Incremental computation and maintenance of the *metric backbone* for fast graph analytics

Master Thesis Proposal

Many graph analytical applications encode metrics of distance or similarity in the edge weights of the input graph. Consider, for instance, a social network where the link weights represent the social proximity of the relationship. The analysis of weighted graphs provides insights in applications such as influential user inference, finding optimal paths to propagate information, community discovery, and fraud detection.

The **metric backbone** is the minimum subgraph of a weighted graph that preserves the shortest paths. Using the metric backbone in place of the original graph for analytics provides significant performance improvements, as the backbone is much smaller in size. By computing on a smaller graph, we achieve lower computation and memory requirements, while also reducing the communication overhead of distributed graph algorithms.

However, using the metric backbone to speed up graph analytics is only beneficial when the cost of its computation can be amortized. Unfortunately, computing the metric backbone for very large graphs can be time-consuming. If the input graph is static, the backbone computation can be amortized by running multiple analysis algorithms on top of the backbone. However, if the input graph is dynamic (edges and nodes can be added or removed), then recomputing the backbone from scratch is not a practical solution.

Thesis Goal

This thesis aims to improve the query evaluation performance of modern graph databases by leveraging the metric backbone. This work will explore incremental computation and maintenance of the metric backbone for dynamic graph inputs. Specifically, the goal is to implement and evaluate an incremental algorithm for computing the metric backbone under arbitrary edge additions and deletions, as described in the Appendix of [1]. Further, the metric backbone algorithm will also be implemented using Differential Dataflow [2] and the two approaches will be compared with regards to performance. The best algorithm will then be integrated in a graph database, such as Neo4j [3].

Additional directions

The metric backbone and distance backbones in general, offer a framework for the design of novel graph algorithms for distance computation and estimation. If time allows, this work will investigate how one could re-design certain algorithms explicitly on top of the backbone. For example, running a transparent shortest paths algorithm on top of the backbone gives the exact same result as it would on the original graph. However, we might be able to exploit the knowledge that every edge in the backbone corresponds to the shortest path between its endpoints to design a better shortest paths algorithm.

References

[1]: Kalavri, Vasiliki, Tiago Simas, and Dionysios Logothetis. "The shortest path is not always a straight line: leveraging semi-metricity in graph analysis." Proceedings of the VLDB Endowment 9.9 (2016): 672-683.

[2]: Differential Dataflow: https://github.com/frankmcsherry/differential-dataflow

[3]: Neo4j: https://neo4j.com/

If you are interested in this project please contact Vasiliki Kalavri (kalavriv@inf.ethz.ch). The proposed thesis will be supervised by Prof. Timothy Roscoe.