International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-3, Issue-3, July 2013

Design and Development of Abstractness in Graph Mining Technique using Structural Datum

S.P.Victor, M. Antony Sundar Singh

Abstract-Graphs are everywhere, ranging from social networks and mobile call networks to biological net-works and the World Wide Web. Mining big graphs leads too many interesting applications including cyber security, fraud detection, Web search, recommendation, and many more. In this paper we describe a technique for the conversion of real-time environment to a Graph Mining pattern. We analyze very large, real world graphs with billions of nodes and edges. Our findings include digraph structures in the connected component size distribution. In the future we will extend our research to propose a GraphTemplateConverter for any real-time complex entities.

Keywords- Graph mining, Graph pattern, Graph template, Graph network.

I. INTRODUCTION

A graph is set of nodes, pairs of which might be connected by edges. In a wide array of disciplines, data can be intuitively cast into this format[1]. For example, computer network consist of routers/computers (nodes) and the links (edges) between them. Social networks consist of individuals and their interconnections (which could be business relationships or kinship or trust, etc)[2]. Protein interaction networks link proteins which must work together to perform some particular biological function. Graphs are seemingly ubiquitous. The problems of detecting abnormalities (outliers) in a given graph and of generating synthetic but realistic graphs have received considerable attention recently.[3]

Table I: Graph notations

Table of Symbols	Symbol Description
N	Number of nodes in the graph
E	Number of edges in the graph
K	Degree for some node
< k >	Average degree of nodes in the graph
CC	Clustering coefficient of the graph
CC(k)	Clustering coefficient of degree-k nodes

Identifying tightly coupled pattern to the problem of finding the distinguishing characteristics of real-world graphs, that is, the patterns that show up frequently in such graphs and can thus be considered as marks of realism. A good generator will create graphs which match these patterns. Patterns and generators are important for many applications.[4]

Manuscript Received on July, 2013

Dr.S.P.Victor, HOD / Department of Computer Science St.Xaviers College,Tiruelveli- Mobile: 9486041594 Tamilandu,India.

M.Antony Sundar Singh, Research Scholar Manonmaniam Sundaranar University, Tirunelveli. Tamilnadu,India.

- Detection of abnormal sub graphs/edges/nodes. Abnormalities should deviate from the normal patterns so understanding the patterns of naturally occurring graphs is a prerequisite for detection of such outliers.[6]
- Simulation studies. Algorithms meant for large real-world graphs can be tested on synthetic graphs which look like the original graphs.[5]
- Realism of samples. We might want to build a small sample graph that is similar to a given large graph. This smaller graph needs to match the patterns of the large graph to be realistic.
- Graph compression. Graph patterns represent regularities in the data. Which can be used to better compress the data.

II. PROPOSED METHODOLOGY

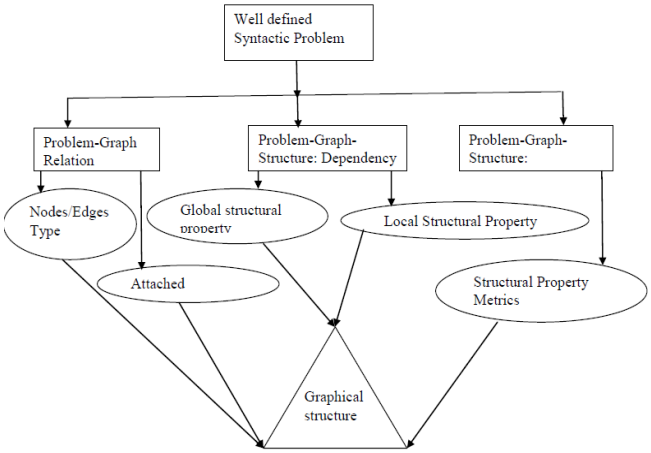

Fig-1: Proposed methodology for Graph conversion

As a mathematical construct, a graph consists of two types of elements: nodes and edges. In translating a problem to a graph-based representation, the first step is to decide how problem elements will translate to these distinct graph elements. As a general guideline, nodes are used to represent entities in a problem – genes, people, cities, businesses – and edges are used to represent relationships between the entities – 'regulates', 'knows', 'exports to', 'sells to'. If these relationships are directional (e.g., the fact that A sells to B doesn't imply that B sells to A), then the result will be a DIRECTED GRAPH; if not, then an UNDIRECTED GRAPH will result. This distinction is important, as some graph-theoretic measurements will treat directed and undirected graphs differently. Unless specifically stated, undirected edges are the assumed norm.

In determining the mapping from problem elements to graph elements, it is sometimes necessary to have multiple types of nodes or edges. While many problems can be adequately represented Without multiple node or edge types, some disciplines, such as social network analysis, make significant use of 2-mode graphs (i.e., graphs having two different node types).

III. III. IMPLEMENTATION RESULT

One example of a 2-mode graph is Teachers and their Class students, in which both teachers and Students are represented as nodes, and a connection between a Teacher and a student means that the teacher is a responsible person of the class. Having multiple node or edge types may complicate analysis, as the techniques and metrics are better developed for single-mode graphs. One solution to this problem is to adapt single-mode for a system by using domain-specific knowledge to, for example, generate and analyse appropriate single-mode sub graphs of the system. An alternative approach is to reduce the system to a single-mode graph by transforming one of the node types into a relationship (e.g., transforming the Teachers and class network by removing classes and transforming all of the edges between teachers and class to edges between teachers). Generally, even in situations where multiple types of nodes or edges would provide a more accurate representation, a simple single-mode graph representation will still provide useful insights.

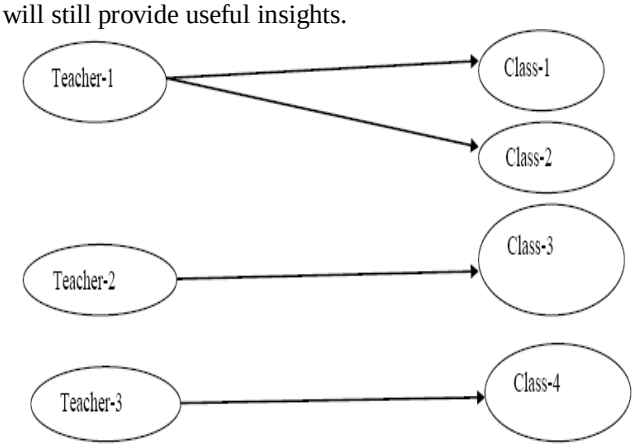

Fig 2: Two mode graph structure

In situations where the graph must represent richer information than simple nodes and edges will allow, it is possible to have nodes and edges with attached properties.

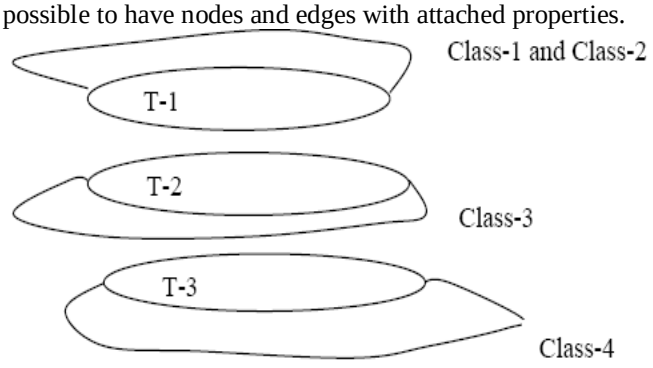

Fig-3: Single Mode Graph structure

Some common examples of properties that may be attached to nodes are the location information of a node in a transportation network or the activation value of a gene in a genetic regulatory network; a link's properties may include a weight, representing the degree of influence a gene exhibits upon the activation of another, or the strength of a relationship in a social network. Some network types, such as transportation networks, have sophisticated techniques for dealing with certain attached properties, such as distance between nodes and flow through edges. In general, however, attached properties will require domain-specific tuning of generic metrics and techniques if such properties are to be taken into account.

IV. DISCUSSION AND RESULTS:

The implementation of a graph-based representation depends on the properties or features of the problem that are being investigated. When viewing at the natural occurrence of real time problem in a genetic regulatory network, investigating node-edge relations and comparing these distributions with those of a grounded structure may be useful; if the targets of inquiry are organisational groups in a social network, usage of clique would be efficient. Identify the different structural properties that may be of interest in a particular problem, it is subsequently necessary to find a suitable metric with which to investigate these features. There is a list of structural characteristics of a graph, along with some suggestions for which metrics to use, in Table 2

Complexity	Metrics
Node –Edge Bond relationship	clique, n-clique, k-plex, k-core
Internal and External Impact with other components.	degree, in-degree, out-degree
Time duration for relationship	shortest-path length
Kernel node or Edge	bridges, pivots
Minimal covering of group nodes	K-Colouring

Table II: Graph complexity with appropriate metrics

Given that there are a set of expected local and/or global structural characteristics to be investigated, selecting appropriate metrics involves identifying those that are likely to provide information about the interesting structures. As most structural characteristics can be measured in many different ways, deciding on appropriate metrics may require a combination of domain-specific knowledge and trial and error. A more complete list of metrics is given in Table 3.

Graph terminologies	Relation
degree, in-degree, out-degree	importance of a node based on how connected it is
degree distribution	set of related properties, such as average shortest path length, probability of creating disconnected components through node/edge removal
shortest path length	distance between two nodes, degree of influence of nodes on each other
clique, n-clique, n-clan, k-plex, k-core	identification of highly interconnected sub graphs
Hub	identification of highly connected nodes
pivots, cut-points	identification of nodes crucial to keeping the graph connected
Bridges	identification of edges crucial to keeping the graph connected
node-connectivity, line-connectivity	how much damage the graph can take before becoming disconnected
Centralization	how much the graph centres on a single node or group of nodes
between's centrality, closeness centrality	importance of a node based on its relationship to other nodes in the network

Table III: Graph structures and metrics

Some real-time structure conversion to graph domain are illustrated in Figure-5.Applying the proposed technique will yield graph patterns for a well defined problem includes several conversion strategies.

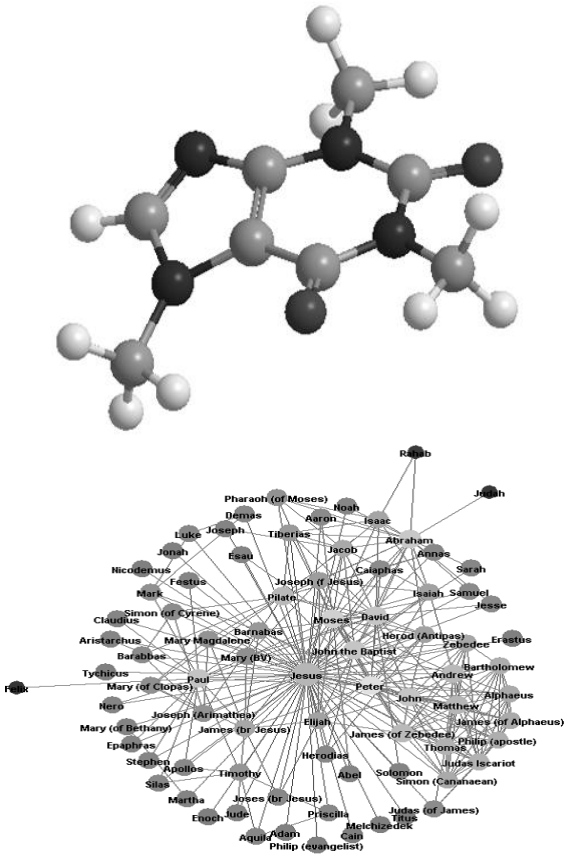

Fig-4: Chemical compound and social network Graph structure sample.

V. CONCLUSION

In this paper we describe the methodology for converting a well defined problem to a Graph pattern, These are only some of the models with modular nature to convert it directly; there are many other models which add new ideas or combine existing models in novel ways. We have looked at many of these and discussed their strengths and weaknesses. In future, we will develop the Graph Template Converter model which can match most of the graph patterns for several real-world graphs. More patterns need to be found, though there is probably a point of diminishing returns where extra patterns do not add much information. We do not think that point has yet been reached. Also, typical generators try to match only one or two patterns; more emphasis needs to be placed on matching the entire patterns. This cycle between finding more patterns and better generators which match these new patterns should eventually help us gain a deep insight into the formation and properties of real-world graphs.

REFERENCES

[1] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Ma-honey. Statistical properties of community structure in large social and information networks. In WWW, pages 695-704, 2008.

[2] C. Liu, F. Guo, and C. Faloutsos. Bbm: Bayesian browsing model from petabyte-scale data. In KDD, pages 537-546, 2009.

[3] Y. Low, J. Gonzalez, A. Kyrola, D. Bick son, C. Guestrin, and J. M. Heller stein. Graph lab: A new framework for parallel machine learning. In UAI, pages 340-349, 2010.

[4] R. Gemulla, E. Nijkamp, P. Haas, and Y. Sisma-nis. Large-scale matrix factorization with distributed stochastic gradient descent. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 69-77. ACM, 2011.

[5] A. Ghoting, R. Krishnamurthy, E. P. D. Pednault,B. Reinwald, V. Sindhwani, S. Tatikonda, Y. Tian,and S. Vaithyanathan. System: Declarative machine learning on map reduce. In ICDE, pages 231-242, 2011

[6] U. Kang, H. Tong, J. Sun, C.-Y. Lin and C. Faloutsos.Gbase: an ancient analysis platform for large graphs.VLDB J., 21(5):637-650, 2012.