A Survey on Assorted Approaches to Graph Data Mining

D. Kavitha
Sr.Asst. Professor, Dept of IT
PVP Siddhartha Inst.of Tech
Vijayawada-520007, A.P., India

B.V. Manikyala Rao
Sr.Asst. Professor, Dept of IT
PVP Siddhartha Inst. Tech
Vijayawada-520007, A.P., India

V.Kishore Babu
I M.Tech I semester, Dept of IT
University of Hyderabad, AP.,
India

ABSTRACT
Graph mining has become a popular area of research in recent years because of its numerous applications in a wide variety of practical fields, including computational biology, sociology, software bug localization, keyword search, and computer networking. Different applications result in graphs of different sizes and complexities. Graph mining is an important tool to transform the graphical data into graphical information. We investigate recurring patterns in real-world graphs, to gain a deeper understanding of their structure. We can extract normal and abnormal subgraphs thereby detecting suspicious nodes and outliers in the existing graphs. In this paper we present a survey of various approaches to mine the graphs. These are used to extract patterns, trends, classes, and clusters from graphs.

Keywords
Data Mining, Graph Mining, Sub Graph, Frequent Sub Graph.

1. INTRODUCTION
Data mining is the extraction of novel and useful knowledge from data. Data mining aims at discovering interesting and previously unknown patterns from data sets. In general, the data can take many forms from a single, time-varying real number to a complex interconnection of entities and relationships. While graphs can represent this entire spectrum of data, they are typically used when relationships are crucial to the domain. Graph-based data mining is the extraction of novel and useful knowledge from a graph representation of data. Graph mining uses the natural structure of the application domain and mines directly over that structure. The most natural form of knowledge that can be extracted from graphs is also a graph. Therefore, the knowledge, sometimes referred to as patterns, mined from the data are typically expressed as graphs, which may be subgraphs of the graphical data, or more abstract expressions of the trends reflected in the data.

Among various kinds of graph patterns, frequent substructures are very basic ones that can be discovered in a set of graphs. They are useful at characterizing graph sets, discriminating different groups of graphs, classifying and clustering graphs, and building graph indices. A number of varied techniques and methodologies have been applied to mining interesting sub graph patterns from graph datasets. These include mathematical graph theory based approaches like FSG and gSpan, greedy search based approaches like Subdue or GBI, inductive logic programming (ILP) approaches like WARMR, inductive database approaches like MolFea and kernel function based approaches.

1.1 Definition
A graph is a pair G = (V, E) where V is a set of vertices and E is a set of edges. Edges connect one vertice to another and can be represented as a pair of vertices. Typically each edge in a graph is given a label. Edges can also be associated with a weight.

We denote the vertex set of a graph g by V (g) and the edge set by E(g). A label function, L, maps a vertex or an edge to a label. A graph g is a subgraph of another graph g' if there exists a subgraph isomorphism from g to g'. (Frequent Graph) Given a labeled graph dataset, D = {G1, G2, . . . , Gn}, support (g) [or frequency(g)] is the percentage (or number) of graphs in D where g is a subgraph. A frequent (sub) graph is a graph whose support is no less than a minimum support threshold, min support.

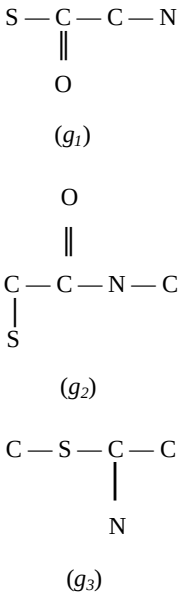

Fig 1: Sample graph dataset.

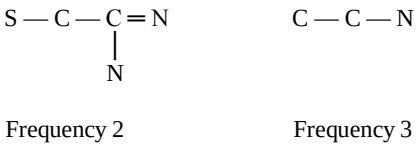

Fig 2: Frequent graphs

2. BASIC APPROACHES OF GRAPH MINING

These are the three basic approaches of graph mining

1. Incomplete beam search 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 use heuristics to find the solution .Inductive logic programming(ILP) mainly 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. In this section , some major studies in each category are described.

Graph Mining Algorithms as shown in the Fig3, basic methods of graph mining and research work done on those till now is presented. Greedy search based approaches use heuristics to evaluate the solution. In greedy method optimal solution is constructed in stages.

These approaches are categorized based on the approach used to search frequent sub graphs in large graph data set. Greedy method use heuristics to find the solution. Inductive logic programming (ILP) mainly uses logic for representation of data and to search.

At each stage we make a decision that appears to be best at the time. A decision made at one stage is not changed in a later stage, so each decision should assure feasibility. The two pioneering works in the field are Subdue[2] and GBI[3]. Subdue uses MDL-based compression heuristics, and GBI uses an empirical graph size-based heuristic. The empirical graph size definition depends on the size of the extracted patterns and the size of the compressed graph.

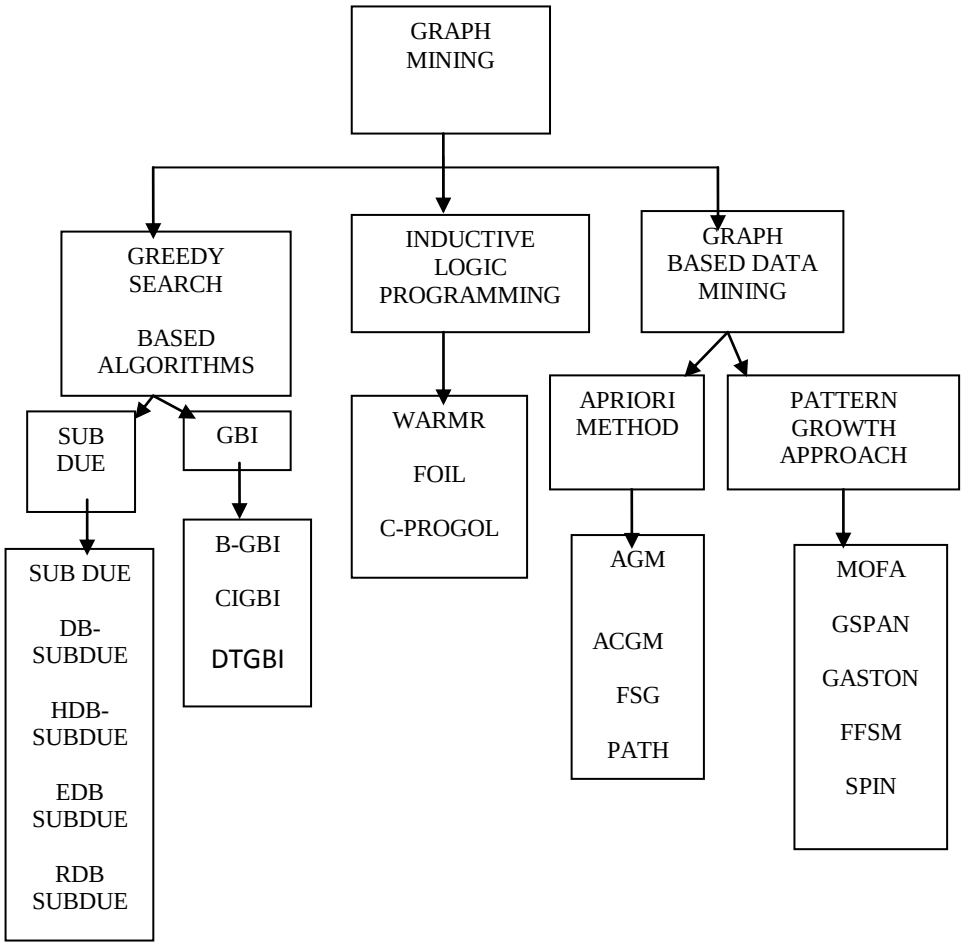

Fig 3: Categorization of Graph Mining.

Subdue[2] is a graph-based relational learning system. Inputs to the Subdue system can be a single graph or a set of graphs. The graphs can be labeled or unlabeled. Subdue outputs substructures that best compress the input dataset according to the Minimum Description Length (MDL) principle. Subdue performs a computationally-constrained beam search which begins from substructures consisting of all vertices with unique labels. The substructures are extended by one vertex and one edge or one edge in all possible ways, as guided by the example graphs, to generate candidate substructures.

Subdue's search is guided by the MDL principle given in Eq. where DL(S) is the description length of the substructure being evaluated, DL (G|S) is the description length of the graph as compressed by the substructure, and DL(G) is the description length of the original graph. The best substructure is the one that minimizes this compression ratio:

$$Compressio\,n = \frac{DL(S) + DL(G/S)}{DL(G)}$$

2.1 Graph-Based Induction (GBI)

Extracts typical patterns from graph data by stepwise pair expansion (pair wise chunking) [3]. It is very efficient because of its greedy search strategy but at the same time it suffers from the incompleteness of search. Improvement is made on its search capability without imposing much computational complexity by 1) incorporating a beam search, 2) using a different evaluation function to extract patterns that are more discriminatory than those simply occurring frequently, and 3) adopting canonical labeling to enumerate identical patterns accurately. This new algorithm, now called Beam-wise GBI[5], B-GBI for short, CI-GBI(recent one).

2.2 Logic-Based Mining

Popularly known as inductive logic programming (ILP) [7], is characterized by the use of logic for the representation of structural data. ILP systems represent examples, background knowledge, hypotheses, and target concepts in Horn clause logic. The core of ILP is the use of logic for representation and the search for syntactically legal hypotheses constructed from predicates provided by the background knowledge. Logic-based approaches rely on the prior identification of the predicate or predicates to be mined. ILP is formalized as follows [7]. Given the background knowledge B and the evidence (the observed data) E where E consists of the positive evidence E+ and the negative evidence E¡, ILP finds a hypothesis H such that the following "normal semantics" conditions hold.

1. Posterior Satisfiability : B ^ H ^ E¡ j= = 2
2. Posterior Sufficiency: B ^ H j= E+

Where 2 is "false", and hence j= = 2 means that the theory is satisfiable. In case of ILP, intentional definitions are ILP systems such as FOIL, CProgol, Golem and WARMR have been extensively applied to supervised learning and to a certain extent to unsupervised learning.

2.3 Graph Theory Based Approaches

It mines a complete set of subgraphs mainly using a support or a frequency measure. Mining the frequent substructures usually consists of two steps. In the first step, we generate frequent substructure candidates. In the second step the frequency of each candidate frequency is checked. Graph based approaches are mainly divided into two categories.

Graph based approaches are mainly of two types. They are

Apriori-based approach and Pattern growth approach.

2.3.1 Apriori-Based Algorithms

This uses apriori property which shares similar characteristics with the Apriori-based itemset mining [4]. In this approach, candidates are generated level wise and uses BFS strategy. To determine whether a size-(k + 1) graph is frequent, it has to 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-based approach usually has to complete the mining of size-k subgraphs. The initial frequent substructure mining algorithm, called AGM[8], was proposed by Inokuchi et al. The Apriori property is also used by other fre-quent substructure discovery algorithms such as FSG [9] and the path-join algorithm [10]. All of them require a join operation to merge two (or more) frequent substructures into one larger substructure candidate. They distinguish themselves by using different building blocks: vertices, edges, and edge-disjoint paths. Various Apriori based algorithms are AGM, FSG and PATH. The Apriori-based approach has to use the breadth-first search (BFS) strategy because of its level-wise candidate generation. Before mining any size-(k + 1) subgraph, the Apriori-based approach usually has to complete the mining of size-k subgraphs. In the context of frequent substructure mining, Apriori-based algorithms have two kinds of considerable overheads: (1) joining two size-k frequent graphs to generate size-(k + 1) graph candidates, and (2) checking the frequency of these candidates separately. These overheads constitute the performance bottleneck of Apriori-based algorithms.

2.3.2 Pattern Growth Approach

To avoid the overheads incurred in Apriori-based algorithms, non-Apriori-based algorithms such as gSpan , MoFa , FFSM , SPIN , and Gaston have been developed recently. These algorithms are inspired by PrefixSpan, TreeMinerV, and FREQT at mining sequences and trees, respectively. All of these algorithms adopt the pattern growth methodology [6], which intends to extend patterns from a single pattern directly. In Pattern growth approach A graph g can be extended by adding a new edge e. The newly formed graph is denoted by g _x e. Edge e may or may not introduce a new vertex to g. Pattern Growth extends a frequent graph in every possible position. 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 any more. It can use both breadth first search as well as depth first search.

3. COMPARISON TO FREQUENT SUB STRUCTURE MINING APPROACHES

Greedy search-based approaches use heuristics to evaluate the solution. Subdue typically produces a smaller number of substructures that best compress the graph dataset and potentially of great interest. (that can provide important insights about the domain). Subdue can accommodate a free-form graph representation. Subdue is preferred over FSG or gSpan when data is presented in one large graph or when a pattern is dominantly present in small or medium size datasets.

Inductive logic programming (ILP) systems work well with complex data. They represent databases in first order logic (FOL), not as graphs, and perform induction on the world of logic statements. Logic based systems make more efficient use of background knowledge and are better at learning semantically complex concepts. Logic-based approaches allow the expression of more complicated patterns involving, for example, recursion, variables, and constraints among variables. These representational limitations of graphs can be overcome, but at a computational cost. Inductive logic programming methods face some limitations because of the explicit encoding of structural information and the prohibitive

size of the search space. Mathematical graph methods are guaranteed to find all subgraphs that satisfy the user-specified constraints. These systems typically generate a large number of substructures, which by themselves provide relatively less insight about the domain. For large databases and those that exhibit a high degree of randomness, FSG or gSpan will likely be better choices.

4. CONCLUSION

Graph-based data mining (GDM) is a fast-growing field due to the increasing interest in mining the relational aspects of data. We have described several approaches to GDM including logic-based approaches in ILP systems, graph-based frequent subgraph mining approaches and a heuristic-based learning approach in Subdue. We can conclude that greedy search based mining algorithms tend to explore the concept hypothesis space more efficiently than logic-based algorithms, which is essential for mining structurally large concepts from databases. However, logic based systems make more efficient use of background knowledge and are better at learning semantically complex concepts. Graph-based approaches are more data-driven, identifying any portion of the graph that has high support.

Many of the graph-theoretic operations inherent in GDM are NP-complete or definitely not in P, scalability is a constant challenge. With the increased need for mining streaming data, the development of new methods for incremental learning from dynamic graphs is important.

5. REFERENCES

[1] Mining Graph Data – Diane J Cook, Lawrence B Holder – 2007 pages.

[2] Subdue [2]-- D. J. Cook and L. B. Holder. Substructure discovery using minimum escription length and background knowledge. J. Artif. Intell. Res. (JAIR), 1:231–255, 1994.

[3] Gbi [3]-- T. Matsuda, T. Horiuchi, H. Motoda, and T. Washio. Extension of graph-based induction for general graph structured data. In PAKDD, pages 420–431, 2000.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proceedings of 1994 International Conference Very Large Data Bases (VLDB'94), pp. 487–499, Santiago, Chile, Sept. 1994

[5] B-Gbi[5]--T. Matsuda, H. Motoda, T. Yoshida, and T. Washio. Mining patterns from structured data by beam-wise graph-based induction. In Proceedings of the 5th International Conference on Discovery ScienceDiscoveery (DS 2002), Vol. 2534 of Lecture Notes in Computer Science, pp. 422–429. Springer, 2002

[6] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Proceedings of 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD'00), pp. 1–12, Dallas, TX, May 2000.

[7] S. Muggleton, ed. Inductive Logic Programming. Academic, London, 1992. S. Muggleton. Stochastic logic programs. In L. de Raedt, ed. Advances in Inductive Logic Programming. IOS Press, Amsterdam, 1996.

[8] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In Proceedings of 2000 European Symposium Principle of Data Mining and Knowledge Discovery (PKDD'00), pp. 13–23, Lyon, France, Sept 2000

[9] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In Proceedings of 2001 International Conference on Data Mining (ICDM'01), pp. 313–320, San Jose, CA, Nov. 2001.

[10] N. Vanetik, E. Gudes, and S. E. Shimony. Computing frequent graph patterns from semistructured data. In Proceedings of 2002 International Conference on Data Mining (ICDM'02), pp. 458–465, Maebashi, Japan, Dec. 2002.