[InnoSpace-2017: Special Edition]

Volume 4, Issue 1, January 2017, pp. 41-45 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

A Comparative Study of Discovering Frequent Subgraphs – Approaches and Techniques

B.Senthilkumaran[1], Dr.K.Thangadurai[2]

[1] Ph.D Research Scholar (Full Time)

[2]Assistant Professor and Head,

P.G. and Research,Department of Computer Science,

Government Arts College (Autonomous), Karur-05.

Email_ID: skumaran.gac16@gmail.com , ktramprasad04@gmail.com

Abstract :-Graph mining is an important research vertical and recently the usage of graphs has become increasingly imperative in modeling problematic complex structures such as electrical circuits, chemical compounds, protein structures, bioinformatics, social networks, workflow diagrams, and XML documents. Plethora of graph mining algorithms has been developed and the primary objective of this paper is to present a detailed survey regarding the approaches and techniques employed to find the issues and complexities involved.

Keywords: Graph, Mining, complex structure, techniques, modelling

1. Introduction

The foremost aim of data mining is to extract and unearth useful hidden knowledge from data [1]. The data extracted can take various forms like vectors, tables, texts, and images and the data can be represented by various means. Structured data and semi-structured data quite naturally suits graphical representation. Since the structured and semi-structured data can be represented without any complexity in graph formats, the graph mining has become a popular research domain.

For example, consider protein-protein structure which can be represented in a graph format in such a way that the vertexes denote genes and edges signify physical interactions or functional associations between them [2]. Usually there are two approaches of measuring similarity between graphs. One approach is to execute a pair wise comparison of the nodes in two networks, and compute an overall similarity score for the two networks. This approach takes time depending upon the number of nodes and edges, and this approach is practically feasible for large graphs. However, this approach has a snag that it totally evades the structure of the networks by treating them as sets of nodes and edges rather than graphs. To overcome this snag, it is imperative to treat two networks similar if they share many common sub-graphs. To treat two networks similar, sub-graph isomorphism problem called NP-complete should be computed. However the computational overhead or cost increases steeply and limits this approach to be employed in small networks. Many heuristics have been developed to increase the speed of calculating sub graph isomorphism by using special canonical labeling of the graphs.

The pioneer in graph mining is frequent sub-graph mining (FSM). The foremost objective of FSM is to discover all frequent sub-graphs in a given dataset whose occurrence is above the threshold count value provided.

The basic methodology behind FSM is to generate candidates (sub-graph candidates) in either a breadth first or depth first manner and determine if the generated candidate sub-graphs occur frequently above the threshold support count provided.

As far as the FSM is considered two important issues have to be handled efficiently (i) discover the candidate frequent sub-graphs and (ii) determine the frequency count of the generated sub-graphs. Here care has to be taken to avoid the generation of duplicate or superfluous candidates.

Support count checking requires repetitive comparison of candidate sub-graphs with sub-graphs in the input data and FSM can be considered as an extension of Frequent Itemset Mining (FIM) popularized in the context of association rule mining. Many researchers proposed solutions to address the issues related to FSM and downward closure property associated with itemset mining is widely adopted for candidate sub-graph generation. This paper deals with many state of the art FSM based algorithm employing different techniques with respect to candidate generation, different support counting process and different mechanism for traversing search space.

2. Preliminaries

FSM can be used in two different graphs, (1) Graph transaction based FSM – Here the input data consists of a collection of graphs (2) Single graph based FSM – Here the input is a single large graph. A sub-graph ϑ is considered to be frequent if the support count is larger than the predefined threshold support count provided by the user. The support of sub-graph ϑ is calculated by either transaction based count or by occurrence based count. In transaction based count, the support is defined by the number of transactions in which the sub-graph ϑ occurs (i.e.) one count per transaction.

Consider the given database $G = \{ G_1, G_2, G_3, \ldots G_N \}$ Where G1,G2,G3 are collection of graph transactions, the support threshold σ $(0 < \sigma \leq 1)$. Then the support of ϑ is

$$\mathrm{Sup}(\vartheta) = |\,\delta(\vartheta)\,| \,/\, N$$

Where $|\,\delta(\vartheta)\,|$ is cardinality of $\delta(\vartheta)$ and N is number of graphs in the database. Here ϑ is frequent, if $\mathrm{Sup}(\vartheta) \geq \sigma$.

The transaction based count utilizes the downward closure property (if a graph is frequent then all sub-graphs will be frequent) to reduce the excess candidate generation overhead and thereby it reduces the memory prints and the execution time considerably.

3. Labeled Graph

A labeled graph can be represented as G (V, E, Lv, Le, δ) , where V is set of vertexes, $E \subseteq V \times V$ is set of edges, Lv and Le is set of vertex and edge labels. δ is label function that denotes the mapping of $V \rightarrow Lv$ and $E \rightarrow Le$.

4. Subgraph

Consider two graphs G_1 $(V_1, E_1, Lv_1, Le_1, \delta_1)$ and G_2 $(V_2, E_2, Lv_2, Le_2, \delta_2)$ where G_1 is sub-graph of G_2 , if G_1 satisfies $V_1 \subseteq V_2$ and $\forall v \in V_1$, $\delta_1 (v) = \delta_2 (v)$. similarly $E_1 \subseteq E_2,$ $\forall (u,v) \in E_1$, $\delta_1 (u,v) = \delta_2 (u,v)$.

G_1 is an induced sub-graph of G_2, if G_1 satisfies $\forall (u,v) \in V_1$, $(u,v) \in E_1 \Leftrightarrow (u,v) \in E_2$

5. Survey Of FSM Algorithms

The frequent sub-graph mining issue has been addressed from many perspectives using approaches like a apriori and pattern growth. The existing algorithm varies with the type of input, search mechanism they utilize and method of representation of graphs. In this paper, we present a comparative survey based on apriori after analyzing various properties and limitations of these algorithms to obtain clear cut knowledge.

6. Algorithms Based On Apriori Approach

The FARMER algorithm [3] uses trie for input graph and uses level-wise search to generate candidates, the FARMER discovers a sub-graph and computes the instances of the sub-graph by one adjacent edge in all possible ways. FARMER utilizes this approach for sub-graph generation. FARMER algorithm is an enhanced version of WARMR, an earlier developed algorithm which works on the basis of ILP approach.

The HSIGRAM algorithm [4] employs adjacency matrix representation of graph. HSIGRAM use iterative merging for sub-graph generation and employs BFS strategy. The main purpose of HSIGRAM is to find the maximal independent set of a graph which is constructed by embedding the frequent sub-graphs and after calculating the frequency count.

The AGM algorithm [5] uses a vertex-based candidate generation approach that during each iteration substructure size is increased by one vertex. Two size-k frequent graphs are joined only when the two graphs have the same size (k - 1). ASM assumes that all vertexes in the graph are distinct. A more efficient version of AGM called AcGM is proposed by Inokuchi in the year 2002 [11] to mine only the frequent connected sub-graphs and the experimental results showcased that AcGM is considerably faster than AGM.

Huan, wang and Prince [6] in the year 2003 proposed a new sub-graph mining algorithm named FFSM, which uses a vertical search mechanism within an algebraic graph framework and restricted join operation to generate candidates to evade sub-graph isomorphism. To count the frequency it uses a sub-optimal canonical adjacency matrix tree. The FFSM is executed on real and synthetic datasets and obtained superior performance when compared with gSpan.

In large graph databases, the total number of frequent sub-graphs can become too large and the computational cost incurred will be huge. And in order to overcome this difficulty Jun Huan, Wei WangPrins, Jiong Yang, Jan [7] proposed an algorithm named SPIN that unearths only maximal frequent sub-graphs, (i.e.) sub-graphs that are not a part of any other frequent sub-graphs. This mechanism substantially decreases the size of the frequent sub-graphs. Initially the SPIN generates all frequent trees from a large graph database and then extracts only the maximal sub-graphs. SPIN performed quite well in large database and provided excellent scalability and efficiency when executed on chemical datasets.

Michihiro Kuramochi and George Karypis [8] in the year 2004 proposed a novel algorithm named GREW to overcome the limitations of existing complete or heuristic frequent sub-graph discovery algorithms. GREW is specially designed and developed to execute on a large graph datasets and to discover patterns corresponding to connected sub-graphs that have a large number of vertex-disjoint embeddings. This algorithm is inexact search based FGM algorithm. The experimental evaluation showed that GREW is efficient and can scale to very large graphs effectively. Another inexact search based algorithm is RAM proposed by Zhang & Yang in the year 2008 and RAM is experimentally proved that it discovers some important patterns that no exact search algorithms can perform.

Table 1: Apriori based algorithms

Algorithm	**Input type**	**Representation**	**Candidate generation**	**Support computation**	**Output**	**Limitation**
FARMER [1999]	Graph set	Trie structure	Level wise	Trie data structure	Frequent sub-graphs	Inefficient
HSIGRAM [1999]	Single graph	Adjacency matrix	Iterative merging	Max independent set	Frequent sub-graphs	Inefficient
AGM [2000]	Graph database	Adjacency matrix	Vertex extension	Canonical labeling	Frequent sub-graphs	NP- complete
FSG [2001]	Set of graphs	Adjacency list	One edge extension	Transaction identifier (TID) lists	Frequent connected sub-graphs	Largely distinct labels on edges needed
FFSM [2003]	Set of graphs	Adjacency matrix	Merging and extension	Suboptimal CAM tree	Frequent sub-graphs	Np-complete
SPIN [2004]	Set of graphs	Adjacency matrix	Join Operation	Canonical Spanning Tree	Maximal frequent sub-graphs	Needs entire DB scan
GREW [2004]	Single large graph	Sparse graph representation	Iterative merging	Maximal independent set	Maximal frequent sub-graphs	Misses many interesting patterns
Dynamic GREW [2005]	Dynamic graphs	Sparse graph representation	Iterative merging	Suffix trees	Dynamic patterns in frequent sub-graphs.	Extra overhead to identify dynamic patterns
MUSE [2009]	Uncertai n set of graphs	Adjaceny Matrix	Disjunctiv e normal forms	DFS coding scheme	Frequent sub-graphs	Frequent sub-graphs are not exact.

Karsten M. Borgwardt, Hans-Peter Kriegel explored the possibility on how pattern mining on static graphs can be extended to time series of graphs (dynamic graphs). They proposed a new technique in which the existing sub-graph mining algorithms can be easily integrated to handle dynamic graphs without any hurdles. The experimental executions on real-world data confirmed the practical feasibility of their approach when executed on dynamic graph datasets.

Mining frequent sub-graphs from uncertain graph data is a tedious and a serious problem to be explored and Zhaonian Zou, Jianzhong Li, and Shuo Zhang [9] in the year 2010 came up with an algorithm MUSE for Mining Frequent Sub-graph Patterns from Uncertain Graph Data. The MUSE algorithm uses efficient methodologies to conclude whether a sub-graph pattern can be given as output and used new pruning technique to reduce the complexity of examining sub-graph patterns. Experimental results showcased that the algorithm is very efficient, accurate, and scalable for large uncertain graph databases.

Lini T Thomas Satyanarayana R Valluri Kamalakar Karlapalem [10] in the year 2006 proposed an algorithm named MARGIN that mines maximal frequent sub-graphs. MARGIN- Maximal frequent mining has sparked much interest as the size of the maximal frequent sub-graphs is much smaller to that of the set of actual frequent sub-graphs. The Margin algorithm generates the candidates efficiently and finds the maximal sub-graphs by a post-processing step. This performance of the MARGIN algorithm is 20 times faster than gSpan for certain datasets.

7. Discussion

A complete survey of the "state of the art" frequent sub-graph mining algorithms is presented in this paper. The most important issue regarding to the FSM algorithms are candidate generation and support computations Largely the characteristic feature of the mining algorithms presented in this survey is how they efficiently and effectively address candidate generation and support counting.

8. Candidate Generation

The candidate generation is the most important phase in frequent sub-graph mining. The primary issue here is to methodically generate candidate sub-graphs without redundancy. Most of the FSM algorithm employs different methods to generate candidates and they are illustrated clearly here.

(1) Level – wise join

Generally a (k + 1) sub-graph candidate is generated by combining two frequent k sub-graphs which share the same (k - 1) sub-graph. The main issue here is that one k sub-graph can have many k different (k - 1) sub-graphs and the joining operation tends to generate many redundant candidates and increases the size of the candidates hugely. Kuramochi & Karypis [8] addressed this issue by limiting the (k - 1) sub-graphs to the two (k - 1) sub-graphs with the smallest and the second smallest canonical labels. By carrying out this adapted join operation, the number of duplicate candidates generated was significantly reduced. Many algorithm [4], [5] used this approach for candidate generation.

(2) Rightmost path expansion

Rightmost path expansion is a most common candidate generation method used in graph mining, it generates (k + 1)sub trees from frequent k-sub trees by adding vertexes only to the rightmost path of the tree as shown in figure 1."RMB" denotes the rightmost branch, which is the path from the root to the rightmost leaf (k - 1), and a new vertex k is added by attaching it to any vertexes along the RMB.

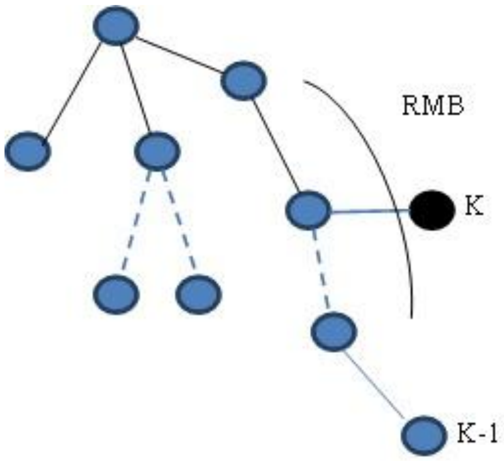

Figure 1: The rightmost path

(3) Extension and join

The extension and join approach was first proposed by Huan et al [12] in the year 2003, and later used by [Chi et al.] in the year 2004. The new approach employs a BFCS representation, whereby a leaf at the bottom level of a BFCF tree is defined as a "leg".

9. Conclusion

The FSM algorithms are extensively used in chemical and bio-informatics, though plethora of research work is carried out in this area, many important issues remains unaddressed. Instead of generating large frequent sub-graphs compact sub-graphs can be generated to avoid runtime expense and large memory prints. For example closed frequent sub-graphs, maximal frequent sub-graphs, approximate frequent sub-graphs and discriminative frequent sub-graphs can be discovered. Finally the exact frequent sub-graphs are not helpful in many real world circumstances and applications. Due to the ever increasing size and complexity of patterns in real world data, the need for an efficient graph mining algorithm is increasing with respect to speed and accuracy.

10. References

[1] Chen, M.S., Han,J.and Yu,P.S. 1996 Data mining – An overview from database perspective, IEEE *Transaction on knowledge and data engineering* 8 , 866-883

[2] Alm, E. and Arkin, A.P. 2003. Biological Networks, Current Opinion in Structural Biology 13(2), 193– 202.

[3] Nijssen, S. and Kok, J., Faster association rules for multiple relations. In IJCAI'01: Seventeenth International Joint Conference on Artificial Intelligence, 2001, vol. 2, pp. 891–896.

[4] Chuntao Jiang, Frans Coenen and Michele Zito, A Survey of Frequent Sub-graph Mining Algorithms:The Knowledge Engineering Review, Vol. 00:0, 1–31.c 2004.

[5] A. Inokuchi, T.Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In PKDD'00.

[6] J. Huan, W.Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. UNC computer science technique report TR03-021, 2003.

[7] J. Huan, W. Wang, J. Prins, and J. Yang. Spin: Mining maximal frequent sub-graphs from graph databases. UNC Technical Report TR04-018, 2004.

[8] M. Kuramochi and G. Karypis. Grew-a scalable frequent subgraph discovery algorithm. In ICDM, pages 439–442,2004.

[9] ZhaonianZou, Jianzhong Li, Hong Gao, and Shuo Zhang : Frequent Subgraph Patterns from Uncertain Graph Data. IEEE Transactions On Knowledge And Data Engineering, Vol. 22, No. 9, September 2010.

[10] L. T. Thomas, S. R. Valluri, and K. Karlapalem. Margin:Maximal frequent subgraph mining. Proc. 6th IEEE Int'l Conf. Data mining (ICDM '06), pp. 1097-1101, 2006.

[11] Inokuchi, A., Washio, T., Nishimura, K. and Motoda, H. 2002. A Fast Algorithm for Mining Frequent Connected Subgraphs, Technical Report RT0448, IBM Research, Tokyo Research Laboratory, Japan.

[12] Huan, J., Wang, W. and Prins, J. 2003. Efficient Mining of Frequent Subgraph in the Presence of Isomorphism, In Proceedings of the 2003 International Conference on Data Mining, 549-552.

About the Authors

B.Senthilkumaran is presently doing Ph.D in P.G. and Research, Department of Computer Science, at Government Arts College (Autonomous), Karur, Tamilnadu, India. He has received his M.Sc (Computer Science) degree from Government Arts College (Autonomous), Karur, Tamilnadu, India. He has published a number of papers in esteemed national/international conferences and journals. His interests are in Data Mining, Graph Mining, etc.,

Dr. K. Thangadurai, is presently working as Assistant professor and Head in P.G. and Research ,Department of Computer Science, Government Arts College (Autonomous), Karur. He has seventeen years of rich teaching experience with ten years of Research experience in the field of Computer Science. He was worked as the HOD of PG Department of Computer Science at Government Arts College (Men), Krishnagiri. He published many technical papers in National and International Conferences and Journals. His areas of interest are Software Engineering, Network Security, Data Mining, etc.,