Electronic Letters on Computer Vision and Image Analysis 15(2):13-15, 2016

Anytime and Distributed Approaches for Graph Matching

Zeina Abu-Aisheh

Laboratoire d'Informatique (LI), Université François Rabelais, 37200, Tours, France
Advisor/s: Romain Raveaux, Patrick Martineau and Jean-Yves Ramel
Date and location of PhD thesis defense: 25 May 2016, Ècole polytechnique de l'universitè de Tours

Received 31st August 2016; accepted 7th September 2016

Abstract

Due to the inherent genericity of graph-based representations, and thanks to the improvement of computer capacities, structural representations have become more and more popular in the field of Pattern Recognition (PR). In a graph-based representation, vertices and their attributes describe objects (or part of them) while edges represent interrelationships between the objects. Representing objects by graphs turns the problem of object comparison into graph matching (GM) where correspondences between vertices and edges of two graphs have to be found [14].

In the domain of GM, over the last decade, Graph Edit Distance (GED) has been given a specific attention due to its flexibility to match many types of graphs [2]. GED has been applied to a wide range of specific applications from molecule recognition to image classification [9]. Researchers have shed light on the approximate methods that can find suboptimal solutions hopefully close to the optimal ones but the gap between optimal and suboptimal solutions has not been deeply studied yet.

Roughly speaking, two main families of GM have been found in the literature: exact and error-tolerant GM. In this thesis, we propose adding a new GM family, called anytime GM. In order to demonstrate the benefit of having such a family, a new optimized GED algorithm which is based on depth-first search is put forward. This algorithm, referred to as *DF*, speeds up the computations of GED thanks to its upper and lower bounds' pruning strategy and its preprocessing step. Moreover, *DF* does not exhaust memory as the number of pending tree search nodes is relatively small thanks to the depth-first search where the number of pending nodes, or so-called partial edit paths, is $|V_1|.|V_2|$ in the worst case where $|V_1|$ and $|V_2|$ are the numbers of vertices in G_1 and G_2, respectively. Accordingly, *DF* outperforms the best-first GED algorithm (A^*) [11] in terms of speed, precision and classification rates. *DF* is able to provide not only one solution but successive solutions for a better and better quality according to available resources. The anytime version of *DF*, denoted by *ADF*, is able to find an initial, possibly suboptimal, solution quickly, keep it in the memory and then continue searching for improved solutions until the convergence to a provably optimal solution. The simplicity of the approach makes it very easy to use; it is also widely applicable. It can be used not only when an optimal solution is desired, but also when we want to see the evolution of the quality of the suboptimal solutions found at each time t. Generally speaking, Anytime GM provides an attractive approach to challenging GM problems, especially when the time

Correspondence to: zeina.abu-aisheh@univ-tours.fr

Recommended for acceptance by David Vázquez Bérmudez
DOI http://dx.doi.org/10.5565/rev/elcvia.986
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

and memory available to compare graphs are limited or uncertain and when we are interested in improving the best solution found so far [17, 6].

This thesis is also considered as a first attempt to reduce the run time of exact GED methods using parallel and distributed fashions. To go one step further and to be able to match larger graphs with better quality, we also propose parallel and distributed GED algorithms. We speed up the computations of *DF* by proposing a multi-threaded algorithm, referred to as *PDFS*, with an efficient load balancing strategy [16]. In *PDFS*, each thread gets one or more partial edit paths and all threads solve their assigned edit paths in a fully parallel manner. A work stealing process is performed whenever a thread finishes all its assigned threads. Moreover, synchronization is applied in order to ensure the upper bound coherence. *PDFS* has a bottleneck since that it cannot be run on several machines. To cope with this problem, a distributed version can be of great interest so as to scale up and to match larger graphs. We propose an exact GED algorithm, referred to as *D-DF*. This algorithm is implemented on the top of Hadoop [15] with a message passing tool [7]. *D-DF* starts with a preprocessing step and the distribution of partial edit paths among workers. Each worker gets one partial edit path and all workers solve their assigned edit paths in a fully distributed manner. In addition, a notification process is integrated. When any worker finds a better upper bound, it notifies the master to share the new upper bound with all workers.

In the literature, error-tolerant GM methods have often been evaluated in a classification context and less deeply assessed in terms of the accuracy of the found solution when scaling up to match large graphs [12, 1, 4, 5, 3]. To evaluate the accuracy of error-tolerant GM methods, graph-level information is required at matching level (i.e., matching quality and similarity deviation) and not only at class level. In this thesis, a performance evaluation tool for GED methods is proposed. This contribution consists of two parts: First, we propose a graph database repository, called GDR4GED, dedicated to scalability. GDR4GED is annotated with graph-level information like graph edit distances and their matching correspondences for some representative graph databases [10, 1]. Since we are interested in testing the scalability of GM methods, we divide the selected datasets into subsets; each of which represents graphs that have the same number of vertices. Second, new metrics have been put forward to characterize GED methods by evaluating the matching correspondences as well as the distance between each pair of graphs. Because of the high complexity of GED methods, we propose evaluating them under time and memory constraints. The aim of this contribution is to make GED methods better comparable against each other and to provide information about their applicability on real-world problems. For that reason, we highly encourage the community not only to use the information provided in GDR4GED, but also to integrate their algorithms' answers when obtaining more accurate results.

To evaluate *ADF*, *PDFS* and *D-DF*, we compared them to both exact and approximate GED approaches, using the datasets proposed in GDR4GED under soft and hard time constraints. Soft constraints are devoted to accuracy tests while hard constraints are devoted to speed tests. Results showed that under soft time constraints *PDFS* and *D-DF* had the minimum deviation and matching dissimilarity. Both of them explored more nodes than *ADF* in parallel and distributed fashions and thus helped in speeding up the exploration of the search tree. As for the tests under hard time constraints, *PDFS* was among the slowest algorithms, however, it was always the most precise one.

In the experiments of anytime GED, we focused on both the deviation when varying the timeout and the minimal time needed by the algorithm to get a first solution on different graph datasets. Results showed that there is a trade-off between time and quality. Even if the fast bipartite GM [13] and the bipartite GM [8] were faster when graphs were sparser; *ADF* was faster when graphs were denser. It is remarkable that anytime algorithms are also effective when we accept some additional time that grantees better solutions to be found.

This thesis brings into question the usual evidences saying that it is impossible to use exact error-tolerant GM methods in real-world applications when matching large graphs, or even in a classification context. However, we argue and show that a new type of GM, referred to as anytime methods, can be successful in a graph-level context as well as a classification one. the anytime videos, the pseudo-codes and the publications related to the thesis are publicly available at: http://www.rfai.li.univ-tours.fr/PagesPerso/zabuaisheh/home.html. The thesis is also publicly available at: http://www.rfai.li.univ-tours.

fr/Documents/Articles_RFAI/PhD2016zeina.pdf.

References

[1] Cmu house and hotel datasets. *http://vasc.ri.cmu.edu/idb/html/motion.*, 2013.

[2] K. Fu. A. Sanfeliu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics.*, pages 353–363, 1983. doi:10.1109/tsmc.1983.6313167.

[3] Vincenzo Carletti, Pasquale Foggia, and Mario Vento. Performance comparison of five exact graph matching algorithms on biological databases. *New Trends in Image Analysis and Processing Proceedings*, pages 409–417, 2013. doi:10.1007/978-3-642-41190-8_44.

[4] Foggia Pasquale Vento Mario Conte, Donatello. Challenging complexity of maximum common subgraph detection algorithms: A performance analysis of three algorithms on a wide database of graphs. *Journal of Graph Algorithms and Applications*, 11(1):99–143, 2007. doi:10.7155/jgaa.00139.

[5] P. Foggia, C. Sansone, and M. Vento. A performance comparison of five algorithms for graph isomorphism. In *Graph-based Representations in Pattern Recognition*, pages 188–199, 2001.

[6] Eric A. Hansen and Rong Zhou. Anytime heuristic search. *Journal of Artificial Intelligence Research*, 28(1):267–297, 2007.

[7] Flavio Junqueira et al. *Zookeeper: Distributed Process Coordination*. 2013.

[8] Bunke H. Riesen, K. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing.*, 27(7):950–959, 2009. doi:10.1016/j.imavis.2008.04.004.

[9] Kaspar Riesen. *Structural Pattern Recognition with Graph Edit Distance - Approximation Algorithms and Applications*. Advances in Computer Vision and Pattern Recognition. Springer, 2015.

[10] Kaspar Riesen et al. Iam graph database repository for graph based pattern recognition and machine learning. *Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 287–297, 2008. doi:10.1007/978-3-540-89689-0_33.

[11] Kaspar Riesen, Stefan Fankhauser, and Horst Bunke. Speeding up graph edit distance computation with a bipartite heuristic. In *Mining and Learning with Graphs*, 2007.

[12] M. De Santo, P. Foggia, C. Sansone, and M. Vento. A large database of graphs and its use for benchmarking graph isomorphism algorithms. *Pattern Recognition Letters*, 24(8):1067 – 1079, 2003. doi:10.1016/s0167-8655(02)00253-2.

[13] Francesc Serratosa. Speeding up fast bipartite graph matching through a new cost matrix. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(2), 2015. doi:10.1142/s021800141550010x.

[14] Mario Vento. A long trip in the charming world of graphs for pattern recognition. *Pattern Recognition*, 48(2):291–301, 2015. doi:10.1016/j.patcog.2014.01.002.

[15] Tom White and Doug Cutting. *Hadoop : the definitive guide*. O'Reilly, 2009.

[16] Chengzhong Xu and Francis C. Lau. *Load Balancing in Parallel Computers: Theory and Practice*. Kluwer Academic Publishers, 1997.

[17] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *Artificial Intellgience Magazine*, 17(3):73–83, 1996.