COMPREHENSIVE ANALYSIS OF COMPLEX DATA STRUCTURES FOR GRAPH DATA MINING

[1]K RAJENDRAN, [2]Dr.T.VENUGOPAL, [3]Dr.A.RAMA MOHAN REDDY

[1]Research Scholar, Dept. of Science & Humanities, SCSVMV University, *Kancheepuram*, India
[2]Reader in Mathematics, Dept. of Science & Humanities, SCSVMV University, *Kancheepuram*, India
[3]Professor in Computer Science and Engineering, SV University, *Tirupati*, India

E-mail: kkuppu_swamy@yahoo.com, venugopal.t@kanchiuniv.ac.in

ABSTRACT

Various methods and techniques are prevailing for processing data to extract useful knowledge; nevertheless they are accounted for business transaction databases. Much data exists in non-business domains also, where there is an ardent need of extracting knowledge. Knowledge is useful for understanding data concisely or comprehending the concepts in a precise manner. Relational characteristics in data complicate the representation, and can be eased through graph structures. As one of the most general forms of data representation, the graph easily represents entities, their attributes, and their relationships to other entities. Envisioning the scope of the problems related to graph data, a collection of representations in graph data mining leads to the development of reference framework. In this paper we propose an integrated analytical reference framework with a comprehensive study and empirical evaluation.

Keywords: *Graph Data Mining, Information Theory, Knowledge Engineering*

1. INTRODUCTION

Since antiquity the intuitive notions of continuous change, growth, and motion, have challenged scientific minds. Yet, the way to understand continuous variations in systems and perceiving various dimensions of data was open only in late 1980s when modern computer science emerged and rapidly developed in close conjunction with electronics and allied sciences. Data and Information are considered to be as of the same form and relevance in most applications, but when the question arises "What is knowledge?" one has to identify the barriers between data and knowledge. Information however is used ultimately for crucial decision making in a management information systems and data at the operational level. Data come in many forms and the place to develop a complete taxonomy persists with data modeling. Indeed, it is not even clear that a complete taxonomy of data model can be developed, since the important of data unstable according to various contexts. Methods and techniques prevail to process data and extract useful knowledge.

The world is deluged by data —scientific data, medical data, demographic data, financial data, and marketing data. A very short time is spent for having a look back at the data, which is having strong historical relevance. Human attention has become the precious resource. Much data exists in non-business domains also, where there is an ardent need of extracting knowledge. So, we must find ways to automatically analyze the data, to automatically classify it, to automatically summarize it, to automatically discover and characterize trends in it, and to automatically flag anomalies.

Knowledge is useful for understanding data concisely or comprehending the concepts in a precise manner. Relational characteristics in data complicate the representation, and can be eased through graph structures. Graphs have come a long way since 1736 when Leonhard Euler applied a graph-theoretic argument to solve the problem of the seven Königsberg bridges. The growth of graph theory during its first two hundred years could in no way foreshadow the spectacular progress which this area was to make. The graph representation, that is, a collection of nodes and links between nodes, does support all aspects of the relational data mining process. Since the years of development of mathematical graph theory, it is said to have characteristics to

represent more qualities of the data. The scope of the problems related to computer theory, graph theory and graph data demands new directions in graph data mining. The conceptual view of such problems can be solved by introducing a reference framework.

With the progress of database technology, various kinds of advanced data and information systems have emerged and are undergoing changes to address the requirements of new applications. These promote the development of advanced data models such as extended-relational, object-oriented, object-relational, and deductive models. Heterogeneous database systems and Internet-based global information systems such as the World Wide Web (WWW) have also emerged and play a vital role in the information industry. Subsequently data can is hosted to store in many different kinds of databases and information repositories. Universally agreed data storage architecture has emerged; the data warehouse, a repository of multiple heterogeneous data sources organized under a unified schema at a single site in order to facilitate management decision making. Data in these repositories is extracted from operational databases as a whole or by business case analysis. For databases containing a huge amount of data, appropriate sampling techniques can be first applied to facilitate interactive data exploration. Mining tasks that employ on such data warehouses can exploit unstructured components of the data.

2. ERSTWHILE ANALYSIS AND SUPPORT

A. Graph Data Structure

Data are collected by mapping entities in the domain of interest to symbolic and structural representation by means of some measurement procedure, which associates the value of a variable with a given property of an entity. The relationships between objects are represented by numerical and hierarchical relationships between variables. These representations of data items are stored as data set.

In many cases multirelational data can be mapped to a single data matrix or table or graph.

One or more graph structures can be combined based on a variable. The transformation of data set into graph makes the sense and representation of the data closely related to domain. This possible transformation leads to development of multirelational structures at all since in principle it is possible to represent the data in one large graph structure. Therefore, this procedure of joining data sets is not only possible for representing as mere structures in memory but also in the persistent form. The advantage of graph structure for representing data is, one can explore the dependence between data, significance of captured dimensional attributes, and merging techniques into a graph structure. More important, from the point of view of efficiency in storage and data access, "flattening" the graph to form a large table (physical structure) involves redundant replication of numerous values.

B. Comprehensive Analysis

Elementary concepts of graph are evaluated comprehensively to ascertain the necessity of the graph structure for graph data mining. Data mining can be the illusive task of mining all possible knowledge from the sources of data. But as the maxim tells, "*Garbage in is garbage out*", the input data for the data mining must be apt to the knowledge that we are expecting derive out of the process. The graph match, isomorphism, labeling and other are the descriptive elementary characteristics of a graph. These characteristics are comprehensively studied. As the properties of the graphs are too wide to describe, all the information is captured completely and a single definition called *information entropy* is derived. The information entropy may also be considered as probability distribution of all the properties that exists in a graph. Otherwise *information entropy* is quoted as Information divergence and information for discrimination - is a very essential phenomenon for identifying diverse characteristics of a graph.

Method of types can be chosen as the procedure of comprehensive analysis from information theory.

Exploring the relationship between information theory and statistics begins by describing the method of types, which is a powerful technique in large deviation theory [1]. Method of types is used to calculate the probability of rare events and shows the existence of universal source codes. The method of types is an even more powerful procedure in which sequences that have the same empirical

distribution are considered. With this restriction, strong bounds can be derived on the number of sequences with a particular empirical distribution and the probability of each sequence. It is then possible to derive strong error bounds for the channel coding theorem and prove a variety of rate distortion results.

In the current problem the method of types determines the probability of characteristics of a graph (*rare* or *frequent*) and develops the sequences with a distribution.

Let X_1, X_2, ... X_n be a sequence of n symbols (characteristics of a graph) from an alphabet $X = \{a_1, a_2 \ldots a_{|X|}\}$. The notation x^n and $\boldsymbol{x}$ interchangeably used to denote a sequence $x_1, x_2 \ldots x_n$.

The type $\boldsymbol{P}_x$ (or empirical probability distribution) of a sequence $x_1, x_2, \ldots x_n$ is the relative proportion of occurrences of each symbol of X. The type of a sequence x is denoted as $\boldsymbol{P}_x$. It is a probability mass function on X.

Let $\boldsymbol{P}_n$ denote the *set of types* with denominator n. For instance, if $X = \{0, 1\}$, the set of possible types with denominator n is

$$P_n = \left\{ (P(0), P(1)) : \left(\frac{0}{n}, \frac{n}{n}\right), \left(\frac{1}{n}, \frac{n-1}{n}\right), \ldots, \left(\frac{n}{n}, \frac{0}{n}\right) \right\}$$

If $\boldsymbol{P} \in \boldsymbol{P}_n$, the set of sequences of length n and type $\boldsymbol{P}$ is called *type class* of $\boldsymbol{P}$, denoted $\boldsymbol{T}(\boldsymbol{P})$:

$$\boldsymbol{T}(\boldsymbol{P}) = \{x \in \boldsymbol{X}^n : \boldsymbol{P}_x = \boldsymbol{P}\}$$

The *type class* is sometimes called the composition class of $\boldsymbol{P}$.
The graph characteristics are evaluated as the probabilistic occurrences in the data set and their weightage for graph structure consideration is estimated.

$$|\boldsymbol{P}_n| \leq (n+1)^{|X|}$$

There are $|\boldsymbol{X}|$ components in the vector that specifies P_x. The numerator in each component can take on only $n + 1$ values. So there are at most $(n + 1)^{|\boldsymbol{X}|}$ choices for the type vector. Of course, these choices are not independent (e.g., the last choice is fixed by the others). But this is a sufficiently good upper bound for our needs.

The crucial point here is that there are only a polynomial number of types of length n. Since the number of sequences is exponential in n, it follows that at least one type has exponentially many sequences in its *type class*. In fact, the largest type class has essentially the same number of elements as the entire set of sequences, to first order in the exponent.

C. Graph Properties

The process of evaluating the structural similarity of graphs is commonly referred to as graph matching. A large variety of methods addressing specific problems of structural matching have been proposed. Graph matching systems can roughly be divided into systems matching structure in an exact manner and systems matching structure in an error-tolerant way. Although exact graph matching offers a rigorous way to describe the graph matching problem in mathematical terms, it is generally only applicable to a restricted set of real-world problems. Error-tolerant graph matching, on the other hand, is able to cope with strong inner-class distortion, which is often present in real-world problems, but is generally computationally less efficient.

Graph matching has successful applications in the field of pattern recognition and machine learning. In the case of exact graph matching, the graph extraction process is assumed to be structurally flawless, that is, the conversion of patterns from a single class into graphs always results in identical structures or substructures. Otherwise graph isomorphism or subgraph isomorphism detection is rather unsuitable, which seriously restricts the applicability of graph isomorphism algorithms. The main advantages of isomorphism algorithms are their mathematically stringent formulation and the existence of well-known procedures to derive optimal solutions.

The properties of the graphs to analyze the isomorphism and matching gives a direction of mining the graph patterns as *sub-graph patterns* or *cliques*. Attributed graphs with an unrestricted label alphabet are one of the most general ways to define graphs.

Given a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, a graph $\boldsymbol{G}_S = (\boldsymbol{V}_S, \boldsymbol{E}_S)$ will be a *subgraph* of $\boldsymbol{G}$ if and only if $\boldsymbol{V}_S \subseteq \boldsymbol{V}$ and $\boldsymbol{E}_S \subseteq \boldsymbol{E}$, and it will be an *induced subgraph* of $\boldsymbol{G}$ if $\boldsymbol{V}_S \subseteq \boldsymbol{V}$ and $\boldsymbol{E}_S$ contains all the edges of $\boldsymbol{E}$ that connect vertices in $\boldsymbol{V}_S$. A graph is *connected* if there is a path between every pair of vertices in the graph. Two graphs $\boldsymbol{G}_1 = (\boldsymbol{V}_1, \boldsymbol{E}_1)$ and $\boldsymbol{G}_2 = (\boldsymbol{V}_2, \boldsymbol{E}_2)$ are *isomorphic* if they are topologically identical to each other, that is, there is a mapping

from V_1 to V_2 such that each edge in E_1 is mapped to a single edge in E_2 and vice versa. In the case of *labeled graphs*, this mapping must also preserve the labels on the vertices and edges. An *automorphism* is an isomorphism mapping where $G_1 = G_2$. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the problem of *subgraph isomorphism* is to find an isomorphism between G_2 and a subgraph of G_1, that is, to determine whether or not G_2 is included in G_1. The *canonical label* of a graph $G = (V, E)$, $cl(G)$, is defined to be a unique code (i.e., a sequence of bits, a string, or a sequence of numbers) that is invariant on the ordering of the vertices and edges in the graph.

3. ANALYSIS

a. Reference Framework

Graphs show up in a surprisingly diverse set of disciplines, ranging from computer networks to sociology, biology, ecology and many more. Informally, a graph is a set of nodes, and a set of edges connecting some node pairs. In database terminology, the nodes represent individual entities, while the edges represent relationships between these entities. This formulation is very general and intuitive, which accounts for the wide variety of real-world datasets which can be easily expressed as graphs.

The crust of the analytical framework concentrates on representation of a graph. As described in the above sections there are various properties of a graph. The structure of graph database is proposed in this section. The properties of a graph are categorized into two, viz., *topological* and *anatomical*. The anatomy of graph represents vertices, edges and their characteristics. The topological properties represent *degree*, *isomorphism*, *complementary*. The anatomical properties of graph are considered to be as the characteristics that are expressed by the constituent parts of the graph. The topological properties describe the shape, geometry, physical dimensions and the expanse of the graph. The schema of the graph database should mandatorily define provisions for both topological and anatomical properties.

Graph-theoretic: Graph-theoretic structures [2] are epitomized by intersection graphs. They subsume concepts as standard as line graphs and as nonstandard as tolerance graphs. Real-time applications such as biological computing, matrix analysis are concentrated on intersection graphs. Intersection in graphs is categorized by intersection classes. As economy is the major concern in the construction and representation of graph, parsimony in constructing the graph classes is observed. Since every graph is an intersection graph, to avoid the illusion of huge structural requirements of graph representation, minimum cardinality set is used to identify a family of subsets of graph. Such set that would represent the cardinality and basic features of the graphs-subset is a parsimonious set. Typically a *parsimonious set* can be connoted with a pattern of machine learning. A graph-pattern exhibit more soft characteristics of the elements of the graph which is juxtaposed with *parsimonious set*.

Segmentation of graphs into fragments such as triangulated graph, chordal graph and tree graphs etc,. Segmentation of a graph, i.e., *a graph that is a complete subgraph that is not properly contained in another complete graph*. A family F = $\{S_1,\ldots,S_k\}$ of subsets of a set S is said to satisfy the following [Helly condition]: for every subfamily $F \subseteq F'$, if the members of F', intersect pairwise, then all the members have a common element – in other words, if every S_i, $S_j \in F'$,has $S_i \cap S_j \neq \phi$, then $\cap \{ S_i : S_i \in F'\} \neq \phi$. This is conceptualized from maxclique of a graph which is a complete subgraph that is not available properly in any another complete subgraph. Those subgraphs that does not form into a pattern and parsimony does not hold may be definitely represented as clique-pattern. A clique-pattern could be a subgraph which does not repeat. In machine learning, we call this as a peculiar or interesting pattern, where interestingness does not meet to any frequency of occurrences of a unit data set in large graphs. Similarly this is extended to a edge-clique where a line graph is identified. As per Krausz theory, A graph G is a line graph if and if it has an edge clique cover ε. A graph with a set of vertices set of family or classes and a cover is a hypergraph, which can be treated as to hold schematic structure of the data.

A supposition is made to data set representation or data bases using graphs, such as a collection of tables, relations, columns and attributes. A hypergraph structures can be used to build constellation schemae for a typical data warehousing application. An acyclic database schema can be constructed, if the relations are arranged as the vertices of a tree, commonly called as join tree, such that the vertices of a graph data set can induced to a subtree. This would be a better scheme to envision for building complex hierarchical and network based database

structures in classical data structures. However hierarchical and network based data structures are found not suitable for conventional database operations, they are best suited for areas of social networks, biological data sets, etc.

4. CONCLUSION

A wide spread analysis have been made on various types of graphs. A mathematical graph theory is used in this paper to converge with the design and development of databases for huge applications. Certainly, data warehousing attracts complex design of databases. The concepts illustrated from our study are to develop the database with maximal characterization of data warehouse data.

REFRENCES:

[1] Elements of Information Theory, Second Edition, By Thomas M. Cover and Joy A. Thomas Copyright © 2006 John Wiley & Sons, Inc.

[2] Topics in Intesection graph theory, Terry A. McKee, F.R.McMorris, © 1999, SIAM, Society for Industrial and Applied Mathematics.

AUTHOR PROFILES

Mr.K.Rajendran has completed M.Sc. and M.Phil in Madras University, Madras. He has profound experience of 16 years in working with defence and teaching. He has worked with graphs and their social applications (social networks). His areas of research are pattern detection and matching in large graphs, graph mining.

Prof. T.Venugopal has headed various programmes in Sri Chandrasekharendra Saraswathi Maha Vidyalaya (University), Kancheepuram. He has worked on various types of problems in graph, functional analysis, and wavelet theory with graph mining.

Dr. A. Rama Mohan Reddy is currently working as Professor, Department of CSE, SVU College of Engineering, S V University, Tirupati, A.P. He has completed his B.Tech from JNT University, Anantapur and M.Tech from NIT, Warangal. He has received his Ph.D. from SV University in Software Architecture. His other areas of interests include Data Mining, softare Architectures and Object Oriented Analysis & Design.