Pattern Recognition Letters 000 (2017) 1–6

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Graph based model for information retrieval using a stochastic local search

Sidali Hocine Farhi*, Dalila Boughaci

LRIA, Dpt. Computer Science, USTHB BP 32 El-Alia, Beb-Ezzouar, Algiers

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Graph mining
Subgraph mining
Frequent subgraphs
Stochastic local search
Information retrieval

ABSTRACT

Graph has become increasingly important in modeling complicated structures and data such as chemical compounds, and social networks. Recent advance of machine learning research has witnessed a number of models based on graphs, from which information retrieval study is also benefited since many of these models have been verified by different information retrieval tasks. In this paper, we investigate the issues of indexing graphs and define a novel solution by applying a stochastic local search (SLS) method to extract subgraphs that will be used for the Information Retrieval process. To reduce the size of the index, we take into consideration the size of the query and the set of the frequent subgraphs. In other words, the subgraphs that will be used to create the index will have a size equal to the size of the query in order to optimize as much as possible the search space and the execution time. The proposed method is evaluated on the CACM collection, which contains the titles, authors and abstracts (where available) of a set of scientific articles and a set of queries and relevancy judgments and compared to the vector-based cosine model proposed in the literature. Our method is able to discover frequent subgraphs serving to establish the index and relevant documents for the Information Retrieval process's output. The numerical results show that our method provides competitive results and finds high quality solutions (documents) compared to the relevant documents cited on the CACM collection.

1. Introduction

The development of information and communication technologies and the increasing number of sectors of human activity has resulted in the production of an unprecedented volume of information, databases size increases in the few past years from some Gigabytes to a thousand Exabyte [9]. In addition, the progressive interconnection of sites via large computer networks such as Internet, as well as the standardization of access and production techniques of these information (URL, HTML, XML,...) make documents available to any Internet user.

This difficulty of access to information has given rise to several information retrieval tools, with the aim of helping the user to find the relevant information he is looking for. It includes search tools by keywords (search engines), by theme (the thematic search tools), by region (geographic search tools) or by the use of several search engines (meta-engines).

Graphs have become increasingly important in modeling complicated structures and schemaless data such as social networks [1], chemical molecule structures [2] and XML documents [15]. A large number of such databases are available on the Web. Data mining and search methods for structured data are needed for users to quickly identify a small subset of relevant data for further analysis and experiments.

The classical graph query problem can be described as follows: Given a graph database D = {G_1, G_2, . . .,G_n} and a graph query q, find all the graphs in which q is a subgraph. The core of the problem is the complexity of subgraph isomorphism [5], a sequential scan is very costly since subgraph isomorphism is NP-complete [14].

In this work, we propose a new model based on stochastic local search (SLS) [10] meta-heuristic. The proposed SLS is used to extract from the set of graphs in the database certain subgraphs (Frequent Subgraphs [8]) that are used to build the index. The extraction process is based on the query size and the support of subgraphs. Then, a graph index is built. Finally, for a given subgraph query, all the indexed subgraphs of the query are determined, and the index is looked up with these subgraphs to obtain a candidate set of graphs containing the indexed subgraphs. The concept of query size is introduced to reduce the complexity of index construction. The proposed method for the graph querying problem is evaluated on the CACM collection.

* Corresponding author.
E-mail address: sid.farhi@gmail.com (S.H. Farhi).

http://dx.doi.org/10.1016/j.patrec.2017.09.019
0167-8655/

The paper is organized as follows. Section 2 defines the graph query problem. Section 3 describes the proposed method. Section 4 discusses results of an experimental study and finally Section 5 provides conclusions and future works.

2. Definitions and problem formulation

In this section, we first give some basic definitions and then describe our graph query problem.

2.1. Definitions

Let us consider G=(N, E) that denotes an input graph where N is the set of nodes and E is the set of edges, where there is an edge e between every pair of nodes (v_i, v_j). Each node $v_i \in N$ has a label from the node label set, and each edge $e \in E$ that connects two nodes v_i, v_j has a label from the edge label set. The edge e can be directed or undirected [4]. In this work, we consider a set of connected graphs $G = \{G_1, G_2, \ldots, G_n\}$.

Definition 1. (Support of subgraph S). The support or (frequency) of subgraph S denoted by Sup(S) in the graph dataset G is the cardinality of the set $\{G_i | S \in G_i, i = 1,..., n\}$.

Definition 2. (Frequent Subgraph). A subgraph S is frequent if its support is no less than a minimum support threshold, *MinSup*.

Definition 3. (Size of subgraph S). The size of subgraph S denoted by size(S) is the number of nodes and edges present in the subgraph S.

Definition 4. (Subgraph Isomorphism). A subgraph isomorphism is an injective function

f: $V(g) \rightarrow V(g')$, such that $\forall u \in V(g), l(u) = l'(f(u))$, and $\forall (u, v) \in E(g)$, $(f(u), f(v)) \in E(g')$ and $l(u, v) = l'(f(u), f(v))$, where l and l' are the label function of g and g', respectively. f is called an embedding of g in g'.

2.2. Problem formulation

The graph query problem tackled in this contribution is defined by considering a general definition of Information Retrieval problem. In our case, the indexing process is based on the frequent subgraphs and the query size.

The processing of graph queries can be divided into three steps:

- Pre-processing, which consists of: (1) constructing the graph database G from the list of the documents in the collection where each document is represented as a graph whose nodes represent the terms of the document and edges correspond to co-occurrence between the terms. Stop words are removed according to the *common_words* file attached to CACM collection. (2) generating randomly a population P of subgraphs of size equal to three (two nodes and one edge) extracted from different graphs and calculates the support *Sup* for each subgraph.
- SLS and Index construction, which is the core of our method, (1): at each iteration, SLS method generates the children of the selected subgraphs where each subgraph is extended according to the graph database G and stores the relevant ones according to their supports in a vector *FSUB*, the process is stopped after a fixed number of iterations. (2) Index construction: the index is elaborated from the final population P generated by the SLS method. Each frequent subgraph is represented by an Id and the related graphs are known.
- Query processing, which enumerates all the frequent subgraphs in *FSUB* as S_q that are isomorphic to the indexed subgraphs of q, then, return the set of related graphs.

The output of our approach is a set of graphs where q is a subgraph ordered by the values of support *Sup*. We store at each iteration, a vector of frequent subgraphs with the best values of support.

3. Proposed approach

In this paper, we propose an index construction method for the problem of graph query. In the following, we detail our proposed method.

3.1. Pre-processing step

Initially, the population P contains randomly generated individuals (subgraphs). In this contribution, a simple program is applied to initialize the population. First, all one-edge subgraphs are created from unique label nodes in graph database G. These subgraphs share the same values of size but may have different values of *Sup*. Then, the set of subgraphs with the highest values of the support are considered, the rest are ignored.

The description of the Pre-processing step is given in Algorithm 1.

Algorithm 1: Pre-processing.

```
Data: Graph Dataset G, pop size |P|
P = Random Selection (one-edge-subgraphs) //Initial subgraph population
Evaluate subgraphs in P using the support function Sup
FSUB = []
FSUB = UpdateFSUB (FSUB, P)
Return FSUB
```

3.2. SLS process

The local search starts with a possible initial solution and tries to improve it by seeking a better solution in the current neighborhood [11]. A neighborhood of a solution A corresponds to adjacent elements to A where each element is achieved by a change in the current configuration. The search process is repeated until no improvement in the current solution could be made.

Stochastic Local Search (SLS) [10] is a widely used approach to solve hard combinatorial optimization problems. Underlying most, if not all, specific SLS algorithms are general methods that can be applied to several optimization problems.

The proposed SLS for the problem of graph query starts with a random solution containing subgraphs with size equals to three (two nodes and one edge), then at each iteration child subgraphs are created according to the support function for the next iteration, any unsatisfactory subgraph must be removed, the selection of subgraphs to include in the next iteration is done according to two criteria:

- The first criterion is to consider similar subgraphs selected randomly with a fixed probability $wp > 0$.
- The second criterion is to choose, with probability 1-*wp*, the best similar subgraphs (maximizing the support function to be considered).

The SLS process is repeated for a number of iterations equal to $(query\_size) - 1$, the different steps of SLS Method are given as follows:

Please cite this article as: S.H. Farhi, D. Boughaci, Graph based model for information retrieval using a stochastic local search, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.09.019