FREQUENT SUB GRAPH MINING AS NEW TREND IN SOCIAL NETWORK

Divya R. Jariwala[1] Nisha G. Medhat[2] Samiksha H. Zaveri[3]

[1]Research Scholar, (computer Science & Applications) Shri JJT University, Jhunjhunu-Churu Road, Vidyanagari, Dist Jhunjhunu, Churela, Rajasthan, India

[2] Assistant Professor, UCCC & SPBCBA & UACCAIT College, Udhna-Navsari Road, Surat, Gujarat, India

[3] Research Scholar, Parul University, Vadodara, Gujarat, India

ABSTRACT:
Mining graph data is the extraction of novel and useful knowledge from a graph representation of data. The most natural form of knowledge that can be extracted from graphs is also a graph, we referred it as patterns. Frequent sub graph pattern mining is a one of the most popular research topics in data mining. Aim of graph mining is finding interesting patterns within data that represent novel knowledge. Now a day frequent sub graph mining used in various domains like in chemical compounds, social networks, biological networks etc. Mining patterns from graph database is difficult because of sub graph testing and their different operations. This paper gives the idea about different sub graph algorithms based on their approaches. This paper investigates on comparison of graph mining algorithms and techniques for finding the frequent patterns. The research goals are directed at: (i) effective mechanisms for generating candidate sub graphs (without generating duplicates) and (ii) how best to process the generated candidate sub graphs so as to identify the desired frequent sub graphs in a way that is computationally efficient and procedurally effective.

Index Terms— Graph, Frequent sub graph, Social Network, A-priori Based approach, Pattern-growth approach

I. INTRODUCTION

Before presenting graph mining methods, it is necessary to first introduce some preliminary definitions of graph.

Two-dimensional drawing showing a relationship (usually between two set of numbers) by means of a line, curve, a series of bars, or other symbols. Typically, an independent variable is represented on the horizontal line (X-axis) and dependent variable on the vertical line (Y-axis). The perpendicular axis intersect at a point called origin, and are calibrated in the units of the quantities represented. Though a graph usually has four quadrants representing the positive and negative values of the variables, usually only the north-east quadrant is shown when the negative values do not exist or are of no interest.

A graph is defined to be a set of vertexes (nodes) which are interred connected by a set of edges (links). The graphs used in FSM are assumed to be labeled simple graphs.

Labeled Graph: A labeled graph can be represented as G(V, E, LV , LE, ϕ), where V is a set of vertexes, $E \subseteq V \times V$ is a set of edges; LV and LE are sets of vertex and edge labels respectively; and ϕ is a label function that defines the mappings $V \rightarrow LV$ and $E \rightarrow LE$. G is (un)directed if $\forall e \in E$, e is an (un)ordered pair of vertexes. A path in G is a sequence of vertexes which can be ordered such that two vertexes form an edge if and only if they are consecutive in the list (West 2000). G is connected, if it contains a path for every pair of vertexes in it and disconnected otherwise. G is complete if each pair of vertexes is joined by an edge and G is acyclic if it contains no cycle. [11]

If a graph is frequent, then all of its sub graphs will also be frequent. A simple graph is an un-weighted and un-directed graph with no loops and no multiple links between any two distinct nodes.

A path in G is a sequence of vertexes which can be ordered such that two vertexes form an edge if and only if they are consecutive in the list. G is connected, if it contains a path for every pair of vertexes in it and disconnected otherwise. G is complete if each pair of vertexes is joined by an edge and G is acyclic if it contains no cycle.

Sub graph: Given two graphs G1 (V1, E1, LV1, LE1, ϕ1) and G2 (V2, E2, LV2, LE2, ϕ2), G1 is a sub graph of G2, if G1 satisfies: (i) $V1 \subseteq V2$, and $\forall v \in V1$, $\phi1(v) = \phi2(v)$, (ii) $E1 \subseteq E2$, and $\forall(u, v) \in E1$, $\phi1(u, v) = \phi2(u, v)$. G1 is an induced sub graph of G2, if G1 further satisfies: $\forall u, v \in V1$,

(u, v) ∈ E1 ⇔ (u, v) ∈ E2, in addition to the above conditions. G2 is also a super graph of G1. (Iokuchi et al. 2002; Huan et al. 2003) [13-18].

Frequent Sub Graph: Given a labeled graph dataset GD = {G1, G2…, GK}, support or frequency of a sub graph g is the percentage of graph in GD where g is a subgraph. A frequent subgraph is a graph whose support is no less than a minimum user specified support threshold. [19]

II. GRAPH BASED DATA MINING

Graph Mining (GM) is essentially the problem of discovering repetitive sub graphs occurring in the input graphs. Graph-based data mining represents a collection of techniques for mining the relational aspects of data represented as a graph. Graph-based data mining (GDM) is the task of finding novel, useful, and understandable graph-theoretic patterns in a graph representation of data. Several approaches to GDM exist based on the task of identifying frequently occurring sub graphs in graph transactions, that is, those sub graphs meeting a minimum level of support.

A. Modelling Data With Graphs…Going Beyond Transactions (6)

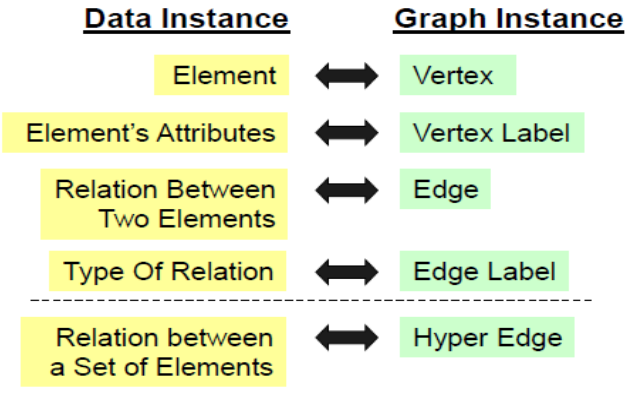


Fig 1. Modelling Data With Graphs…Going Beyond Transactions [6]

B. Applications of Graph Mining/Domains Of Graph Mining:

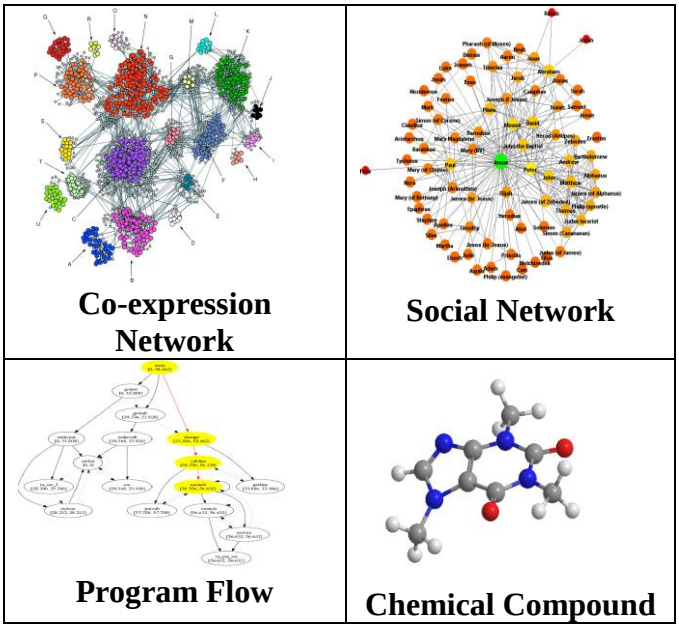


Fig 2. Applications of Graph Mining [21]

III. SOCIAL NETWORKS ANALYSIS:

The notion of social networks, where relationships between entities are represented as links in a graph, has attracted increasing attention in the past decades. Thus social network analysis, from a data mining perspective, is also called link analysis or link mining.

From data mining point of view, a social network is a heterogeneous and multi-relational data set represented by a graph. The graph is typically very large, with nodes corresponding to objects and edges corresponding to links representing relationships or interactions between objects. Both nodes and links have attributes. Objects may have class labels. Links can be one-directional and are not required to be binary. Social networks need not be social in context. There are many real-world instances of technological, business, economic, and biologic social networks. Examples include electrical power grids, telephone call graphs, the spread of computer viruses, and the World Wide Web, and co authorship and citation networks of scientists.

They reflect the concept of "small worlds," which originally focused on networks among individuals.

The motivation here is the popularity of social networking sites such as Facebook, and the consequent desire to identify groupings (communities) within these networks. However, there are many other forms of social networks, such as transport and co-authoring (bibliographic) networks, to which social network mining techniques can be applied. [11]

Efficient methods have been developed for mining frequent sub graph patterns. They can be categorized into Apriori-based and pattern growth–based approaches. The Apriori-based approach has to use the breadth-first search (BFS) strategy because of bits level-wise candidate generation. The pattern-growth approach is more flexible with respect to the search method. A typical pattern-growth method is gSpan, which explores additional optimization techniques in pattern growth and achieves high performance.

The further extension of gSpan for mining closed frequent graph patterns leads to the Close Graph algorithm, which mines more compressed but complete sets of graph patterns, given the minimum support threshold. [12]

IV. Approaches Of Frequent Sub Graph:

The approaches of graph based data mining are fall into various categories. In this paper we focuses on two categories i.e. Apriori and Pattern based approach. And we included different algorithms for this approaches [21].

Generation of Candidate Patterns

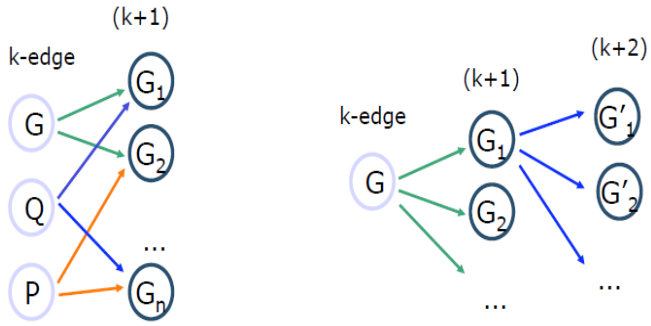


Fig 3. Apriori-Based Approach v/s Pattern-Growth Approach

A. Apriori-Based Approach

It uses a generate-and-test approach – generates candidate item sets and tests if they are frequent: One is Generation of candidate item sets is expensive (in both space and time) second Support counting is expensive i.e., Subset checking, Multiple Database scans (I/O) [8].

The Apriori-based approach has to use the breadth-first search (BFS) strategy because of its level-wise candidate generation. In order to determine whether a size-(k+1) graph is frequent; it must check all of its corresponding size-k subgraphs to obtain an upper bound of its frequency. Thus, before mining any size-(k+1) subgraph, the Apriori approach usually has to complete the mining of size-k subgraphs. Therefore, BFS is necessary in the Apriori-like approach.

Apriori Property

If a graph is frequent, all of its subgraphs are frequent [21].

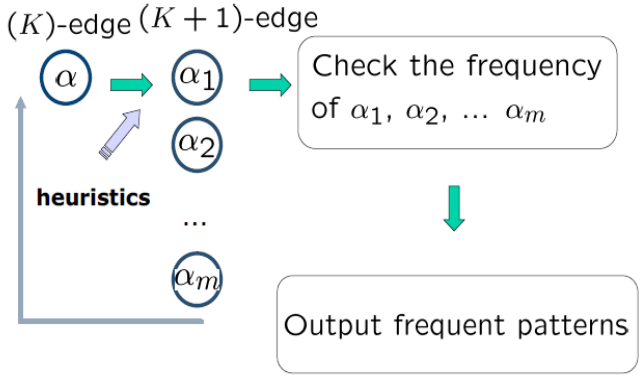


Fig 4. [21]

Depth-First Search (DFS)

Depth-first search (DFS) starts from a node vi, selects one of its neighbors vj 2 N(vi), and performs DFS on vj before visiting other neighbors in N(vi). In other words, DFS explores as deep as possible in the graph using one neighbor before backtracking to other neighbors. Consider a node vi that has neighbors vj and vk; that is, vj; vk 2 N(vi). Let vj(1) 2 N(vj) and vj(2) 2 N(vj) denote neighbors of vj such that vi , vj(1) , vj(2). Then for a depth-first search starting at vi, that visits vj next, nodes vj(1) and vj(2) are visited before visiting vk. In other words, a deeper node vj(1) is preferred to a neighbor vk that is closer to vi. Depth-first search can be used both for trees and graphs, but is better visualized using trees. The DFS execution on a tree is shown in Figure 2.19(a).

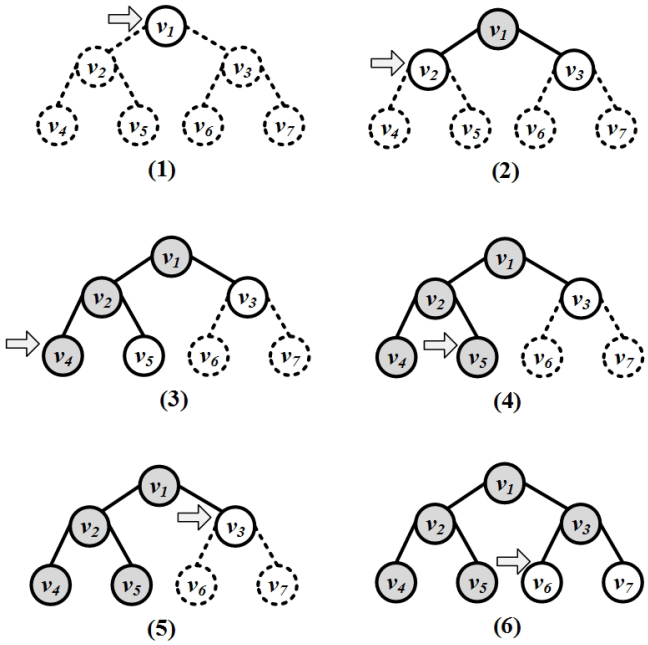
The DFS algorithm is provided in Algorithm 2.2. The algorithm uses a stack structure to visit nonvisited nodes in a depth-first fashion [20].

Algorithm 2.2 Depth-First Search (DFS)
Require: Initial node v, graph/tree G(V; E), stack S
1: return An ordering on how nodes in G are visited
2: Push v into S;
3: visitOrder = 0;
4: while S not empty do
5: node = pop from S;
6: if node not visited then
7: visitOrder = visitOrder +1;
8: Mark node as visited with order visitOrder; //or print node
9: Push all neighbors/children of node into S;
10: end if
11: end while
12: Return all nodes with their visit order.[20]

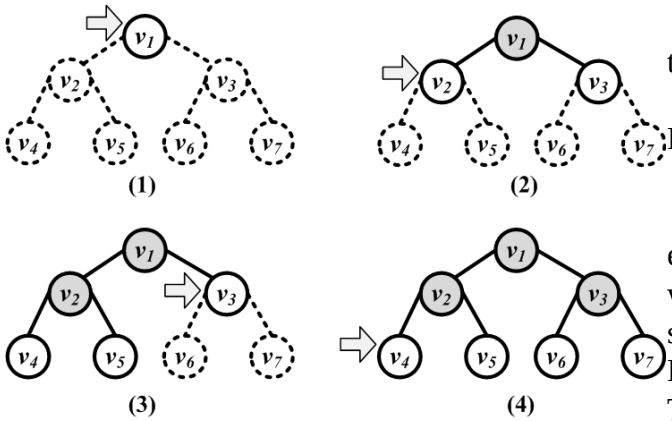
Breadth-First Search (BFS)

Breadth-first search (BFS) starts from a node, visits all its immediate neighbors first, and then moves to the second level by traversing their neighbors. Like DFS, the algorithm can be used both for trees and graphs and is provided in Algorithm 2.3.The algorithm uses a queue data structure to achieve its goal of breadth traversal. Its execution on a tree is shown in Figure 2.19(b) [20].

In social media, we can use BFS or DFS to traverse a social network: the algorithm choice depends on which nodes we are interested in visiting first. In social media, immediate neighbors (i.e., friends) are often more important to visit first; therefore, it is more common to use breadth-first search [20].



(a) Depth-First Search (DFS) [20]



(b) Breadth-First Search (BFS)

Fig 5. Graph Traversal Example.

Algorithm 2.3 Breadth-First Search (BFS)
Require: Initial node v, graph/tree G(V; E), queue Q
1: return An ordering on how nodes are visited
2: Enqueue v into queue Q;
3: visitOrder = 0;
4: while Q not empty do
5: node = dequeue from Q;
6: if node not visited then
7: visitOrder = visitOrder +1;
8: Mark node as visited with order visitOrder; //or print node
9: Enqueue all neighbors/children of node into Q;
10: end if
11: end while[20]

B. Pattern-Growth Approach

The graph representation has gained popularity in pattern recognition and machine learning. Frequent pattern mining (FPM) is an important part of graph mining that helps to discover patterns that conceptually represent relations among discrete entities. Developing algorithms that discover all frequently occurring sub graph in a large graph dataset is particularly challenging and computationally intensive, as graph and sub graph isomorphism play a key role throughout the computations.
It allows frequent item set discovery without candidate generation. Two steps:

1. Build a compact data structure called the FP-tree.
2. Extracts frequent item sets directly from the FP-tree [8].

For each discovered graph g, it performs extensions recursively until all the frequent graphs with g embedded are discovered. The recursion stops once no frequent graph can be generated. Pattern Growth Graph is simple, but not efficient. The bottleneck is at the inefficiency of extending a graph. The same graph can be discovered many times. For example, there may exist n different (n□1)-edge graphs that can be extended to the same n-edge graph. The repeated discovery of the same graph is computationally inefficient. We call a graph that is discovered a second time a duplicate graph. Although Pattern Growth Graph gets rid of duplicate graphs, the generation and detection of duplicate graphs may increase the workload. In order to reduce the generation of duplicate graphs, each frequent graph should be extended as conservatively as possible. This principle leads to the design of several new algorithms.

A typical such example is the gSpan algorithm, as described below. The gSpan algorithm is designed to reduce the generation of duplicate graphs. It need not search previously discovered frequent graphs for duplicate detection. It does not extend any duplicate graph, yet still guarantees the discovery of the complete set of frequent graphs.[12]

Gspan (graph-based Substructure pattern mining) [8] developed by Xifeng Yan, Jiawei Han in 2002. Gspan use DFS strategy, lexicographic order, minimum DFS code and rightmost extension. So that, it discovers frequent substructures without candidate generation. gspan works on label simple graph. Gspan use adjacency list for graph representation. In gspan use 340 chemical compound data set for evolution of performance of algorithm. [10]

The g Span algorithm

Algorithm g Span Mining (D, Min Sup, S)
1: sort labels of the vertices and edges in D by frequency;
2: remove infrequent vertices and edges;
3: relabel the remaining vertices and edges (descending);
4:S0=code of all frequent graphs with single edge;
5: sort S0 in DFS lexicographic order; S=S0;
6: for each code s in S0 does
7: gSpan(s, D, MinSup, S);
8: D: =D-s;
9: if |D|<Minus;
10: break;[19]

Algorithm gSpan(s, D, MinSup, S)
1: if s! =min(s), then
2: return
3: insert s into S
4: set C to {}
5: scan D once; find every edge e such that s can be right-most
Extended to frequent s*e;
Insert s*e into C;
6: sort C in DFS lexicographic order;
7: for each s*e in C do
8: Call gSpan(s*e, D, MinSup, S);
9: return [19]

V. SUMMARY:

Graphs represent a more general class of structures than sets, sequences, lattices, and trees. Graph mining is used to mine frequent graph patterns, and perform characterization, discrimination, classification, and cluster analysis over large graph data sets. Graph mining has a broad spectrum of applications in chemical informatics, bioinformatics, computer vision, video indexing, text retrieval, and Web analysis. Efficient methods have been developed for mining frequent sub graph patterns. They can be categorized into apriori-based and pattern growth–based approaches. The apriori-based approach has to use the breadth-first search (BFS) strategy because of its level-wise candidate generation. The pattern-growth approach is more flexible with respect to the search method. A typical pattern-growth method is gSpan, which explores additional optimization techniques in pattern growth and achieves high performance. The further extension of gSpan for mining closed frequent graph patterns leads to the CloseGraph algorithm, which mines more compressed but complete sets of graph patterns, given the minimum support threshold.

A social network is a heterogeneous and multirelational data set represented by a graph, which is typically very large, with nodes corresponding to objects, and edges (or links) representing relationships between objects.

VI. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract in the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

References

[1] Takashi Washio, Hiroshi Motoda,State of the Art of Graph based Data Mining,
[2] Prof. Ehud Gudes,Graph and Web Mining - Motivation, Applications and Algorithms , Department of Computer Science
[3] Prof. Mahesh Panchal, Dr. Monal J. Patel, A Survey of Graph Pattern Mining Algorithm and Techniques,Harsh J. Patel, Rakesh Prajapati, International Journal of Application or Innovation in Engineering & Management (IJAIEM),Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com,Volume 2, Issue 1, January 2013 ISSN 2319 – 4847
[4] 4. Yasunari Kishimoto, Hiroaki Shiokawa,Yasuhiro Fujiwara, and Makoto Onizuka NIT technical Review, Vol. 11 No. 12 Dec. 2013 REfficient Mining Algorithms for Large-scale Graphs
[5] Amit Kr. Mishra, Pradeep Gupta, Ashutosh Bhatt, Jainendra Singh Rana ,ISSN: 2277-3754 ISO 9001:2008 Certified International

Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 2, August 2012 47 ,Innovative Study to the

[6] Chuntao Jiang, Frans Coenen and Michele Zito,The Knowledge Engineering Review, Vol. 00:0, 1–31.c 2004, Cambridge University Press DOI: 10.1017/S000000000000000 ,A Survey of Frequent Subgraph Mining Algorithms

[7] Hemant Kumar Sharma ,GRAPH BASED APPROACHES USED IN ASSOCIATION RULE MINING Thesis submitted in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering ,

[8] TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, OCTOBER 2013 1 An Iterative MapReduce based Frequent Subgraph Mining Algorithm Mansurul A Bhuiyan, and Mohammad Al Hasan

[9] International Journal of Engineering Inventions ISSN: 2278-7461, www.ijeijournal.com Volume 1, Issue 5 (September2012) PP: 60-63 60 , A Survey on Algorithms of Mining Frequent Subgraphs Maryam Gholami 1, Afshin Salajegheh,

[10] International Journal of Engineering Research & Technology (IJERT),Vol. 2 Issue 10, October – 2013,IJERTIJERT,ISSN: 2278-0181,IJERTV2IS100903 ,Review on Frequent Subgraph Pattern Mining Algorithms,Janki K. Bhut,M.E.C.E., Faculty of PG Studies & Research in Engg. (run by MEFGI), Rajkot ,India Mansi Vithalani

[11] The Knowledge Engineering Review, Vol. 00:0, 1-24. (2004) "Data Mining: Past, Present and Future", FRANS COENEN

[12] Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques", The Morgan Kaufmann Series in Data Management Systems (Second Edition) Chapter 9, Graph Mining, Social Network Analysis, and Multi relational Data Mining

[13] Huan, J., Wang, W. and Prins, J. 2003. Efficient Mining of Frequent Subgraph in the Presence of Isomorphism, In Proceedings of the 2003 International Conference on Data Graph-based Data Mining: Application of the Data Mining.

Mining, 549-552.Mining Frequent Patterns in Graph Databases 29

[14] Huan, J., Wang, W., Prins, J. and Yang, J. 2004a. SPIN: Mining Maximal Frequent Subgraphs from Graph Databases, In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 581–586.

[15] Huang, X. and Lai, W. 2006. Clustering Graphs for Visualization via Node Similarities, Visual Language and Computing 17, 225–253.

[16] Inokuchi, A., Washio, T. and Motoda, H. 2000. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data, In Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, 13–23.

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

[18] Inokuchi, A., Washio, T. and Motoda, H. 2003. Complete Mining of Frequent Patterns from Graphs:Mining Graph Data, Journal of Machine Learning, 50(3), 321–354.

[19] K.Lakshmi, T.Meyyappan, Efficient Algorithm for Mining Frequent Subgraphs (Static and Dynamic) based on gSpan, International Journal of Computer Applications (0975 – 8887) Volume 63– No.19, February 2013

[20] This chapter is from Social Media Mining: An Introduction. By Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. Cambridge University Press, 2014. Draft version: April 20, 2014.

[21] Karsten Borgwardt and Xifeng Yan, GRAPH MINING ,Interdepartmental Bioinformatics Group Max Planck Institute for Biological Cybernetics Max Planck Institute for Developmental Biology.