Postprint of a Graph Neural Network-Based Symbolic Algorithm for Subgraph Matching

Authors: Yang Xin, Xu Zhoubo, Chen Puqing, Liu Huadong, Xu Zhoubo

Date: 2022-11-02T00:00:00+00:00

Abstract

Subgraph matching is a fundamental problem in graph data analysis with significant research importance. To address the issue of extensive redundant search in subgraph matching solution algorithms, we propose a symbolic subgraph matching algorithm based on graph neural networks. This algorithm leverages graph neural network technology to aggregate neighborhood information of nodes, obtaining feature vectors that incorporate local graph properties and structures, which are then used as filtering conditions to derive the node candidate set C for the query graph. Additionally, by optimizing the matching order and utilizing symbolic ADD operations to construct the respective candidate regions of C within the data graph, redundant search during the subgraph enumeration and verification process is reduced. Experimental results demonstrate that, compared with the VF3 algorithm, this algorithm effectively improves the solution efficiency of subgraph matching.

Full Text

Abstract

Subgraph matching is a fundamental problem in graph data analysis with significant research importance. To address the issue of extensive redundant search in existing subgraph matching algorithms, this paper proposes a subgraph matching symbolic algorithm based on graph neural networks (SSMGNN). The algorithm utilizes graph neural network technology to aggregate neighborhood information of nodes, obtaining feature vectors that incorporate local attributes and structures of the graph. These vectors serve as filtering conditions to generate node candidate sets C for the query graph, thereby reducing redundant searches during the subgraph enumeration and verification process. By constructing various candidate regions for C in the data graph and optimizing the matching order using symbolic operations, the algorithm significantly improves

the efficiency of subgraph matching. Experimental results demonstrate the effectiveness of the proposed approach.

Introduction

Graphs serve as an effective data structure for modeling relationships between entities, enabling the representation of numerous complex real-world problems. Subgraph matching, which aims to find all subgraphs in a data graph g that are isomorphic to a query graph q, constitutes one of the most fundamental problems in graph analysis with widespread applications in chemical formula retrieval, image retrieval, social network analysis, and other domains.

As the size of data grows rapidly, the computational complexity of subgraph matching increases exponentially, prompting researchers to focus on expanding the solvable scale and improving the efficiency of subgraph matching algorithms. Over the past decades, numerous algorithms have been proposed. The Ullmann algorithm, based on backtracking tree search, enumerates all matches of query graph q in data graph g but employs only simple pruning strategies that cannot efficiently reduce the search space. The VF2 algorithm enhances pruning effectiveness by incorporating neighbor information as constraints, thereby effectively reducing the search space. GraphQL generates a depth-first search tree for the query graph to filter nodes in the data graph, performing hierarchical traversal while using node information at each level as constraints. Spath utilizes complex structural and semantic information to filter nodes in the data graph.

While these algorithms improve search efficiency through various pruning techniques, they struggle to solve subgraph matching on large-scale data graphs within reasonable time. To expand the solvable scale, the VF3 algorithm computes matching orders based on the occurrence probability of query graph nodes in the data graph and introduces classification concepts to categorize nodes by degree and label, further reducing the search space. CFL-Match proposes decomposing the query graph to delay Cartesian products, effectively reducing redundant intermediate results. CECI stores partial edges from candidate sets and partitions the data graph into multiple embedding clusters for parallel processing, employing auxiliary data structures and pruning techniques to repeatedly trim embedding clusters. Although constructing auxiliary data structures improves performance on large-scale data graphs, it consumes substantial memory space and preprocessing time.

To balance the time and space complexity of subgraph matching algorithms while optimizing matching order and reducing redundant searches, this paper proposes the SSMGNN algorithm. Unlike existing approaches, SSMGNN introduces graph neural networks to extract node features that serve as filtering conditions to reduce candidate set sizes. The algorithm also incorporates algebraic decision diagrams (ADD) as an implicit storage structure for parallel processing of candidate regions.

Algorithm Framework

The SSMGNN algorithm consists of three main components: candidate set generation, matching order optimization, and subgraph isomorphism solving. The basic framework is illustrated in [Figure 1: see original paper]. From the perspective of improving filtering efficiency, the algorithm introduces graph neural networks to extract node features and uses these vectors as filtering conditions to generate candidates for the query graph. Matching order optimization considers candidate set sizes, node degrees, and the hierarchical structure of the query graph to determine an optimal matching sequence. Symbolic operations construct candidate regions in the data graph within the radius of the query graph, enabling parallel processing and reducing redundant searches. Due to the implicit storage characteristic of symbolic structures, the algorithm can operate on large-scale graphs within limited memory space.

Candidate Set Generation

Existing subgraph matching algorithms primarily generate candidate sets based on node labels and degree information while neglecting local neighborhood structure and attribute information. SSMGNN employs graph neural networks to aggregate neighborhood information, enabling nodes to incorporate more local information into their feature vectors and thereby enhancing filtering efficiency.

The algorithm first utilizes graph attention networks to learn node features for both the data graph and pattern graph, obtaining their representation vectors. We design a function $p(\cdot)$ that compares each dimension of the representation vectors to determine whether the neighborhood structures of two nodes may have an inclusion relationship. Based on this subgraph prediction function, we examine whether nodes u and v have consistent labels and degrees. If consistent and the neighborhood information suggests an inclusion relationship, node v is added to the candidate set of node u.

Matching Order Optimization

The matching order significantly impacts search efficiency. Given a query graph q and data graph g, if the matching order is $P = \{(u_1, C(u_1)), (u_2, C(u_2)), ...\}$, the search process may need to traverse the entire search space. An appropriate matching order can effectively reduce search iterations and improve efficiency.

SSMGNN optimizes the matching order by comprehensively considering three aspects: candidate set sizes, node degrees, and the query graph structure. Priority is given to nodes with small candidate sets and large degrees, which reduces intermediate results. Since the search process checks whether structural relationships between candidate nodes and already-matched nodes are consistent with the query graph, prioritizing nodes with many connections to matched nodes also improves efficiency.

The matching order optimization function incorporates candidate set size, node

degree, and the number of matched neighbors. The root node is selected first based on the ratio of degree to candidate set size. For subsequent nodes, the algorithm selects the node with the maximum value of:

$$\text{order}(u) = \max\left(\frac{\deg(u)}{|C(u)|} + |N(u) \cap M|\right)$$

where $C(u)$ is the candidate set of node u, $N(u)$ is the neighbor set of u, $\deg(u)$ is the degree of u, and M is the set of matched nodes (initially empty). When multiple nodes have the same ratio, priority is given to nodes with more connections to M.

Candidate Region Construction Using ADD

Research indicates that performing subgraph matching within candidate regions can effectively reduce search space, and candidate sets can be further filtered during region construction. To obtain all isomorphic subgraphs in the data graph, SSMGNN uses each node in the root node' s candidate set as a center to mine subgraphs of size equal to the query graph' s radius as candidate regions. This approach introduces symbolic operations to compress storage and enable parallel construction of all candidate regions, improving efficiency.

An Algebraic Decision Diagram (ADD) is a highly compact data structure that enables compressed storage and parallel operations. For graph G, node encoding uses binary strings of length m to represent nodes and edges. The edge set E_g is transformed into a representation $E_g(x_m, y_1, y_2, \dots, y_m)$, where X represents the start node and Y represents the end node of an edge. To distinguish different candidate regions, a variable Z is added as a region marker, resulting in the representation $E_g(x_m, y_1, y_2, \dots, y_m, z)$.

The candidate region construction process uses breadth-first search from each root candidate node to build regions of radius r. During construction, nodes at each layer are intersected with the candidate sets of corresponding query graph nodes to retain only valid candidates. The pseudocode for region construction is shown in Algorithm 1.

Overall Algorithm

The complete SSMGNN algorithm proceeds as follows:

1. **Node Embedding**: Generate node embeddings for each vertex in the data graph and query graph using graph neural networks.
2. **Candidate Computation**: Compute candidate sets C for query graph nodes based on feature vectors and neighborhood inclusion relationships.
3. **Order Optimization**: Determine the matching order and root node using candidate sets, degrees, and query graph structure.
4. **ADD Construction**: Convert candidate sets and root nodes into ADD representation.

5. **Region Partition**: Build multiple candidate regions R in parallel using symbolic operations.
6. **Subgraph Matching**: For each candidate region, perform backtracking search to find feasible solutions and add them to the solution set.

The algorithm returns all subgraph isomorphic solutions.

Experiments

Experimental Setup

The experimental environment uses protein-protein interaction datasets. The hardware configuration includes an Intel Core i5-1038NG7 CPU and 16GB RAM, running Windows operating system. Implementation uses Python and C/C++.

Two public datasets are employed: **human** and **yeast**. The **human** dataset is an undirected graph with multiple labels, while the **yeast** dataset contains a different set of labels. For each dataset, query sets Q_i are generated, where Q_i represents a set of connected query graphs containing i nodes. Each query set contains the same number of connected graphs with identical node counts.

The VF3 algorithm is selected for comparison as it is a highly efficient single-machine algorithm that outperforms many existing methods.

Results and Analysis

Experimental results are evaluated using average runtime, calculated as the total time to find all isomorphic solutions for a query group divided by the number of queries. This metric better reflects the time efficiency of subgraph isomorphism algorithms.

Figure 2 shows the average runtime on the yeast dataset, while Figure 3 presents results on the human dataset. When the query set is Q_4, SSMGNN demonstrates relatively low average solving time. Through comparative experiments, SSMGNN outperforms VF3 on both datasets, with the performance advantage becoming more pronounced as query graph size increases.

Although SSMGNN requires additional time for graph neural network-based neighborhood information aggregation, this preprocessing step consumes minimal time (as shown in Table 1). The neighborhood aggregation time for query sets of different scales remains below 0.1 seconds, even for 20-node queries. Experimental observations reveal that aggregating more neighborhood information does not necessarily improve filtering effectiveness. When the k-value (representing k-hop neighbors) becomes too large, all nodes' neighborhood information tends to converge, reducing filtering effectiveness. The optimal k-value is typically within the query graph' s radius range, which rarely exceeds 3 in these datasets.

The **yeast** dataset is denser with fewer labels, requiring more search time. However, SSMGNN' s optimization of matching order and incorporation of neighborhood information as filtering conditions significantly reduces search space. Consequently, SSMGNN solves the **yeast** dataset more efficiently than VF3, despite both algorithms' time complexity increasing with query graph size. SSMGNN' s time increase is smaller than VF3' s, demonstrating its advantage for larger query graphs.

Conclusion

This paper proposes SSMGNN, a subgraph isomorphism algorithm that combines graph neural networks with symbolic algebraic decision diagrams. The algorithm leverages graph neural networks to aggregate neighbor information, improving the efficiency of candidate node filtering. Based on candidate set characteristics and subgraph structure, SSMGNN introduces an optimized matching order and constructs candidate regions using symbolic ADD operations, enabling parallel partitioning and backtracking search on a single machine. Experimental results demonstrate that the algorithm effectively improves the solving efficiency of subgraph isomorphism problems.

Future work will investigate advanced symbolic techniques to further enhance single-machine solving efficiency for subgraph isomorphism problems.

References

[1] SUN Z,LUO X.In-memory subgraph matching:an in-depth study[C]//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data.Portland,Oregon:ACM Press,2020:1083-1098.

[2] HAN M,KIM H,GU G,et al.Efficient subgraph matching:harmonizing dynamic programming,adaptive matching order,and failing sets together[C]//Proceedings of the 2019 International Conference on Management of Data.Amsterdam:ACM Press,2019:1429-1446.

[3] ULLMANN J R.An algorithm for subgraph isomorphism[J].Journal of the ACM,1976,23(1):31-42.

[4] CORDELLA P,FOGGIA P,SANSONE C,et al.A (sub) graph isomorphism algorithm for matching large graphs[J].IEEE Transactions on Pattern Analysis and Machine Intelligence,2004,26(10):1367-1372.

[5] HE H,SINGH A K.Graphs-at-a-time:query language and access methods for graph databases[C]//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data.Vancouver,Canada:ACM Press,2008:405-418.

[6] ZHAO X,HAN C,LI X,et al.Spath:graph query optimization with large neighborhood signatures[J].Proceedings of the VLDB Endowment,2010,3(1/2):340-351.

[7] CARLETTI V,FOGGIA P,SAGGESE A,et al.Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with VF3[J].IEEE Transactions on Pattern Analysis and Machine Intelligence,2017,40(4):804-818.

[8] BI F,CHANG J,LIN L,et al.Efficient subgraph matching by postponing cartesian products[C]//Proceedings of the 2016 International Conference on Management of Data.San Francisco:ACM Press,2016:1199-1214.

[9] BHATTARAI B,LIU H,HUANG H.Ceci:compact embedding cluster index for scalable subgraph matching[C]//Proceedings of the 2019 International Conference on Management of Data.Amsterdam,Netherlands:ACM Press,2019:1447-1462.

[10] BONNICI V,GIUGNO R,PULVIRENTI A,et al.A subgraph isomorphism algorithm and its application to biochemical data[J].BMC Bioinformatics,2013,14(7):S13.

[11] SOLNON C.AllDifferent-based filtering for subgraph isomorphism[J].Artificial Intelligence,2010,174(12-13):850-864.

[12] SHANG Y,LIN C,ZHANG Y,et al.Taming verification hardness:an efficient algorithm for testing subgraph isomorphism[J].Proceedings of the VLDB Endowment,2008,1(1):364-375.

[13] REN G,WANG H.Exploiting vertex relationships for speeding up subgraph isomorphism on large graphs[J].Proceedings of the VLDB Endowment,2015,8(5):617-628.

[14] VELICKOVIC P,CUCURULL G,CASANOVA A,et al.Graph attention networks[EB/OL].(2018-02-04)[2022-04-02].https://arxiv.org/pdf/1710.10903.

[15] SCARSELLI F,GORI M,TSOI A C,et al.The graph neural network model[J].IEEE Transactions on Neural Networks,2008,20(1):61-80.

[16] WU Z,PAN S,CHEN F,et al.A comprehensive survey on graph neural networks[J].IEEE Transactions on Neural Networks and Learning Systems,2020,32(1):4-24.

[17] BAHAR R I,FROHM E A,GAONA C M,et al.Algebraic decision diagrams and their applications[J].Formal Methods in System Design,1997,10(2):171-206.

[18] HAN S,LEE J,LEE H.Turboiso:towards ultrafast and robust subgraph isomorphism search in large graph databases[C]//Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data.New York:ACM Press,2013:337-348.

[19] YING R,HE R,CHEN K,et al.Graph convolutional neural networks for web-scale recommender systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.London,UK:ACM Press,2018:974-983.

[20] YOU J,YING R,RE C,et al.Graph convolutional policy network for goal-directed molecular graph generation[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems.Montréal,Canada:Curran Associates Inc.,2018:6412-6422.

[21] ALMASRI I,GAO X,FEDOROFF N.Quick mining of isomorphic exact large patterns from large graphs[C]//2014 IEEE International Conference on Data Mining Workshop.Piscataway,NJ:IEEE Press,2014:315-322.

[22] 杨欣, 李周波, 蒲庆, 等. 基于邻居信息聚合的子图同构约束求解算法 [J]. 计算机应用,2021,41(1):43-47.

[23] 王浩, 张鹏. 有序二叉树决策图及应用 [M]. 北京: 科学出版社,2009:92-111.

[24] 李华, 张明. 代数决策图在图匹配中的应用研究 [J]. 桂林电子科技大学学报,2019,39(5):357-363.

Note: Figure translations are in progress. See original paper for figures.