

# Graph Exploration: From Users to Large Graphs

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

Emmanuel Müller  
Hasso Plattner Institute  
emmanuel.mueller@hpi.de

## ABSTRACT

The increasing interest in social networks, knowledge graphs, protein-interaction, and many other types of networks has raised the question how users can explore such large and complex graph structures easily. Current tools focus on graph management, graph mining, or graph visualization but lack user-driven methods for graph exploration. In many cases graph methods try to scale to the size and complexity of a real network. However, methods miss user requirements such as exploratory graph query processing, intuitive graph explanation, and interactivity in graph exploration. While there is consensus in database and data mining communities on the definition of data exploration practices for relational and semi-structured data, graph exploration practices are still indeterminate.

In this tutorial, we will discuss a set of techniques, which have been developed in the last few years for independent purposes, within a unified graph exploration taxonomy. The tutorial will provide a generalized definition of graph exploration in which the user interacts directly with the system either providing feedback or a partial query. We will discuss common, diverse, and missing properties of graph exploration techniques based on this definition, our taxonomy, and multiple applications for graph exploration. Concluding this discussion we will highlight interesting and relevant challenges for data scientists in graph exploration.

## 1. SCOPE OF THE TUTORIAL

The continuously increasing interest in graphs and the growing amount of graph data available on the web require a careful design of data analysis techniques. However, from the user perspective most of the existing techniques appear as a black box that returns results without any explanation. For these reasons our community has resorted to data exploration techniques. In particular, while a huge effort has been devoted to text, relational, and semi-structured data [8], data exploration on graphs (*graph exploration* in short) is still in its infancy. Although many techniques

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGMOD'17, May 14-19, 2017, Chicago, IL, USA

© 2017 ACM. ISBN 978-1-4503-4197-4/17/05...\$15.00

DOI: <http://dx.doi.org/10.1145/3035918.3054778>

for graphs have been studied in different domains, there is still lack of a unified graph exploration taxonomy. We abstracted user-driven graph exploration properties from techniques proposed in the literature and defined such a unified taxonomy. Our taxonomy consists of three strategies that form the backbone of our presentation along with relevant literature identified so far: exploratory graph analysis, refinement of graph query results, and focused graph mining.

**Exploratory Graph Analysis** entails the process of casting an incomplete or imperfect pattern query to let the system find the closest match. Such exploratory analysis may return a huge number of results, e.g., structures matching the pattern. Thus, the system is required to provide intelligent support. One such strategy is the well known query-by-example paradigm, in which the user provides the template for the tuples and let the system infer the others.

**Refinement of Graph Query Results** is needed to deal with the overwhelming amount of results that is typical in subgraph processing. It includes approaches designed to present comprehensive result sets to the user or intermediate results that can be refined further. Instantiations of this kind are graph summaries, top-k methods, query reformulation, and skyline queries.

**Focused Graph Mining** guides the users to a specific portion of the graph they are interested in. It requires the user to provide feedback in the process to restrict the computation to some portion of the graph. Ego-networks mining belongs to this strategy, since the user search is limited to a particular area of the graph and the algorithms focus on that specific area.

We conclude the tutorial with a number of open research questions, highlighting the huge potential of graph exploration with many challenges still unsolved.

## 2. TUTORIAL OUTLINE

The tutorial provides a gently introduction to the concept of graph exploration, considering the data exploration perspective and combining with the recent advances in graph analytics. No previous knowledge is required although basic database and graph mining concepts are beneficial for the full understanding of the topic. The tutorial is organized as follows.

**I. Introduction and motivation:** The first part of the tutorial introduces the benefits of data exploration for extracting knowledge from data without requiring any specific expertise or having in mind a clear task [8]. We also show how graphs are important for modeling complex information

in more natural way. However, current techniques for graph exploration are limited to visualization tools that do not scale to real graphs or complex (and non standard) graph query languages. Instead, we claim the necessity of algorithms for graph exploration that take the user in the loop and at the same time scale to real world graphs.

**II. Graph Exploration Taxonomy:** We examine graph exploration methodologies, identifying in the existing works the three strategies introduced earlier and discussed in more detail in Section 3: exploratory graph analysis, focused graph mining, and refinement of query results.

**III. User-driven Graph Exploration:** We discuss algorithmic solutions proposed in the last years that fit our graph exploration taxonomy. In this light, we first present *exploratory graph queries* in terms of approximate search [3,11,13,30,31], by-example methods [9,17], and path learning [1]. Second, we introduce methods for *refining the results* of incorrectly specified queries, in case they return little or no results, or a large number of irrelevant results. Three main solutions have been proposed in this regard: query reformulation and refinement [16,21,27,29], ranking and top-k [4,5,10,25], and skyline queries [34,36]. Finally, we present techniques developed in the context of graph mining for *selectively restrict* the graph to a relevant portion; they include focused graph clustering and outlier detection [5,12,19,23,35], and space restriction methods [2,22,24,26,32].

**IV. Open challenges:** The last part of the tutorial discusses open problems and visions about graph exploration. We propose four missing tiles in graph exploration that should become the focus of future research: interactivity, scalability, adaptivity, and personalization.

### 3. GRAPH EXPLORATION TAXONOMY

The main part of the tutorial introduces techniques for graph exploration. We identify and categorize existing solutions into a graph exploration taxonomy that gathers the most important research developments in this area. Our intent is to showcase a number of algorithms and to demonstrate how independent research can properly fit into the graph exploration taxonomy. A summary of the discussed algorithms is presented in Table 1.

#### 3.1 Exploratory Graph Analysis

Exploratory graph analysis offers some degree of freedom in query formulation, allowing the user to explore the graph without bothering about the strict semantic of query languages. In this category falls two main query strategies, namely *approximate graph search* and *searching by example*.

**Approximate graph search** includes methods that relax the way in which a query answer is found in the graph, allowing potential mistakes or imprecisions. A strict query on a graph is about finding occurrences of a particular structure (or pattern) in a graph through subgraph isomorphism. As opposed to the rigidity of subgraph isomorphism, approximate graph search finds structures that are similar to the query structure to some extent. One kind of approximate graph search is strong simulation [13] that relaxes the condition for which a match is a bijective function over the nodes in the query and in the graph. Similarly, p-homomorphism [3] includes the notion of matching-path to

map edges in the query to paths in the graph. As a further extension, NeMa [11] and SLQ [30] propose the use of similarity among nodes, encompassing cases in which the query does not exactly match any subgraph.

**Searching by example** approaches the problem from a different angle: Instead of considering the query as a specification of the answers, the by-example method assumes that the input is a representative of the intended results. The main by-example methods in graphs are exemplar queries [17] and graph query by-example [9]. In exemplar queries [17] the user provides a structure as example and the method finds other structures with the same characteristics. Graph Query by-example (GQBE) [9] on the other hand assumes the input is a tuple instead of a subgraph.

#### 3.2 Refinement of Graph Query Results

In an exploratory task, the query might be vague and return a large number of results (many-answer problem) or over-specific and return little or no results (empty-answer problem). In both cases the results are not useful for the user who in turn is forced to manually modify the query to find different results. For this reason three different approaches have been proposed: *query reformulation and refinement*, *top-k results*, and *skyline queries*.

**Query reformulation** (a.k.a. query reformulation, query modification) modifies the user query and returns an alternative set of queries that are more (or less) specific. One of the earliest methods for query reformulation in graphs proposes a set theoretical notion [16] to find meaningful and expressive reformulations. A preliminary study of empty-answer queries in graphs has been also recently proposed [27]. That method is based on the idea of differential graphs, i.e., the largest subgraph of the query that produces answers. Similar to query reformulation is result summarization [21,29] which aims at finding a compact representation of the results returned by a graph query.

**Top-k results** returns the  $k$  best results according to a ranking function that should ideally express the user preferences. Top-k approaches either try to return results that cover different aspects or topics [4,5,10] or learn a preference function from user feedback [25].

**Skyline queries** decomposes the query conditions in dimensions and return results that maximize each condition individually. Skyline approaches require the definition of a dominance relation which, in case of graphs, can be based on the distance of the other nodes from the query nodes [34,36].

#### 3.3 Focused Graph Mining

The last graph exploration strategy entails a smart restriction of the graph to the subgraph that contain only relevant information for the user. These methods have been studied in the context of graph mining and include *focused graph clustering and outlier detection* and *space restriction methods*.

**Focused graph clustering and outlier detection** includes analytic methods that are driven by the user who provides an initial seed set of nodes as searching criteria. The initial nodes can then be used to detect communities of users that share some characteristics with the seed nodes [12]. Other techniques take a more conservative approach, combining community detection to subspace clustering [19,23] to discover communities with different characteristics. An or-

<i>Exploratory graph analysis</i>	<i>Refinement of graph results</i>	<i>Focused graph mining</i>
<ul style="list-style-type: none"> <li>• Approximate graph search [3, 11, 13, 30, 31]</li> <li>• Searching by example [1, 9, 17]</li> </ul>	<ul style="list-style-type: none"> <li>• Query reformulation [16, 21, 27, 29]</li> <li>• Top-k results [4, 5, 10, 25]</li> <li>• Skyline queries [34, 36]</li> </ul>	<ul style="list-style-type: none"> <li>• Focused graph clustering and outlier detection [3, 11, 13, 30, 31]</li> <li>• Space restriction methods [2, 22, 24, 26, 32]</li> </ul>

Table 1: Approaches in graph exploration, categorized using the graph exploration taxonomy.

thogonal approach to focused clustering is outlier detection based on input template queries [5, 35] or seed nodes [19].

**Space restriction methods** focus on the analysis of specific portions of the graph. Common approaches include ego-network analysis methods [2] and local community detection [22, 24]. An Ego-network is the induced subgraph of nodes adjacent to a target node and are useful tools for many applications, such as node similarity and community evolution. Similarly, local community detection [22, 24] focuses on small communities around a set of input nodes. Center-piece subgraphs [26] expand the idea of finding a node (i.e., center-piece node) that is in some path connecting a set of query nodes. With center-piece subgraphs one can for instance understand the connections among a set of interesting users in a social networks. We also present query-driven graph compression in which the graph is compacted in way that the results relevant to the query are highlighted [32].

## 4. OPEN RESEARCH CHALLENGES

The last part of the tutorial discusses open research questions and challenges. While the tutorial covers the recent advances in graph exploration, identifying existing research in this area, it also highlights differences with data exploration methods. Data exploration identifies adaptivity, especially in indexing techniques, as a key component for fast data access [7, 20], however no graph counterpart currently exists. Existing graph indexes [6, 33] mostly assume the data and the query workload to be static, dismissing important optimizations that can be performed on-the-fly in an *adaptive* manner.

Another desiderata for current systems is a support to *interactive* and *personalized* exploration. With interactivity we require that the system promptly reacts to user feedback and offers minimum effort strategies to reach the correct answers quickly. This direction has been explored in relational databases for query relaxation [18], and itinerary planning [15]. Recently, the use of machine learning and more specifically active search has been rediscovered for graph exploration [14, 28]. The use of machine learning for structure discovery has not been proposed, yet it represents a promising ground for research.

Interactivity and adaptivity should also be coupled with *scalable* solutions, allowing for fast access to large data. Scalability is only partially addressed by present solutions and is a bottleneck to real size graphs, such as social, biological, information networks.

We discuss the aforementioned open challenges and highlight opportunities for innovation and applicability in modern database systems and visualization methods.

## 5. BIOGRAPHIES

**Davide Mottin** is a postdoctoral researcher in the Knowledge Discovery and Data Mining group at Hasso Plattner Institute (Germany). His research interests include graph mining, novel query paradigms, and interactive methods. He received his Master degree and PhD in 2015 from the University of Trento. In 2011 he interned Microsoft Research, Beijing, and in 2013 Yahoo! Labs, Barcelona. Davide is a recipient of PhD on the Move Programme and two travel awards. He is actively engaged in teaching graph mining, database, and big data analytics for Bachelor and Master courses.

**Emmanuel Müller** is professor and head of the Knowledge Discovery and Data Mining group at Hasso Plattner Institute. His research interests include graph mining, stream mining, clustering and outlier mining on graphs, streams, and traditional databases. He holds a best paper award at SSDBM 2016, and several awards for teaching activities. He presented tutorials in database, data mining, and machine learning conferences such as SDM, ICDM, and ICML. He received his PhD in 2010 from RWTH Aachen University, had been independent group leader at Karlsruhe Institute of Technology (2010 - 2015) and postdoctoral fellow at University of Antwerp (2012 - 2015).

## 6. REFERENCES

- [1] A. Bonifati, R. Ciucanu, and A. Lemay. Learning path queries on graph databases. In *EDBT*, 2014.
- [2] A. Epasto, S. Lattanzi, V. Mirrokni, I. O. Sebe, A. Taei, and S. Verma. Ego-net community mining applied to friend suggestion. *PVLDB*, 9(4):324–335, 2015.
- [3] W. Fan, J. Li, S. Ma, H. Wang, and Y. Wu. Graph homomorphism revisited for graph matching. *PVLDB*, 3(1-2):1161–1172, 2010.
- [4] W. Fan, X. Wang, and Y. Wu. Diversified top-k graph pattern matching. *PVLDB*, 6(13):1510–1521, 2013.
- [5] M. Gupta, J. Gao, X. Yan, H. Cam, and J. Han. Top-k interesting subgraph discovery in information networks. In *ICDE*, pages 820–831, 2014.
- [6] W.-S. Han, J. Lee, and J.-H. Lee. Turbo iso: towards ultrafast and robust subgraph isomorphism search in large graph databases. In *SIGMOD*, pages 337–348, 2013.
- [7] S. Idreos, S. Manegold, and G. Graefe. Adaptive indexing in modern database kernels. In *EDBT*, pages 566–569, 2012.
- [8] S. Idreos, O. Papaemmanoil, and S. Chaudhuri. Overview of data exploration techniques. In *SIGMOD*, pages 277–281, 2015.

- [9] N. Jayaram, A. Khan, C. Li, X. Yan, and R. Elmasri. Querying knowledge graphs by example entity tuples. *TKDE*, 27(10):2797–2811, Oct 2015.
- [10] J. Jin, S. Khemmarat, L. Gao, and J. Luo. Querying web-scale information networks through bounding matching scores. In *WWW*, pages 527–537, 2015.
- [11] A. Khan, Y. Wu, C. C. Aggarwal, and X. Yan. Nema: Fast graph search with label similarity. In *PVLDB*, volume 6, pages 181–192, 2013.
- [12] I. M. Kloumann and J. M. Kleinberg. Community membership identification from small seed sets. In *KDD*, pages 1366–1375, 2014.
- [13] S. Ma, Y. Cao, W. Fan, J. Huai, and T. Wo. Strong simulation: Capturing topology in graph pattern matching. *TODS*, 39(1):4, 2014.
- [14] Y. Ma, T.-K. Huang, and J. G. Schneider. Active search and bandits on graphs using sigma-optimality. In *UAI*, pages 542–551, 2015.
- [15] C. Mishra and N. Koudas. Interactive query refinement. In *EDBT*, pages 862–873, 2009.
- [16] D. Mottin, F. Bonchi, and F. Gullo. Graph query reformulation with diversity. In *KDD*, pages 825–834, 2015.
- [17] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. Exemplar queries: Give me an example of what you need. *PVLDB*, 7(5):365–376, 2014.
- [18] D. Mottin, A. Marascu, S. B. Roy, G. Das, T. Palpanas, and Y. Velegrakis. A probabilistic optimization framework for the empty-answer problem. *PVLDB*, 6(14):1762–1773, 2013.
- [19] 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.
- [20] W. Qin and S. Idreos. Adaptive data skipping in main-memory systems. In *SIGMOD*, pages 2255–2256, 2016.
- [21] S. Ranu, M. Hoang, and A. Singh. Answering top-k representative queries on graph databases. In *SIGMOD*, pages 1163–1174, 2014.
- [22] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In *WWW*, pages 1089–1098, 2013.
- [23] P. I. Sanchez, E. Müller, U. L. Korn, K. Böhm, A. Kappes, T. Hartmann, and D. Wagner. Efficient algorithms for a robust modularity-driven clustering of attributed graphs. In *SDM*, pages 100–108, 2015.
- [24] C. L. Staudt, Y. Marrakchi, and H. Meyerhenke. Detecting communities around seed nodes in complex networks. In *Big Data*, pages 62–69, 2014.
- [25] 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.
- [26] H. Tong and C. Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *SIGKDD*, pages 404–413, 2006.
- [27] E. Vasilyeva, M. Thiele, C. Bornhövd, and W. Lehner. Answering “why empty?” and “why so many?” queries in graph databases. *JCSS*, 82(1):3–22, 2016.
- [28] X. Wang, R. Garnett, and J. Schneider. Active search on graphs. In *KDD*, pages 731–738, 2013.
- [29] Y. Wu, S. Yang, M. Srivatsa, A. Iyengar, and X. Yan. Summarizing answer graphs induced by keyword queries. *PVLDB*, 6(14):1774–1785, 2013.
- [30] S. Yang, Y. Xie, Y. Wu, T. Wu, H. Sun, J. Wu, and X. Yan. Slq: a user-friendly graph querying system. In *SIGMOD*, pages 893–896, 2014.
- [31] Y. Yuan, G. Wang, L. Chen, and H. Wang. Efficient subgraph similarity search on large probabilistic graph databases. *PVLDB*, 5(9):800–811, 2012.
- [32] N. Zhang, Y. Tian, and J. M. Patel. Discovery-driven graph summarization. In *ICDE*, pages 880–891, 2010.
- [33] P. Zhao, J. X. Yu, and P. S. Yu. Graph indexing: tree+ delta<= graph. In *VLDB*, pages 938–949, 2007.
- [34] W. Zheng, L. Zou, X. Lian, L. Hong, and D. Zhao. Efficient subgraph skyline search over large graphs. In *CIKM*, pages 1529–1538, 2014.
- [35] H. Zhuang, J. Zhang, G. Brova, J. Tang, H. Cam, X. Yan, and J. Han. Mining query-based subnetwork outliers in heterogeneous information networks. In *ICDM*, pages 1127–1132, 2014.
- [36] L. Zou, L. Chen, M. T. Özsü, and D. Zhao. Dynamic skyline queries in large graphs. In *DASFAA*, pages 62–78, 2010.