Graph Mining Sub Domains and a Framework for Indexing – A Graphical Approach

K. Vivekanandan
Professor
BSMED
Bharathiar University
Coimbatore
India

A. Pankaj Moses Monickaraj
(Correspoding author)
Doctoral Scholar
Department of Computer Science
Bharathiar University
Coimbatore
India

D. Ramya Chithra
Assistant Professor
Department of Computer Science
Bharathiar University
Coimbatore
India

ABSTRACT: **Graphs are one of the popular models for effective representation of complex structured huge data and the similarity search for graphs has become a fundamental research problem in Graph Mining. In this paper initially, the preliminary graph related basic theorems are brushed and showcased on with various research sub domains such as Graph Classification, Graph Searching, Graph Indexing, and Graph Clustering. These are discussed with few of the most dominant algorithms in their respective sub domains. Finally a model is proposed along with various algorithms with their future projection.**

Keywords: Graph; Graph Mining; Graph Classification; Graph Searching; Graph Indexing; Graph Clustering

I. INTRODUCTION

The primary goal of data mining is to extract statistically significant and useful knowledge from data [1][2][3] which may be in any of the forms like image, text, links, vectors, tables and so on. Various forms of representing the data are available for both structured and semi-structured form. But both forms of data can be represented by a graph. Naturally this caused the vast area of research known as Graph Mining.

Raymond Kosala, Hendrik Blockeel in "Mining Research: A Survey", explore the connection between the web mining categories, and related agents. Interesting fact is graph structure occurs everywhere in the web mining research which is still at the budding stage [25].

From table I. , web graph is a form of representation propelled in web structure and usage mining research. In this paper, we show case the various sub domains in the field of graph mining and a model to index, update and upgrade without performance degradation.

II. RELATING GRAPH SUBSTRUCTURES WITH MATHEMATICS THEOREMS

A Graph is defined to be a set of vertexes (nodes) which are interconnected by a set of edges (links) [23].

TABLE I. Web Mining category [25]

	Web Mining			
	Web Content Mining		Web Structure Mining	Web Usage Mining
	IR View	DB View		
View of Data	· Unstructured · Semi structured	· Semi structured · Web site as DB	· Links structure	· Interactivity
Main Data	· Text documents · Hypertext documents	· Hypertext documents	· Links structure	· Server logs · Browser logs
Representation	· Bag of words, n-grams · Terms, phrases · Concepts or ontology · Relational	· Edge-labeled graph (OEM) · Relational	· Graph	· Relational table · Graph
Method	· TFIDF and variants · Machine learning · Statistical (including NLP)	· Proprietary algorithms · ILP · (Modified) association rules	· Proprietary algorithms	· Machine Learning · Statistical · (Modified) association rules
Application Categories	· Categorization · Clustering · Finding extraction rules · Finding patterns in text · User modeling	· Finding frequent sub-structures · Web site schema discovery	· Categorization · Clustering	· Site construction, adaptation, and management · Marketing · User modeling

Theorem: 1 The graph G = (V,E), where V = $\{v_1, \ldots, v_n\}$ and E = $\{e_1, \ldots, e_m\}$, satisfies

$$\sum_{i=1}^{n} d(v_i) = 2m$$

Corollary: Every graph has an even number of vertices of odd degree. [Figure 1]

The total sum of degree of each vertex in a graph is equal to twice the number of edges. From the number of vertices and their degrees, the number of connectivity which may be present among the vertices in the graph can be predicted which would be more useful while indexing and searching.

Theorem: 2 The vertex v is a cut vertex of the connected graph G if and only if there exist

two vertices u and w in the graph G such that (i) $u \neq v$, $v \neq w$ and $u \neq w$, but (ii) v is on every u–w path.

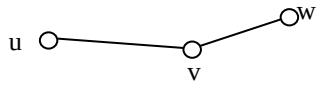

Figure 1

In this graph, u is connected to v and v is connected to w. If v is removed the connectivity is incomplete. Hence, here v is called cut vertex.

Theorem: 2 play a key role in graph classification, soon after the data are categorized according to the various conditions. The association among the content in the graph can be effectively refined by this theorem.

Theorem: 3 Every vertex of a graph G belongs to exactly one component of G. Similarly, every edge of G belongs to exactly one component of G.

Theorem:3 role comes in a graph database, when updates has to be inserted into an index, data features should be abstracted and categorized such that they can be inserted at right position in the index. Here, updates refer to the vertices and their relationship refers to the edges.

III. GLIMPSES OF RESEARCH SUB DOMAINS IN GRAPH MINING:

Using graphs as a strong method to model complex datasets, various disciplines have been recognized by various researchers in domains such as chemical [23, 24, 25], computer vision [5, 6], image and object retrieval [6, 9], and machine learning [8, 7, 9].

Enormous amount of graph data found throughout, many data mining process can be imparted but for a graph databases, it comes in different dimension. Graph classification [12], graph indexing [10][11], and graph clustering [13][18], sub graphs patterns as features are some of the major key areas of research in Graph Mining.

For example, biological structures can be stored as graphs, and in order to classify these structural graphs as active or inactive format, number of subgraph patterns are needed to build classification model [14], [15], [16].

Subgraph Isomorphism, Video Indexing, Correlated Graph Pattern Mining, Optimal Graph Pattern Mining, Approximate Graph Pattern Mining, Graph Pattern Summarization, Graph Classification, Graph Clustering, Graph Indexing, Graph Searching, Graph Kernels, Link Mining, Web Structure Mining, Work-Flow Mining, Biological Network Mining, , Improving Storage Efficiency Of Semi-Structured Databases, Efficient Indexing And Web Information Management are also some of the sub domains [23] in the field of graph mining of which few are discussed.

A. Graph Classification:

Xifeng Yan and Jiawei Han has proposed GSpan [29] (graph-based Substructure pattern mining) finds frequent substructures without candidate generation. Subgraph Mining is recursively called to grow the graphs and to find all their frequent descendants. It terminates its search when the support of a graph is less than the minimum support. It builds a new lexicographic order and maps each graph to a unique minimum Depth First Search code as its canonical label. Through this lexicographic order, it adopts the depth First search strategy to mine frequent connected sub graphs and uses a sparse adjacency list representation to store graphs.

Let {A,B,C….} be the vertices and {a,b,c….} be the connecting edges. The algorithm discovers A-^{a}A and then A-^{a}B until all frequent subgraph are discovered.

Michihiro Kuramochi and George Karyused proposed Frequent Sub Graph (FSG) [12] to find all connected subgraphs that appear frequently in a large graph database. It finds frequent subgraphs using the same level-by-level expansion adopted in Apriori [17][24].

Key features of FSG are

(1) uses a sparse graph representation minimizing both storage and computation.

(2) increases the size of frequent subgraphs by adding one edge at a time, allowing to generate the candidates efficiently

(3) uses simple algorithms of canonical labeling and graph isomorphism which work efficiently for small graphs

(4) incorporates various optimizations for candidate generation and counting which allow it to scale to large graph databases.

B. Graph Clustering:

Brian Kulis et.al has proposed a kernel approach [13] unify vector-based and graph-based approaches. The objective function for semi-supervised clustering based on Hidden Markov Random Fields, with squared Euclidean distance and a certain class of constraint penalty functions, are expressed as a special case of the weighted kernel k-means objective. It is an extension of probabilistic framework for semi supervised clustering with pairwise constraints.

This paper was based on Hidden Markov Random Fields [18]. This framework with semi-supervised clustering algorithm SS-Kernel-k means unifies vector-based and graph-based approaches using a kernel approach.

SS-Kernel-kmeans(S, k, M, C, W, tmax)

(1) Form the matrix $K = S + W$.

(2) Diagonal-shift K by adding σI to guarantee positive definiteness of K.

(3) Get initial clusters $\{\pi_c\}^{k}_{c}=1$ using constraints.

(4) Return $\{\pi_c^{(0)}\}^{k}_{c}=1$ = Kernel-kmeans (K, k, tmax,1, $\{\pi_c^{(0)}\}^{k}_{c=1}$, where 1 is the vector of all ones

C. Graph Searching:

Rosalba Giugno and Dennis Shasha has proposed an algorithm GraphGrep [20] which is an application-independent method for querying graphs, (i.e) for finding all the occurrences of a subgraph in a graph database. The interface is a regular expression graph query language Glide (a graph linear query language) the combined features from XPath and Smart acts as interface. Glide incorporates both single node and variable-length.

Steps of GraphGrep are:

(1) Build the database to represent the graphs as sets of paths

(2) Filter the database based on the submitted query to reduce the search space

(3) Perform exact matching.

The algorithm first extract all Cycle structures in a graph g, then extract all Star structures, and finally, identify the remaining structures as either Line structures or as attachments to the extracted basic structures.

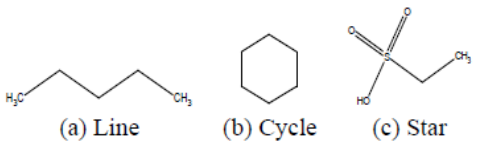

Fig. 1. Basic Structure [20]

Haoliang Jiang et.al in this paper [21] describes the transformation of a graph into a string representation, or capturing the semantics in graph data. The meaningful components in graph structures are found and are used for the most basic units in sequencing. It reduces the size of resulting sequences, but also enables semantic-based searching. Here it is approached with chemical compounds which can also be tested with protein structures as well.

D. Graph Indexing:

There are plenty of research efforts to solve the sub graph isomorphism problem for a large graph database by utilizing graph indexes of which few are listed below:

In this paper [28], Peixiang Zhao et.al proposed a new cost-effective graph indexing method based on frequent tree-features of the graph database. Effectiveness and efficiency are analyzed in three critical aspects: feature size, feature selection cost, and pruning power. To achieve better pruning, frequent tree-features (Tree),a small number of discriminative graphs (¢) are selected on demand. It has two implications: (1) the index construction by (Tree+¢) is efficient, and (2) the graph containment query processing by (Tree+¢) is efficient.

Wook Shin Han et.al has proposed iGraph [19], a framework with binary executables , heap files, B+-trees, inverted indexes, disk-based prefix trees, binary large object (BLOB) files, an LRU buffer manager, m-way posting list intersection, and external sorting.

Xifeng Yan et.al has proposed an algorithm gindex [10] which makes use of frequent substructure as the basic indexing feature.

Frequent substructures are ideal candidates as they explore the intrinsic characteristics of the data. Two techniques such as size-increasing support con straint and discriminative fragments, are introduced to reduce the size of index structure.

The design and implementation of gIndex algorithm is segmented to 5 sub sections:

(1) Discriminative fragment selection
(2) Index construction
(3) Search
(4) Verification and
(5) Incremental maintenance.

James Cheng et.al has proposed FG-index [11], novel indexing technique that constructs a nested inverted-index based on the set of Frequent subGraphs (FGs). For a graph query, FG-index returns the exact set of query answers without performing candidate verification. In case, if the query is an infrequent graph, the algorithm a candidate answer set as output which is close to the exact answer set.

The algorithm is divided into three parts:

(1) computation of T (where T is a sub graph)
(2) construction of the core FG-index,
(3) creation of Edge-index.

IV. A FRAME WORK FOR INDEXING:

Irrespective of the type of graph data, there are various mine at once algorithms to build index for any large database. After indexing, due to various updates, the index has to be restructured such that the retrieving efficiency or speed doesn't get degraded (performance). If the changes cause major performance issues, then the complete work has to be indexed from the scratch which is quite expensive and tedious.

Therefore, we propose a framework which can index with its features and update the right features at right place through search algorithms at the index.

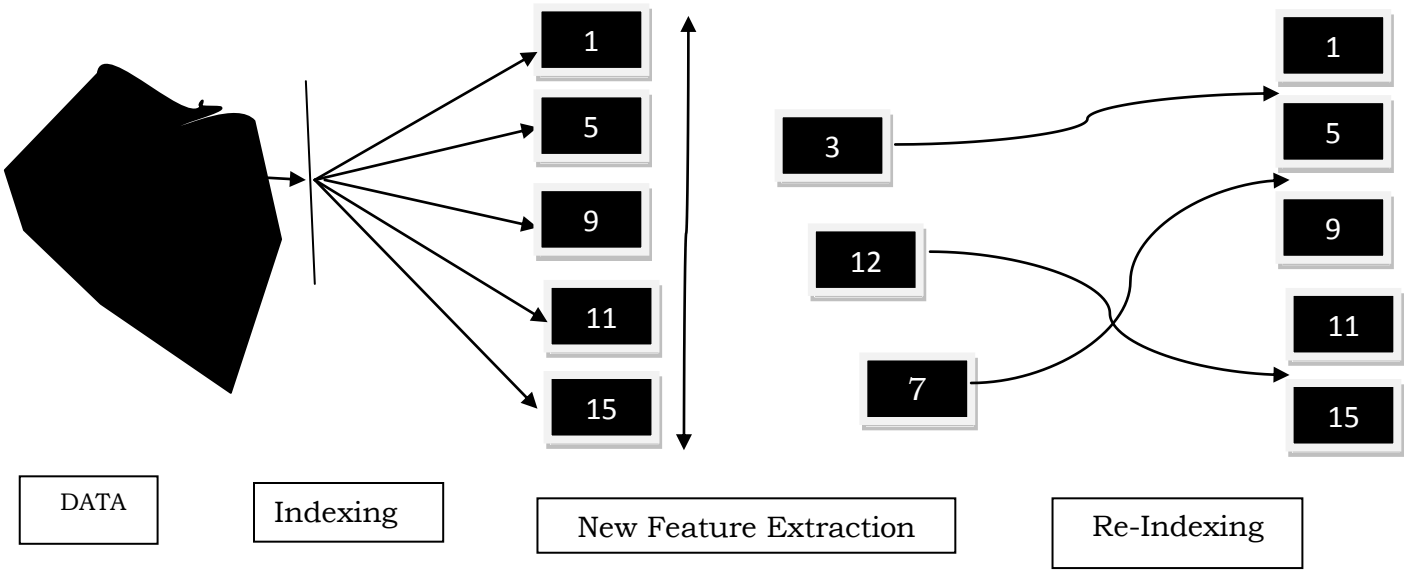

Fig. 2. ARCHITECTURAL FRAME WORK FOR GRAPH INDEXING

Mine at once indexing algorithm index any type of data. Most of the algorithms are extension or improved version of some basic techniques so a hybrid model for indexing can be built, such that indexing will be much more effective.

To upgrade the indexes with updates, the feature mining is one of the technique, in which iterative sub graph feature mining algorithm [22] is more effective in finding the upgraded parts in a graph.

Once the changes in the graph are extracted by any of the feature mining technique, right place has to be found out where the feature has to be pushed into or popped off from the index for which the basic searching techniques like BFS, DFS, G-string can be used to find the exact location where the particular extracted feature has to be pushed or popped into or off the index.

V. CONCLUSION

This paper includes the various areas of research fields in graph mining along with a model or architectural Framework which includes Graph Searching, Indexing and feature mining techniques. As there are plenty of mine at once algorithm, according to type of the data, effective indexing can be done by imparting the particular type of algorithm for particular data. Irrespective to the field of any applications, this model can act as a core algorithmic structure for effective indexing and upgrading the index.

REFERENCES

[1] R. N. Chittimoori, L. B. Holder, and D. J. Cook. Applying the SUBDUE substructure discovery system to the chemical toxicity domain. In Proc. of the 12th International Florida AI Research Society Conference,pages 90–94, 1999

[2] A. Srinivasan, R. D. King, S. Muggleton, and M. J. E. Sternberg. Carcinogenesis predictions using ILP. In S. Dˇzeroski and N. Lavraˇc, editors, Proc. of the 7th International Workshop on Inductive Logic Programming, volume 1297, pages 273–287. Springer-Verlag, Berlin, 1997.

[3] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds, Proc. of the 4th International Conference on Knowledge Discovery and Data Mining, pages 30–36. AAAI Press, 1998

[4] A. Srinivasan, R. D. King, S. H. Muggleton, and M. Sternberg. The predictive toxicology evaluation challenge. In Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI), pages 1–6. Morgan-Kaufmann, 1997.

[5] H. K¨alvi¨ainen and E. Oja. Comparisons of attributed graph matching algorithms for computer vision. In Proc. of STEP-90, Finnish Artificial Intelligence Symposium, pages 354–368, Oulu, Finland, June 1990.

[6] D. A. L. Piriyakumar and P. Levi. An efficient A* based algorithm for optimal graph matching applied to computer vision. In GRWSIA-98, Munich, 1998.

[7] C.-W. K. Chen and D. Y. Y. Yun. Unifying graph-matching problem with a practical solution. In Proceedings of International Conference on Systems, Signals, Control, Computers, September 1998.

[8] L. Holder, D. Cook, and S. Djoko. Substructure discovery in the SUBDUE system. In Proc. of the Workshop on Knowledge Discovery in Databases, pages 169–180, 1994.

[9] K. Yoshida and H. Motoda. CLIP: Concept learning from inference patterns. Artificial Intelligence, 75(1):63–92, 1995.

[10] X. Van, P. S. Yu, and J. Han. "Graph indexing: a frequent structure-based approach," In Proc. of the ACM SIGMOD international conference on Management of data, pages 335-346, 2004.

[11] J. Cheng, Y. Ke, W. Ng, and A. Lu. "Fg-index: towards verification-free query processing on graph databases," In Proc. Of the ACM SIGMOD international conference on Management of data, pp. 857-872,2007.

[12] M. Deshpande, M. Kuramochi, and G. Karypis. "Frequent substructure discovery,"Proc. 3rd IEEE Int'l Conf. Data Mining (ICDM '02), 2001.

[13] Brian Kulis, Sugato Basu, Indeljit Dhillon and Raymond Mooney "Semi-supervised graph clustering: a kernel approach " in: . Proc. Proceedings of the 22nd international conference on Machine learning ICML '05

[14] C.-W. K. Chen and D. Y. Y. Yun. Unifying graph-matching problem with a practical solution. In Proceedings of International Conference on Systems, Signals, Control, Computers, September 1998.

[15] R. N. Chittimoori, L. B. Holder, and D. J. Cook. Applying the SUBDUE ubstructure discovery system to the chemical toxicity domain. In Proc. of the 12th International Florida AI Research Society Conference, pages 90–94, 1999.

[16] V. A. Cicirello. Intelligent retrieval of solid models. Master's thesis, Drexel University, Philadelphia, PA, 1999.

[17] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, Proc. of the 20th Int. Conf. on Very Large Databases (VLDB), pages 487–499. Morgan Kaufmann, September 1994

[18] Basu, S., Bilenko, M., & Mooney, R. (2004). A probabilistic framework for semi-supervised clustering. Proc. 10th Intl. Conf. on Knowledge Discovery and Data Mining.

[19] Wook Shin Han, Jinsoo Lee, Minh Duc Pham, Jeffrey Xu Yu "iGraph: A Framework for Comparisons of Disk Based Graph Indexing Techniques", The 36th International Conference on Very Large Data Bases, September13-17,2010, Singapore. Proceedings of the VLDB Endowment, Vol. 3,

[20] Rosalba Giugno, Dennis Shasha, GraphGrep: A Fast and Universal Method for Querying Graphs, Pattern Recognition, 2002.

[21] Haoliang Jiang, Haixun Wang, Philip S. Yu, Shuigeng Zhou, GString: A Novel Approach for Efficient Search in Graph Databases, IEEE Pattern Recognition, 2002.

[22] Dayu Yuan, Prasenjit Mitra , Huiwen Yu, C. Lee Giles Iterative Graph Feature Mining for Graph Indexing, 2012 IEEE 28th International Conference on Data Engineering.

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

[24] Chen, M.S., Han, J. and Yu, P.S. 1996. Data Mining: An Overview from Database Perspective, IEEE Transaction on Knowledge and Data Engineering 8, 866–883.

[25] Raymond Kosala, Hendrik Block, Web Mining Research: A Survey, SGIKDD, Explorations, ACM, 2000.

[26] Akihiro Inokuchi, Takashi Washio and Hiroshi Motoda, An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data Currently being in Tokyo Research Institute, IBM, 1623-14 Shimotsuruma, Yamatoshi, Kanagawa, 242-8502, Japan.

[27] [27] Huan, 1., Wang, W., Prins, 1. Efficient mining of frequent subgraphs in the presence of isomorphism. In: Proceedings of the 3rd IEEE IntI. Conf. on Data Mining ICDM, (2003) 549-552

[28] Peixiang Zhao, Jeffrey Xu Yu, Philip S. Yu , Graph Indexing: Tree + Delta >= Graph, ACM. VLDB '07, September 2328, 2007, Vienna, Austria.

[29] Xifeng Yan Jiawei Han, gSpan: Graph-Based Substructure Pattern Mining, ICDM 2003. Proceedings,2002.