International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 5 Issue 8 ǁ Aug. 2017 ǁ PP.66-74

A Study on Subtree and Subgraph Mining Methods

Geetidarshini Panigrahi[1], Pratyush RanjanMohapatra[2]
*[*1] Department of Computer Science & Engineering, Gandhi Engineering College, Odisha, India*
[2] Department of Computer Science & Engineering, Gandhi Institute For Technology, Odisha, India

ABSTRACT: *Graph is a basic data structure which, can be used to model complex structures and the relationships between the data elements, such as XML documents, social networks, communication networks, chemical informatics, biology networks, and structure of web pages. In graph mining scenario Frequent subgraph pattern mining is one of the important aspect. Researchers have found may research oriented topic 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 mining algorithms use various methods on different datasets, patterns mining types, graph and tree representations.This paper presents a brief report of an intensive investigation of frequent subgraphs and subtrees mining algorithms.*
Keywords: *Graph Mining, Subgraph, Frequent Pattern, Graph indexing*.

I. 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 documentsclustering andclassification[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.

II. 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.

BASICCONCEPTS

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).

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.

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'⊆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:
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
Embeddedpattern:
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.

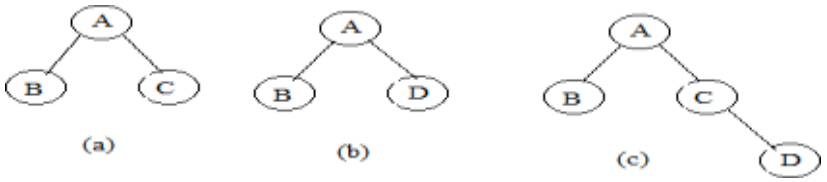

Figure.1. An example of the Induced and embedded subtree pattern

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

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

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

III. AN OVERVIEW OF FREQUENT SUBGRAPH MINING ALGORITHM ACCORDING TO DIFFERENTCRITERIA

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.

GraphRepresentation
AdjacencyMatrix

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 bythejunctionoftherowandthecolumnisfilledby"1"andotherwiseby "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.

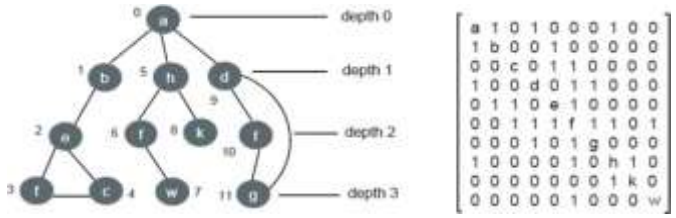

Figure.2. Left side a graph and right side corresponding adjacency matrix

AdjacencyList

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 memorydynamically

SubgraphGeneration

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 isgenerated.

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.

IV. A SURVEY OF FREQUENT SUBGRAPH MINING ALGORITHMS

Classification Based on AlgorithmicApproach

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 withlength 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].

PatternGrowth-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]	

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).

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

Single GraphDatabase
Database consists of a single large graph

Transactional GraphDatabase

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 (rightside).

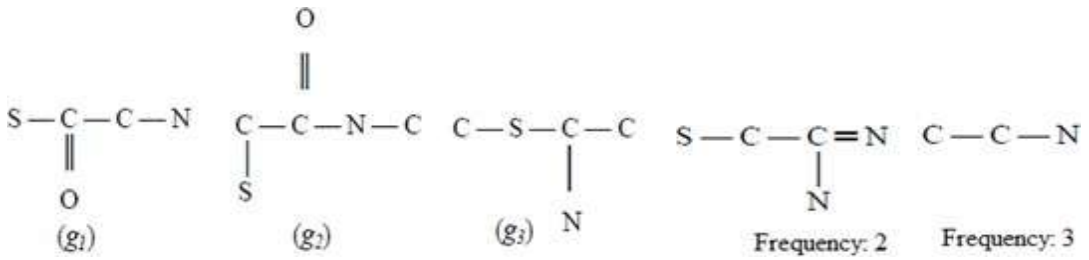

Figure.3. A database consisting of three graph g1, g2, g3 and two subgraph and frequency of each

Classification Based on Nature of theOutput
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

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:
Maximal Pattern
Subgraph g1 is maximal pattern if the pattern is frequent and does not consist of any super-pattern, so that g2 ↳g2⊃g1.
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	

Logic-BasedMining

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

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

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

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

- **Gp-GrowthAlgorithm**

The algorithm consists of three main steps:

1. Candidate generation by joinoperation.

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

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 MinerAlgorithm**

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].

V. FREQUENT SUBTREES MINING ALGORITHMS CLASSIFICATION

TreesRepresentation

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

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 asfollow:

{(0,a),(1,b),(2,e),(3,a),(1,c),(2,f),(3,b),(3,d),(2,a),(1,d),(2,f)(3,c)}

DFS – LS (Depth First Sequence)-(LabelSequence)

Assumed a labeled ordered tree, Labels is added to string ofs during the DFS traversal of Tree T. During backtrack '-1'or'$'or '/' isadded to the string, DFS-LS code for tree T is illustrated in in Figure.4

{abea$$$cfb$d$$a$$dfc$$$}

BFCS (Breadth First CanonicalString)

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#}

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 obtainedforthetreepresentedin Figure.4isasfollow 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].

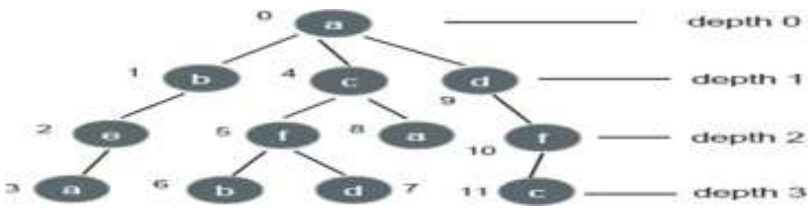

Figure.4. A Tree Example

Table 6. Frequent subtree Mining Algorithms (Tree Representation)

Algorithms	**Tree Representation**
uFreqtSLEUTH Unot PathJoin RootedTreeMiner[**66**] FREQT TreeMinerChopper XSPanner AMIOT IMB3Miner TRIPS FreeTreeMinerCMTreeMin erHybridTreeMiner [**67**] GP-Growth	DLS DFS-LS DLS FST-Forest BFCS DLS DFS-LS DLS DLS DFS string DFS-LS CPS BFCS DFS-LS BFCS DFS- LS

InputTypes

Rooted OrderedTrees

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 isshown.

Table 7. Rooted Ordered Tree mining Algorithms

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

Rooted UnorderedTrees

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**]

Tree Base DataMining

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

VI. CONCLUSION

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 theresults,itisconcludedthatinabsenceofgeneratingpatternsbypattern-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 datavolume.

REFERENCES

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

[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 andSecurity.

[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 intrusiondetection, 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 universityPress.

[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 ScienceReference.
[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.IvancsyandI.Vajk,2006.FrequentPatternMininginWebLogData,in Acta PolytechnicaHungarica, 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–