Review on Frequent Subgraph Pattern Mining Algorithms

Janki K. Bhut
M.E.C.E., Faculty of PG Studies & Research in Engg. (run by MEFGI), Rajkot ,India

Mansi Vithalani
Assi. Prof., Faculty of PG Studies & Research in Engg. (run by MEFGI), Rajkot, India

Abstract

Frequent subgraph pattern mining is a one of the most popular research topics in data mining. Aim of graph mining is finding interesting patterns within data that represent novel knowledge. Now a day frequent subgraph mining used in various domains like in chemical compounds, social networks, biological networks etc. Mining patterns from graph database is difficult because of subgraph testing and their different operations. This paper gives the idea about different subgraph algorithms based on their approaches.

1. Introduction

Data mining is the procedure of extracting knowledge from raw data. Due to increasing number of complex objects, data mining algorithms are facing challenges. To model such complex object, graph data structure is used. A graph based structure representation combined with a substructure discovery technique. Most important concept in graph mining is to find frequent subgraph from graph database. Frequent subgraph pattern mining is procedure of extracting all frequent subgraphs from graph dataset who have occurrence count greater than or equal to the specified threshold.

A simple example of frequent subgraph pattern mining is as below. In which one graph database as input with two different graphs (i.e. g1 and g2) is given and third graph (g3) represent the output which is display the frequent subgraph pattern of input graphs.

The frequent subgraph mining algorithms mainly divided into two different approaches: (1) Apriory based approach and (2) pattern growth approach [1]. Algorithms of both approaches are describing in section-2.1 and section-2.2 respectively.

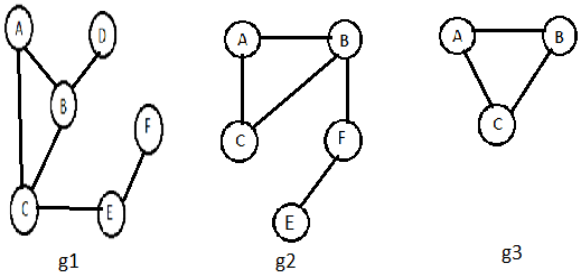

Figure 1. Example of frequent subgraph mining

2. Frequent subgraph algorithms

Frequent subgraph algorithms uses two different search strategies for finding frequent patterns. In which generally apriory based approach uses BFS search strategy and pattern growth approach uses DFS search strategy.

2.1. Apriory based algorithms

The search for frequent graphs starts with graphs of small "size," and proceeds in a bottom-up manner by generating candidates having an extra vertex, edge, or path [1]. The complexity of apriory-based substructure mining algorithm is based on candidate generation steps. Apriory-based algorithms of frequent subgraph are as below.

2.1.1. AGM algorithm. AGM [2] developed by Inokuchi, T. Washio, and H. Motoda in 2000. AGM uses the level-wise search strategy. So, they generate candidate based on vertex that increases the substructure size. AGM uses adjacency matrix for graph representation. In the experiment large graph of chemical compound discovered by AGM having size 13 atoms. AGM efficiently mine frequent subgraph from given dataset but complexity is high due to multiple candidate generation.

2.1.2. FSG algorithm. FSG [3] developed by M. Kuramochi and G. Karypis in 2001. FSG generate candidate based on edge. So, increases the complexity by adding edge. Every time candidate subgraphs are adding edge to previous subgraph. FSG uses transaction identifier lists for frequency counting and uses adjacency list for graph representation. In FSG, canonical labels used to check isomorphic graphs. FSG uses isomorphic testing so it is very costly and FSG also generate multiple candidates.

2.1.3. FFSM algorithm. FFSM (Fast Frequent Subgraph Mining) [4] developed by Jun Huan, Wei Wang, Jan Prins in 2003. FFSM uses vertical level search strategy for reduce number of candidate generation. Methods used in FFSM algorithms are: canonical adjacency matrix with FFSM-join and FFSM-extension, suboptimal CAM tree. So, it uses adjacency matrix for graph representation. Limitation of FFSM algorithm is that it is NP-complete problem.

2.1.4. SPIN algorithm. SPIN (**SP**anning tree based maximal graph m**IN**ing) [5] developed by Jun Huan, Wei Wang, Jan Prins,Jiong Yang in 2004. SPIN algorithm only mines maximal frequent subgraph. SPIN algorithm save space and subsequent analysis effort using several pruning techniques. Pruning technique used in SPIN algorithm are bottom-up pruning, tail shrink and external-edge pruning. SPIN uses the adjacency matrix for graph representation. Two different dataset used for evolution of performance of SPIN algorithm and they are: DTP AIDS and DTP CM data set.

2.1.5. GREW algorithm. GREW [6] developed by Michihiro Kuramochi and George Karypis in 2004. GREW is efficient, can scale to very large graphs, and find non-trivial patterns that cover large portions of the input graph and the lattice of frequent patterns. Two versions of GREW are: GREW-SE (single-edge collasping) and GREW-ME (multi-edge collasping). GREW can generally operate effectively on very large graphs. GREW uses sparse graph for the graph representation. Four different data sets used for evolution of performance and they are: Aviation, VSLI data set, Citation data set and Web data set.

2.1.6. Dynamic GREW algorithm. Dynamic GREW [7] developed by Karsten M. Borgwardt, Hans-Peter Kriegel, Peter Wackersreuther in 2006. In which consider dynamic graphs with edge insertions and edge deletions over time. It is also uses the adjacency matrix for graph representation. Limitation of this algorithm is extra overhead in identify dynamic patterns.

2.2. Pattern growth approach

The pattern growth approach use breath-first search as well as depth-first search for consumes less memory. Pattern growth based algorithms of frequent subgraph are as below.

2.2.1. Gspan algorithm. Gspan (graph-based Substructure pattern mining) [8] developed by Xifeng Yan, Jiawei Han in 2002. Gspan use DFS strategy, lexicographic order, minimum DFS code and rightmost extension. So that, it discovers frequent substructures without candidate generation. Gspan works on label simple graph. Gspan use adjacency list for graph representation. In Gspan use 340 chemical compound data set for evolution of performance of algorithm.

2.2.2. CloseGraph algorithm. CloseGraph [9] developed by Xifeng Yan, Jiawei Han in 2003. Instead of mining all subgraphs, CloseGraph algorithm mine only closed frequent graph pattern. A frequent pattern is closed if there exists no proper super-pattern with the same support in the data set. CloseGraph is more efficient than Gspan. CloseGraph also use DFS strategy, lexicographic order, minimum DFS code and rightmost extension for finding closed frequent patterns. So, it also use adjacency list for graph representation.

2.2.3. Gaston algorithm. Gaston is carried out with the help of embedding lists, where all the occurrences of a particular label are stored in the embedding lists. It is use hash table for the graph representation. But in which interesting pattern may be lost. Gaston tool [10] is available for mining cyclic subgraphs.

2.2.4. Subdue algorithm. Subdue [11] developed by Nikhil S. Ketkar, Lawrence B. Holder, Diane J. Cook in 2005. Subdue focuses on not only frequent pattern discovery but also on compress the graph data set. For that use minimum description length (MDL) approach and compression based methodology in Subdue algorithm. Subdue uses the adjacency matrix for graph representation. In Subdue chemical toxicity dataset and chemical compound data set used for evolution of performance.

2.2.5. RING algorithm. RING [12] developed by Shijie Zhang ,Jiong Yang , Shirong Li in 2009. Two steps of RING algorithm are (1)compute pattern distribution (2) depth first searching algorithm to mine representative to a pre-set space limit. RING use invariant vectors methodology for subgraph pattern

mining. RING also uses the adjacency matrix for graph representation and used DFS search strategy. In RING algorithm protein-protein interaction data set used for evolution of performance.

2.2.6. GraphSig algorithm. GraphSig [13] developed by Sayan Ranu , Ambuj K. Singh in 2009. In GraphSig convert each graph into a set of feature vectors where each vector represents a region within the graph. Prior probabilities of features are computed empirically to evaluate statistical significance of patterns in the feature space. It accesses only a small portion of the exponential search space, and groups candidate subgraphs into sets based on their similarity. So, GraphSig uses feature vector for graph representation. In GraphSig algorithm different chemical compound data set used for evolution of performance.

2.2.7. RP-GD algorithm. RP-GD [14] developed by Jianzhong Li, Yong Liu, and Hong Gao in 2011. It is mines a representative set from graph databases directly. RP-GD uses delta-jump pattern technique. RP-GD mines closed frequent subgraph patterns. It is used rightmost extension and DFS search strategy. RP-GD uses adjacency list for graph representation. In RP-GD algorithm PTE contain 340 chemical compound and AIDS data set used for evolution of performance.

2.2.8. RP-FP algorithm. RP-FP [14] developed by Jianzhong Li, Yong Liu, and Hong Gao in 2011. RP-FP derives a representative set from frequent closed subgraphs. RP-FP uses jump pattern. For find jump pattern, RP-FP algorithm uses close Graph algorithm to mine closed frequent sub graph after getting closed frequent subgraph, using substantial sub graph isomorphism testing, it find jump pattern. It is also used rightmost extension and DFS search strategy. RP-FP uses adjacency list for graph representation. In RP-FP algorithm PTE contain 340 chemical compound and AIDS data set used for evolution of performance.

2.2.9. TSP algorithm. TSP (temporal subgraph pattern mining) [15] developed by Hsun-Ping Hsieh,Cheng-Te Li in 2010. TSP used to mine the patterns which contain temporal information and forms a connective subgraph. TSP-algorithm only needs to scan the database once and does not generate unnecessary candidates. In TSP algorithm used projected database, concatenation function, super pattern, forward unnecessary checking scheme, backward unnecessary checking scheme for find out closed frequent temporal subgraph. It is used extension and DFS search strategy. TSP algorithm uses adjacency list for graph representation.

Based on analysis of above all algorithms, summarize two tables which give information about input, output and graph representation of algorithms.

Table 1. Apriory approach based algorithm

Algorithm	Input	Output	Graph Representation
AGM	Graph database	Frequent subgraph	Adjacency matrix
FSG	Set of graphs	Frequent subgraph	Adjacency list
FFSM	Set of graphs	Frequent subgraph	Adjacency matrix
SPIN	Set of graphs	Maximal Frequent subgraph	Adjacency matrix
GREW	Large Graph	Maximal Frequent subgraph	Sparse graph
Dynamic GREW	Dynamic graphs	Dynamic Frequent subgraph	Sparse graph

Table 2. PatternGrowth approach based algorithm

Algorithm	Input	Output	Graph Representation
Gspan	Set of graphs	Frequent subgraph	Adjacency list
CloseGraph	Set of graphs	closed Frequent subgraph	Adjacency list
Gaston	Set of graphs	maximal Frequent Subgraph	Hash table
Subdue	Large graph	Frequent subgraph	Adjacency matrix
RING	Set of graphs	Representative graph	Adjacency matrix
GraphSig	Set of graphs	Frequent pattern	Feature Vector
RP-GD	Set of graphs	Representative graph	Adjacency list
GP-FP	Set of graphs	Representative graph	Adjacency list
TSP	Set of graphs	Closed temporal frequent subgraph	Adjacency list

3. Conclusion

Graph mining algorithms generally concentrate on frequent substructures in all domains. For graph based methods, the goal is to find those parts of the graph

which have more frequent. In this paper we have studied different graph algorithms for finding frequent subgraph patterns. From this paper it is clear that algorithms used Pattern growth approach are more efficient in case of time complexity than algorithms used apriory based approach. Reducing the number of graph isomorphism is a promising direction which saves computational time.

4. References

[1] Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques," 2nd Edition.

[2] A. Inokuchi, T. Washio, and H. Motoda, "An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data," Proc. Fourth European Conf. Principles of Data Mining and Knowledge Discovery (PKDD), pp. 13-23, 2000.

[3]M. Kuramochi and G. Karypis, "Frequent Subgraph Discovery," Proc. IEEE Int'l Conf. Data Mining (ICDM), pp. 313-320, 2001.

[4] Jun Huan, Wei Wang, Jan Prins,"Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism".

[5] Jun Huan, Wei Wang, Jan Prins,Jiong Yang, "SPIN: Mining Maximal Frequent Subgraphs from Graph Databases",KDD'04, August 22–25, 2004.

[6] Michihiro Kuramochi and George Karypis,"GREW—A Scalable Frequent Subgraph Discovery Algorithm",2004.

[7] Karsten M. Borgwardt, Hans-Peter Kriegel, Peter Wackersreuther,"Pattern Mining in Frequent Dynamic Subgraphs",ICDM 2006.

[8] Xifeng Yan, Jiawei Han,"gSpan: Graph-Based Substructure Pattern Mining", Proc. 2nd IEEE Int'1 Conf,2002.

[9] Xifeng Yan, Jiawei Han, "CloseGraph: Mining Closed Frequent Graph Patterns",Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 286-295, 2003.

[10] Siegfried Nijssen and Joost N. Kok, "The Gaston Tool for Frequent Subgraph Mining",GraBaTs'04 Preliminary Version,2004.

[11] Nikhil S. Ketkar, Lawrence B. Holder, Diane J. Cook,"Subdue: CompressionBased Frequent Pattern Discovery in Graph Data",2005.

[12] Shijie Zhang , Jiong Yang , Shirong Li,"RING: An Integrated Method for Frequent Representative Subgraph Mining",In the proceedings of Ninth IEEE International Conference on Data Mining 2009.

[13] Sayan Ranu , Ambuj K. Singh,"GraphSig: A Scalable Approach to Mining Significant Subgraphs in Large Graph Databases", In the proceedings IEEE International Conference on Data Engineering 2009.

[14] Jianzhong Li, Yong Liu, and Hong Gao, "Efficient Algorithms for Summarizing Graph Patterns",IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 9, September 2011.

[15] Hsun-Ping Hsieh,Cheng-Te Li, "Mining Temporal Subgraph Patterns in Heterogeneous Information Networks" IEEE International Conference on Social Computing / IEEE International Conference on Privacy, Security, Risk and Trust, 2010.