Web Communities in Big Data Era. Editorial

ABSTRACT

Web-based community is a self-defined web-based network of interactive communication organized around a shared interest or purpose. It provides the means of interactions among people in which they create, share, and exchange information and ideas in virtual space and networks. Working with big data often requires querying and reasoning that data to isolate information of interest and manipulate it in various ways. This editorial paper explores recent big data research topics – stream querying and reasoning – over data from web based communities. It combines aspects from some well-studied research domains, such as, social network analysis, graph databases, and data streams. We provide a brief synopsis of some research issues in supporting reasoning and querying tasks. This editorial also presents the WI&C'16 workshop's goal and programme.

Keywords

Web communities, datastream, querying, and reasoning

1. INTRODUCTION

Web communities form out of groups of people (typically called users) from all different backgrounds and histories, and they use online technologies to communicate with each other and share information. From a social standpoint, communities are the most interesting to study because they consist of people who probably have never met yet are held together by a common interest or goal. People join online communities for all sorts of reasons – perhaps they share a preference for similar ideas or a similar lifestyle. The web communities can be characterized by a complex overlapping and *nested* structure.

Big data refers to a collection of data sets so large and complex that it becomes difficult to process using traditional database management tools or data processing applications. More precisely, big data is often characterized with four V's: Volume for the scale of the data, Velocity for its streaming or dynamic nature, Variety for its heterogeneity, and Veracity for the uncertainty of the data. As this big data gets bigger, it becomes a challenge to gain insights through traditional database queries and reasoning.

In this paper, data refers to be the data arising in the context of web communities that includes both the embedded structural information as well as the data generated by users. Web based communities generate such large amount of data that includes both structural changes to the network and updates that are associated with the nodes or the edges of the network. The task of querying and reasoning refers to consumption and management of generated data, and most of the time such task is often performed in real-time as the data arrives.

We briefly discuss some of the important trends and research challenges in supporting querying and reasoning on communities' data in next section.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
DOI: http://dx.doi.org/10.1145/2872518.2890583

2. SOME TRENDS & CHALLENGES

From a technical perspective, we need to explore the right types of infrastructures such as a real-time system with the desired expressiveness and scalability. Although we want to carry out our reasoning/analysis in web communities, we still need to store instances of the social underling graph and be able to access it with good response rates. So, we need graph databases in order to store and make computations on top of huge graphs. Graph Databases provide the means to store massive graphs in a distributed manner, and perform queries as well as complex computations on them. Similarly to key value stores, they still work with a store and process approach and do not provide much support for temporal analyses.

Graphs are extensively used for representing data, with the result that a number of query languages for graphs have been proposed over the past few decades. Well-established query languages such as relational or XML query language are not suitable for graph structured data since they fall short to provide support for specifying graph traversals. By the way, there have been proposals for graph query languages, but so far except SPARQL no other language has secured broad recognition .However the use of SPARQL language has been mostly restricted to RDF datasets. There are some languages have been proposed recently that build upon SPARQL, such as, Streaming SPARQL [7], EP-SPARQL [8], Continuous SPARQL (C-SPARQL) [9].

Rapidly growing social networks, online communities and other graph datasets require a scalable processing infrastructure. MapReduce, despite its popularity for big data computation, is problematic at supporting iterative graph algorithms. The MapReduce framework provides austere abstractions as well as mature infrastructure to utilize very large computer clusters for computation and data management. However, the limited persuasiveness and state modelling limit the suitability of Map-Reduce for workloads with complex expressions and complex state.

NoSQL scalable data stores provide a good trade off in data and query persuasiveness, consistency, architecture and scalability. For instance, distributed key value stores utilize a restricted set of operations on a minimal data model to achieve scalability, high performance and ease of use.

Irrespective of how the reasoning tasks are quantified, we must devise efficient execution strategies that can handle the excessive update rates expected in online communities.

Usually re-executing a query or a reasoning job when a new update arrives is likely to be infeasible with the exception of very low rate data streams. Instead the purpose of incremental computation is to preserve sufficient transitional state in memory so that the new answer can be computed in an incremental manner with least work. Such incremental techniques are often restricted to the task at hand. A central research challenge here is to find incremental methods that are applicable to a wide variety of jobs.

Another key challenge is designing appropriate distributed programming frameworks to support specifying general-purpose stream querying and reasoning tasks. Recently this has been addressed in Kineograph [1], GraphInc [2], Trinity [10], and Grace [11]. However there is a still lot to be done to scalable support a variety of complex stream querying and reasoning jobs.

In stream query processing systems, we may have several uninterrupted queries running concurrently. For instance, a personalized query for trend detection where the aim is to detect trending local topics in each user's network can be seen as a pool of a significant number of independent queries. Sharing of computation across these queries is crucial in order to limit the computational cost. Such sharing has been proved to be an effective way to handle high rate data streams [3, 4]. However, such techniques have not been well studied in the backdrop of online communities.

One more way to manage the high update rates is use random sampling to reduce the size of the data that needs to be processed. One could sample at two different levels in a web community: sample from the network structure itself to reduce the size of the graph that needs to be processed, or sample from the updates to the content. Ahmed et al. [5] present a comprehensive treatment of network sampling, both in static and streaming situations.

Another interesting stream query on social data is a publish/subscribe query. Publish/Subscribe is a popular paradigm for users to express interests ("subscriptions") in events ("publications"). It allows efficient asynchronous interaction. For instance, a query that asks to fetch all updates from all friends. Responding these queries with very low latencies is tricky if the data is distributed across different machines forcing expensive distributed traversals. One possibility is to replicate the data appropriately so that, for each user, the required data is located on some machine [6]. But cutting-edge techniques for partitioning and replica maintenance must be developed for wide-ranging stream reasoning and querying jobs.

Stream querying and reasoning over web communities' data is still an evolving research area that brings together web community analysis, social networks, social data streams, graph databases, and real-time processing. The convergence of these fields offers novel and stimulating ideas for research.

3. PAPERS IN THE WI&C'16 WORKSHOP

The papers, selected for this workshop, deal with some of the significant issues in the field. The keynote will be presented by Babak Esfandiarion (Carlton University) on *Distributed Wikis and Social Networks: a Good Fit*.

In the workshop, four papers are selected for presentation. These papers are:

-*Enriching how-to guides by linking actionable phrases* is presented by Nikolaos Lagos (Xerox Research Centre Europe), Alexandr Chernov (University of Tübingen), Matthias Gallé (Xerox Research Centre Europe) and Agnes Sandor (Xerox Research Centre Europe). In this paper authors present a method for enriching community-specific procedural knowledge entries that can be found on the Web. They achieve actionable phrase extraction with an F-score of more than 67%, and they provide a higher linking performance than state-of-the-art methods.

-*Detection of Multiple Identity Manipulation in Collaborative Projects* writen by Zaher Yamak (INSA de Rouen), Laurent Vercouter (INSA de Rouen) and Julien Saunier (INSA de Rouen). Authors are interested in detecting fake accounts that try to bypass the Online Social Networks regulations. The presented methodology detects 99% of fake accounts (on a base of 10.000) on English Wikipedia.

-*Ontological Networks: Mapping Ontological Knowledge Bases into Graphs proposed by* Lucas Navarro (Federal University of Sao Carlos), Estevam Hruschka Junior (Federal University of Sao Carlos) and Ana Paula (IBM Research Brazil). In this paper, authors describe a graph structure called Ontological Network which can be used generically to map Ontological Knowledge Bases (OKB). It is shown that Ontological Network are convenient for implementing graph-mining based algorithms to find new facts and to extend the OKB.

-*StarrySky: A Practical System to Track Millions of High-Precision Query Intents by* Qi Ye (Sogou Inc.), Feng Wang (Sogou Inc.) and Bo Li (Sogou Inc.). StarrySky is a practical system for identifying and inferring millions of query intents with high precision and acceptable recall. The inference algorithm achieves up to 96% precision and 68% recall on daily search requests.

4. REFERENCES

[1] Zhuhua Cai, Dionysios Logothetis, and Georgos Siganos. Facilitating real-time graph mining. In Proceedings of the fourth international workshop on Cloud data management, CloudDB '12, pages 1–8, 2012.

[2] Raymond Cheng, Ji Hong, Aapo Kyrola, Youshan Miao, Xuetian Weng, Ming Wu, Fan Yang, Lidong Zhou, Feng Zhao, and Enhong Chen. Kineograph: taking the pulse of a fast-changing and connected world. In Proceedings of the 7th ACM european conference on Computer Systems, EuroSys '12, pages 85–98, 2012.

[3] Yanlei Diao, Peter Fischer, Michael J Franklin, and Raymond To. Yfilter: Efficient and scalable filtering of XML documents. In Data Engineering, 2002. Proceedings. 18th International Conference on, pages 341–342. IEEE, 2002.

[4] Samuel Madden, Mehul A. Shah, Joseph M. Hellerstein, and Vijayshankar Raman. Continuously adaptive continuous queries over streams. In SIGMOD, 2002.
[5] Nesreen K. Ahmed, Jennifer Neville, and Ramana Rao Kompella. Network sampling: From static to streaming graphs. CoRR, abs/1211.3412, 2012.
[6] Josep M Pujol, Vijay Erramilli, Georgos Siganos, Xiaoyuan Yang, Nikos Laoutaris, Parminder Chhabra, and Pablo Rodriguez. The little engine (s) that could: scaling online social networks. In SIGCOMM, 2010.
[7] Andre Bolles, Marco Grawunder, and Jonas Jacobi. Streaming SPARQL: extending SPARQL to process data streams. The Semantic Web: Research and Applications, 2008.
[8] Darko Anicic, Paul Fodor, Sebastian Rudolph, Nenad Stojanovic. EP-SPARQL: A Unified Language for Event Processing and Stream Reasoning In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, Ravi Kumar, eds., *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, 635-644, 2011. ACM.
[9] D.F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. C-SPARQL: SPARQL for continuous querying. In WWW, 2009.
[10] B. Shao, H. Wang, and Y. Li. Trinity: A Distributed Graph Engine on a Memory Cloud. In SIGMOD'13, pages 505–516, 2013.
[11] G. Wang, W. Xie, A. Demers, and J. Gehrke. Asynchronous large-scale graph processing made easy. In CIDR'13, 2013.