Mining Graph Patterns in Massive Networks

Nilothpal Talukder
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th st, Troy, NY 12180
talukn@rpi.edu

Keywords:
Graph Mining,
Massive Network,
Subgraph Isomorphism

Background

Graphs are widely used to represent relationships among the entities, such as friendships in social networks, interactions in biological networks, and so on. Mining commonly occurring subgraph patterns from a massive social graph or citation network can help discover similar groups/behaviors, which may be of interest to social scientists. Likewise, a bioinformatics researcher may be interested in finding the common sub-structures within gene/protein networks. In the literature, this task is known as frequent subgraph mining (FSM). Although the problem has a great deal of importance, unfortunately, it is computationally hard due to the following major facts:

1) The search space for enumerating the subgraph patterns is exponential.
2) It requires subgraph isomorphism checking, which belongs to the class of NP-complete problems.

The task has become highly challenging since the size of the graphs (e.g., social networks) has grown very large. For instance, the popular social network, Facebook, currently has 1.4 billion monthly active users. There is no extant FSM algorithm in the literature that can handle a graph that large.

In our research we are developing a scalable and distributed approach for mining frequent subgraph patterns from a single, massive network. There is a related problem of mining patterns in a database of very many, but usually much smaller, graphs. We refer this latter case as the transactional setting, which is relatively easy to handle in a distributed manner. We therefore focus on the much more challenging, single large network, scenario.

The FSM task is to enumerate all subgraphs with frequency, or support, above some minimum support threshold. However, the definition of support in a single graph has to be carefully defined. First, we need to find all distinct isomorphisms of a pattern in the input graph, and then choose a support measure that conforms to the anti-monotonicity principle, required for effective pruning of the output pattern space.

Most existing parallel [BPC06], distributed/mapreduce-based [LXG14] and GPU-based [KTAZ14] approaches have focused on the relatively easier transactional setting where there are many moderately sized input graphs, which can be horizontally partitioned among the processors and mined in parallel. However, in the single-graph setting the size of the graph can be massive which poses a huge challenge.

There are currently no distributed algorithms for mining a single large graph. Existing solution include SiGraM [KK05] and the state-of-art method, GraMi [EASK14], which are both sequential; and a shared-memory parallel approach [RK07]. All these approaches assume that the input graph and the intermediate data structures can be fit into the memory of a single shared-memory system, which is practically not feasible with very large graphs consisting of billions of vertices.

We propose a novel distributed algorithm for mining frequent subgraphs from a single, very large, labeled network. Our approach introduces several firsts:

i) It is the first approach to consider mining over a partitioned input graph, which introduces significant challenges,
ii) It is the first hybrid approach, which leverages both thread-based parallelism and MPI based distributed computation,
iii) It is the first approach demonstrated to scale to graphs with over a billion nodes and edges. Our approach minimizes the amount of communication, and uses efficient MPI primitives to extract performance. Results on up to 2048 IBM Blue Gene/Q compute nodes (with 16 cores each) show very good speedup.

Problem Statement

We briefly describe the notion of support in the context of FSM. In the case of a single graph, we use the most restricted node (MRN) support [BN08], also called as the minimum image based support. It is defined as the minimum number of unique vertex mappings over any of the vertices in a pattern, given its subgraph isomorphisms in the input graph. For the FSM problem, the input is a single graph and a user specified parameter called minimum support threshold, or in short, *minsup*. The task is to discover all subgraph patterns from the input graph, such that each pattern has a support greater than or equal to *minsup*.

Mining a large graph is a challenging task primarily because the graph and the intermediate states of mining, such as the set of isomorphisms, cannot be kept in memory of a single system. Here, we briefly describe the challenges and how we address them.

1*) Large input graph*: Our method allows for a massive input graph that need not fit in the memory of any individual compute node. We adopt a partitioned approach, where we split the input graph into multiple partitions and perform the mining task in a distributed fashion. While any partitioning scheme, even a random one, can be used for graph partitioning, for good performance we need the partitions to be balanced. Graph partitioning is a well-studied problem and solutions to these problems are generally derived using heuristics and approximation algorithms, since the balanced partitioning problem is NP-complete.

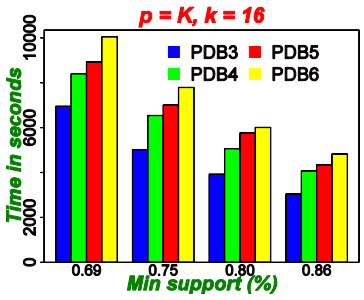
2) *False negatives*: A challenge in mining a partitioned graph is that there can be false negative patterns, i.e., a pattern *P* that is globally frequent can be missed due to the fact that certain edges involved in the isomorphisms for *P* span across different partitions. We address this problem by suitably extending the local partitions to external neighbors and make sure that no isomorphism is missed.

3) *Local support and pruning*: Since the support of a pattern depends on the minimum number of unique mappings of its vertices, at first glance, it appears that will not be able to prune any pattern locally, i.e., we would have to compute the isomorphisms for all possible patterns in each partition, followed by a reduction step to compute the global support. We develop a notion of local support that allows us to locally prune patterns that cannot be globally frequent.

4) *False positives*: The next challenge we face is how to cut down on the cost of communicating the isomorphisms in a distributed system, since a pattern can have an exponential number of isomorphisms (across different partitions). However, we need to communicate only the vertex mappings information, which is smaller, but still expensive. Therefore, we first estimate the global support based on only the cardinality information of the mappings. Since there will be patterns that are false positives, i.e., not actually frequent, we need to communicate vertex mappings to eliminate those.

Broader Impacts

Our distributed mining approach made it possible to discover frequent patterns from massive networks, at a scale that was not previously feasible. In the left figure, we show the timings from our data scaling experiment on an IBM Blue Gene/Q system. Here, PDB3 is a large protein structural interaction graph consisting of 17.4 million vertices and 68.5 million edges. PDB4, PDB5 and PDB6 graphs are 16x, 32x and 64x scaled versions of PDB3, respectively. *Thus, PDB6 has over 1 billion nodes and over 4 billion edges.* The timings are shown for different support values; *p* and *k* indicate the number of compute nodes and threads, respectively. We use p=32 compute nodes for PDB3, but 64x, i.e., p=2048 compute nodes for PDB6. The results show that our algorithm can efficiently scale to billion node/edge networks. *To the best of our knowledge this is the largest network considered, to date, for subgraph pattern mining*.



References

[BN08] B. Bringmann and S. Nijssen, "What is frequent in a single graph?" in Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 2008.

[BPC06] G. Buehrer, S. Parthasarathy, and Y. K. Chen, "Adaptive parallel graph mining for cmp architectures," in IEEE International Conference on Data Mining, 2006.

[EASK14] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis, "Grami: Frequent subgraph and pattern mining in a single large graph," Proceedings of the VLDB Endowment, vol. 7, no. 7, pp.517–528, 2014.

[KK05] M. Kuramochi and G. Karypis, "Finding frequent patterns in a large sparse graph," Data Mining and Knowledge Discovery, vol. 11, no. 3, pp. 243–271, 2005.

[KTAZ14] R. Kessl, N. Talukder, P. Anchuri, and M. J. Zaki, "Parallel graph mining with GPUs," Proceedings of the BigMine Workshop (ACM SIGKDD), pp. 1–16, 2014.

[LXG14] W. Lin, X. Xiao, and G. Ghinita, "Large-scale frequent subgraph mining in mapreduce," in IEEE International Conference on Data Engineering, 2014.

[RK07] S. Reinhardt and G. Karypis, "A multi-level parallel implementation of a program for finding frequent patterns in a large sparse graph." in IEEE International Parallel and Distributed Processing Symposium, 2007.