Category: Doctorate

Author: Vinícius Vitor dos Santos Dias

Title: Graph Pattern Mining: consolidating models, systems, and abstractions

Graduate program: Graduate Program in Computer Science of the Federal University of Minas Gerais (DCC/UFMG)

Advisor: Dorgival Guedes Neto (DCC/UFMG)

Date of defense e approval: March 24th, 2023

Members of committee:

- Prof. Dorgival Guedes (Advisor)
  Departamento de Ciência da Computação - Universidade Federal de Minas Gerais
- Prof. Srinivasan Parthasarathy
  Department of Computer Science and Engineering - Ohio State University
- Prof. Arlei Lopes da Silva
  Department of Computer Science - Rice University
- Prof. Italo Fernando Scotá Cunha
  Departamento de Ciência da Computação - Universidade Federal de Minas Gerais
- Prof. Vinícius Fernandes dos Santos
  Departamento de Ciência da Computação - Universidade Federal de Minas Gerais
- Prof. Wagner Meira Júnior
  Departamento de Ciência da Computação - Universidade Federal de Minas Gerais

Grade and Honors: The dissertation was unconditionally approved, there is no grade assignment in the program. One of the publications [6] was selected to be presented in the best paper session, ranking among the top 6 full papers out of a total of 36. Despite the nomination, the publication was not selected for an award.

Highlights:

1. A proposed primitive-based model standardizes the design of GPM algorithms.
2. Fractal system offers a good trade-off between user experience and efficiency.
3. Memory demand is mitigated in Fractal, improving performance and reliability.
4. A dynamic load balancing in Fractal improves resource utilization.
5. Our experimental evaluation unveils challenges and opportunities in GPM.

1. Context and problem

Graphs are widely used to model problems in various areas, including web applications, social media, biological networks, brain networks, conceptual graphs, among others. In this work we navigate the trade-off between abstractions and system performance in the context of Graph Pattern Mining (GPM): a class of problems marked by the processing of subgraphs extracted from larger graphs. The relevance of GPM computation is multidisciplinary, including applications such as motif extraction from biological networks [1], frequent subgraph mining [9], subgraph searching over semantic data [8], social media network characterization [17], community discovery [2], periodic community discovery [15], temporal hotspot identification [18], identification of dense subgraphs in social networks [10], link spam detection [12], recommendation systems [19], graph learning [14], to cite a few.

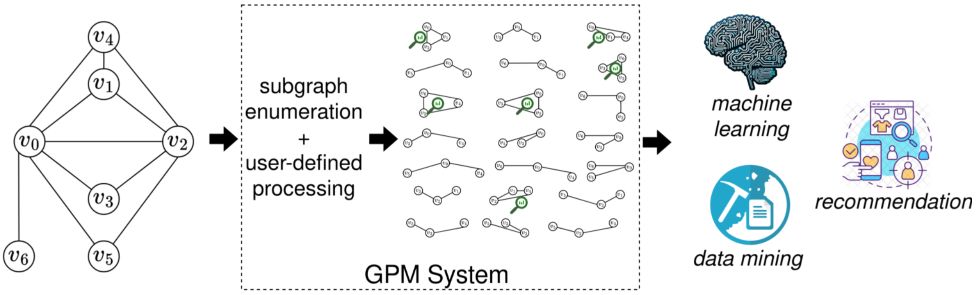

Figure 1. High-level operation of a Graph Pattern Mining (GPM) system.

In this context, general-purpose graph pattern mining systems (Fig. 1) emerge as an alternative for programming and for maintaining parallel GPM applications. Next we highlight the main challenges concerning GPM processing. *[The lack of a standard algorithm model]:* The space of existing general-purpose GPM systems is diverse and little effort has been made to model GPM algorithms in such a way that is independent of system implementation details. This limits the generalization, the proper evaluation, and the extensibility of existing GPM solutions. *[The need for an efficient, productive, and integrated GPM system]:* GPM tasks are computational intensive, irregular in terms of load balancing and memory access. They are also complex to develop from scratch since it often include non-trivial concepts from graph theory and mathematics (e.g., isomorphism and combinatorics) and often used as a pre-processing step in data analytics pipelines. It is not trivial to address all these aspects altogether. *[The need for a fair and informative evaluation of GPM paradigms]:* Existing GPM systems are not ideal for a wide experimental characterization of GPM paradigms since implementation details are too merged into application design, this challenges the fairness of performance comparisons and the identification of opportunities for research directions.

2. Objectives

The thesis statement of this work is that GPM systems can benefit from a strong, well-defined model for algorithms that is independent of implementation details such as system architecture, programming language, and parallelization strategies. Specific objectives: (1) *Propose* a simple and expressive algorithm model for general-purpose GPM; (2) *Design and implement* a GPM system that adopts the proposed model and that deals with system challenges concerning the efficiency and the programming productivity of GPM applications. (3) *Present* an evaluation study to consolidate collective knowledge about GPM processing and to identify promising future work.

3. Contributions

We highlight three contributions of these work (Fig. 2). First, we propose a model for representing GPM algorithms, unveiling important building blocks for standardized application design, which besides improving productivity also allows a more consistent access to performance diagnostics and optimizations. Second, we provide the design and the implementation of *Fractal*, a general-purpose distributed and parallel system for GPM. Fractal offers an expressive and compositional programming interface, bounds memory demand via a stateless subgraph enumeration algorithm, and includes an adaptive and dynamic load balancing layer via work stealing. Third, we leverage our well defined model and system to provide an extensive experimental study of GPM workloads, including a wide range of application scenarios considering multiple algorithms and over real-world datasets.

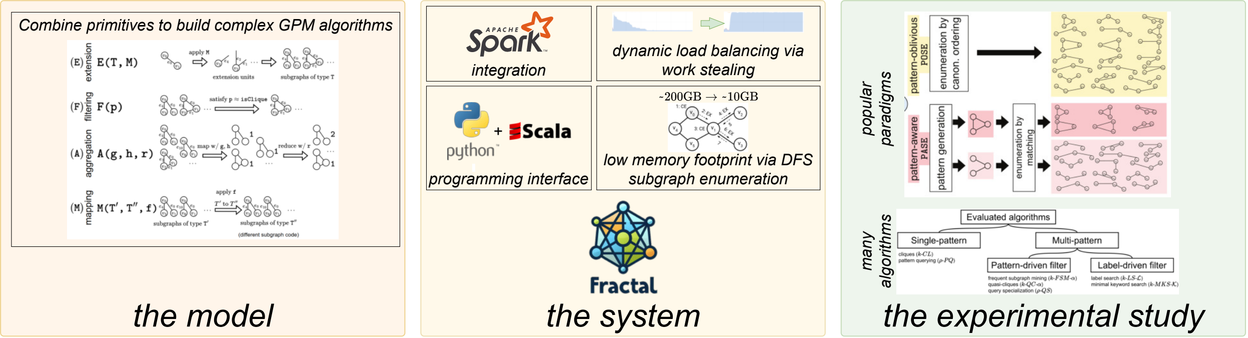

Figure 2. Overview of our contributions and solutions.

4. Relevance to the state-of-the-art

Overall, existing systems has its very particular application model and/or makes strong assumptions about the GPM tasks supported. In this work we focus on a system design that not only can be easily integrated to existing data analysis pipelines but also exhibit optimal and competitive performance for various tasks – none of the existing related work accomplishes these requirements altogether nor are adequate for a fair experimental study of GPM algorithms. For instance, first-generation system Arabesque [16] suffers from intractable memory demands. Peregrine [11] and Automine [13] are hand-tuned approaches that do not handle distributed environments nor offer integrated solution capabilities. G2Miner [5] (GPU) and G-Miner [4] (distributed) offer an efficient system design but not easily integrated into data processing tools. Tesseract [3] handles distributed and dynamic graphs but lacks the flexibility for supporting multiple paradigms in its subgraph enumeration engine.

5. Solution

Our model (Fig. 2-left) is based on a set of concise primitives that can be combined in such a way that abstracts all the low level implementation details and parallel/distributed deployment. Our system (Fig. 2-center) implements the primitive based model on top of Spark framework. We extended Spark's execution model to accomodate our optimizations: load balancing via dynamic work stealing, depth-first subgraph enumeration for bounded memory, and integration with Resilient Distributed Datasets (RDD) abstraction. Our experimental study (Fig. 2-right) include multiple GPM algorithms (categorized by semantics) and a wide range of real-world graphs. We reimplemented the algorithms under the same conditions to enable a fair comparison and an informative set of conclusions.

6. Evaluation

Fractal is efficient and competitive with existing baselines: We evaluated Fractal's performance against specialized baselines (able to solve a single problem) and other general-purpose GPM systems over real-world graphs. Fractal outperforms some baselines and stays competitive against hand-optimized ones – although Fractal is not always more efficient, it allows solving various problems with a reduced programming effort. Our evaluation also unveils that system optimizations proposed are able to enhance resource utilization by near-perfect load balancing and orders of magnitude improvement in memory demand. *Experimental study enhance knowledge about trade-offs between GPM paradigms:* We implemented and evaluated popular GPM algorithms using our solutions, enforcing comparison fairness and unveiling challenges and opportunities for the area. Our main conclusion is that there is no silver bullet in terms of GPM paradigms, contrary to existing claims in the literature.

Scientific production

[publication] Full paper (Qualis A1) [7]: ***Dias, V.***, *Teixeira, C. H. C., Guedes, D., Meira Jr., W., and Parthasarathy, S. (2019). Fractal: A general-purpose graph pattern mining system. In Proceedings of the 2019 International Conference on Management of Data* ***(SIGMOD)****.*

Link:
`https://doi.org/10.1145/3299869.3319875`

[publication] Full paper with best paper nomination (Qualis A3) [6]: ***Dias, V.***, *Ferraz, S., Vadlamani, A., Erfanian, M., Teixeira, C. H., Guedes, D., Meira, W., and Parthasarathy, S. (2023). Graph pattern mining paradigms: Consolidation and renewed bearing. In 2023 31st IEEE International Conf. on High Performance Computing, Data, and Analytics* ***(HiPC)****.*

Link:
`https://doi.org/10.1109/HiPC58850.2023.00040`

[software] Fractal project is publicly available, including reproducibility artifacts and guides on how to deploy and to use our system to produce new integrated solutions to custom GPM problems.

Code link (including reproducibility of above papers):
`https://github.com/dccspeed/fractal`

Data link:
`https://drive.google.com/drive/folders/1ViLAlQt45hFDtqTCJnOfqk4WZ71E3IUN`

[submission] Short course and hence, also a book chapter, submitted to this year's 39th Brazilian Database Symposium (SBBD '24): "Practical Graph Pattern Mining: Systems, Applications, and Challenges". The decision has not been made yet, however, if accepted we intend to share the knowledge produced with this dissertation with the database community through practical examples with Fractal.

References

[1] Agrawal, M., Zitnik, M., and Leskovec, J. Large-scale analysis of disease pathways in the human interactome. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing 23* (2018), 111–122.

[2] Benson, A. R., Gleich, D. F., and Leskovec, J. Higher-order organization of complex networks. *Science* (2016).

[3] Bindschaedler, L., Malicevic, J., Lepers, B., Goel, A., and Zwaenepoel, W. *Tesseract: Distributed, General Graph Pattern Mining on Evolving Graphs.* Association for Computing Machinery, New York, NY, USA, 2021, p. 458–473.

[4] Chen, H., Liu, M., Zhao, Y., Yan, X., Yan, D., and Cheng, J. G-miner: An efficient task-oriented graph mining system. In *Proceedings of the Thirteenth EuroSys Conference* (2018), EuroSys '18.

[5] Chen, X., and Arvind. Efficient and scalable graph pattern mining on GPUs. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)* (Carlsbad, CA, July 2022), USENIX Association, pp. 857–877.

[6] Dias, V., Ferraz, S., Vadlamani, A., Erfanian, M., Teixeira, C. H., Guedes, D., Meira, W., and Parthasarathy, S. Graph pattern mining paradigms: Consolidation and renewed bearing. In *2023 31st IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC)* (2023).

[7] Dias, V., Teixeira, C. H. C., Guedes, D., Meira Jr., W., and Parthasarathy, S. Fractal: A general-purpose graph pattern mining system. In *Proceedings of the 2019 International Conference on Management of Data (SIGMOD)* (2019).

[8] Elbassuoni, S., and Blanco, R. Keyword search over rdf graphs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2011), CIKM '11, ACM, pp. 237–242.

[9] Elseidy, M., Abdelhamid, E., Skiadopoulos, S., and Kalnis, P. Grami: Frequent subgraph and pattern mining in a single large graph. *Proc. VLDB Endow. 7*, 7 (Mar. 2014), 517–528.

[10] Hooi, B., Shin, K., Lamba, H., and Faloutsos, C. Telltail: Fast scoring and detection of dense subgraphs. *Proceedings of the AAAI Conference on Artificial Intelligence 34*, 04 (Apr. 2020), 4150–4157.

[11] Jamshidi, K., Mahadasa, R., and Vora, K. Peregrine: A pattern-aware graph mining system. In *Proceedings of the Fifteenth European Conference on Computer Systems* (New York, NY, USA, 2020), EuroSys '20, Association for Computing Machinery.

[12] Leon-Suematsu, Y. I., Inui, K., Kurohashi, S., and Kidawara, Y. Web Spam Detection by Exploring Densely Connected Subgraphs. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* (Aug. 2011), vol. 1, pp. 124–129.

[13] Mawhirter, D., and Wu, B. Automine: Harmonizing high-level abstraction and high performance for graph mining. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles* (New York, NY, USA, 2019), SOSP '19, ACM, pp. 509–523.

[14] Meng, C., Mouli, S. C., Ribeiro, B., and Neville, J. Subgraph pattern neural networks for high-order graph evolution prediction, 2018.

[15] Qin, H., Li, R.-H., Wang, G., Qin, L., Cheng, Y., and Yuan, Y. Mining periodic cliques in temporal networks. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (2019), pp. 1130–1141.

[16] Teixeira, C. H. C., Fonseca, A. J., Serafini, M., Siganos, G., Zaki, M. J., and Aboulnaga, A. Arabesque: A system for distributed graph mining. In *Proceedings of the 25th Symposium on Operating Systems Principles* (2015), SOSP '15, pp. 425–440.

[17] Ugander, J., Backstrom, L., and Kleinberg, J. Subgraph frequencies: mapping the empirical and extremal geography of large graph collections. In *WWW* (2013).

[18] Yang, Y., Yan, D., Wu, H., Cheng, J., Zhou, S., and Lui, J. C. Diversified temporal subgraph pattern mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD '16, Association for Computing Machinery, p. 1965–1974.

[19] Zhao, H., Zhou, Y., Song, Y., and Lee, D. L. Motif enhanced recommendation over heterogeneous information network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2019), CIKM '19, ACM, pp. 2189–2192.