Graph mining: A survey of graph mining techniques

Conference Paper · August 2012

DOI: 10.1109/ICDIM.2012.6360146

Saif ur Rehman
PMAS - Arid Agriculture University
20 PUBLICATIONS **36** CITATIONS

Simon Fong
University of Macau
585 PUBLICATIONS **2,587** CITATIONS

Graph Mining: A Survey of Graph Mining Techniques

Saif Ur Rehman
CORD: Center for Research in Data Engineering
Mohammad Ali Jinnah University
Islamabad, Pakistan
saifi.ur.rehman@gmail.com

Asmat Ullah Khan
Department of Computer Science
Shaheed Zulifiqar Ali Bhutto Institute of Science and Technology (SZABIST)
Islamabad, Pakistan
Loyalms@hotmail.com

Simon Fong
Department of Computer and Information Science
University of Macau
Taipa, Macau SAR
ccfong@umac.mo

***Abstract*—Data mining is comprised of many data analysis techniques. Its basic objective is to discover the hidden and useful data pattern from very large set of data. Graph mining, which has gained much attention in the last few decades, is one of the novel approaches for mining the dataset represented by graph structure. Graph mining finds its applications in various problem domains, including: bioinformatics, chemical reactions, Program flow structures, computer networks, social networks etc. Different data mining approaches are used for mining the graph-based data and performing useful analysis on these mined data. In literature various graph mining approaches have been proposed. Each of these approaches is based on either classification; clustering or decision trees data mining techniques. In this study, we present a comprehensive review of various graph mining techniques. These different graph mining techniques have been critically evaluated in this study. This evaluation is based on different parameters. In our future work, we will provide our own classification based graph mining technique which will efficiently and accurately perform mining on the graph structured data.**

***Index Terms*—Graph Mining, Sub graphs, frequent graphs, Data Mining**

I. INTRODUCTION

Over the last few years there has been a number of research work on data mining in seeking for better performance and innovation. One innovation includes mining from structured data, which is a new challenge. Since a structure is represented by proper relations and a graph can easily represent such relations, knowledge discovery from graph-structured data poses a general problem for mining from structured data. Some examples amenable to graph mining are finding typical web browsing patterns, identifying typical substructures of chemical compounds, finding typical subsequences of DNA and discovering diagnostic rules from patient history records [21].

Graph mining techniques have been categorized into following groups. (1) Graph clustering; is the task of grouping the vertices of the graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters? Graph clustering in the sense of grouping the vertices of a given input graph into clusters [22] graph clustering is based on unsupervised learning technique in which the classes are not known in prior to clustering. The graph clusters are formed based on some similarities in the underlying graph structured data graph. (2) Graph Classification; in graph classification the main task is to classify separate, individual graphs in a graph database into two or more categories/classes [22]. Classification is based on supervised/semi supervised learning technique in which the classes of the data are defined in prior. (3) Sub graph mining; sub graph is a graph whose vertices and edges are subsets of another graph. The frequent sub graph mining problem is to produce the set of sub graphs occurring in at least some given threshold of the given n input example graphs [23].

In this study we have provided comprehensive summary details of the different graph mining techniques. Each of these techniques has been outlined with their techniques details, their major research contributions along with the limitation of the proposed techniques. These techniques have been further critically evaluated.

The rest of this paper is organized as follow: In section II the underlying terminologies used in graph theory is provided. In section III a detailed literature review is provided on the graph mining techniques proposed in the last few decades. Section IV focuses on the critical analysis of these different graph mining techniques, whose details are discussed in section II. This study will end with the conclusion of our work with some future directions in section V

II. BASIC GRAPH THEORY

A graph G is a pair of sets $G = (V, E)$. V is the set of vertices and the number of vertices $n = |V|$ is the order of the graph. The set E contains the edges of the graph. In an undirected graph, each edge is an unordered pair $\{v, w\}$. In a directed graph (also called a digraph in much literature), edges are ordered pairs. The vertices v and w are called the endpoints of the edge. The edge count $|E| = m$ is the size of the graph. In a weighted graph, a weight functions $\omega: \mathrm{E} \rightarrow \mathrm{R}$ is defined that assigns a weight on each edge. A graph is planar if it can be drawn in a plane without any of the edges crossing [22]. The

978-1-4673-2430-4/12/$31.00 ©2012 IEEE

density of a graph G = (V, E) is defined as the ratio of the number of edges present in a graph to the maximum possible,

$$\delta(G)=\frac{m}{\substack{n\\2}} \text{ For } n\in\{0,1\} \text{ , we set } \delta(G)=0 .$$

A graph of density one is called complete [22].

The adjacency matrix AG of a given graph G = (V, E) of order n is an n × n matrix $AG=(a_{V,u}^{G})$ where

$$AG=(a_{V,u}^{G})=\begin{cases}1, & \mathit{if}\ (v,u)\in E,\\ 0, & \mathit{otherwise}\end{cases}$$ the diagonal matrix of graph $G(V,E)$ is

$$D=\begin{bmatrix} \deg(v2) & 0 & 0 & 0 & 0 \\ 0 & \deg(v2) & 0 & 0 & 0 \\ 0 & 0 & \deg(v2) & 0 & 0 \\ 0 & 0 & 0 & \deg(v2) & \deg(v2) \end{bmatrix}$$

The length of a path is the number of edges on it, and the distance between v and u is the length of the shortest path connecting them in G. The distance from a vertex to itself is zero: the path from a vertex to itself is an empty edge sequence. A graph is connected if there exist paths between all pairs of vertices. If there are vertices that cannot be reached from others, the graph is disconnected. The minimum number of edges that would need to be removed from G in order to make it disconnected is the edge connectivity of the graph. A cycle is a simple path that begins and ends at the same vertex. A graph that contains no cycle is acyclic and is also called a forest. A connected forest is called a tree [21].

A sub graph $G^S=(S, E_S)$ of $G=(V, E)$ is composed of a set of vertices $S \subseteq V$ and a set of edges $E_S \subseteq E$ such that $\{v, u\} \subseteq E_S$ implies u, $v \in S$; the graph G is a super graph of G^S. A connected acyclic sub graph that includes all vertices is called a spanning tree of the graph. A spanning tree has necessarily exactly n−1 edges. If the edges are assigned weights, the spanning tree with smallest total weight is called the minimum spanning tree.

Note that there may exist several minimum spanning trees that may even be edge disjoint [22].

Two graphs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$ are isomorphic if there exists a bi-jective (one to one) mapping $f: V_i \rightarrow V_j$ (called an isomorphism) such that $\{u, v\} \in E_i$; if and only if $\{f(v), f(w)\} \in E_i$. A bipartite graph is a graph where the vertex set V can be split in two sets A and B such that all edges lie between those two sets: if $\{u, v\} \in E_i$, either $v \in A$ and $w \in B$ or $v \in B$ and $W \in A$ [23]. A complete graph is a graph where every pair of distinct vertices is adjacent. A complete graph on n vertices is denoted by Kn (or sometimes by $K(n)$) and The complete graph Kn of order n is a simple graph with n vertices in which every vertex is adjacent to every other is called clique.

III. Literature Review

This section summarizes the different proposed graph mining algorithms with their major research contributions and limitations.

In [1], Callut *et al.* have proposed a new technique called D-Walks. This technique can efficiently handle the semi-supervised classification issues associated with the graphs of large size. Their technique is based on the betweeness measures. The detail of betweeness can be found in [1]. The *D-walks* can classify the unlabeled nodes of different types of graphs including directed or undirected graphs. This classification has a linear time complexity with respect to the (1) number of edges in the graphs, (2) the maximum walk length considered and (3) the number of classes [1]. The unlabeled nodes of the graphs are predicted by comparing its betweeness measure with that of maximum betweeness measure. The technique proposed in [1] has been implemented on the CORA database. Then different experiments have been performed using this database. All of these experiments showed that [1] is more efficient and can accurately classify the unlabeled nodes of the graphs and outperforms the existing techniques available in the literature such as [2] and [3]. Their main achievement is to handle the graphs having large number of nodes and edges as compare to [2] and [3] techniques

In [4], Kashima *et al.* have proposed a new method that can handle the classification problem of graphs that have extremely large no of nodes and edges. Their graph classification method is based on kernel method. The details of kernel methods can be found in [5]. The method proposed in [4] efficiently computes the inner product of two graphs to make a feature space for classifying the graphs. This technique takes an unknown graph as input and classifies the unknown graph into an appropriate class. Their proposed method calculates the similarity of two graphs based on nodes of the graphs and labels of the edges in the graphs. In [4] graphs are classified into same group if their similarities are identical. The technique proposed in [4] has been implemented for the prediction of properties of chemical compound using the mutag and PTC dataset. Then different experiments have been performed using these datasets. All of these experiments showed that [4] is not as efficient as [6] for mutag dataset but for PTC dataset it is more efficient then existing techniques available in the literature such as [6] and [7].

In [8], Dhillon *et al.* have presented an efficient and fast technique for graph clustering. This technique can handles graph having large number of nodes and very large number of edges. Their graph clustering technique is based on multilevel methods using weighted kernel K-means objective function as refinement algorithms .The details of weighted Kernel k-means objective function for multilevel methods can be found in [9]. The technique proposed in [8] does not restricts the size of the cluster be nearly equal as compared to existing graph clustering techniques available in literature. Furthermore, the graph clustering objective functions proposed in [8] can be specialized for all phases of the algorithm according to situation. The technique proposed in [8] has been implemented on the IMDB Movie dataset. The dataset has 1.2 million nodes and 7.6 million edges. Furthermore different experiments have been performed using this dataset. The proposed techniques compute 5000 cluster and 5000 eigenvectors [8] which is impractical for the algorithm in [9] due to requirements of main memory up to 25 GB. All of these experiments showed that [8] is more efficient not only in memory consumption but also in running time compared to the existing techniques available such as [9]. Their main achievement is to handle the graphs having large number of nodes and edges which is impractical to be handled in existing graph clustering techniques [9].

In [10], Dias and Ochi have presented enhancement in the basic Genetic Algorithms (GAs). Their proposed technique can efficiently handle the issues of graph partitioning in large graph databases. The [10] proposed different procedure as evolutionary steps for the improvement in the performance of the basic GA. The proposed modifications in [10] to the basic GA algorithms do not alter the global acting of the basic technique for GA. Therefore these modifications are implemented as fittings to the Basic GA. The proposed procedures in [10] modify the local search and other diversification procedures [10]. The proposed procedures in [10] are implemented in 7 different versions. The performance of the proposed algorithms was evaluated for different no of nodes in graph. The results established that the proposed algorithms produces high quality clusters while maintaining the same running time as compared to existing GA in the literature. The main contribution of the proposed procedures has good performance when the no of nodes are high as 500 nodes.

In [11], Zhao *et al* have proposed a new technique for mining closed free tree in large graphs. Their technique is called *CFFTree* (Closed Frequent Free Tree). This technique efficiently mine frequent closed free tree in large graph database whose nodes are labeled. The technique proposed in [11] can handle the issues of mining frequent free trees in large graph database which is NP complete the details of NP problem is found in [12]. A tree *t* with no designated root is called a free tree and a free tree *t* is closed if no super tree of *t* that has the same frequency of *t* [11] exists. The authors suggested that closed free trees are very few in graph but can maintain the same useful information as free trees. Furthermore, they established that the computational time of closed frequent free trees mining algorithm is polynomial and closed free trees are more efficient. The [11] proposed efficient pruning methods such as safe labeling pruning , safe positioning pruning, auto-morphisim-based pruning and canonical mapping-based pruning the details of these methods can be found in [11] to prune free trees that cannot generate closed free tree in order to tune the mining process of closed free trees. The technique proposed in [11] has been implemented on the AIDS antiviral screen chemical compound from Development Therapeutics program in NCI/NIH. Different experiments have been performed by using this database. All of these experiments proved that [11] is more efficient and can accurately computes free trees compared to [13]. While the proposed technique is the only technique developed for mining closed frequent free tree in time the paper was written. The main contribution of their proposed technique is working on the novel concept of closed frequent free trees mining and designing an algorithm for mining closed trees from graph databases.

In [14], Le *et al.* have proposed a new method for clustering of bi-partite graph. This technique is called Coring technique. The proposed technique can handle the issues of partitioning a large graph into small sub graphs. The nodes of the clustered sub graphs are strongly interconnected within graph and weakly connected to the nodes of other graphs. Their method is called coring method that can handle both weighted and unweighted graphs. The technique in [14] can computes clusters that have a highly dense core region and encircled by lower density region. The proposed method in [14] works in following steps Step 1: In this step, the coring method computes the density variation sequence .The method iteratively computes the minimum density *D* and set of nodes having minimum density M. The output of this step is sequence of *D,s* and *M,s*. Setp2: Following step 1, the coring method identify the core nodes . To identify the method calculates the rate of decrease/increase in value of minimum density. If the rate of increase/decrease in the *D* value is greater than the threshold and the sequence of *M* is also in some order then the nodes are identified as core nodes. Step 3: In this step, the coring method partition the graph nodes into clusters. The set of core nodes is the output to the next step. Step 4: it is the final step of this technique the core groups are expended into full clusters. The core nodes are the center of the clusters and the lower density nodes are encircles these core nodes. The technique proposed in [14] has been implemented on the microarray dataset containing 62 samples including 40 tumor and 22 normal colon tissues. Each sample consists of 2000 gene expressions database. The [14] successfully cluster the tumor tissues and normal tissues in the database further the method was evaluated using image of size 200×300 and the [14] efficiently cluster the core region from the image. The proposed method was also evaluated for introducing noise into the image. The method successfully clusters the core region. The main strength of the proposed work is that this method can efficiently be used for noisy data.

In [15], Chen *et al.* have proposed a graph model that can efficiently handle the many to many correspondences problem among concepts in ontologies. Their proposed technique used weighted bi-partite graph to model ontologies. The similarity measure is computed for the all the edges using similarity measure techniques such as in [16]. The proposed technique, assigns the similarity degree as weights of the edges in the graph. In the proposed technique, edges of the bi-partite graph having weight greater than the threshold are maintained other edges are purged. The [15] uses graph partitioning technique [15] to co-cluster the vertex of the graph as concept cluster for two ontologies. The concept cluster produced by [15] in previous step contains all common concepts from ontologies. In next step the concept cluster is used to set up mappings among ontologies. The contribution of the proposed techniques is that many-to-many mapping can be establish among ontologies.

In [16], Barber has proposed a new graph clustering mechanism for representing graph in the form of matrix. Their proposed technique extends the incidence matrix (showing joining vertex of graph as matrix) to clique matrix. The clique matrix shows that which nodes of the graph can form a clique. The clique matrix can be efficiently used for graph clustering. The proposed technique executes in the following steps: (1) in first step, it calculates the maximal clique. (2) In this step, the clustering is performed by [16] as it identified the matrix with smallest no of columns. The size of the clique is controlled by using threshold parameter that controls how large the clique should be. Their technique is successfully applied to find the large well-connected group in social network and cluster gene-expression that exists in large population. The main contribution of the proposed work is the clique matrix notation for graphs.

In [17], Kraus *et al.* have proposed a new algorithm for handling the graph clustering. Their algorithm is called semi supervised divisive hierarchical Graph clustering algorithm. Their proposed technique can effectively handle the problem of clustering with having no knowledge of the structure of underlying dataset. The authors proposed a hierarchical algorithm that incorporates background knowledge into the graph. The technique in [17] is used with weighted undirected graph. The Euclidian distance between two adjacent nodes is calculated. To calculate the Euclidian distance formula is given below:

$$d(p,q) = d(q,p) = \sqrt{(q_1-q_1)^2+(q_2-p_2)^2+....+(q_n-p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i-p_i)^2}$$

and the ratio is computed by dividing the distance with average Euclidian distance of all the nodes in a graph. The must link indicates that two data item must be placed in same groups, and can-not links – two data item cannot be placed in same group, are identified. Links with less weight are removed to control the chaining effect of the nodes on the clusters. To propagate background knowledge in adjacent nodes the probability of the visiting nodes with some threshold steps are calculated for two nodes and neighborhood similarity is measured for the nodes. The proposed algorithm increases the weight of the edge if two nodes are similar else the weight of the edge is decrease. Afterward, nodes with small neighborhood are removed for creating clusters. Nodes having similar neighborhood values are cluster in same group. The main contribution of the proposed work is the including of background knowledge in the clustering process.

In [18], Schenker *et al.* have proposed a graph model for classification of web documents. The proposed method is based on k-NN [18] that successfully classifies unknown documents to its respective classes automatically. The experiments on [18] is conducted which reveals that the graph based model for document classification computation time is parallel to other vector based k-NN model. The experiments showed that for small nodes up to 30 the classification time of the proposed technique is as efficient as vector based k-NN techniques but the technique in [18] out performed vector based k-NN methods for large no of nodes both in performance and accuracy.

In [19] T.Ozaki et al, have proposed a new method for sub graph mining in graph-structured database. Their method is called HSG. The algorithm proposed in [19] is based frequent hyper clique patterns; which tries to find the dependencies among graph in the large. The method proposed in [19] efficiently mine correlation in structured database. The authors proposed efficient pruning methods based on h-confidence measures and depth-first and breadth-depth search methods, the details of these methods can be found in [19]. The technique proposed in [19] has been implemented on the PTE and DTP_CM datasets. The [19] efficiently mine frequent hyperclique patterns in these datasets in reasonable time. The main contribution of the proposed work is that [19] introduces a new concept of hyperclique to mine correlation in graph databases and proposed an algorithm to mine frequent hyper clique patterns in large graph databases.

In [20] Fatta *et al*, proposed a new method for sub graph mining in large graph database. The method is called distributed algorithm. Their algorithm is based on distributed peer to peer communication framework. The [20] can handle to very high workload in distributed manner. The distributed algorithm proposed in [20] efficiently mines sub graph in molecular compounds, the molecular compounds have very large trees and very large no of sub graph. The [20] first partitioned the search space dynamically to partition a large tree. In the second step the [20] distributes the portioned tree in peer-to-peer communication framework and in the last step the distributed algorithm uses load balancing and receiver initiated methods [20] to execute the sub graph mining process in distributed environment. To further test the effectiveness of the proposed method. The proposed technique has been implemented on the DPT dataset. Then different experiments have been performed using this dataset. All of these experiments showed that [1] is more efficient and can accurately mind sub graph in highly distributed and heterogeneous environment. The method proposed in [20] also has been tested for fault tolerant and the results showed that the proposed method has handled the situation very efficiently. The main contribution of the proposed work is that it works in highly distributed and heterogeneous environment.

IV. Critical Evaluation

In this section we comment about the techniques, critically. The critical evaluation is based on the observation of the following metrics: parameters, technique, method, implementation, features, comparison and efficiency. The details are shown in Table 1. According to the comparison in Table 1, the work in [8] seems to be more efficient in computation time and memory usage during the clustering process than [1] and [4] for classification. The model in [10] is capable of handling larger nodes than [16] and [18] for clustering in a large graph in efficiency and features. The results generated from the model [14] is however more accurate than those in [1] and [8] for feature support. Thus the above discussion reveals that [14] may be more accurate for noisy data and [8] may be more efficient for larger graphs.

TABLE I. COMPARISON OF RECENT WORKS ON GRAPH MINING

Paper	Technique/Method	Implementation	Features	Efficiency	Comparison
Callut *et al*.[1]	D-Walks	CORA	Capable of handling large graphs	1.4 seconds per graph	yes
Kashima *et al*[4]	Multi-level Kernel k-means	Mutag, PTC	Reduced chaining effect ; Computes similarity both on label and edges		yes
Dhillon *et al* [8]	Multi-level Kernel k-means	IMDB Movie	Memory efficient Efficient in running time	25 minutes for 1.2 million nodes and 7.6 million edges	yes
Dias and Ochi[10]	Genetic Algorithm	C++	Tracked the performance of GA for different type of graph	98 % for 500 nodes	yes
Zhao *et al*.[11]	CFFTree	C++,VS	More efficient for graph with large no of nodes	10 to 1.5 free tree and closed	yes
Le *et al*.[14]	Coring Method	MicroArray dataset, image	Efficiently clustered core region in noisy data		yes
Chen *et al*.[15]	Bi-partite graph co-clustering				yes
Barber[16]	Clique matrix	DIMACS	Clique matrix notation for graphs. Clustering based on clique matrix notations		no
Kraus *et al*.[17]	SSHGCA	MicroArray dataset	Including of background knowledge in clustering process		yes
Schenker *et al*.[18]	K-NN	Yahoo News C++	More efficient and accurate for large size graph		Yes
T. Ozaki *et al*[19]	HSG	PTE, DTP_CM Java	Mine correlation in graphs		No
Fatta et al. [20]	Distributed Algorithm	PTE, DTP_CM Java	Efficient; Distributed; Heterogeneous		No

V. CONCLUSION AND FUTURE WORK

In this study, we have presented the summary information of the different graph mining techniques. These graph mining techniques are based on the classification, clustering, decision tree approaches, which are the data mining fundamentals. In addition, we also have highlighted the research contributions and found out some limitations in different research works. Consequently, this work also depicts the critical evaluation in which comparison and contrast have been taken out to show the similarities and differences among different author's works. The spatiality of this work is that it reveals the literature review of different graph mining techniques and provides a vast amount of information under a single paper. In our future work, we have planned to propose a new classification method based on graph mining technique, provide its implementation and compare its results with the different existing classification based graph mining algorithms.

REFERENCES

[1] J. Callut, K. Françoisse, M. Saerens and P. Dupont, "Semi-supervised Classification from Discriminative Random Walks", Lecture Notes in Artificial Intelligence No. 5211, Springer, 2008., pp. 162-177,

[2] S.Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study", J. Mach. Learn. Res., 8, pp935-983,2007

[3] M.Newman, "A measure of betweenness centrality based on random walks", Social networks 27,pp39-54, 2005

[4] H. Kashima and A. Inokuchi, "Kernels for graph classification", ICDM Workshop on Active Mining 2002, 2002.

[5] M.Swell ,"Kernel Methods",2009

[6] J. Han.,X. Yan and P.S. Yu, "Mining and searching graphs and structures", in the proceedings of 12th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD'2006),2006

[7] S. Kramer and L. D. Raedt., "Feature construction with version spaces for biochemical application", In Proc. of the 18th ICML,2001

[8] Dhillon, Y. Guan and B. Kulis, "A Fast Kernel-based Multilevel Algorithm for Graph Clustering", Proceedings of The 11th ACM SIGKDD, Chicago, IL, Aug. 21 - 24, 2005

[9] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs". SIAM J. Sci. Comput.,20(1):359–392, 1999

[10] C. R.Dias, and L. S.Ochi, "Efficient Evolutionary Algorithms for the Clustering Problem in Directed Graphs", Proceedings of the 2003 IEEE Congress on Evolutionary Computation, v.1, pp. 983-988, 2003

[11] P. Zhao and J. X. Yu, "Mining Closed Frequent Free Trees in Graph Databases", Proceeding of Database Systems for Advance Application 2007, pp. 91–102, 2007

[12] Yun Chi, Yirong Yang, and Richard R. Muntz. , "Indexing and mining free trees", In Proceedings of ICDM03, 2003.

[13] Wache, H., Vogele, T., Visser, U., Stuckenschmidt, "Ontology-Based Integration of Information - A Survey of Existing Approaches", In Proceedings of IJCAI-01 Workshop on Ontologies and Information Sharing, 108-117, 2001.

[14] T. V. Le, C. A. Kulikowaski and I. B. Muchnik, "Coring Method for Clustering a Graph", In proceedings of IEEE 2008, 2008

[15] Y. Chen and F. Fonseca , "A Bipartite Graph Co-Clustering Approach to Ontology Mapping",2004

[16] D. Barber. Clique Matrices for Statistical Graph Decomposition and Parame- nite Matrices. In D. A. McAllester and P. Myllymaki, editors, AUAI Press, pp 26-33, 2008.

[17] J. M. Kraus, G. Palm and H. A. Kestler, "On the robustness of semi-supervised hierarchical graph clustering in functional genomics", 2007

[18] A. Schenker, M. Last, H. Bonke and A. Kandel "Classification of Web Documents Using a Graph Mode",Proceedings of the Seventh International Conference on Document Analysis and Recognition, 2003

[19] T.Ozaki and T.Ohkawa , "Mining Correlated Subgraphsin Graph Databases", PAKDD 2008,pp 272-283, 2008

[20] G.D. Fatat and M.R. Berthold "High Performance Subgraph Mining in Molecular Compounds",HPCC 2005, pp 866-877 ,2005

[21] S.E Schaeffer, "Graph Clustering", Computer Science Review 2007, pp 27-64, 2007

[22] H. Motoda, "What Can We Do with Graph-Structured Data A Data Mining Perspective", Springe 2006, pp 1-2,2006

[23] N. S. Ketkar, L.B.Holder and D.J. Cook," Empirical Comparison of Graph Classification Algorithms",IEEE,2009