Arabesque A System for Distributed Graph Mining

The paper presents a new system for distributed graph mining called Arabesque. Aim of this algorithm is to solve problems like frequent subgraph mining, counting motifs and finding cliques with higher performance than the other algorithms. Arabesque uses “Think like an embedding“ (TLE) method for efficient and faster computation results. Also, algorithm is scalable and Arabesque system uses multithreading to increase overall performance. Lastly, paper contains different experiments and their results for Arabesque and TLE’s performance comparison against similar algorithms.

Arabesque itself isn’t just an algorithm but it’s a system for graph mining like entity matching for Magellan. That’s why users can easily implement different methods in Arabesque. The authors mentioned that 20 lines of code in Arabesque is equivalent to ~4K lines of code. This paper also introduces TLE method and its advantages. It has better performance, more efficient and more scalable than the TLV and TLP methods. For memory space efficiency, the authors used ODAG method for storing embedding instances. Also, for faster computation, Arabesque uses multithreading and it significantly improves overall system performance. For multithreading, Arabesque uses workers and load balancing can be used on these workers. Arabesque is an automatic system and while running it doesn’t create each embeddings. It only loads necessary embeddings and systematically visits those created ones. Lastly, the authors tested Arabesques performance with 6 different datasets. Each dataset has distinct number of vertexes and edges (1K to 200M). The experiment results show that TLE is more efficient and scalable than the TLV and TLP. They also compared Arabesque with other systems and even on single thread run, Arabesque has close or even better performance.

This paper was hard for me to understand because of the graph data terms. That’s why my weakness points are mostly focuses on context not the algorithm or system itself. The authors could simplify the paper and could add more examples. For instance, there should be more examples about “Think like an embedding” subject. There was only one example about it and it couldn’t explain the TLE methods much.

More examples would benefit some of the readers. Also, Arabesque system uses lots of different outside libraries or systems (C++ and Java). I think, the system itself is very dependant on these outside sources and that can affect it. Finally, this paper focuses on only 3 different problems. However, adding more or graph data problems in the experiments would be nice.

2019-03-09

Baran Kaya, kayab@mcmaster.ca, 400284996