International Journal of Computer Science and Business Informatics

IJCSBI.ORG

A Survey of Frequent Subgraphs and Subtree Mining Methods

Hamed Dinari and Hassan Naderi
Department of Computer Engineering
Iran University of Science and Technology
Tehran, Iran

ABSTRACT
A graph is a basic data structure which, can be used to model complex structures and the relationships between them, such as XML documents, social networks, communication networks, chemical informatics, biology networks, and structure of web pages. Frequent subgraph pattern mining is one of the most important fields in graph mining. In light of many applications for it, there are extensive researches in this area, such as analysis and processing of XML documents, documents clustering and classification, images and video indexing, graph indexing for graph querying, routing in computer networks, web links analysis, drugs design, and carcinogenesis. Several frequent pattern mining algorithms have been proposed in recent years and every day a new one is introduced. The fact that these algorithms use various methods on different datasets, patterns mining types, graph and tree representations, it is not easy to study them in terms of features and performance. This paper presents a brief report of an intensive investigation of actual frequent subgraphs and subtrees mining algorithms. The algorithms were also categorized based on different features.

Keywords
Graph Mining, Subgraph, Frequent Pattern, Graph indexing.

1. INTRODUCTION
Today we are faced with ever-increasing volumes of data. Most of these data naturally are of graph or tree structure. The process of extracting new and useful knowledge from graph data is known as graph mining [1] [2] Frequent subgraph patterns mining [3] is an important part of graph mining. It is defined as "process of pattern extraction from a database that the number frequency of which is greater than or equal to a threshold defined by the user." Due to its wide utilization in various fields, including social network analysis [4] [5] [6], XML documents clustering and classification [7] [8], network intrusion [9] [10], VLSI reverse [11], behavioral modeling [12], semantic web [13], graph indexing [14] [15] [16] [17] [18], web logs analysis[19], links analysis[20], drug design [21] [22] [23], and Classification of chemical compounds[24] [25] [26], this field has been subject matter of several works.

The present paper is an attempt to survey subtree and subgraph mining algorithms. A comparison and classification of these algorithms, according to their different features, is also made. The next section discusses the literature review followed by section three that deals with the basic ideas and concepts of graphs and trees. Mining algorithms, frequent subgraphs are discussed in section four from different viewpoint such as criteria of representing graphs (adjacency matrix and adjacency list), generation of subgraphs, number of replications, pattern growth-based and apriori-based classifications, classification based on search method, classification based on transactional and single inputs, classification based on type of output, and also Mining based on the logic. Fifth section focuses on frequent Mining algorithm from different angles such as trees representation method, type of algorithms input, tree-based Mining, and Mining based on Constraints on outputs.

2. RELATED WORKS

H.J.Patel1, R.Prajapati,et al. [27] Classified graph mining and mentioned two types of the algorithms, apriori-based and pattern growth based. K.Lakshmi1,T.Meyyappan [28] studied apriori based and pattern growth based, taking into account aspects such as input/output type, how to display a graph, how to generate candidates, and how many times a candidates is repeated in the graph dataset. In [29] D.Kavitha, B.V.Manikyala, et al. suggested the third type of graph mining algorithms named as inductive logic programming. Here a complete survey of graph mining concepts and a very useful set of examples to ease the understanding of the concept come next.

3. BASIC CONCEPTS

3.1 Garph

A graph G (V, E) is composed of a set of vertices (V) connected to each other by and a set of edges (E).

3.2 Tree

A tree T is a connected graph that has no cycle. In other words, there is only and only one path between any two vertices.

3.3 Subgraph

A **subgraph** G '(V', E') is a subgraph of G (V, E), which vertices and edges are subsets of V and E respectively:

- $V' \subseteq V$
- $E' \subseteq E \wedge ((v_1, v_2) \in E' \rightarrow v_1, v_2 \in V')$

One may say that a subgraph of a graph is a pattern of that graph. Concerning trees two types of patterns can be defined:

3.3.1 Induced pattern

The definition is exactly the same as the definition of subtree in a tree (Figure.1.a, Figure.1.c). It means that the vertices and the edges of Figure.1.a. Can be seen in Figure.1.c as well

3.3.2 Embedded pattern:

Almost the same as induced pattern, except that there may be one or more supplementary vertices between the two parents and child nodes of pattern, For example vertex **A** in Figure.1.c is parent of vertex D; and in Figure.1.b an embedded pattern of Figure 1.c is seen.

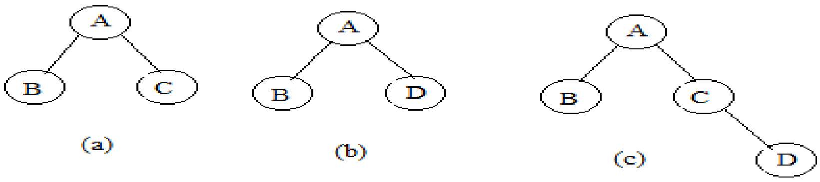


Figure.1. An example of the Induced and embedded subtree pattern

3.3.3 Isomorphism

Two graphs are **isomorph**, if there are one to one relationships among their vertices and edges.

3.3.4 Frequent Subgraph

Suppose a graph G and a set of graphs D = {g1, g2, g3,…, gn} are given, support(G) is:

$$Support\ (G) = \frac{\textbf{\textit{number of graphs in D that contain G}}}{|\boldsymbol{D}|}$$

A graph G in a dataset D is called **Frequent** if its support is not less than of a predefined threshold.

4. AN OVERVIEW OF FREQUENT SUBGRAPH MINING ALGORITHM ACCORDING TO DIFFERENT CRITERIA

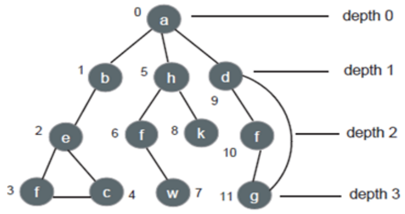
This section discusses different criteria for classification of frequent graph mining algorithms, including: graph representation, input type, constraint-based, inductive logic programming, search strategy, and completeness/-incompleteness of outputs.

4.1 Graph Representation

4.1.1 Adjacency Matrix

A graph can be demonstrated as an adjacency matrix, in this case the row and the column represent vertex of graph and the entries represents edges of graph (i.e. when there is an edge between two vertices, entries constituted by the junction of the row and the column is filled by “1” and otherwise by

”0”). Furthermore, the nodes are represented on the main diameter of the matrix (Figure.2).To Show the graph as a string a combination of nodes and edges as a sequence in particular order can be used, and since every permutation of the nodes may generate a specific string, a series of maximum or minimum canonical adjacency matrix (CAM) must be taken into account. An advantage of this is that two isomorphism graphs will have the same maximum/minimum CAM.



$$\begin{bmatrix} a & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & b & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & d & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & e & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & f & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & g & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & h & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & k & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & w \end{bmatrix}$$

Figure.2. Left side a graph and right side corresponding adjacency matrix

4.1.2 Adjacency List

Another way to represent a graph is adjacency list. When the graph is sparse, several “zeros” are generated in the adjacency matrix, which is a great waste of memory and to avoid this, adjacency list is an answer as it assigns memory dynamically

4.2 Subgraph Generation

Two subgraphs can be mixed to generate candidate subgraph and the result will be a new subgraph. However, given that many copied subgraphs might be generated in the mixing process, the way of generating candid subgraphs is critical. Among the available methods are extension and right most expansion. In the latter case, the subgraphs are expanded in one direction and no duplicate candidate is generated.

4.3 Frequency Counting

To check if the generated candidates is a duplicate or not, the frequency of each must be determined and compared with the support value. Of the data structures, which are used to count the frequency of each candidate are embedding list and TSP tree.

5. A SURVEY OF FREQUENT SUBGRAPH MINING ALGORITHMS

5.1 Classification Based on Algorithmic Approach

5.1.1 Apriori-Based (Breadth First Search)

This category of algorithms uses generates and test method and surface search to find a subgraph from the network that consist the database. Therefore, before the subgraph with length of k +1 ((k+1)-candidate), all frequent subgraphs with length of k must be found. Thus, each candidate with length of k +1 is obtained by connecting two frequent subgraphs with length of k. However, in this method, all state of candidate subgraph generated is considered .Maintenance and processing need plenty of time and memory, which tackles the performance [30] [2].

5.1.2 Pattern Growth-Based

In FP-growth-based methods a candidate subgraph with length of k+1 is obtained by extending a frequent pattern with length of *k*. Since extending a frequent subgraph with length of *k* may generate several candidate of length k+1, thus the way a frequent subgraph is expanded is critical in reducing generation of copied subgraphs.Table1 lists apriori and pattern growth algorithms [2].

Table 1. Frequent Subgraph Mining Algorithms

Pattern Growth	Apriori
GSPan [41]	FARMER [31]
CloseGraph [42]	FSG [3]
Gaston [43]	HSIGRAM
TSP [44]	GREW [32]
MOFa [45]	FFSM [4]
RP-FP [46]	ISG
RP-GD [46]	SPIN [33]
JPMiner [47]	Dynamic GREW [34]
MSPAN	AGM [35]
VSIGRAM [48]	MUSE [36]
FPF [49]	SUBDUE [37]
Gapprox [50]	AcGM [38]
HybridGMiner	DPMine
FCPMiner [51]	gFSG [39]
RING [52]	MARGIN [40]
SCMiner [53]	
GraphSig [54]	
FP-GraphMiner [55]	
gPrune [56]	
CLOSECUT [57]	
FSMA [58]	

5.2 Classification Based on Search Strategy

There are two search strategies to find frequent subgraphs. These two methods include breadth first search (BFS) and depth first search (DFS).

5.3 Classification Based on Nature of the Input

Depending on input type of algorithms, here tried to be divided two categories presented as following:

5.3.1 Single Graph Database

Database consists of a single large graph

5.3.2 Transactional Graph Database

Database consists of a large number of small graphs. Figure.3 shows a database consist of a set of graphs and two subgraphs and their frequency. In Figure.3 (left side, g, g2, and g2) demonstrates a transactional graph database and frequency of two frequent subgraphs (right side).

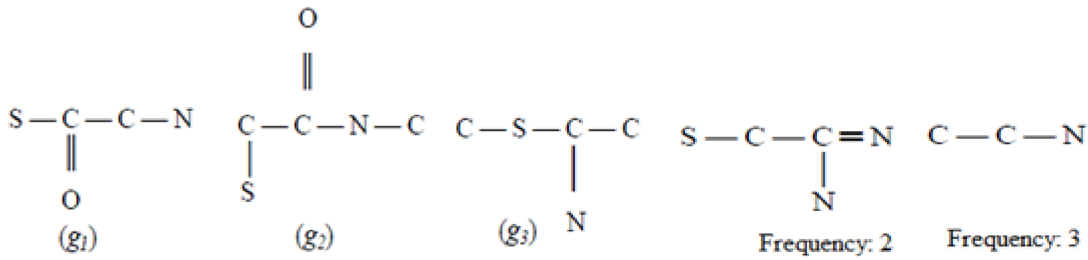


Figure.3. A database consisting of three graph g1, g2, g3 and two subgraph and frequency of each

5.4 Classification Based on Nature of the Output

5.4.1 Completeness of the Output

While, some algorithms find all frequent patterns, some other only mines part of frequent patterns. Frequent patterns mining is closely related to performance. When the total size of dataset is too high, it is better to use algorithms that are faster in execution so that reduction of the performance is avoided, although, not all frequent patterns are minded. Table 2 lists the completeness of output [29].

Table 2. Completeness of Output

Incomplete Output	Complete Output
SUBDUE	FARMER
GREW	gSpan
CloseGraph	FFSM
ISG	Gaston
	FSG
	HSIGRAM

5.4.2 Constraint-Based

With increase of size database, the number of frequent pattern is increased. This makes maintenance and analyzing more difficult as it needs more memory space. Reducing the number of frequent patterns without losing the data is achievable through mining and maintains more comprehensive patterns. Given that each pattern satisfies the condition of being frequent the whole subset satisfies the condition, to achieve more comprehend patterns we can use the following terms:

5.4.2.1 Maximal Pattern

Subgraph *g1* is maximal pattern if the pattern is frequent and does not consist of any super-pattern, so that g2 با g2⊃g1.

5.4.2.2 Closed Pattern

Subgraph g1 is closed if it is frequent and does not consist of any frequent super-pattern such as g2, g2⊃g1 (i.e. support).Table3 lists maximal and closed subgraph algorithms.

Table 3. Frequent Subgraph Mining (Constraintd)

Closed	Maximal
CloseGraph	SPIN
CLOSECUT	MARGIN
TSP	ISG
RP-FP	GREW
RP-GD	

5.5 Logic-Based Mining

Also known as inductive logic programming, which also an area of machine learning, mainly in biology. This method uses inductive logic to display structured data. ILP core uses the logic to display for search and the basic assumptions of that structured way (e.g. WARMR, FOIL, and C-PROGOL), which is derived from background knowledge [29]. Table 4 lists the Pattern Growth and Table 5 indicates apriori-based algorithms categorized from different aspects [59] [27] [60] [61] [62] [28] [30] [63].

Table 4. Frequent Subgraph Mining Algorithms (Pattern Growth-based)

Algorithms	Input Type	Graph Representation	Subgraph Generation	Frequency Counting

GSpan	Set of graphs	Adjacence Matrix	Rightmost Extension	DFS
CloseGeaph	Set of graphs	Adjacence Matrix	Rightmost Extension	DFS
Gaston	Set of graphs	Hash Table	Extension	DFS
TSP	Set of graphs	Adjacence Matrix	Extension	TSP Tree
MOFA	Set of graphs	Adjacence Matrix	Rightmost Extension	DFS
RP-FP	Set of graphs	Adjacence Matrix	Rightmost Extension	DFS
RP-GD	Set of graphs	Adjacence Matrix	Rightmost Extension	DFS
JPMiner	Set of graphs	Adjacence Matrix	Rightmost Extension	DFS
MSPAN	Set of graphs	Adjacence Matrix	Rightmost Extension	DFS
FP-Graph-Miner	Set of graphs	BitCode	Extension	DFS
gPrune	Set of graphs	Adjacency matrix	Iteration	M-DFSC
FSMA	Set of graphs	incident matrix	Extension	Normalize Matrix
RING	Set of graphs	Invariant vector	Extension	R-tree, DFS
GraphSig	Set of graphs	Feature vector	Merge and Extension	DFS

Table 5. Frequent Subgraph Mining Algorithms (Apriori-based)

Algorithms	Input Type	Graph Representation	Subgraph Generation	Frequency Counting
SUBDUE	Single Large Graph	Adjacence Matrix	Level-wise Search	MDFS
FARMER	Set of graphs	Trie structure	Level-wise Search, ILP	Trie data structure
FSG	Set of graphs	Adjacency List	One Edge Extension	TID list
HSIGRAM	Single large Graph	Adjacency Matrix	Iterative Merging	Maximal independent set
GREW	Single large Graph	Sparse graph	Iterative Merging	Maximal independent set
FFSM	Set of graphs	Adjacency Matrix	Merging and Extension	Suboptimal canonical adjacency matrix tree
ISG	Set of graphs	Edge Triple	Edge Triple Extension	TID List
SPIN	Set of graphs	Adjacency Matrix	Join Operation	Canonical Spanning Tree
Dynamic GREW	Set of graphs	Sparse graph	Iterative Merging	Suffix trees
AGM	Set of graphs	Adjacency Matrix	Vertex Extension	Canonical Labeling
MUSE	Set of graphs	Search Tee	Disjunctive Normal Form	DFS coding
MARGIN	Set of graphs	Lattice	Join	CAM
AcGM	Set of graphs	Adjacency Matrix	Join	CAM
gFSG	Set of graphs	Adjacency Matrix	Iterative Merging	Hashtree

Here several algorithms related to graph/tree mining are discussed in more details.

- ***Gp-Growth Algorithm***

The algorithm consists of three main steps:

1. Candidate generation by join operation.

2. Using a new method for tree representation and look up table that allows quick access to the information nodes in the candidate generation phase without having to read the trees of the database.

3. using right most expansion to candidate generation that guaranteed not generate duplicate candidate.

This algorithm uses lookup table that is implemented as Hash table to store input trees information. It is the key part, represented as the pair of (T,pos), where T is identification of input tree and *pos* is number in preorder traversal, and value part, represented as (l,s), where l is label and *s* is scope of node. In this algorithm a new candidate is generated using scope of each node That means, first node, which is added to the other node should be added along the right most expansion and that within the scope of the first node to be added continually this process other frequent pattern is found [64].

- ***Fp-Graph Miner Algorithm***

This algorithm uses FP-growth method to find frequent subgraphs, Its input is a set of graphs (Transactional database). First a BitCode for each edge is defined, then a set of edge is defined for each edge .When, edge is found in the each of graphs, the BitCode is '1' and otherwise '0'. Then a frequency table is sorted in ascending order based on equivalent BitCode belongs to each edge and afterward, FP tree is constructed and frequent subgraphs are obtained through depth traversal [55].

6. FREQUENT SUBTREES MINING ALGORITHMS CLASSIFICATION

6.1 Trees Representation

A tree can be encoded as a sequence of nodes and edges. Some of most important ways of encoding trees are introduced below:

6.1.1 DLS (Depth Label Sequence)

Let T be a Labeled Ordered Tree and depth-label pairs including labels and depth for each node are belonged to V. For example, (d(vi),l(vi)) are added to string *s* throughout DFS traversal of tree *T*. Depth-label sequence of tree *T* is obtained as { d(v1), l(v1)), …,(d(vk), l(vk) }. For instance, DLS for tree in **Figure**.4 can be presented as follow:
{(0,a),(1,b),(2,e),(3,a),(1,c),(2,f),(3,b),(3,d),(2,a),(1,d),(2,f)(3,c)}

6.1.2 DFS – LS (Depth First Sequence)-(Label Sequence)

Assumed a labeled ordered tree, Labels $\forall vi \in V$ is added to string of *s* during the DFS traversal of Tree *T*. During backtrack '-1'or'$'or '/' is added to the string, DFS-LS code for tree *T* is illustrated in in Figure.4 {abea$$$cfb$d$$a$$dfc$$$}

6.1.3 BFCS (Breadth First Canonical String)

Let T be an unordered tree. Several sequence encoded string can be generated using the BFS method and through changing the order of children of a node. Thus, one may say that BFCS tree T equals to the smallest lexicographic order of this encoded string. BFCS of tree T is showed in Figure.4. {abcdefaf$a$bd$$c#}

6.1.4 CPS (Consolidate Prufer Sequence)

Let T be a labeled tree T, and CPS encoding method consists of two parts: NPS as extended prufer sequence, which uses vertex numbers traversal as set of unique label is obtained; an d LS (Label Sequence) as a sequence consisting of labels in prefix traversal after the leafs is removed is achieved. Both NPS and LS generate a unique encoding for labeled tree. NPS and LS obtained for the tree presented in Figure.4 is as follow respectively: {ebaffccafda-}, {aebbdfaccfda}. To obtain NPS, a leaf from the tree is removed in each step and the parent of the leaf is taken get as output. This is repeated until only the roots remain and '-' is added to note as the end of the string. Regarding LS (Label Sequence) the same postfix traversal of the tree is taken as LS. Table9 remarks this category of trees [65].

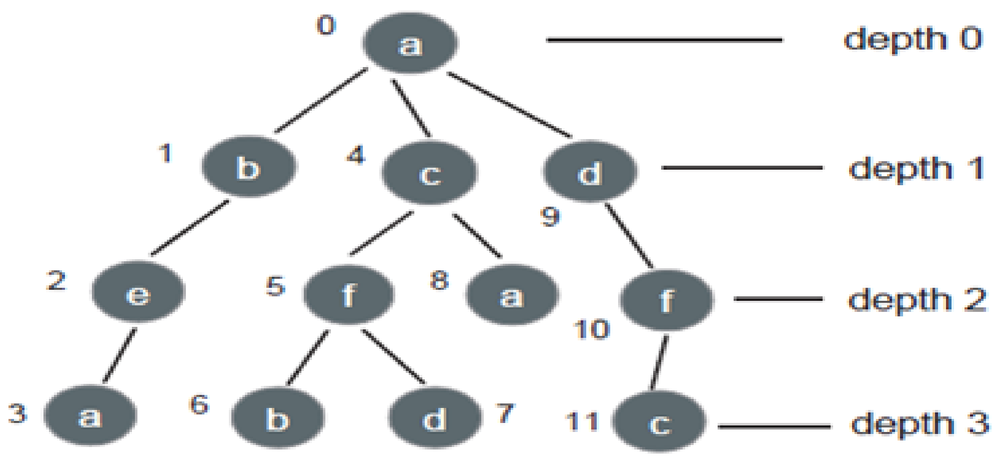


Figure.4. A Tree Example

Table 6. Frequent subtree Mining Algorithms (Tree Representation)

Algorithms	Tree Representation
uFreqt	DLS
SLEUTH	DFS-LS
Unot	DLS
Path Join	FST-Forest
RootedTreeMiner [66]	BFCS
FREQT	DLS
TreeMiner	DFS-LS
Chopper	DLS
XSPanner	DLS
AMIOT	DFS string
IMB3Miner	DFS-LS
TRIPS	CPS
FreeTreeMiner	BFCS
CMTreeMiner	DFS-LS
HybridTreeMiner [67]	BFCS
GP-Growth	DFS-LS

6.2 Input Types

6.2.1 Rooted Ordered Trees

Rooted Ordered sub-tree is a kind of tree in which a single node is considered as the "root" of the tree and there is a relationship between children of each node so that each child is greater than or equal to its siblings that are placed at the left hand side of it; moreover it is less than or equal to ones that are placed at its right hand side. If we elate or definition of rooted ordered tree such that there was no need to consider the relationship between siblings we have a rooted unordered sub-tree.in Table7 rooted ordered tree mining algorithms is shown.

Table 7. Rooted Ordered Tree mining Algorithms

Embedded	Induced
TreeMiner [71]	FREQT [68]
Chopper [72]	AMIOT [69]
XSPanner [72]	IMB3Miner [70]
IMB3-Miner	TRIPES [65]
	TIDES [65]

6.2.2 Rooted Unordered Trees

In this type of trees, a node is considered as the root, however, there is no particular order between the descendants of each node,In Table 8 rooted unordered tree mining algorithms is listed.

Table 8. Rooted unordered Tree mining Algorithms

Embedded	Induced
TreeFinder Cousin Pair [77] SLEUTH [78]	uFreqT [73] Unot [74] PathJoin [65] Rooted TreeMiner [75] TreeFinder [76]

6.3 Tree Base Data Mining

Frequent subtrees mining algorithm can be categorized into two major categories, aprior-based and pattern growth-based. Table 9 lists the apriori and pattern growth algorithms of trees [79] [76] [80].

Table 9. Frequent Subtree Mining Algorithms

Pattern Growth	Apriori
uFreqt Unot FREQT TRIPS TIDES Path Join XSPanner Chopper PrefixTreeISpan [86] PCITMiner [87] F3TM [88] GP-Growth [64]	TreeFinder AMIOT FreeTreeMiner TreeMine [81] SLEUTH CMTreeMiner [82] Pattern Matcher [71] W3Miner [83] FTMiner [84] CFFTree [85] IMB3-Miner

7. CONCLUSIONS AND FUTURE WORKS

Frequent subgraph Mining algorithms were first examined from different viewpoints such as different ways of representing a graph (e.g. adjacency matrix and adjacency list), generation of subgraphs, frequency counting, pattern growth-based and apriori-based algorithm classification, search based classification, input-based classification (single, transactional), output based classification. Furthermore, Mining based on logic was discussed. Afterward, frequent subtrees traversal algorithms were examined from different viewpoints such as trees representation methods, type of inputs, tree-based traversal, and also Mining based on Constraints of outputs. Given the results, it is concluded that in absence of generating patterns by pattern-growth, it is featured with less computation work and needs smaller memory size. Moreover, these algorithms are specifically designed for trees and graphs and cannot be used for other purposes. On the other hand, as they work on variety of datasets, it is not easy to find tradeoffs between them. The same frequent patterns can be used for searching similarity, indexing, classifying graphs and documents in future studies. Parallel methods and technologies such as Hadoop can also be needed when working with excessive data volume.

8. ACKNOWLEDGMENTS

Authors are thankful to Mohammad Reza Abbasifard for their support of the investigations.

REFERENCES

[1] A.Rajaraman, J.D.Ullman, 2012. *Mining of Massive Datasets*, 2nd ed.

[2] J.Han, M.Kamber, 2006, *Data Mining Concepts and Techniques*. USA: Diane Cerra.

[3] Kuramochi, Michihiro, and G.Karypis., 2004. An efficient algorithm for discovering frequent subgraphs, in *IEEE Transactions on Knowledge and Data Engineering*, pp. 1038-1051.

[4] J.Huan, W.Wang, J. Prins, 2003. Efficient Mining of Frequent Subgraph in the presence of isomorphism, in *Third IEEE International Conference on Data Minign (ICDM)*.

[5] (2013, Dec.) Trust Network Datasets - TrustLet. [Online]. http://www.trustlet.org

[6] L.YAN, J.WANG, 2011. Extracting regular behaviors from social media networks, in *Third International Conference on Multimedia Information Networking and Security*.

[7] Ivancsy,I. Renata, I.Vajk., 2009. Clustering XML documents using frequent subtrees, *Advances in Focused Retrieval*, Vol. 3, pp. 436-445.

[8] J.Yuan, X.Li, L.Ma, 2008. An Improved XML Document Clustering Using Path Features, in *Fifth International Conference on Fuzzy Systems and knowledge Discovery*, Vol. 2.

[9] Lee, Wenke, and Salvatore J. Stolfo, 2000. A framework for constructing features and models for intrusion detection systems, in *ACM transactions on Information and system security (TiSSEC)*, pp. 227-261.

[10] Ko, C, Logic induction of valid behavior specifications for intrusion detection , 2000. in *In IEEE Symposium on Security and Privacy (S&P)*, pp. 142–155.

[11] Yoshida, K. and Motoda, 1995. CLIP: Concept learning from inference patterns, in *Artificial Intelligence*, pp. 63–92.

[12] Wasserman, S., Faust, K., and Iacobucci. D, 1994. *Social network analysis : Methods and applications*. Cambridge university Press.

[13] Berendt, B., Hotho, A., and Stumme, G., 2002. semantic web mining, in *In Conference International Semantic Web (ISWC)*, pp. 264–278.

[14] S.C.Manekar, M.Narnaware, May 2013. Indexing Frequent Subgraphs in Large graph Database using Parallelization, *International Journal of Science and Research (IJSR)*, Vol. 2 , No. 5.

[15] Peng, Tao, et al., 2010. A Graph Indexing Approach for Content-Based Recommendation System, in *IEEE Second International Conference on Multimedia and Information Technology (MMIT)*, pp. 93-97.

[16] S.Sakr, E.Pardede, 2011. Graph Data Management: Techniques and Applications, in *Published in the United States of America by Information Science Reference*.

[17] Y.Xiaogang, T.Ye, P.Tao, C.Canfeng, M.Jian, 2010. Semantic-Based Graph Index for Mobile Photo Search," in *Second International Workshop on Education Technology and Computer Science*, pp. 193-197.

[18] Yildirim, Hilmi, and Mohammed Javeed Zaki., 2010. Graph indexing for reachability queries, in *26th International Conference on Data Engineering Workshops (ICDEW)IEEE*, pp. 321-324.

[19] R.Ivancsy and I.Vajk, 2006. Frequent Pattern Mining in Web Log Data, in *Acta Polytechnica Hungarica*, pp. 77-90.

[20] G.XU, Y.zhang, L.li, 2010. *Web mining and Social Networking*. melbourn: Springer.

[21] S.Ranu, A.K. Singh, 2010. Indexing and mining topological patterns for drug, in *ACM, Data mining and knowlodge discovery*, Berlin, Germany.

[22] (2013, Dec.) Drug Information Portal. [Online]. http://druginfo.nlm.nih.gov

[23] (2013, Dec.) DrugBank. [Online]. http://www.drugbank.ca

[24] Dehaspe,Toivonen, and King, R.D., 1998. Finding frequent substructures in chemical compounds, in *In Proc. of the 4th ACM International Conference on Knowledge Discovery and Data Mining*, pp.30-36.

[25] Kramer, S., De Raedt, L., and Helma, C., 2001. Molecular feature mining in HIV data, in *In Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-01)*, pp. 136–143.

[26] Gonzalez, J., Holder, L.B. and Cook, 2001. Application of graph-based concept learning to the predictive toxicology domain, in *In Proc. of the*

Predictive Toxicology Challenge Workshop.

[27] H.J.Patel, R.Prajapati, M.Panchal, M.Patel, Jan. 2013. A Survey of Graph Pattern Mining Algorithm and Techniques, *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Vol. 2, No. 1.

[28] K.Lakshmi, T. Meyyappan, 2012. FREQUENT SUBGRAPH MINING ALGORITHMS - A SURVEY AND FRAMEWORK FOR CLASSIFICATION, *computer science and information technology*, pp. 189–202.

[29] D.Kavitha, B.V.Manikyala Rao and V. Kishore Babu, 2011. A Survey on Assorted Approaches to Graph Data Mining, in *International Journal of Computer Applications*, pp. 43-46.

[30] C.C.Aggarwal,Wang, Haixun, 2010. *Managing and Mining Graph Data*. Springer,.

[31] B.Wackersreuther, Bianca, et al. , 2010. Frequent subgraph discovery in dynamic networks, in *ACM, Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, Washington DC USA, pp. 155-162.

[32] Kuramochi, Michihiro, and G.Karypis, 2004. Grew-a scalable frequent subgraph discovery algorithm, in *Fourth IEEE International Conference on Data Mining (ICDM)*, pp. 439-442.

[33] Huan, Jun, SPIN: mining maximal frequent subgraphs from graph databases, 2004. in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

[34] Borgwardt, Karsten M., H-P. Kriegel, and P.Wackersreuther, 2006. Pattern mining in frequent dynamic subgraphs, in *Sixth International Conference on Data Mining (ICDM)*, pp. 818-822.

[35] Inokuchi, Akihiro, T.Washio, and H.Motoda, 2000. An apriori-based algorithm for mining frequent substructures from graph data, in *Principles of Data Mining and Knowledge Discovery*, pp. 13-23, Springer Berlin Heidelberg.

[36] Zou, Zhaonian, et al, 2009. Frequent subgraph pattern mining on uncertain graph data, in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 583-592.

[37] Ketkar, N.S, Lawrence B.Holder, and D.J.Cook, 2005. Subdue: compression-based frequent pattern discovery in graph data, in *ACM, Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pp. 71-76.

[38] A. Inokuchi, T. Washio, and H. Motoda, 2003. Complete mining of frequent patterns from graphs: Mining graph data, in *Machine Learning*, pp. 321-354.

[39] Kuramochi, Michihiro, and G.Karypis, 2007. Discovering frequent geometric subgraphs, in *Information Systems*, pp. 1101-1120.

[40] Thomas, Lini T, Satyanarayana R. Valluri, and K.Karlapalem, 2006. Margin: Maximal frequent subgraph mining, in *IEEE Sixth International Conference on Data Mining (ICDM)*, pp. 1097-1101.

[41] Yan, Xifeng, and J.Han, 2002. gspan: Graph-based substructure pattern mining, in *Proceedings International Conference on Data Mining.IEEE*, pp. 721-724.

[42] Yan, Xifeng, and Jiawei Han, 2003. CloseGraph: mining closed frequent graph patterns, in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 286-295.

[43] Nijssen, Siegfried, and J.N. Kok., 2005. The gaston tool for frequent subgraph mining, in *Electronic Notes in Theoretical Computer Science*, pp. 77-87.

[44] Hsieh, Hsun-Ping, and Cheng-Te Li, 2010. Mining temporal subgraph patterns in heterogeneous information networks, in *IEEE Second International Conference on Social Computing (SocialCom)*, pp. 282-287.

[45] Wörlein, Marc, et al, 2005. A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston, in *Knowledge Discovery in Databases: PKDD* , Springer Berlin Heidelberg, pp. 392-403.

[46] S.J.Suryawanshi,S.M.Kamalapur, Mar 2013. Algorithms for Frequent Subgraph Mining, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, No. 3.

[47] Liu, Yong, Jianzhong Li, and Hong Gao, 2009. JPMiner: mining frequent jump patterns from graph databases, in *IEEE, Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 114-118.

[48] Reinhardt, Steve, and G.Karypis, 2007. A multi-level parallel implementation of a program for finding frequent patterns in a large sparse graph, in *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1-8.

[49] Schreiber, Falk, and H.Schwobbermeyer., 2005. Frequency concepts and pattern detection for the analysis of motifs in networks, in *Transactions on computational systems biology III*, pp. 89-104, Springer Berlin Heidelberg.

[50] Chent, Chen, et al., 2007. gapprox: Mining frequent approximate patterns from a massive network, in *Seventh IEEE International Conference on Data Mining (ICDM),* pp. 445-450.

[51] Ke, Yiping, J.Cheng, and Jeffrey Xu Yu, 2009. Efficient discovery of frequent correlated subgraph pairs, in *Ninth IEEE International Conference on Data Mining (ICDM)*, pp. 239-248.

[52] Zhang, Shijie, J.Yang, and Shirong Li, 2009. Ring: An integrated method for frequent representative subgraph mining, in *Ninth IEEE International Conference on Data Mining (ICDM)*, pp. 1082-1087.

[53] Fromont, Elisa, Céline Robardet, and A.Prado, 2009. Constraint-based subspace clustering, in *International conference on data mining*, pp. 26-37.

[54] Ranu, Sayan, and Ambuj K. Singh., 2009. Graphsig: A scalable approach to mining significant subgraphs in large graph databases, in *IEEE 25th International Conference on Data Engineering (ICDE)*, pp. 844-855.

[55] R. Vijayalakshmi,R. Nadarajan, J.F.Roddick,M. Thilaga, 2011. FP-GraphMiner, A Fast Frequent Pattern Mining Algorithm for Network Graphs, *Journal of Graph Algorithms and Applications*, Vol. 15, pp. 753-776.

[56] Zhu, Feida, et al., 2007. gPrune: a constraint pushing framework for graph pattern mining, in *Advances in Knowledge Discovery and Data Mining*, , pp. 388-400, Springer Berlin Heidelberg.

[57] Yan, Xifeng, X. Zhou, and Jiawei Han, 2005. Mining closed relational graphs with connectivity constraints, in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 324-333.

[58] Wu, Jia, and Ling Chen, 2008. A fast frequent subgraph mining algorithm, in *The 9th International Conference for Young Computer Scientists (ICYCS)*, pp. 82-87.

[59] Krishna, Varun, N. N. R. R. Suri, G. Athithan, 2011. A comparative survey of algorithms for frequent subgraph discovery, *Current Science(Bangalore)*, pp. 1980-1988.

[60] K.Lakshmi, T. Meyyappan, Apr. 2012. A COMPARATIVE STUDY OF FREQUENT SUBGRAPH MINING ALGORITHMS, *International Journal of Information Technology Convergence and Services (IJITCS)*, Vol. 2, No. 2.

[61] C.Jiang, F.Coenen, M.Zito, 2004. A Survey of Frequent Subgraph Mining Algorithms, *The Knowledge Engineering Review*, pp. 1-31.

[62] M.Gholami, A.Salajegheh, Sep. 2012. A Survey on Algorithms of Mining Frequent Subgraphs, *International Journal of Engineering Inventions*, Vol. 1, No. 5, pp. 60-63.

[63] V.Singh, D.Garg, Jul. 2011. Survey of Finding Frequent Patterns in Graph Mining: Algorithms and Techniques, *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 1, No. 3.

[64] Hussein, M.MA, T. H.Soliman, O.H. Karam, 2007. GP-Growth: A New Algorithm for Mining Frequent Embedded Subtrees. *12th IEEE Symposium on Computers and Communications*.

[65] Tatikonda, Shirish, S.Parthasarathy,T.Kurc., 2006. TRIPS and TIDES: new algorithms for tree mining, in *Proceedings of the 15th ACM international conference on Information and knowledge management*.

[66] Tung, Jiun-Hung, 2006. MINT: Mining Frequent Rooted Induced Unordered Tree without Candidate Generation.

[67] Chi, Yun, Y.Yang, and Richard R. Muntz., 2004. HybridTreeMiner: An efficient algorithm for mining frequent rooted trees and free trees using canonical forms, in *Proceedings 16th International Conference on Scientific and Statistical Database Management*.

[68] T.Asai, H.Arimura, T.Uno, S.Nakano and K.Satoh, 2008. Efficient tree mining using reverse search.

[69] S.Hido, and H. Kawano., 2005. AMIOT: Induced Ordered Tree Mining in Tree-structured Databases, in *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*.

[70] H.Tan, T.S. Dillon, F.Hadzic, E.Chang, and L.Feng, 2006. IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding, in *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, pp. 450–461.

[71] M.J.Zaki, 2002. Efficiently mining frequent trees in a forest, in *In Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, pp. 71-80.

[72] C.Wang, M.Hong, J.Pei, H.Zhou, W.Wang, 2004. Efficient pattern-growth methods for frequent tree pattern mining, in *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, pp. 441-451.

[73] S.Nijssen and J.N.Kok, 2003. Efficient Discovery of Frequent Unordered Trees, in *Proc. First Intl Workshop on Mining Graphs Trees and Sequences*, pp. 55-64.

[74] T. Asai, H. Arimura, T.Uno and S. Nakano., 2003. Discovering Frequent Substructures in Large Unordered Trees, in *procceding sixth conference on Discovery Science*, pp. 47-61.

[75] Y.Chi, Y.Yang, and R. Muntz., May 2004. Canonical Forms for Labeled Trees and Their Applications in Frequent Subtree Mining, *Knowledge and Information Systems*, No. 8.2, pp. 203-234.

[76] Chi, Yun, et al.,2005. Frequent subtree mining-an overview, in *Fundamenta Informaticae*, pp. 161-198.

[77] Shasha, Dennis, J.Tsong-Li Wang and Sen Zhang.,2004. Unordered tree mining with applications to phylogeny, in *IEEE Proceedings 20th International Conference on Data Engineering*, pp. 708-719.

[78] M.J.Zaki., 2005. Efficiently Mining Frequent Embedded Unordered Trees, in *IOS Press*, pp. 1-20.

[79] Jimenez, Aida, F.Berzal,J.Cubero.,2008. Mining induced and embedded subtrees in ordered, unordered, and partially-ordered trees, in *EEE Transactions on Knowledge and Data Engineering*, Springer Berlin Heidelberg, pp. 111-120.

[80] Jimenez, Aida,F. Berzal Juan-Carlos Cubero.,2006. Mining Different Kinds of Trees: A Tree Mining Overview, in *Data Mining*.

[81] B.Bringmann.,2004. Matching in Frequent Tree Discovery, in *Fourth IEEE International Conference on Data Mining*.

[82] Chi, Yun, et al. Mining.,2004. Cmtreeminer: Mining both closed and maximal frequent subtrees, in *Advances in Knowledge Discovery and Data* , Springer Berlin Heidelberg, pp. 63-73.

[83] AliMohammadzadeh, Rahman, et al., Aug 2006. Complete Discovery of Weighted Frequent Subtrees in Tree-Structured Datasets, *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 6, No. 8, pp. 188-196.

[84] J.HU, X.Y.LI., Mar 2009. Association Rules Mining Including Weak-Support Modes Using Novel Measures," *WSEAS Transactions on Computers*, Vol. 8, No. 3, pp. 559-568.

[85] Zhao, Peixiang, and J.X.Yu.,2007. Mining closed frequent free trees in graph databases, in *Advances in Databases: Concepts, Systems and Applications*, Springer Berlin Heidelberg, pp. 91-102.

[86] Zou, Lei, et al.,2006. PrefixTreeESpan: A pattern growth algorithm for mining embedded subtrees, in *Web Information Systems (WISE)*, Springer Berlin Heidelberg, pp. 499-505.

[87] Kutty, Sangeetha, R.Nayak, Y.Li., 2007. PCITMiner: prefix-based closed induced tree miner for finding closed induced frequent subtrees, in *Proceedings of the sixth Australasian conference on Data mining and analytics*, Vol. 70, Australian Computer Society.

[88] Zhao, Peixiang, and J.X.Yu., 2008. Fast frequent free tree mining in graph databases, in S*pringer World Wide Web*, Hong Kong, pp. 71-92.

This paper may be cited as:

Dinari, H. and Naderi, H. 2014. A Survey of Frequent Subgraphs and Subtree Mining Methods. *International Journal of Computer Science and Business Informatics. Vol. 14, No. 1, pp. 39-57.*