Graph characterisation for explaining runtime problems in graph pattern mining

Master thesis proposal

Supervision: Francesco Bariatti & Matthijs van Leeuwen
f.bariatti@liacs.leidenuniv.nl

June 2023

1 Context and motivation

Graph pattern mining [5] helps users make sense of large quantities of graph data by extracting recurring structures —the patterns— on which the user can focus their analysis, instead of having to take into account the entirety of the data at once. A major challenge for graph mining approaches is that computing pattern occurrences in the data has a potentially high complexity compared to other types of data/patterns: due to combinatorics, the number of occurrences of a pattern can be exponential, and computing them (known as the *subgraph isomorphism problem* [4]) is a NP-complete task which can take a long time.

In practice, *in most cases* the actual time required for graph pattern matching is not prohibitive, and approaches based on graph patterns can execute in reasonable time. However, *in some cases* (specifically, for *some patterns only*) the runtime can explode and significantly hinder the execution. For example, in our current research we want to get an idea of the size of the pattern space, so we randomly sample n patterns and compute their occurrences. The runtime of this approach tends to be linear: if it takes one minute for n samples, it will take roughly two minutes for $2n$ samples. But sometimes, a pattern will be sampled that is problematic and will take several minutes to compute *the occurrences of that pattern alone*. This makes the approach very fragile, as its runtime is difficult to predict and depends on luck. In another research project we were extracting all patterns from some graph data, and the runtime was reasonable enough. However, when we moved to a different dataset, the runtime suddenly exploded, making the approach impractical. Could have we predicted that beforehand by comparing the two datasets?

During this thesis, the student will work with Francesco Bariatti and Matthijs van Leeuwen, in the Explanatory Data Analysis group[1].

[1] https://eda.liacs.nl/

2 Goal and challenges

For this master thesis, we are interested in characterising and explaining what causes the runtime problems mentioned above. For that, we want to compare the data/patterns combinations which generate high runtime with the ones that do not. The analysis will be mainly focused on the approaches that we use for our research, and the real runtime problems that they exhibit in our experiments.

Currently, we plan to undertake the following steps during this master thesis[2]:

- Since many measures exist in the literature to compare graphs (e.g. centrality measures, attribute distributions, ...), a part of the work will be to review existing measures and what they can detect/express
  - Note that we aim to use existing graph manipulation libraries (e.g. NetworkX[3]), so implementing them from scratch will not be needed.
- Compare problematic and safe patterns with the chosen measures, and try to explain where the runtime problems can come from.
  - This could either be done by manual observation, or by training a machine learning model (or both!).
- Produce heuristics that can detect whether a candidate pattern would cause runtime problems or not, so as to avoid problematic patterns during execution without incurring too large of a runtime cost.
- Ideally, make this process generalisable enough that it could be applied to other graph mining approaches.
- Ideally, be able to produce patterns that do or do not cause runtime problems at will.

3 Relevant reading material

In addition to the material cited in previous sections, the following reading material could be interesting as an entry point to the subject:

- The material for the "Social Network Analysis for Computer Scientists" course at LIACS [2]
- Very complete book about graphs (long, it is suggested to select relevant chapters from the index): [3]
- The "algorithms" section of the documentation of the NetworkX library, to get an idea of existing measures: [1]

[2] Which can be subject to discussion based on the student's preferences and ideas, and preliminary results.

[3] `https://networkx.org/`

References

[1] Algorithms – NetworkX documentation. https://networkx.org/documentation/stable/reference/algorithms/index.html.

[2] Social Network Analysis for Computer Scientists course at leiden university. https://liacs.leidenuniv.nl/~takesfw/SNACS/.

[3] Michele Coscia. The Atlas for the Aspiring Network Scientist, February 2021. doi: 10.48550/arXiv.2101.00863.

[4] Scott Fortin. The Graph Isomorphism Problem. page 25, July 1996.

[5] Chuntao Jiang, Frans Coenen, and Michele Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(1):75–105, March 2013. doi: 10.1017/S0269888912000331.