Knowledge-Based Systems 109 (2016) 256–265

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Efficient pattern matching for graphs with multi-Labeled nodes

Ali Shemshadi[a], Quan Z. Sheng[a], Yongrui Qin[b,*]

[a] School of Computer Science, The University of Adelaide, Adelaide, Australia
[b] School of Computing and Engineering, University of Huddersfield, Huddersfield, United Kingdom

ARTICLE INFO

Article history:
Received 13 December 2015
Revised 3 July 2016
Accepted 5 July 2016
Available online 9 July 2016

Keywords:
Graph matching
Multi-labeled graph
Graph simulation
Metropolis hastings

ABSTRACT

Graph matching is important for a wide variety of applications in different domains such as social network analysis and knowledge discovery. Despite extensive research over the last few decades, graph matching is still challenging particularly when it comes with new conditions and constraints. In this paper, we focus on a new class of graph matching, in which each node can accept multiple labels instead of one. In particular, we address the problem of finding the top-*k* nodes of a data graph which best match a labeled query node from a given pattern graph. We firstly prove this to be an NP-Complete problem. Then, to address this issue and improve the scalability of our approach, we introduce a more flexible graph simulation, namely *surjective simulation*. This new graph simulation reduces the unnecessary complexity that is due to the unnecessary constraints imposed by the existing definitions while achieving high-quality matching results. In addition, our approach is associated with an *early stop* strategy to further boost the performance. To approximate the maximum size of a simulation, our approach utilizes Metropolis Hastings algorithm and ranks the top-k matches after computing the set of surjective simulations. The experimental results over social network graphs demonstrate the efficiency of the proposed approach and superiority over existing approaches.

1. Introduction

Graph pattern matching is fundamental to various applications in many domains including, but not limited to, social computing [1], computer vision [2] and computational chemistry [3]. It has been extensively studied in different contexts within the past few years. Assorted types of conditions and requirements impose different constraints on pattern matching techniques. In general, pattern matching aims to solve the problem of finding subgraphs of a given data graph *G*, which match a given pattern graph *Q*. One of the particular conditions, i.e., processing graphs with labeled nodes [4], is increasingly receiving attention.

The existing approaches for labeled graph pattern matching use a particular definition of labeled nodes where each node is associated with a single label [1]. Thus, for the pattern *Q* and the data graph *G*, the nodes of *G* can be categorized based on the set of labels of nodes in *Q* without any conflict. This approach is useful for many applications in social computing but does not cover some of the new domains, such as the Internet of Things [5], and some sophisticated problems in social networks, i.e., when the labels are uncertain or when the data is incomplete. In any of these cases, assigning a single label to each node could be unrealistic and we need to define graphs with *multi-labeled* nodes. However, using multi-labeled graphs compared to the single-labeled graphs can potentially increase the complexity of the problem. In this case, revising the current pattern matching approaches for multi-labeled graphs is beneficial.

Example 1. To provide a clear image of the problem, in this paper we explain an application in the context of the Internet of Things. For example, a search service extracts the pattern graph from one network of things and the data graph from another. Fig. 1 illustrates two graphs that are obtained from these two networks. Each edge represents a relationship between two nodes in the same network.

Each node (i.e., a thing) is described with a set of metadata about its sensors and actuators. We can take each tag as a label due to some reasons including (1) there is not universal description for things connected to the network, and (2) each node can be registered partially on different networks. The labels are selected from a language $\Sigma.s=$ {sensor/thermal, sensor/weather, sensor/signal, sensor/current, sensor/motion} and $\Sigma.a=$ {actuator/screen, actuator/switch, actuator/fan, actuator/speaker, actuator/alarm}. Table 1 shows the labels assigned to each node.

[*] Corresponding author.
E-mail addresses: ali.shemshadi@adelaide.edu.au (A. Shemshadi), michael.sheng@adelaide.edu.au (Q.Z. Sheng), y.qin2@hud.ac.uk (Y. Qin).

http://dx.doi.org/10.1016/j.knosys.2016.07.009
0950-7051/© 2016 Elsevier B.V. All rights reserved.

Table 1
The set of label assignments.

Node	Labels
u_1	$l.s_0$, $l.s_1$, $l.a_0$, $l.a_3$
u_2	$l.s_2$, $l.s_3$, $l.a_1$, $l.a_3$
u_3	$l.s_1$, $l.s_3$, $l.a_2$, $l.a_3$
u_4	$l.s_0$, $l.s_1$, $l.a_0$, $l.a_1$
v_1	$l.s_2$, $l.s_3$, $l.a_1$, $l.a_3$
v_2	$l.s_0$, $l.s_1$, $l.a_0$, $l.a_2$
v_3	$l.s_1$, $l.s_2$, $l.s_3$, $l.a_1$, $l.a_2$
v_4	$l.s_2$, $l.a_0$, $l.a_1$, $l.a_2$
v_5	$l.s_3$, $l.a_0$, $l.a_1$, $l.a_3$
v_6	$l.s_0$, $l.s_1$, $l.a_0$, $l.a_1$
v_7	$l.s_1$, $l.s_2$, $l.s_3$, $l.a_3$
v_8	$l.s_0$, $l.s_2$, $l.a_0$, $l.a_1$
v_9	$l.s_1$, $l.s_2$, $l.s_3$, $l.a_0$
v_{10}	$l.s_0$, $l.s_2$, $l.a_0$, $l.a_1$, $l.a_2$
v_{11}	$l.s_1$, $l.s_2$, $l.a_1$, $l.a_2$
v_{12}	$l.s_0$, $l.s_1$

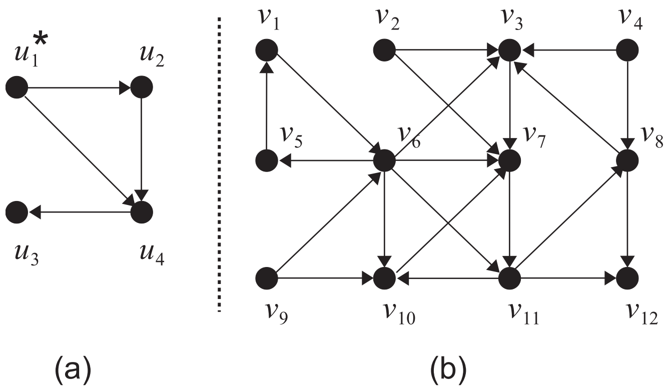

Fig. 1. Querying two networks of things (a) the pattern graph, (b) the data graph.

Table 2
Nodes with label similarity above threshold.

Query graph node	Similar data graph node
u_1	v_2, v_6, v_{12}
u_2	v_1, v_3, v_5, v_7
u_3	v_3, v_7
u_4	v_2, v_6, v_8, v_{10}, v_{12}

To examine the similarity between two nodes, we use a similarity ratio as follows:

$$s(v_i, v_j) = \frac{|L(v_i) \cap L(v_j)|}{|L(v_i) \cup L(v_j)|} \quad (1)$$

where $|L(v_i) \cap Ł(v_j)|$ denotes the number of common labels between two nodes and $|L(v_i)|$ denotes the number of the labels of v_i. We can compare the similarity score with a threshold t.

Based on the present model, if we set the similarity threshold as 0.5, for each node $u_i \in Q$, Table 2 lists the nodes $v_i \in G$ that can be a match. All of the nodes from the data graph appear more than once in the set of similarity lists. Due to this conflict, no unique label can be associated with any of the nodes. For instance, although v_2 is specified a match of the query node (u_1), it can match node u_4 as well. The set of similarity lists is more complex than the case when each node is assigned only one table. This is because in that case, each node of the data graph would appear only once in the final list. Therefore, the pattern matching process can be more complex and new situations must be considered.

With the increasing complexity and volume of graphs in new contexts such as social networks and the Internet of Things, performance will be the main issue that we should tackle when we revise graph pattern matching for multi-labeled graphs. The typical definitions of graph simulation are generally too restrictive to be applied for this purpose. Nonetheless, computing all of the possible simulations will result in the inefficiency of any solution. In order to tackle these challenges, we propose a novel approach for graph pattern matching for multi-labeled graphs, and briefly, our contributions in this paper are as follows:

- We introduce the concept of *surjective simulation*, which is more flexible than graph simulation and bounded simulation. Through the use of surjective simulation, the proposed approach can notably reduce the complexity of simulation creation step to $|V_p||V|$. With this concept, we can optimize the process via removing the simulations that do not contain any match of the query nodes.
- To avoid going through the computation of each simulation to get its size, we propose an approximation procedure based on the Metropolis-Hastings Algorithm. We also devise an early stop mechanism when (1) we have at least k nodes in the results and (2) the size of the smallest simulation is equal to or greater than the rest of surjective simulations.
- We evaluate the proposed approach via extensive experimental studies. The results show the efficiency of our approach and verify the superiority of our approach over existing approaches.

The rest of this paper is organized as follows. In Section 2 we define the problem. A naive approach based on a very recent work is provided in Section 3. We provide necessary background in Section 4. Then in Section 5 we show how we can compute the set of the top-k results using the proposed concept of surjective simulation with the Metropolis–Hastings algorithm. Section 6 presents the experimental results. Finally, Section 7 reviews the related works and Section 8 provides some concluding remarks.

2. Problem formulation

Before presenting our approach for graph pattern matching, we first formally define the problem that we are going to investigate. Multi-labeled graph pattern matching is the task of matching the nodes of a given pattern graph Q with a data graph G based on structural similarity, where each node is given a set of labels.

Definition 1. Graph A graph is represented as $G = (V, E, L)$ where (1) $V = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes; (2) $E \subseteq V \times V$ is a set of edges; and (3) $L = \{l_i : l_i \in V \times \Sigma\}$ is a mapping that relates each node to a set of assigned labels from language Σ.

Definition 2. Pattern Graph [4,6] A pattern graph is a directed and connected graph $Q = (V_p, E_p, L_p, u^*)$, where (1) V_p is a set of query nodes; (2) E_p is a set of query edges; (3) $L_p \subseteq V_p \times \Sigma$ is a mapping that links every node $u \in V_p$ to a set of labels in $\Sigma_p \subseteq \Sigma$; and (4) $u^* \in V_p$ that specifies the query node.

Definition 3. Graph Simulation [6] A graph G matches a pattern Q iff there exists a binary relation $S \subseteq V_p \times V$ such that (1) for each node $u \in V_p$, there exists a node $v \in V$ such that $(u, v) \in S$, referred to as a match of u; (2) for each pair $(u, v) \in S$, $L(u) = L(v)$, and for each edge (u, u') in E_p, there exists an edge (v, v') in G such that $(u', v') \in S$.

A top-k problem can be defined in the following. Given a surface $\Gamma(x, y)$, a function $f(x, y)$: $x, y \to D$, a scoring function $\delta(f)$, and a positive integer k, it is to find a subset $\phi \subseteq D$, such that $|\phi| = k$ and

$$\phi = \underset{\phi \subseteq \Gamma, |\phi| = k}{\operatorname{argmax}} \sum_{x,y \in \Gamma} \delta(f) \quad (2)$$