Graph summarizing using quasi-cliques

Julien Baste* Mohammed Haddad† Hamida Seba ‡

1 Context of the intership

This intership will take place within the ANR project COREGRAPHIE[1] and will be co-supervised by Julien Baste, Mohammed Haddad and Hamida Seba. The student can be hosted either in the GOAL (Graphes Algorithmes et Applications) team at LIRIS (Lyon) or in the ORKAD team at Cristal(Lille). This intership can be followed by a PhD thesis.

2 Description

Graph compression, also known as graph summarizing attracts more and more interests in various domains [5,7,9]. Existing works show that graph compression is not only a tool to reduce the amount of storage required to store massive graphs but also an interesting pre-processing step that can enhance graph algorithms. Graph compression provides another point of view from which we can tackle scalability and performance issues. Beyond the reduction of the volume of data which is the main aim of compression, we are looking for significant summaries that can be used, in graph analysis, without decompression. A graph summary can be lossy or lossless according to wether it preserves all the semantic and structural information of the original graph or not. Several approaches have been used to summarise graphs such as graph separators [2], modular decomposition [4,6,8], k2-trees [1,3], frequent substructures [9], etc.

During this intership, we will start from the fact that storing the vertex set of a clique is smaller than storing all the edges of this clique and the same applies to dense substructures of the graph. Based on this, we want to compress a graph using its cliques and quasi-cliques. The cliques or quasi-cliques we consider can overlap as long as they always contains edges that are not already encoded (i.e., marked).

We proceed as follows for a graph G:

Algorithm 1: Greedy cliques

Result: set S of cliques
unmark the edges of G;
$S = \emptyset$;
while *there exists an unmarked edge* **do**
  select an unmarked edge, this will be a clique K;
  while *there exists v a common neighbors of every vertex of K and there exists $u \in V(K)$ such that $\{u, v\}$ is unmarked* **do**
    add v to K;
  end
  add K to the set of remembered cliques S;
end
Return the set S of cliques

We also want to consider variant of this algorithm where at step 3 we select v that maximizes the set of unmarked edges. We also consider the case where K is not a clique but a quasi-clique and so at step 3, we may allow that v is not neighbor of all K but just a fraction such that K respect the conditions we want for the quasi-clique.

*julien.baste@univ-lille.fr
†mohammed.haddad@univ-lyon1.fr
‡hamida.seba@univ-lyon1.fr
[1] `https://coregraphie.projet.liris.cnrs.fr`

We expect that this procedure will reduce significantly the size of the graph for community graph in particular.

Moreover, we then can construct the compressed graph as the graph with vertex set S, the set of cliques, where there is an edge between $K_1 \in S$ and $K_2 \in S$ if and only if $V(K_1) \cap V(K_2) \neq \emptyset$. In this compressed graph we can compute for instance the shortest path in an exact way by adding a source and a sink.

As a matter of fact, in order to mitigate appearance of *false* edges in the decompressed graph, we also consider quasi-cliques in the complementary of the original graph. Thus finding the right decomposition of the graph and/or its complementary will have crucial importance in the context of a lossy compression designed for a specific application. We also believe that we can compute communities directly in the compressed graph.

N.B.: The study to be carried out during this internship could be purely theoretical or much more guided by practical applications with strong programming and benchmarking track. This will be defined in details with the candidate according to her/his profile.

References

[1] Sandra Álvarez-García, Borja Freire, Susana Ladra, and Oscar Pedreira. Compact and efficient representation of general graph databases. *Knowl Inf Syst*, 2018.

[2] Daniel K. Blandford, Guy E. Blelloch, and Ian A. Kash. Compact representations of separable graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pages 679–688, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[3] Nieves R. Brisaboa, Susana Ladra, and Gonzalo Navarro. *k2-Trees for Compact Web Graph Representation*, pages 18–30. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[4] B.-M. Bui-Xuan, M. Habib, V. Limouzy, and F. De Montgolfier. Algorithmic aspects of a general modular decomposition theory. *Discrete Applied Mathematics*, 157(9):1993–2009, 2009.

[5] Sofia Fernandes, Hadi Fanaee-T, and João Gama. Dynamic graph summarization: a tensor decomposition approach. *Data Mining and Knowledge Discovery*, 32(5):1397–1420, Sep 2018.

[6] T. Gallai. Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, 18:25–66, 1967.

[7] Kifayat Ullah Khan, Waqas Nawaz, and Young-Koo Lee. Set-based unified approach for summarization of a multi-attributed graph. *World Wide Web*, 20(3):543–570, May 2017.

[8] S. Lagraa and H. Seba. An efficient exact algorithm for triangle listing in large graphs. *Data Mining and Knowledge Discovery*, 30(5):1350–1369, Sep 2016.

[9] Yike Liu, Tara Safavi, Neil Shah, and Danai Koutra. Reducing large graphs to small supergraphs: a unified approach. *Social Netw. Analys. Mining*, 8(1):17, 2018.