Machine Learning for Subgraph Extraction: Methods, Applications and Challenges

Kai Siong Yow
School of Computer Science and Engineering
Nanyang Technological University
kaisiong.yow@ntu.edu.sg

Ningyi Liao
School of Computer Science and Engineering
Nanyang Technological University
liao0090@e.ntu.edu.sg

Siqiang Luo
School of Computer Science and Engineering
Nanyang Technological University
siqiang.luo@ntu.edu.sg

Reynold Cheng
Department of Computer Science
Guangdong-Hong Kong-Macau Joint Laboratory
HKU Musketeers Foundation Institute of Data Science
The University of Hong Kong
ckcheng@cs.hku.hk

ABSTRACT

Subgraphs are obtained by extracting a subset of vertices and a subset of edges from the associated original graphs, and many graph properties are known to be inherited by subgraphs. Subgraphs can be applied in many areas such as social networks, recommender systems, biochemistry and fraud discovery. Researchers from various communities have paid a great deal of attention to investigate numerous subgraph problems, by proposing algorithms that mainly extract important structures of a given graph. There are however some limitations that should be addressed, with regard to the efficiency, effectiveness and scalability of these traditional algorithms. As a consequence, machine learning techniques—one of the most latest trends—have recently been employed in the database community to address various subgraph problems considering that they have been shown to be beneficial in dealing with graph-related problems. We discuss learning-based approaches for four well known subgraph problems in this tutorial, namely subgraph isomorphism, maximum common subgraph, community detection and community search problems. We give a general description of each proposed model, and analyse its design and performance. To allow further investigations on relevant subgraph problems, we suggest some potential future directions in this area. We believe that this work can be used as one of the primary resources, for researchers who intend to develop learning models in solving problems that are closely related to subgraphs.

PVLDB Reference Format:
Kai Siong Yow, Ningyi Liao, Siqiang Luo, and Reynold Cheng. Machine Learning for Subgraph Extraction: Methods, Applications and Challenges. PVLDB, 16(12): 3864 - 3867, 2023.
doi:10.14778/3611540.3611571

1 INTRODUCTION

Graphs are commonly used in representing pairwise relationships between objects in which they involve in numerous real-world applications [4, 12, 13] such as information systems, financial markets and community networks. Many problems are visualised using graph models based on various graph properties, to understand the process and to predict the solutions of these problems. This approach has also been extended to subgraphs given that many interesting graph properties (e.g., being a triangle-free or bipartite graph) are known to be hereditary for subgraphs.

Subgraph problems are often solved by extracting some regular patterns and structures from original graphs. Due to the complexity [15] of the problems and the difficulty in extracting structures for certain types of graphs, problems related to these classes of graphs have been studied extensively via various graph representations by using machine learning (ML) [20] techniques, to enhance the overall performances particularly the efficiency and effectiveness of conventional algorithms. In view of the abilities of ML in dealing with numerous real-world problems and applications, applying ML techniques on graph problems is therefore crucial to enrich solutions to the related applications.

There are several benefits in employing ML techniques to address graph related problems compared to conventional approaches. First, ML techniques are more flexible and scalable. Various training strategies can be employed and they can usually be used in dealing with large and complex query graphs that have non-specific query patterns. Second, ML techniques can be combined with conventional frameworks to enhance their efficiency, e.g., RL strategies and heuristic search. Third, learning models yield solutions that have higher precision even for large query graphs, as evidenced by recent findings [24]. In addition, they have been shown to be more effective by employing various training strategies.

Given the popularity and significance of this research direction, there are quite a number of surveys [21, 31] where ML frameworks are used in solving graph and combinatorial optimisation problems. Despite the increasing importance of subgraph extraction, no tutorials or surveys focus on it. Hence, we provide a summary of ML frameworks that have been developed over the past few years in dealing with subgraph extraction. We focus on four representative

Proceedings of the VLDB Endowment, Vol. 16, No. 12 ISSN 2150-8097.
doi:10.14778/3611540.3611571

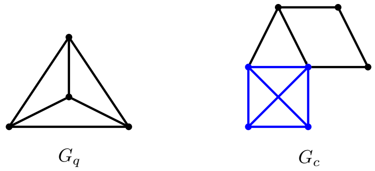

Figure 1: The graph G_c contains a subgraph (a complete graph of size four K_4 coloured in blue) that is isomorphic to G_q

subgraph problems that cover a wide range of applications (see the respective sections) in different fields. Based on the proposed frameworks, we then suggest some future directions so that related graph problems can be explored using analogous strategies.

Through this tutorial, we will convey the key concept that is proposed to extract graph information in each framework to audiences, and discuss some of their connections. We aim to trigger some creative ideas to (1) enhance further the proposed frameworks, (2) develop new learning models in solving related graph problems and (3) improve data management based on ML. This work may also serve as one of the references in perceiving graph problems that have been solved by learning-based approaches, and ultimately contribute practical concepts to relevant research communities in managing data and solving real-world problems.

Tutorial overview. We split this 1.5-hour lecture-style tutorial into the following sections:

(1) **Introduction** (5 mins): We introduce the concepts of ML and graphs, as well as the connections between them.
(2) **Subgraph isomorphism counting** (20 mins): We cover general graphs with both vertex and edge labels based on three learning models in this problem.
(3) **Maximum common subgraph** (15 mins): We discuss two learning frameworks in solving the MCS problem, where one of them is developed based on a conventional approach.
(4) **Community detection** (20 mins): We cover three learning methods that are proposed to solve overlapping CD as well as CD in attributed graphs.
(5) **Community search** (15 mins): We focus on two learning frameworks that are designed to address interactive and attributed CS problems.
(6) **Future directions** (10 mins): We conclude with some potential problems and challenges in solving subgraph problems using learning methods.

Target audience. The tutorial is designed for researchers and practitioners who are interested on graphs, data management and ML techniques.

2 TUTORIAL OUTLINE

2.1 Subgraph Isomorphism Counting

The *subgraph isomorphism* (SI) problem determines if a *corpus graph* G_c contains a subgraph that is isomorphic to a *query graph* G_q. (see Figure 1 for example). Its applications [2] can be found in recommender systems, bioinformatics and even social networks. It is however known that the SI problem belongs to the class NP-complete [6], and the counting version of this problem is hence more complex. One such problem is the *subgraph isomorphism counting* (SIC) problem that takes a corpus graph and a query graph as input, and then determines the frequency of subgraphs of the corpus graph that is isomorphic to the query graph.

We discuss three learning models in tackling the SIC problem.

(1) **DIAMNet**. Liu et al. [17] proposed a representation model to retrieve essential information from the corpus and/or query graphs. They introduced neural network structures that learn to predict a count for a corpus graph based on the encoded elements from both corpus and query graphs. An interaction layer is proposed to aggregate information and output the prediction.

(2) **ALSS**. A semi-supervised framework ALSS that employs the question-answering framework is proposed by Zhao et al. [33], by utilising both sketch and active learnings. To count a query in a corpus graph, the learned sketch extracts the corpus graph to a series of vectorised features, each corresponds to a basic substructure. For the task-specific prediction, a multilayer perceptron (MLP) is used to aggregate the corpus graph representation and the query substructures, and to estimate the final count of queries. The active learner queries the sketch model itself to choose the query graph.

(3) **NeurSC**. Wang et al. [27] proposed a semi-supervised approach NeurSC to extract and integrate different representations from both corpus and query graphs, and produce outputs by using an estimator. The extraction module only extracts useful vertices from the corpus graph according to vertices in query graphs. The estimator on the other hand utilises a graph isomorphism network to capture substructures for the corpus graph and the individual information for query graphs. To make predictions, features learned by both GNNs are combined and passed to an MLP. Adversarial trainings are also conducted given that representations between query graphs and structures of the corpus graph could be different.

2.2 Maximum Common Subgraph

The *maximum common subgraph* (MCS) problem finds isomorphic subgraphs in two graphs to identify the largest subgraph in common [5]. The MCS of G_q and G_c in Figure 1 is a K_4 (there are however other common subgraphs of smaller size), which is also the graph G_q. This problem is known to be NP-hard and computationally challenging. It is used to measure graph similarity and to identify the degree of structural overlaps between networks, and has a broad application [7] in various disciplines such as biochemistry, information retrieval and programme analysis.

We focus on two learning frameworks in addressing this problem.

(1) **McSplit+RL**. The state-of-the-art algorithm McSplit [19] employs a branch and bound (BnB) heuristic search framework that exhausts the search space efficiently in solving the MCS problem. Liu et al. [18] proposed to improve the McSplit by combining it with reinforcement learning (RL) strategies. The framework McSplit+RL is developed based on the observation that the search algorithm can be regarded as an agent that performs a sequence of actions in finding maximum common subgraphs. RL strategies are used together with the BnB algorithm in growing a candidate subgraph, so that the optimal vertex is chosen upon learning. The search space is also minimised by designing an appropriate reward function.

(2) **GLSearch**. Another RL-powered model GLSearch is proposed by Bai et al. [3]. The GLSearch model deeply reshapes the process of

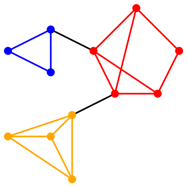

Figure 2: A small community that has three clusters in social network analysis, coloured in three different colours

reaching the optimal solution to solve the MCS problem directly instead of just enhancing conventional search algorithms. The model primarily learns according to a quality function, and conducts semi-supervised learning by first pre-training small datasets with sole BnB search, before it performs predictions on large datasets. The RL algorithm not only performs branching point choices, but also adjusts the order of the search so that solutions can be found effectively without involving much backtracking and pruning processes.

2.3 Community Detection

Let G be a graph. The *community detection* (CD) [9] problem aims to partition G into multiple groups such that the vertex set $U \subseteq V(G)$ are densely connected among themselves within a group, but sparsely connected with vertices in $V(G) \setminus U$. It could be used in identifying missing links in a network, fraud detection [22] as well as social network analysis. A small example where a community is partitioned into three groups that appears in social network analysis can be found in Figure 2.

We discuss two learning frameworks in addressing overlapping CD, and one for the CD problem in attributed networks.

(1) **SMACD**. Gujral and Papalexakis [11] developed SMACD, a semi-supervised model that detects both non-overlapping and overlapping communities in multi-view graphs. Two constraints that are meant to use for CD are included into the coupled matrix-tensor factorisation [1] model, to develope an algorithm. An automated mechanism is introduced so that a suitable sparsity regulariser penalty λ will be selected based on the fact that λ and the sparsity levels in the latent vectors are correlated.

(2) **NOCD**. Shchur and Günnemann [24] introduced a framework NOCD that combines the Bernoulli-Poisson (BP) model [28] and GNNs in discovering overlapping communities in undirected graphs. They discovered a number of benefits by using GNN architectures in the CD problem. First, better outputs can be obtained by using GNN models that produce identical affiliation matrices for adjacent vertices, compared to simpler models such as free variable and MLP. Second, vertex features can be integrated into GNN models easily. Third, during the training phase, communities for unseen vertices could also be predicted inductively.

(3) **CE-MOEA**. A continuous encoding multi-objective evolutionary algorithm (CE-MOEA) that uses a GNN encoding method was proposed by Sun et al. [26] to transform discrete problems to continuous problems. The benefits of their learning framework in dealing with attributed networks are as follows. First, for both attributed and non-attributed graphs, the encoding method can be applied regardless of whether they are undirected or directed. Second, during the GNN encoding, the information of neighbouring vertices are fully exploited, which makes it more robust. Third, by dealing with multi-objective continuous optimisation problems, any MOEA can be employed resulting in a smoother fitness landscape.

2.4 Community Search

Let G be a graph and $v \in V(G)$ be a query vertex. The *community search* (CS) (or *query-based CD*) problem [8, 25] is a variant of CD problems which aims to determine the most likely subgraph $H \subseteq G$ such that $v \in V(H)$ and H satisfies the cohesiveness and connectivity constraints. Community search appears in many real-world applications [8] including friend recommendation, e-commerce and fraudulent group discovery.

To enhance further the performance of conventional algorithms, learning-based frameworks have recently been proposed [10, 14]. We now discuss two frameworks that are designed to address (1) interactive CS and (2) attributed community search (ACS) problems.

(1) **ICS-GNN**. Gao et al. [10] developed an interactive CS algorithm ICS-GNN that uses a GNN to capture similarities between vertices in an online social network. The model involves several rounds of community search where feedback from users are incorporated during the search process. A ranking loss is introduced to integrate implicit feedback from users so that a correct label can be decided. In handling the special case where a query vertex is equivalent to a boundary vertex of a community, a greedy measure is introduced in which the authors used the global relative benefit of a vertex to determine its membership.

(2) **QD-GNN & AQD-GNN**. Jiang et al. [14] proposed a supervised learning method namely QD-GNN to encode information from both query vertices and graphs, which can be used in dealing with both CS and ACS problems (with appropriate extensions). Their model can be applied on attributed graphs, which extends the ICS-GNN [10] framework. The QD-GNN model consists of three encoders (for graph, query and attribute) and one feature fusion operator, which encode graph information and utilise local query information as well as global graph knowledge in obtaining the final output. To solve the ACS problem, another variant namely AQD-GNN is developed by incorporating query attributes into QD-GNN. This new model consists of one extra encoder that provides an interface for query attributes, and a revised fusion operator. To support interactive attributed CS, the GNN model in ICS-GNN [10] is replaced with QD-GNN and AQD-GNN.

2.5 Future Direction

Given that ML frameworks in solving subgraph problems are relatively limited compared to other areas, we now suggest some general research questions so that more learning models can be explored in addressing similar graph problems.

Learning strategy. Since different learning strategies are adapted in ML models, they could be characterised so that the most appropriate learning strategy that should be used in solving certain types of graph problems can be determined, which may lead to a more promising outcome.

Graph type. Existing learning frameworks for graph problems could possibly be transformed to handle a wide range of subgraph problems, i.e., problems that involve dynamic graphs [16] and directed graphs [4, 29, 32] that are also used to model many real-world applications. This is somehow more challenging due to their restrictive graph properties.

GNN. Knowing that the proposed learning models using GNNs outperform traditional algorithms in different aspects, it is natural to explore the power of GNNs in relevant graph problems from different perspectives such as the approximation ratios [23].

Model extension. Learning-based approaches have been shown in providing better solutions in various circumstances. It is hence worth to extend it to other subgraph problems, particularly those related to a problem that has been addressed by learning models. For instance, design learning frameworks for the densest subgraph problem that are related to the maximum weight clique problem.

3 PRESENTERS

(1) **Reynold Cheng** is a Professor of the Department of Computer Science in the University of Hong Kong (HKU). His research interests are in data science, big graph analytics and uncertain data management. He received his BEng in 1998, and MPhil in 2000 from HKU. He then obtained his MSc and PhD degrees from Department of Computer Science of Purdue University in 2003 and 2005.
(2) **Siqiang Luo** is a Nanyang Assistant Professor at the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. He was a postdoc researcher at Harvard University from 2019 to 2020. He received his Ph.D. degree in computer science from HKU in 2019. His research interest lies in big data management algorithms and systems, including graph algorithms/systems, key-value systems, and ML-driven data management.
(3) **Kai Siong Yow** is an SASEA Fellow hosted by the School of Computer Science and Engineering, NTU, Singapore. He received his Ph.D. degree in mathematics from Monash University, Australia. His research interest lies in graph theory, data management and computational mathematics.
(4) **Ningyi Liao** is currently a Ph.D student at the School of Computer Science and Engineering, NTU, Singapore. His research interest lies in ML graph algorithms.
Note: This tutorial is designed based on our long survey paper [30], and it has not been offered elsewhere before.

ACKNOWLEDGMENTS

This research is supported by Singapore MOE AcRF Tier 1 Seed Funding (RS05/21), NTU startup grant and in part by MOE AcRF Tier 2 Grant (MOE-T2EP20122-0003). The first author is most grateful to Singapore National Academy of Science for appointing him as an SASEA Fellow through a grant supported by NRF Singapore, and Universiti Putra Malaysia (GP-IPM/2020/9684300) for granting him Leave of Absence in completing this work in NTU. Reynold Cheng was supported by the University of Hong Kong (Projects 104005858 and 10400599), the Guangdong-Hong Kong-Macau Joint Laboratory Program 2020 (Project No: 2020B1212030009), and The Hong Kong Jockey Club Charities Trust (HKJC), No. 260920140.

REFERENCES

[1] E. Acar, T. G. Kolda, and D. M. Dunlavy. 2011. All-at-once optimization for coupled matrix and tensor factorizations. *Preprint.* arXiv:1105.34226.

[2] N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S. C. Sahinalp. 2008. Biomolecular network motif counting and discovery by color coding. *Bioinformatics* 24, 13 (2008), i241–i249.

[3] Y. Bai, D. Xu, Y. Sun, and W. Wang. 2020. GLSearch: Maximum Common Subgraph Detection via Learning to Search. In *ICML*. 588–598.

[4] J. A. Bondy and U. S. R Murty. 1976. *Graph theory with applications.* The Macmillan Press Ltd, New York.

[5] H. Bunke. 1997. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters* 18, 8 (1997), 689–694.

[6] S. A. Cook. 1971. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM STOC.* 151–158. doi:10.1145/800157.805047.

[7] A. P. Cootes, S. H. Muggleton, and M. J. E. Sternberg. 2007. The Identification of Similarities between Biological Networks: Application to the Metabolome and Interactome. *Journal of Molecular Biology* 369, 4 (2007), 1126–1139.

[8] Y. Fang, X. Huang, L. Qin, Y. Zhang, W. Zhang, R. Cheng, and X. Lin. 2020. A survey of community search over big graphs. *The VLDB Journal* 29, 1 (2020), 353–392. doi:10.1007/s00778-019-00556-x.

[9] S. Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174. doi:10.1016/j.physrep.2009.11.002.

[10] J. Gao, J. Chen, Z. Li, and J. Zhang. 2021. ICS-GNN: Lightweight interactive community search via graph neural network. *Proc. VLDB Endow* 14, 6 (2021), 1006–1018. doi:10.14778/3447689.3447704.

[11] E. Gujral and E. E. Papalexakis. 2018. SMACD: Semi-supervised multi-aspect community detection. In *Proceedings of the 2018 SIAM ICDM.* 702–710. doi:10.1137/1.9781611975321.79.

[12] F. Harary and R. Z. Norman. 1953. *Graph theory as a mathematical model in social science.* University of Michigan, Institute for Social Research, Ann Arbor.

[13] B. Hayes. 2000. Graph theory in practice: Part I. *Amer. Scientist* 88, 1 (2000), 9–13.

[14] Y. Jiang, Y. Rong, H. Cheng, X. Huang, K. Zhao, and J. Huang. 2022. Query driven-graph neural networks for community search: From non-attributed, attributed, to interactive attributed. *Proc. VLDB Endow* 15, 6 (2022), 1243–1255.

[15] R. M. Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations.* Springer, Boston, MA, 85–103. doi:10.1007/978-1-4684-2001-2_9.

[16] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart. 2020. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research* 21, 70 (2020), 1–73.

[17] X. Liu, H. Pan, M. He, Y.u Song, X. Jiang, and L. Shang. 2020. Neural Subgraph Isomorphism Counting. In *SIGKDD.* 1959–1969.

[18] Y. Liu, C. M. Li, H. Jiang, and K. He. 2020. A Learning Based Branch and Bound for Maximum Common Subgraph Related Problems. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 03 (2020), 2392–2399.

[19] C. McCreesh, P. Prosser, and J. Trimble. 2017. A Partitioning Algorithm for Maximum Common Subgraph Problems. In *26th IJCAI.* 712–719.

[20] M. Mohri, A. Rostamizadeh, and A. Talwalkar. 2018. *Foundations of machine learning.* MIT Press.

[21] Y. Peng, B. Choi, and J. Xu. 2021. Graph Learning for Combinatorial Optimization: A Survey of State-of-the-Art. *Data Science and Engineering* 6 (2021), 119–141.

[22] C. A. R. Pinheiro. 2012. Community detection to identify fraud events in telecommunications networks. In *SAS SUGI Proceedings: Customer Intelligence.*

[23] R. Sato, M. Yamada, and H. Kashima. 2019. Approximation Ratios of Graph Neural Networks for Combinatorial Problems. In *Advances in Neural Information Processing Systems*, Vol. 32.

[24] O. Shchur and S. Günnemann. 2019. Overlapping Community Detection with Graph Neural Networks. In *Proceedings of the First International Workshop on Deep Learning for Graphs (DLG '19).* 1–7.

[25] M. Sozio and A. Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *SIGKDD.* 939–948.

[26] J. Sun, W. Zheng, Q. Zhang, and Z. Xu. 2022. Graph Neural Network Encoding for Community Detection in Attribute Networks. *IEEE Transactions on Cybernetics* 52, 8 (2022), 7791–7804. doi:10.1109/TCYB.2021.3051021.

[27] H. Wang, R. Hu, Y. Zhang, L. Qin, W. Wang, and W. Zhang. 2022. Neural Subgraph Counting with Wasserstein Estimator. In *SIGMOD.* 160–175.

[28] J. Yang and J. Leskovec. 2013. Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization Approach. In *Proceedings of the Sixth ACM WSDM.* 587–596.

[29] K. S. Yow, G. E. Farr, and K. J. Morgan. 2018. Tutte invariants for alternating dimaps. *Preprint.* arXiv:1803.05539.

[30] K. S. Yow, N. Liao, S. Luo, R. Cheng, C. Ma, and X. Han. 2023. A Survey on Machine Learning Solutions for Graph Pattern Extraction. *Preprint.* arXiv:2204.01057v3.

[31] K. S. Yow and S. Luo. 2022. Learning-Based Approaches for Graph Problems: A Survey. *Preprint.* arXiv:2204.01057v2.

[32] K. S. Yow, K. J. Morgan, and G. E. Farr. 2021. Factorisation of greedoid polynomials of rooted digraphs. *Graphs and Combinatorics* 37, 6 (2021), 2245–2264.

[33] K. Zhao, J. X. Yu, H. Zhang, Q. Li, and Y. Rong. 2021. A Learned Sketch for Subgraph Counting. In *SIGMOD.* 2142–2155.