Survey on Graph Pattern Mining Approach

[1]Jignesh Patel,[2]Bhavesh A. Oza
[1]Student, [2]Assistant Professor
[1]Computer Science and Engineering, L.D. Engineering College, Ahmedabad, India
[2]Computer Department, L.D. Engineering College, Ahmedabad, India
[1]jigneshj204@gmail.com, [2]bhavesh.oza.123@gmail.com

***Abstract*— Aim of Data Mining is to extract significant and Useful knowledge from the Data. Data Stored in the database may be any type such as text, images and videos so on. Due to increasing the size of the data and storing such data is becoming complex. Data mining algorithms are facing the challenges for storing such data. Graph become more important in storing and visualizing this complicated data (i.e. chemical datasets, biological dataset, XML datasets, Social networks datasets, the web datasets etc.) In this paper it is discussed about different Algorithm used for Graph Mining and different techniques used for Graph Mining.**

Index Terms — Graph Mining, Graph Mining Algorithms, Graph Theory based Approach, Greedy Approach, Inductive Logic Programming, Frequent Pattern Growth, Apriori based Approach

I. INTRODUCTION

The primary goal of data mining is to extract statistically significant and useful knowledge from data. The data may be any form such as text, images, and videos and so on. Graph Mining is an active area of research. Frequent pattern mining is important part of which help to discover patterns which represents relations among discrete entities. In the graph related domain, the requirement of different applications is not very uniform. Thus, graph mining algorithms which work well in one domain may not work well in another. In this paper we first introduce basics of Graph Theory than different Techniques for the Graph mining and different Algorithms used for the Graph mining.

II. BASIC GRAPH THEORY

Graph

A Graph G (V, E) is defined as set vertices V (nodes) which are interconnected by a set of edges E (links). Below. figure (a) shows the simple graph with three nodes and three vertices.

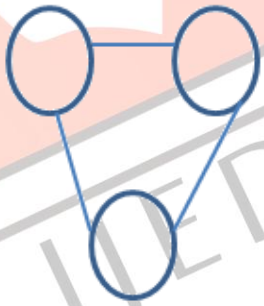

Figure (a) Graph G

Labeled Graph

A labeled Graph can be represented as
G (V,E,L_v,L_E,μ) , Where V is set of Vertex ,
$E \subseteq V \times V$ is a set of edges, L_V and L_E are sets of vertex and edge labels respectively, μ is label function that defines the mapping $V \rightarrow L_v$ and $E \rightarrow L_E$. 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. 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. Below figure (b) shows simple labeled graph with three nodes and three vertices.

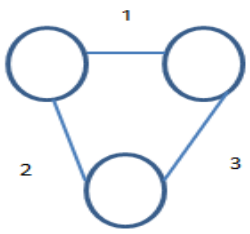

Figure (b) Labeled Graph G

Subgraph

Given two Graph G1 (V1, E1, LV1, LE1, μ1) and G2 (V2, E2, LV2, LE2, μ2), G1 is a subgraph of G2, if G1 satisfies:

(i) $V1 \subseteq V2$, and $\forall v \in V1$, $\mu1(v) = \mu2(v)$,

(ii) (ii) $E1 \subseteq E2$, and $\forall (u, v) \in E1$, $\mu1 (u, v) = \mu2 (u, v)$.

G1 is an induced subgraph of G2, if G1 further satisfies: $\forall u, v \in V1$, $(u, v) \in E1 \Leftrightarrow (u, v) \in E2$, in addition to the above conditions. G2 is also a supergraph of G1.

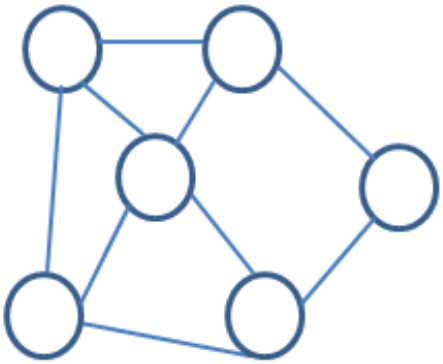

Figure (c) Graph G

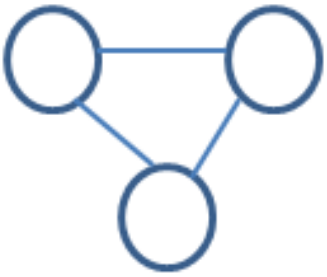

Figure (d) subgraph G1

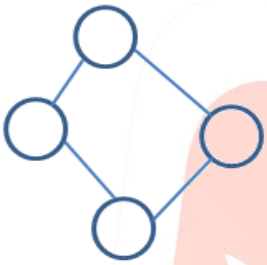

Figure (e) Subgraph G2

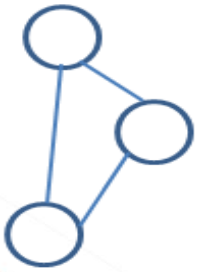

Figure (f) subgraph G3

As shown in figure (d), (e) and (f) different subgraphs for main graph G (figure (c)) is given. It may be possible that there are many subgraphs generated from a given supergraph.

Graph Isomorphism

A graph G1 (V1, E1, LV1, LE1, μ1) is isomorphic to another graph G2 (V2, E2, LV2, LE2, μ2), if and only if a bijection f: $V1 \rightarrow V2$ exists such that:

(i) $\forall u \in V1$, $\mu1 (u) = \mu2 (f (u))$,

(ii) $\forall (u, v) \in E1 \Leftrightarrow (f (u), f (v)) \in E2$,

(iii) $\forall (u, v) \in E1$, $\mu1 (u, v) = \mu2 (f (u), f (v))$.

The bijection f is an isomorphism between G1 and G2. A graph G1 is subgraph isomorphic to a graph G2, if and only if there exists a subgraph $g \subseteq G2$ such that G1 is isomorphic to g. In this case g is called an embedding of G1in G2.

III. BASIC APPROACH FOR GRAPH MINING

Various techniques and algorithm were developed by researchers. There are mainly three types of Graph Mining as given below

1. Greedy Method
2. Inductive Logic Programming
3. Graph theory based approaches

These approaches are categorized based on the approach used to search frequent subgraphs in large graph data set. Greedy Method as suggest it will use heuristic approach for find the solution. Inductive Logic Programming uses logic for representation of data and to search. Mathematical Graph theory based approaches mine a complete set of subgraphs mainly using a support or a frequency measure.

Different Graph mining algorithm as shown in fig1, basic methods for graph mining are representing as given in fig 1.

1. Greedy Based Algorithm

A Greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding optimum solution [16]. Different algorithm comes under this categories are SUBDUE and GBI.

SUBDUE [3] algorithm is focuses on the sub-graph which are not frequent but it is also compress the graph data, using heuristic approach. SUBDUE Algorithm uses Adjacency Matrix for Graph Representation. For Frequency Evolution it uses Minimum Description Length. It uses Breadth First Search Strategy for Searching purpose. Extension of SUBDUE algorithm are DB-SUBDUE, HDB-SUBDUE, EDB-SUBDUE, and RDB-SUBDUE.

HDB-SUBDUE Algorithm is extension of SUBDUE algorithm. Database SUBDUE (DB-SUBDUE) algorithm closely follows the SUBDUE algorithm in candidate generation using expanding substructure of size n to n+1 joins. EDB-SUBDUE algorithm is extension of DB-SUBDUE Algorithm. It was developed in an effort to handle certain aspects of graph like overlap and inexact graph match, cycles which may not considered in DB-SUBDUE algorithm. EDB-SUBDUE uses cursors to implement the beam for limiting the number substructures considered for next pass.

Another algorithm in this category is Graph Based Induction (GBI). As GBI uses Greedy based method to mining the given graph data. It uses Adjacency Matrix for Graph Representation. For Frequency Evolution it uses Minimum Description Length. It uses Breadth First Search Strategy for searching purpose. GBI efficiently extracts typical patterns from Directed Graph Data by stepwise pair expansion [6]. The Extension of GBI is as given classified as B-GBI, CIGBI, and DTGBI. Decision Tree Graph Based Induction (DTGBI) [7] is used for constructing a decision tree for graph structured data. [6] Chunkingless Graph Based Induction(CIGBI)[5] is extended version of the GBI which employs stepwise pair expansion to extract typical patterns from graph structured data, and can find overlapping patterns that cannot not be found in GBI.[7]

2. Inductive Logic Programming

Inductive Logic Programming (ILP) [17] is the intersection of Machine Learning and Logic Programming. ILP is characterized by the use of logic for the representation of multirelational data. ILP systems represent examples, background knowledge, hypotheses and target concepts in Horn clause logic. An ILP system can be characterized by the way the hypothesis space is structured and the search strategy used to explore the hypothesis space. ILP systems may learn a single concept or multiple concepts. ILP systems may be batch or incremental depending on how they accept examples. Different algorithm for graph mining that follows the Inductive Logic Programming is WARMER, FOIL, C-PROGOL.

C-Progol is an ILP system, characterized by the use of mode-directed inverse entailment and a hybrid search mechanism. [13] Inverse entailment is a procedure which generates a single, most specific clause that, together with the background knowledge, entails the observed data. The inverse entailment in C-Progol is mode-directed that is, it uses mode definitions. CProgol first computes the most specific clause which covers the seed example and belongs to the hypothesis language. The hypothesis space explored by CProgol consists of every hypothesis defined by the hypothesis language.

First Order Inductive Logic Algorithm (FOIL) is logic based algorithm. FOIL learns function-free Horn clauses, a subset of first-order predicate calculus. Given positive and negative examples of some concept and a set of background-knowledge predicates, FOIL inductively generates a logical concept definition or rule for the concept. The induced rule must not involve any constants (color(X,red) becomescolor(X,Y), red(Y)) or function symbols, but may allow negated predicates; recursive concepts are also learnable. Like the ID3 algorithm, FOIL hill climbs using a metric based on information theory to construct a rule that covers the data. Unlike ID3, however, FOIL uses a separate-and-conquer method rather than divide-and-conquer, focusing on creating one rule at a time and collecting uncovered examples for the next iteration of the algorithm.

WARMER specialized mining algorithms often concentrate on one type of database, for example databases of labeled undirected graphs. For different kinds of structures, modified algorithms are required. In ILP algorithms any structure can be expressed easily. The incorporation of background knowledge is also straightforward. The choice for traditional clause based query evaluation in WARMER two variables may have the same value during evaluation. In the subgraph mining algorithms two nodes in a subgraph cannot be mapped to one node in a database graph. The WARMER algorithm can be considered as a proof-of-concept of a framework; efficiency issues have not been given too much attention.

3. Graph Based Data Mining

Graph-based approaches are characterized by representation of multi- relational data in the form of graphs. Graph-based systems have been extensively applied to the task of unsupervised learning, popularly known as frequent subgraph mining and to a certain extent to supervised learning. Graph-based approaches represent examples, background knowledge, hypotheses and target concepts as graphs. These approaches include mathematical graph theory based approaches. Mathematical graph theory based approaches mine a complete set of subgraphs mainly using a support or frequency measure. Graph based approach can be classified as Apriori based approach and Frequent Pattern Growth Approach.

3.1 Apriori based Approach

A k-item set is frequent if and only if all of its sub-item sets are frequent. This implies that frequent item sets can be mined by first scanning the database to find the frequent 1-itemsets, then using the frequent 1-itemsets to generate candidate frequent 2-itemsets, and check against the database to obtain the frequent 2-itemsets. This process iterates until no more frequent k-itemsets can be generated for some k. This is the essence of the Apriori algorithm. Different Algorithm that follows these techniques is AGM, ACGM, FSG, and PATH. This Algorithms uses Adjacency Matrix for Graph Representation. It uses Canonical form for Frequency Evolution. It uses BFS for Searching Purpose.

AGM [21] (Apriori-based Graph Mining) in which the knowledge representation and the search operations are highly dedicated to the graph structure mining. It can efficiently discover all frequent patterns in terms of induced subgraphs contained in a dataset of labeled graphs.

ACGM is an extension of AGM algorithm. The algorithm can handle complete search of frequent connected subgraphs in a labeled graphs and perform the search highly efficiently.

FSG [20] finds all connected subgraphs that appear frequently in a large graph database and finds frequent subgraphs using the same level-by-level expansion strategy adopted by Apriori [2]. The key features of FSG are the following: (i) it uses a sparse graph

representation that minimizes both storage and computation; (ii) it increases the size of frequent subgraphs by adding one edge at a time, allowing it to generate the candidates efficiently; (iii) it incorporates various optimizations for candidate generation and frequency counting which enables it to scale to large graph databases; and (iv) it uses sophisticated algorithms for canonical labeling to uniquely identify the various generated subgraphs without having to resort to computationally expensive graph- and subgraph-isomorphism computations.

PATH algorithm is Apriori based algorithm which is used for mining a frequent pattern mining in graph data.

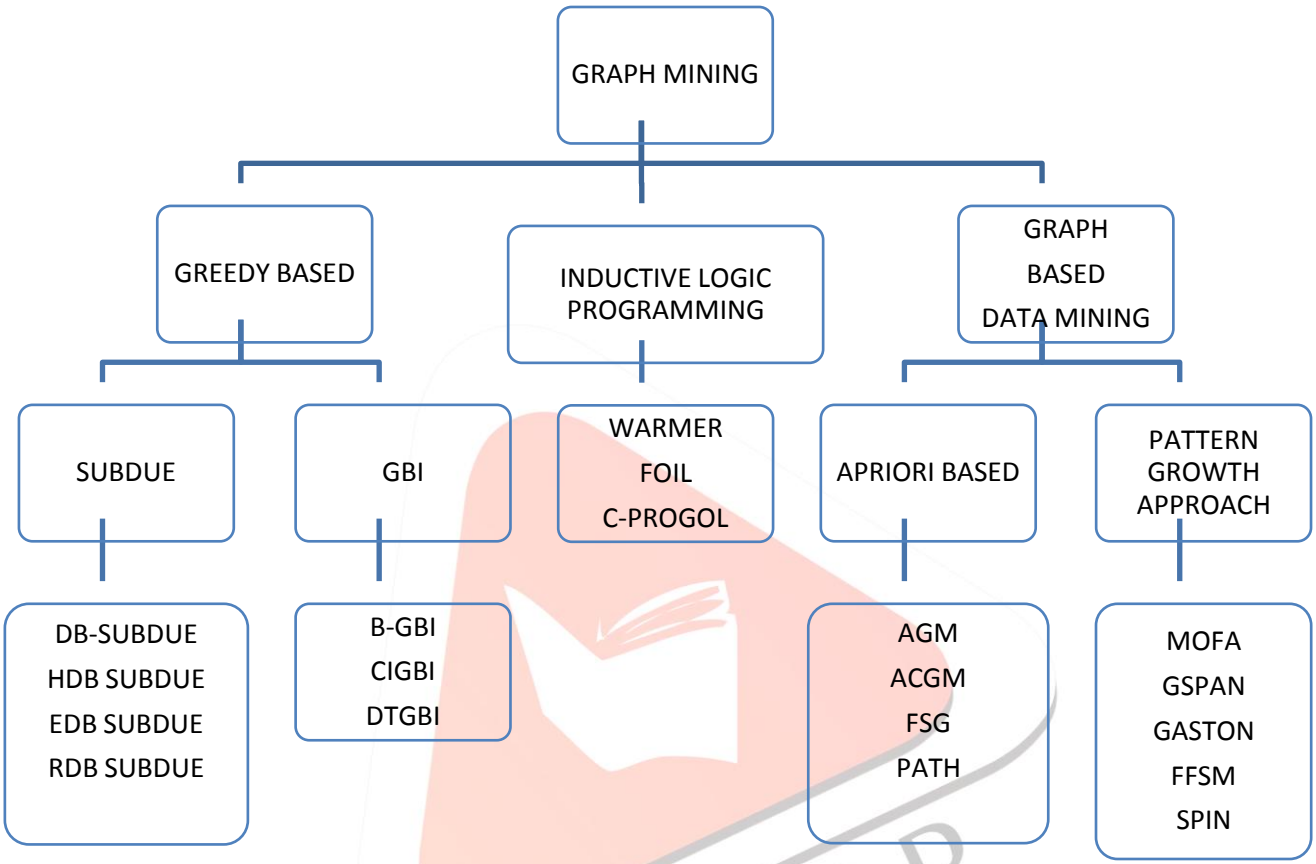

Figure 1: Categorization of Graph Mining

3.2 Frequent Pattern Growth Approach

FP-growth uses divide-and-conquer method for mining available graph. First step of this approach scan whole available database and derives list of frequent items then arrange them by frequency descending order. According to the frequency-descending list, the database is compressed into a frequent-pattern tree which retains the its set association information. The Frequent Pattern tree is mined by starting from each frequent length-1 pattern constructing its conditional pattern base then constructing its conditional Frequent Pattern tree, and performing mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree. Different Algorithm that follows these techniques is MOFA, GSPAN, GASTON, FFSM, and SPIN.

The goal of Molecular Fragment Mining (MOFA [11]) is to find discriminative fragments in a database of molecules, which are classified as either active or inactive. To achieve this goal, the algorithm presented in represents molecules as attributed graphs and carries out a depth first search on a tree of fragments. Going down one level in this search tree means extending a fragment by adding a bond and maybe an atom to it.

Graph-Based Substructure Pattern Mining (gSpan [12]). gSpan is the first algorithm that explores depth-first search (DFS) strategy for mining frequent subgraph from available graph data. In gSpan frequent subgraph generate without candidate generation and false positives pruning. It provides the growing and checking of frequent subgraphs into single procedure, so it accelerates the mining process.

GASTON [19] is new efficient frequent graph mining algorithm**.** GASTON focuses on enumerating simple frequent structures first, such as paths and trees, and then moves to identifying general graphs.

Fast Frequent Sub Graph Mining (FFSM) [18] targeting efficient subgraph testing and a better candidate subgraph enumeration scheme. The key features of our method are: (i) a novel graph canonical form and two efficient candidate proposing operations: FFSM Join and FFSM-Extension, (ii) an algebraic graph framework (suboptimal CAM tree) to guarantee that all frequent

subgraphs are enumerated unambiguously and (iii) completely avoiding subgraph isomorphism testing by maintaining an embedding set for each frequent subgraph.

SPIN [15] (Spanning tree based maximal graph mining) mines only maximal frequent subgraphs of large graph databases. SPIN algorithm offers good scalability to large graph structure data

IV. CONCLUSION

The main challenge to develop any algorithm for gaining high performance to enhance graph mining process is graph isomorphism which is the most costly step since it is an NP-complete problem in case of Graph. Hence, reducing the number of graph isomorphism is a promising direction which would save computational time. Due to increasing size of data and computational complexity of pattern in computer sciences the need for efficient graph mining algorithm is increasing. Still there is a scope of improvement in graph mining algorithm; the improvement can be in speed or sensitivity.

REFERENCES

[1] Xifeng Yan, Jiawei Han "Gspan: Graph-Based Substructure Pattern Mining" September 2002.
[2] Takashi Matsuda, Tadashi Horiuchi, Hiroshi Motoda and Takshi Washio "Extension of Graph-Based Induction for General Graph Structured Data" 2006
[3] Kuramochi, M. and Karypis, G., Finding frequent patterns in a large sparse graph. *Data Min. Knowledge Discovery*, 2005, 11(3), 243–271.
[4] Takashi Matsuda, Hiroshi Motoda, Tetsuya Yoshida and Takashi Washio "Mining Patterns from Structured Data by Beam-wise Graph-Based Induction" 2004
[5] Phu Chien Nguyen, Kouzou Ohara, Akira Mogi, Hiroshi Motoda, Takashi Washio "Constructing Decision Trees for Graph-Structured Data by Chunkingless Graph-Based Induction" 2005
[6] Warodom Geamsakul, Takashi Matsuda, Tetsuya Yoshida, Hiroshi Motoda and Takashi Washio "Constructing a Decision Tree for Graph Structured Data" 2005
[7] Nikhil S. Ketkar, Lawrence B. Holder, Diane J. Cook "Comparison of Graph based and Logic based Multi relational Data Mining" SIGKDD Explorations Volume 7, Issue 2, 2007
[8] Akihiro Inokuchi, Takashi Washio and Hiroshi Motoda "Complete Mining of Frequent Patterns from Graphs: Mining Graph Data"2004
[9] Akihiro Inokuchi, Takashi Washio, Kunio Nishimura and Hiroshi Motoda "A fast Algorithm for Mining Frequent Connected Subgraphs"2004
[10] Michihiro Kuramochi and George Karypis "An Efficient Algorithm for Discovering Frequent Subgraphs" IEEE Trans. Knowl. Data Eng. 16(9): 1038-1051, 2004
[11] C. Borgelt, M. R. Berthold. Mining molecular fragments: Finding Relevant Substructures of Molecules ICDM Conference, 2002.
[12] X. Yan, J. Han. Gspan: Graph-based Substructure Pattern Mining. ICDM Conference, 2002.
[13] Siegfried Nijssen, Joost N. Kok "A Quickstart in Frequent Structure Mining can make a Difference"2004
[14] Jun Huan, Wei Wang, Jan Prins" Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism" 2004
[15] J. Huan, W. Wang, J. Prins, J. Yang. Spin: Mining Maximal Frequent Subgraphs from Graph Databases. KDD Conference, 2004.
[16] http://en.wikipedia.org/wiki/Greedy_algorithm
[17] Muggleton, S., Inductive Logic Programming, Academic Press, 1992.
[18] Huan, J., Wang, W. and Prins, J., Efficient mining of frequent subgraphs in the presence of isomorphism. In Proceedings of the Third IEEE International Conference on Data Mining, ICDM '03, IEEE Computer Society, Washington DC, USA, 2003.
[19] Nijssen, S. and Kok, J., A quickstart in frequent structure mining can make a difference. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 647–652.
[20] Yan, X. and Han, J., gSpan: graph-based substructure pattern mining. In Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM '02, IEEE Computer Society, Washington, DC, USA, 2002, p. 721.
[21] A. Inokuchi, T. Washio, and H. Motoda, \Complete mining of frequent patterns from graphs: Mining graph data," Mach. Learn., vol. 50, no. 3, pp. 321{354, 2003.