Letter from the Special Issue Editor

In the last decade, a tremendous amount of work has been devoted to managing and mining large graphs. We have witnessed during this golden time the dramatic rise of social networks, knowledge graphs, and data of increasingly rich relationships. Publications on graph related research also thrived. In this issue, we review some of the biggest challenges, survey a few brilliant solutions, and reflect on the current status and the future of this field.

Big graphs bring a lot of algorithmic challenges, and as a result, topics such as graph partitioning, graph reachability, keyword search, subgraph matching, etc. have attracted a lot of attention. In this issue, we chose not to focus on any specific algorithms, partly because there are just too many to cover. Instead, we take a system oriented approach, that is, we focus on work on managing and understanding graphs that lead to general purpose systems that support a variety of algorithms on big graphs. When the data we are dealing with contains billions of records (nodes) and trillions of relationships (edges), how to manage the data needs to take precedence over how to design ad-hoc algorithms that assume certain data organization tailored for the algorithms.

In "Graph Processing in RDBMSs," Zhao and Yu showed that, instead of devising specific algorithms with specific data organization, many graph processing needs can be supported in RDBMs with an advanced query language. They revisit and enhance SQL recursive queries and show that a set of fundamental graph operations are ensured to have a fixpoint. In "Trinity Graph Engine and its Applications," Shao, Li, Wang and Xia introduced the Trinity Graph Engine, whichh is an open-source distributed in-memory data processing engine, underpinned by a strongly-typed in-memory key-value store and a general distributed computation engine. It is used to serve real-time queries for many real-life big graphs such as Microsoft Knowledge Graph and Microsoft Academic Graph. In "GRAPE: Conducting Parallel Graph Computations without Developing Parallel Algorithms", Fan, Xu, Luo, Wu, Yu, and Xu develop a general purpose framework to parallelize existing sequential graph algorithms, without recasting the algorithms into a parallel model. It is clear that the two most important tasks are managing and mining graph data. In "Towards A Unified Graph Model for Supporting Data Management and Usable Machine Learning," Li, Zhang, Chen, and Ooi present a preliminary design of a graph model for supporting both data management and usable machine learning at the same time.

In many graph systems including the above, graphs are stored in their native forms (nodes and edges). Machine learning tasks on graphs may require a very different representation of graphs. In "Representation Learning on Graphs: Methods and Applications," Hamilton, Ying, and Leskovec tried to find a way to represent or encode graph structure so that it can be easily exploited by machine learning models. They develop a unified framework to describe recent approaches, and highlighted a number of important applications and directions for future work. In "On Summarizing Large-Scale Dynamic Graphs," Shah, Koutra, Jin, Zou, Gallagher, and Faloutsos focused on how to describe a large, dynamic graph over time using an information-theoretic approach. Specifically, it compresses graphs by summarizing important temporal structures and finds patterns that agree with intuition.

Finally, we also highlight the biggest challenges in the field. In "Billion-Node Graph Challenges", Xiao and Shao describe the challenges in the managing and mining billon-node graphs in a distributed environment, while in "Mining Social Streams: Models and Applications," Subbian and Aggarwal focus specifically on challenges in online social networks.

Haixun Wang
Facebook