Data & Knowledge Engineering 91 (2014) 17–33

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Editorial

Reducing the bottleneck of graph-based data mining by improving the efficiency of labeled graph isomorphism testing

Shu-Ming Hsieh [a,b], Chiun-Chieh Hsu [a,*], Yen-Wu Ti [b], Chi-Jung Kuo [a,c]

[a] Department of Information Management, National Taiwan University of Science and Technology, No. 43, Sec. 4, Keelung Rd., Taipei, Taiwan
[b] Department of Computer Science and Information Engineering, Hwa Hsia Institute of Technology, No. 111, Gongzhuan Rd., Zhonghe Dist., New Taipei, Taiwan
[c] Department of Information Management, Taipei Chengshih University of Science and Technology, Xueyuan Rd., Taipei, Taiwan

ARTICLE INFO

Article history:
Received 26 January 2013
Accepted 19 February 2014
Available online 3 March 2014

Keywords:
Data mining
Mining methods and algorithm
Isomorphism testing
Graph signature
Search state-space tree

ABSTRACT

Due to the complex nature of graph representations, the *isomorphism testing* between a pair of labeled graphs becomes one of the most time-consuming procedures during the process of graph-based data mining. In order to reduce this bottleneck, in this paper we propose a novel efficient algorithm to perform isomorphism testing of labeled graphs which in general performs less state-space tree searching. The proposed method uses *graph signatures* as the first-step filter, and it limits the backtracking occurring only between each pair of corresponding vertex classes, based on the proposed data structures and the vertex partition method. We compared the proposed method with state-of-the-art methods to verify its efficiency for several datasets each with different aspects of characteristics. The experimental results show that for irregular graphs, either labeled or unlabeled, the proposed method outperforms the compared methods in efficiency. For graphs with multiple labels but high regularity, the proposed method is still better than the compared methods. The result of this algorithm is directly applicable to those emerging applications related to graph-based data mining which need to perform isomorphism testing of labeled graphs in large databases.

1. Introduction

Graphs are suitable for modeling a variety of real-world complex structures with the benefits of being capable of characterizing the spatial, topological, geometric, or abstract properties among the objects in the modeled data sets [1]. In many application areas, objects are symbolized as vertices in a graph and the relationships between object pairs are represented as edges joining the vertex pairs. In such graphs, each of vertices and edges is assigned a label to denote the object class and to convey the relation held between a pair of objects, respectively. For example, in a *molecular structure* an atom corresponds to a vertex and a bond between two atoms is represented as an edge between two vertices. The vertices are assigned labels that stand for the atom types (e.g. C, H) and the labels of edges are the bond types or the relative 3D orientations between a pair of atoms in such structures [2,3]. In *geographical information systems*, objects or spatial elements are vertices and the relationships between object pairs are edges that hold the information of adjacency, connectivity, and containment [4]. Another application is *image retrieval*, in which vertices denote the recognized objects in an image and edges are the spatial or topological relationships between pairs of objects [5–7].

For the fields appropriately modeled by graphs, one of the emerging topics is to discover the recurring substructures or *patterns* in the graph database, which is the collection of all the modeled instances. Taking the molecular biology as an example,

* Corresponding author. Tel.: +886 2 27376766; fax: +886 2 27376777.
E-mail addresses: stanley@cc.hwh.edu.tw (S.-M. Hsieh), cchsu@mail.ntust.edu.tw (C.-C. Hsu), tiyenwu@cc.hwh.edu.tw (Y.-W. Ti), cjkuo@tpcu.edu.tw (C.-J. Kuo).

http://dx.doi.org/10.1016/j.datak.2014.02.003
0169-023X/

mining the frequent topological structures in order to identify novel RNAs or protein structures is vital for researchers in this field [2,3]. Finding the frequent common chemical substructures is also helpful for toxicologists to classify new toxic substances [8]. There are still more applications such as computer networks, pattern recognition, social network analysis, and Web mining which are also of great need to find frequently occurring patterns.

The approaches to graph-based data mining can be categorized into five groups, among which the *mathematical graph theory based* approach is closely related to the substance of graph based modeling [23]. The recent graph theory based methods include AGM (Apriori-based Graph Mining) [9], FSG (Frequent SubGraph discovery) [1], gSpan (graph-based Substructure pattern mining) [10], FFSM (Fast Frequent Subgraph Mining) [8], etc. The graph/subgraph isomorphism testing plays the key role during the whole process of frequent subgraph (pattern) mining, which usually comprises *candidate generation* and *frequency counting* [1,10]. The subgraph isomorphism testing problem is NP-complete whereas there is no known polynomial-time algorithm for the isomorphism testing of *general graphs* [11–14]. However, for some *specific* kinds of graphs, polynomial-time testing algorithms had been proposed. For example, trees can be tested in linear time [15], and so are planar graphs [16]. In addition, an $O(n^3 \log n)$ algorithm was proposed for graphs with at most three edges incident on every vertex [17].

For the methods handling general kinds of graphs, most of the previous works are dedicated to *unlabeled* graphs [18–20], which are graphs that only convey structural (adjacent or not) information. Among these efforts, the NAUTY algorithm [21,22] is known to be one of the fastest algorithms for unlabeled graph isomorphism testing [1,23]. However, NAUTY does not allow graphs to have edge labels; hence it is not applicable to labeled graphs [1].

Algorithms for the isomorphism testing between two general labeled graphs can be roughly divided into two types. The first type of methods such as Ullmann [24] and VF2 [19] can be characterized by the *backtracking* on the search state-space tree. They employ both structural and label constraints for matching each promising vertex pair. Although the theoretical worst time complexity is $O\left(n^{\sqrt{n}}\right)$, the effort made by these methods can cut the size of the search tree effectively. The second type of methods such as FSG [1] uses the *canonical code* as the representation of a graph, by which two graphs are isomorphic if and only if their canonical codes are identical. Below are the brief discussions of these methods.

Ullmann proposed a backtracking method that significantly reduces the size of the search tree. In this method, a refinement procedure is employed that eliminates unpromising successor nodes in the state space tree searching. It is still one of the most commonly used methods for exact graph matching [20].

Another outstanding algorithm VF2, also based on depth-first search (DFS) strategy, employs some sophisticated feasibility rules that prune unpromising vertex pairs in the searching on a state-space tree. According to the experimental results reported in Foggia et al. and De Santo et al. [25], VF2 performs better for sparse or high structural regularity graphs. VF2, as well as Ullmann, can work under the structural constraints as *synthetic rules* and label constraints as *semantic rules*.

The work of FSG combines several types of vertex invariants to partition the vertices into equivalence classes, by which the vertices in the same class have the same vertex invariants such as vertex degrees, vertex labels, and vertex neighbor lists. In FSG, the canonical code of a graph is defined to be the smallest code composed of the vertex labels followed by the concatenation of the columns of the upper triangular adjacency matrix over all possible permutations of the vertices. If the vertices of a graph with n vertices are partitioned into c classes $\pi_1, \pi_2, \ldots, \pi_c$, then the number of different permutations need to be considered in order to find that the canonical code is $\Pi_{i=1\sim c}(|\pi_i|!)$, which is substantially faster than the $n!$ permutations required by the earlier approaches [1].

In this paper, we propose a new algorithm CISC (*C*lass-wise *IS*omorphism testing with limited ba*C*ktracking) to perform the isomorphism testing of *general labeled* graphs. By the best usage of the features of labeled graphs, CISC combines several techniques to improve the overall performance. In the worst case, the amount of considered candidates (nodes visited in the search state-space tree) is $O(\Sigma_{i=1\sim c}(|\pi_i|!))$, in which the product of all $|\pi_i|!$ terms is replaced by the sum of the terms, compared with $O(\Pi_{i=1\sim c}(|\pi_i|!))$ [1]. In addition to its theoretical advantages, CISC also possesses the efficiency benefit in practical cases. According to the results of the conducted experiments, CISC outperforms Ullmann, FSG, and VF2 for *multiple labeled* graphs or *irregular unlabeled* graphs. The main contribution of CISC is that it can be directly applied to the work that needs to perform labeled graph isomorphism testing such as graph-based data mining.

The remaining of the paper is organized as follows. Section 2 discusses the definitions about labeled graphs and proofs of some lemmas for the proposed algorithm. In Section 3, we will introduce the algorithm and the related procedures. The experimental results of comparing the average matching times with three well-known methods are in Section 4. Section 5 concludes our work with future direction.

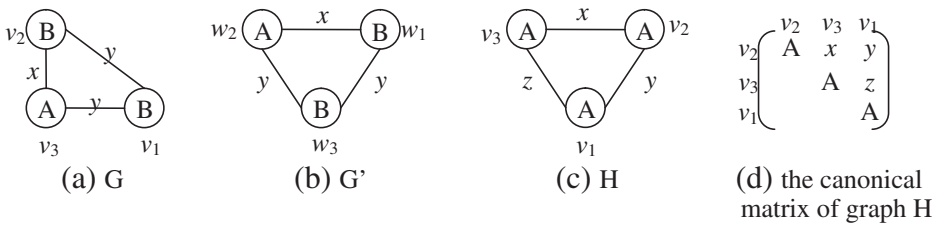

Fig. 1. Three labeled graphs and the canonical matrix of graph H in (c).