Information Systems 48 (2015) 213–223

Contents lists available at ScienceDirect

Information Systems

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

Density-based data partitioning strategy to approximate large-scale subgraph mining

Sabeur Aridhi [a,b,c,*], Laurent d'Orazio [a,b], Mondher Maddouri [c,d], Engelbert Mephu Nguifo [a,b,*]

[a] Clermont University, Blaise Pascal University, LIMOS, BP 10448, F-63000 Clermont-Ferrand, France
[b] CNRS, UMR 6158, LIMOS, F-63173 Aubiere, France
[c] University of Tunis El Manar, LIPAH – FST, Academic Campus, Tunis 2092, Tunisia
[d] Taibah University, Almadinah, Kingdom of Saudi Arabia

ARTICLE INFO

Available online 14 September 2013

Keywords:
Frequent subgraph mining
Graph partitioning
Graph density
MapReduce
Cloud computing

ABSTRACT

Recently, graph mining approaches have become very popular, especially in certain domains such as bioinformatics, chemoinformatics and social networks. One of the most challenging tasks is frequent subgraph discovery. This task has been highly motivated by the tremendously increasing size of existing graph databases. Due to this fact, there is an urgent need of efficient and scaling approaches for frequent subgraph discovery. In this paper, we propose a novel approach for large-scale subgraph mining by means of a density-based partitioning technique, using the MapReduce framework. Our partitioning aims to balance computational load on a collection of machines. We experimentally show that our approach decreases significantly the execution time and scales the subgraph discovery process to large graph databases.

1. Introduction

Graphs show up in diverse set of disciplines, ranging from computer networks, social networks to bioinformatics, chemoinformatics and others. These fields exploit the representation power of graph format to describe their associated data, e.g., social networks consist of individuals and their relationships. In bioinformatics, the protein structure can be considered as a graph where nodes represent the amino acids and edges represent the interactions between them. Finding recurrent and frequent substructures may give important insights on the data under consideration. These substructures may correspond to important functional fragments in proteins such as active sites, feature positions, junction sites. Moreover, in a social network, frequent substructures can help to identify the few most likely paths of transmission for a rumor or joke from one person to another [1]. Mining these substructures from data in a graph perspective falls in the field of graph mining and more specifically in frequent subgraph mining.

Frequent subgraph mining is a main task in the area of graph mining and it has attracted much interest. Consequently, several subgraph mining algorithms have been developed, such as FSG [2], Gaston [3] and gSpan [4]. However, existing approaches are mainly used on centralized computing systems and evaluated on relatively small databases [5]. Nowadays, there is an exponential growth in both the graph size and the number of graphs in databases, which makes the above cited approaches face the scalability issue. Several parallel or distributed solutions have been proposed for frequent subgraph mining on a single large graph [6–9]. However, the problem of subgraph mining from large-scale graph databases is still challenging.

In this paper, we propose a scalable and distributed approach for large scale frequent subgraph mining based

* Corresponding author at: Clermont University, Blaise Pascal University, LIMOS, BP 10448, F-63000 Clermont-Ferrand, France. Tel.: +33 473407629.
E-mail addresses: aridhi@isima.fr (S. Aridhi), dorazio@isima.fr (L. d'Orazio), mondher.maddouri@fst.rnu.tn (M. Maddouri), mephu@isima.fr (E. Mephu Nguifo).

0306-4379/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2013.08.005

on MapReduce framework [10]. Our approach is not the first one to use MapReduce to solve the distributed frequent subgraph mining task, it differs from and enhances previous works in two crucial aspects. First, previous attempts try to construct the final set of frequent subgraphs iteratively using MapReduce, possibly resulting in a big number of MapReduce passes and an exponential growth of intermediate data especially with large-scale datasets. On the contrary, our approach mines the final set of frequent subgraphs from different partitions of the original dataset by a unique execution of a subgraph mining algorithm. In addition, it offers the possibility to apply any of the known subgraph mining algorithms in a distributed way. Second, our approach differs from previous algorithms by providing a density-based data partitioning technique of the input data. In previous works, the default MapReduce partitioning technique was used to partition the input data, which can be the origin of imbalanced computational load among map tasks [11].

The contributions of this paper are as follows:

- We propose a MapReduce-based framework for approximate large-scale frequent subgraph mining.
- We propose a density-based data partitioning technique using MapReduce in order to enhance the default data partitioning technique provided by MapReduce.
- We experimentally show that the proposed solution is reliable and scalable in the case of huge graph datasets.

This paper is organized as follows. In the next section, we define the problem of large-scale subgraph mining. In Section 3, we present our approach of large-scale subgraph mining with MapReduce. Then, we describe our experimental study and we discuss the obtained results in Section 4. Finally, in Section 5, we present an overview of some related works dealing with the concept of large scale subgraph mining.

2. Problem definition

In this section, we present definitions and notations used in this paper. Then, we present the MapReduce framework. Finally, we define the problem we are addressing and specify our assumptions.

2.1. Definitions

A graph is a collection of objects denoted as $G=(V,E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of edges. A graph G' is a subgraph of another graph G, if there exists a subgraph isomorphism from G' to G, denoted as $G' \subseteq G$. The definitions of subgraph and subgraph isomorphism are given as follows.

Definition 1 (*Subgraph*). A graph $G'=(V',E')$ is a subgraph of another graph $G=(V,E)$ iff $V' \subseteq V$ and $E' \subseteq E$.

Definition 2 (*Graph and subgraph isomorphism*). An isomorphism of graphs G and H is a bijection $f: V(G) \longrightarrow V(H)$ such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. A graph G' has a subgraph isomorphism with G if:

- G' is a subgraph of G, and
- there exists an isomorphism between G' and G.

A task of major interest in this setting is frequent subgraph mining (FSM) with respect to a minimum support threshold. There are two separate problem formulations for FSM: (1) graph transaction based FSM and (2) single graph based FSM. In graph transaction based FSM, the input data comprises a collection of medium-size graphs called transactions. In single graph based FSM, the input data, as the name implies, comprises one very large graph. In this work, we are interested in large scale graph transaction based FSM. The definitions of subgraph support and the graph transaction based FSM are given as follows.

Definition 3 (*Subgraph relative support*). Given a graph database $DB=\{G_1,\ldots,G_K\}$, the relative support of a subgraph G' is defined by

$$Support(G',DB)=\frac{\sum_{i=1}^{n}\sigma(G',G_i)}{|DB|}, \tag{1}$$

where

$$\sigma(G',G_i)=\begin{cases}1 & \text{if } G' \text{ has a subgraph isomorphism with } G_i,\\ 0 & \text{otherwise.}\end{cases}$$

In the following, support refers to relative support.

Definition 4 (*Graph transaction based FSM*). Given a minimum support threshold $\theta \in [0,1]$, the frequent subgraph mining task with respect to θ is finding all subgraphs with a support greater than θ, i.e., the set $SG(DB,\theta)=\{(A, Support(A,DB)): A$ is a subgraph of DB and $Support(A, DB) \geq \theta\}$.

Definition 5 (*Graph density*). The graph density measures the ratio of the number of edges compared to the maximal number of edges. A graph is said to be dense if the ratio is close to 1, and is said to be sparse if the ratio is close to 0. The density of a graph $G=(V,E)$ is calculated by

$$density(G)=2\cdot\frac{|E|}{(|V|\cdot(|V|-1))}.$$

2.2. MapReduce

MapReduce is a framework for processing highly distributable problems across huge datasets using a large number of computers. It was developed within Google as a mechanism for processing large amounts of raw data, for example, crawled documents or web request logs. This data is so large, it must be distributed across thousands of machines in order to be processed in a reasonable amount of time. This distribution implies parallel computing since the same computations are performed on each CPU, but with a different dataset. We notice that the data distribution technique of MapReduce consists of the decomposition of the input data into equal-size partitions called