Graph Mining (MPRI): Finding quasi-cliques

Mauro Sozio
Télécom Paris, IP Paris, France
sozio@telecom-paris.fr

January 7, 2020

General information and main heuristic. Given $\alpha \in [0,1]$, a graph $G = (V, E)$ is an α-quasi-clique if its *clique density* (also called edge density) is at least α, i.e. $\frac{2 \cdot |E|}{|V| \cdot |V-1|} \geq \alpha$. The main goal of this project is to find quasi-cliques with at least s nodes, where s is provided in input. Finding quasi cliques is an interesting problem from both a theoretical and practical point of view.

One heuristic that works well in practice is the following one. Given a graph $G = (V, E)$, the k-clique degree of a node v is defined to be the number of k-cliques in G containing v. Observe that a k-clique is a 1-quasi clique with k nodes, while the 2-clique degree of a node is equal to its degree. Let $G_0 = G$. At any step $t > 0$, if G_{t-1} is non-empty obtain G_t by removing a node with minimum k-clique degree (and all its edges) from G_{t-1}. The heuristic terminates when the graph becomes empty, at which step a graph with maximum clique density is returned among all G_t's of size at least s.

Tasks. Your tasks are the following:

1. Implement the heuristic described above. To this end, you should implement first the algorithm for enumerating all k-cliques in a graph described in Algorithm 2 of the paper provided for the project. Then, use such an algorithm for computing the k-clique degrees.

2. choose at least three graphs from `http://konect.uni-koblenz.de/`. For each of the graphs you chose, report the number of nodes and the number of edges. Report also the specifications of your machine (CPU, main memory, operative system, etc.). The graphs should be as large as possible, depending on the specifications of your machine. Let $s = 10$. For every input graph and for each value of k in $\{2, 3, 4\}$ report a) the clique density, b) the number of nodes of the graph found by the heuristic, c) the running time of the heuristic. Does the clique density increase or decrease as a function of k? What about the running time? Explain why those values increase or decrease.

3. Perform one of the following tasks: a) Devise your own heuristic for finding quasi cliques. This could be for example a variant of the heuristic described above. You should try to argue that your heuristic has an advantage with respect to the previous heuristic in terms of running time or clique density; b) in case you have access to some "interesting" graph (e.g. biological data, co-authorship graph, etc.), perform some data analysis on such a graph and argue that you found something "interesting". For example, you found a set of proteins with similar functions or a set of scientists working on a same research area.

What to send. You should send a report (1-3 pages) in pdf with the answers to the previous questions and your findings and send all the code as well (Python, Java or C/C++). You should send one single file via email to `sozio@telecom-paris.fr` with subject: "MPRI Project 2020" by **February 15th 2020**.

Evaluation. You will be evaluated by a) how well your report is written b) the heuristic you developed or the data analysis task c) how efficient is your code (C code is usually more efficient) d) how large are the graphs you considered relatively to the specifications of your machine. This is an individual project (it cannot be done in group).

Plagiarism. You are free to discuss with other students and the teacher however you should write your own code and your own report. In case, we suspected you copied from other students or from the internet you will fail the exam.