODS*, 40(4):24, 2016.

[4] A. Bonifati, U. Comignani, E. Coquery, and R. Thion. Interactive mapping specification with exemplar tuples. In *SIGMOD*, pages 667–682, 2017.

[5] I. Bordino, G. De Francisci Morales, I. Weber, and F. Bonchi. From machu_picchu to rafting the urubamba river: anticipating information needs via the entity-query graph. In *WSDM*, pages 275–284, 2013.

[6] D. Deutch and A. Gilad. Qplain: Query by explanation. In *ICDE*, pages 1358–1361, 2016.

[7] G. Diaz, M. Arenas, and M. Benedikt. Sparqlbye: Querying rdf data by example. *Proceedings of the VLDB Endowment*, 9(13):1533–1536, 2016.

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

[9] B. Eravci and H. Ferhatosmanoglu. Diversity based relevance feedback for time series search. *PVLDB*, 7(2):109–120, 2013.

[10] A. Gionis, M. Mathioudakis, and A. Ukkonen. Bump hunting in the dark: Local discrepancy maximization on graphs. In *ICDE*, pages 1155–1166, 2015.

[11] M. F. Hanafi, A. Abouzied, L. Chiticariu, and Y. Li. Synthesizing extraction rules from user examples with seer. In *SIGMOD*, pages 1687–1690, 2017.

[12] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *VLDB*, 1998.

[13] N. Jayaram, A. Khan, C. Li, X. Yan, and R. Elmasri. Querying knowledge graphs by example entity tuples. *TKDE*, 27(10):2797–2811, 2015.

[14] I. M. Kloumann and J. M. Kleinberg. Community membership identification from small seed sets. In *KDD*, 2014.

[15] H. Li, C.-Y. Chan, and D. Maier. Query from examples: An iterative, data-driven approach to query construction. *PVLDB*, 8(13):2158–2169, 2015.

[16] Y. Ma, T.-K. Huang, and J. G. Schneider. Active search and bandits on graphs using sigma-optimality. In *UAI*, pages 542–551, 2015.

[17] S. Metzger, R. Schenkel, and M. Sydow. Qbees: query by entity examples. In *CIKM*, pages 1829–1832, 2013.

[18] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. Searching with xq: the exemplar query search engine. In *SIGMOD*, pages 901–904. ACM, 2014.

[19] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. Exemplar queries: a new way of searching. *VLDB J.*, pages 1–25, 2016.

[20] F. Murai, D. Rennó, B. Ribeiro, G. L. Pappa, D. Towsley, and K. Gile. Selective harvesting over networks. *arXiv preprint arXiv:1703.05082*, 2017.

[21] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, and E. Müller. Focused clustering and outlier detection in large attributed graphs. In *KDD*, pages 1346–1355, 2014.

[22] F. Psallidas, B. Ding, K. Chakrabarti, and S. Chaudhuri. S4: Top-k spreadsheet-style search for query discovery. In *SIGMOD*, pages 2001–2016, 2015.

[23] R. Rolim, G. Soares, L. D'Antoni, O. Polozov, S. Gulwani, R. Gheyi, R. Suzuki, and B. Hartmann. Learning syntactic program transformations from examples. In *ICSE*, pages 404–415. IEEE Press, 2017.

[24] N. Ruchansky, F. Bonchi, D. García-Soriano, F. Gullo, and N. Kourtellis. The minimum wiener connector problem. In *SIGMOD*, pages 1587–1602, 2015.

[25] T. Sellam and M. Kersten. Cluster-driven navigation of the query space. *TKDE*, 28(5):1118–1131, 2016.

[26] Y. Shen, K. Chakrabarti, S. Chaudhuri, B. Ding, and L. Novik. Discovering queries based on example tuples. In *SIGMOD*, pages 493–504, 2014.

[27] R. Singh. Blinkfill: Semi-supervised programming by example for syntactic string transformations. *PVLDB*, 9(10):816–827, 2016.

[28] G. Sobczak, M. Chochół, R. Schenkel, and M. Sydow. iqbees: Towards interactive semantic entity search based on maximal aspects. In *Foundations of Intelligent Systems*, pages 259–264. Springer, 2015.

[29] Y. Su, S. Yang, H. Sun, M. Srivatsa, S. Kase, M. Vanni, and X. Yan. Exploiting relevance feedback in knowledge graph search. In *KDD*, pages 1135–1144, 2015.

[30] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MM*, pages 107–118, 2001.

[31] Q. T. Tran, C.-Y. Chan, and S. Parthasarathy. Query reverse engineering. *VLDB J.*, 23(5):721–746, 2014.

[32] H. P. Vanchinathan, A. Marfurt, C.-A. Robelin, D. Kossmann, and A. Krause. Discovering valuable items from massive data. In *KDD*, pages 1195–1204, 2015.

[33] C. Wang, A. Cheung, and R. Bodik. Interactive query synthesis from input-output examples. In *SIGMOD*, pages 1631–1634, 2017.

[34] C. Wang, A. Cheung, and R. Bodik. Synthesizing highly expressive sql queries from input-output examples. In *PLDI*, 2017.

[35] A. Wasay, M. Athanassoulis, and S. Idreos. Queriosity: Automated data exploration. In *Proceedings of the IEEE International Congress on Big Data*, 2015.

[36] Y. Y. Weiss and S. Cohen. Reverse engineering spj-queries from examples. In *SIGMOD*, pages 151–166, 2017.

[37] E. Wu, L. Battle, and S. R. Madden. The case for data visualization management systems: Vision paper. *Proc. VLDB Endow.*, 7(10):903–906, June 2014.

[38] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. Infogather: Entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*, 2012.

[39] M. Zhu and Y.-F. B. Wu. Search by multiple examples. In *WSDM*, pages 667–672, 2014.

[40] M. M. Zloof. Query by example. In *AFIPS NCC*, pages 431–438, 1975.