Information Sciences 459 (2018) 117–134

Contents lists available at ScienceDirect

Information Sciences

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

Mining diversified association rules in big datasets: A cluster/GPU/genetic approach

Youcef Djenouri[a,b], Asma Belhadi[c], Philippe Fournier-Viger[d,*], Hamido Fujita[e]

[a] IMADA, Southern Denmark University, Odense, Denmark
[b] CERIST Research Center, Algiers, Algeria
[c] RIMA, USTHB, Algiers, Algeria
[d] Harbin Institute of Technology (Shenzhen), School of Humanities and Social Sciences, Shenzhen, China
[e] Iwate Prefectural University, 152-52 Sugo, Takizawa, Iwate 020-0193, Japan

ARTICLE INFO

Article history:
Received 23 September 2017
Revised 26 February 2018
Accepted 13 May 2018

Keywords:
Association rule mining
GPU-based algorithm
Genetic algorithm
Cluster of GPUs

ABSTRACT

Association rule mining is a popular data mining task, which has important in many domains. Because the task of association rule mining is very time consuming, evolutionary and swarm based algorithms have been designed to find approximate solutions. However, these approaches still have long execution times, especially when applied on dense and big databases, or when low minsup and minconf threshold values are used. Moreover, these approaches suffer from the lack of diversity in the rules presented to the user. To address these drawbacks of previous algorithms, this paper proposes an efficient parallel algorithm named CGPUGA. It is a genetic algorithm that runs on clusters of GPUs to efficiently discover diversified association rules. It benefits from cluster computing to generate rules. Then, to evaluate rules, which is the most time consuming task, the designed algorithm relies on the massively parallel GPU threads. Furthermore, to deal with the issue of rule quality, the search space of rules is partitioned into several regions assigned to different workers, and rules found by each workers are the merged to ensure diversification. The designed approach has been empirically compared with state-of-the-art algorithms using small, medium, large and big datasets. Results reveal that CGPUGA is 600 times faster than the sequential version of the algorithm for big datasets. Moreover, it outperforms state-of-the-art high performance computing based association rule mining algorithms for real big datasets such as Pokec, Webdocs and Wikilinks. In terms of rule quality, results show that the designed CGPUGA algorithm provides rules of higher quality compared to the state-of-the-art NIGGAR, MSP-MPSO and MPGA algorithms for diversified association rule mining.

1. Introduction

Association Rule Mining (ARM) [20] can be generally defined as the process of extracting frequent patterns, associations or causal relationships between sets of items in a transactional database. Transactions are a common type of data found in databases of numerous companies such as retail stores, all over the world. Thus, there is a huge need for analyzing this type of data. Several sequential algorithms have been proposed for association rule mining such as Apriori [2], FP-Growth [23] and SSFIM [14]. However, these algorithms often have very long runtimes for large or dense datasets, or when

* Corresponding author.
E-mail addresses: djenouri@imada.sdu.dk (Y. Djenouri), abelhadi@usthb.dz (A. Belhadi), philfv@hit.cn (P. Fournier-Viger), hfujita-799@acm.org (H. Fujita).

https://doi.org/10.1016/j.ins.2018.05.031
0020-0255/

low threshold values are used. In fact, most of the research on sequential algorithms for association rule mining have been evaluated with relatively small datasets often containing from 5000 to less than 1 million transactions [48].

To address the problem of long execution times and the size of datasets, the development of parallel algorithms has been studied. Several researchers have parallelized the Apriori and FP-Growth algorithms for cluster-based architectures [8,24,28,38,50]. However, these algorithms remains inefficient on big instances such as the Webdocs benchmark dataset. To decrease the runtime of association rule mining, evolutionary and swarm intelligence techniques based approaches have been used in ARM such as Genetic Algorithms (GA), Genetic Programming (GP), Mimetic Algorithms (MA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Bees Swarm Optimization (BSO), and Bat Algorithms (BA). Some of the most popular ARM algorithms based on these techniques are GAR [31], G3PARM [39], GA-GD [10], PSOARM [26] for PSO, ACO_R [27] for ACO, and BSO-ARM [13] for BSO. According to recent surveys [32,33], genetic algorithms are more efficient than other evolutionary and swarm intelligence ARM algorithms. However, two major issues have been observed:

(1) *Solution quality.* Evolutionary and swarm intelligence based approaches are approximate algorithms. i.e. they may miss some valid association rules that would be found by traditional association rule mining algorithms. To address this issue, three algorithms have been developed to find a set of diversified association rules (NIGGAR [29], MSP-MPSO [21] and MPGA [22]). The idea is to adopt strategies to ensure diversification of the rules found when exploring the search space of association rules. Although these approaches provides an improvement in terms of rule quality, it is desirable to further improve this aspect.
(2) *Runtime performance.* Though, evolutionary and swarm intelligence based approaches find solutions in reasonable time on medium datasets, the runtime performance greatly decreases for large and big datasets. To deal with this issue, many High Performance Computing (HPC) based algorithms have been proposed [43,50]. However, these approaches are still inefficient for big datasets containing several millions of transactions such as the Webdocs benchmark dataset.

This paper addresses both of these issues of previous association rule mining algorithms, which are to decrease the runtime performance and increase the quality of rules presented to the user. Motivated by the success of genetic algorithms for the association rule mining problem, as well as our successful preliminary results using cluster and GPUs for association rule mining, this paper proposes a cluster-computing and GPU-computing based algorithm, named CGPUGA, for mining big transactional databases. To solve the runtime performance issue, it generates rules using a computer cluster. Then, to evaluate rules, which is the most time consuming task, the designed algorithm utilizes massively parallel GPU threads. Furthermore, to find rules of high quality, the search space of rules is partitioned into several regions assigned to different workers, and rules found by each workers are the merged.This ensures the diversification of the rules presented to the user.

To validate the proposed approach, several experiments have been carried out on large and big datasets, having up to 50 million transactions. The results are very promising in terms of speed up and rule quality. They reveal that CGPUGA is 600 times faster than the sequential version of the algorithm when dealing with synthetic big datasets. Moreover, it outperforms state-of-the-art HPC-based ARM approaches for big datasets such as Pokec, Webdocs and Wikilinks. In terms of rule quality based on the criteria of diversification, results also show the efficiency of the proposed algorithm compared to the state-of-the-art NIGGAR, MSP-MPSO and MPGA algorithms.

The rest of the paper is organized as follows. Section 2 introduces preliminaries related to the ARM problem, genetic algorithms, GPU and cluster computing. Section 3 reviews recent parallel ARM algorithms. Section 4 describes the proposed CPUGA algorithm. Then, Section 5 presents a performance evaluation. Finally, Section 6 draws a conclusion.

2. Preliminaries

This section introduces important preliminaries related to association rule mining, genetic algorithms, cluster computing and GPU computing.

2.1. Association rule mining

Association rule mining [2] consists of discovering frequent patterns, associations or causal structures among sets of items from a given transactional database. The ARM problem is defined as follows. Let T be a transactional database, defined as a set of transactions, $\{t_1, t_2, \ldots, t_m\}$, and I be a set of n different items (symbols or attribute values) $\{i_1, i_2, \ldots, i_n\}$. An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$. The itemsets X and Y are called the rule antecedent and consequent, respectively. The interpretation of a rule $X \rightarrow Y$ is that if an itemset X appears in a transaction, it is likely to co-occur with the itemset Y according to some interestingness measure.

Two interestingness measures are traditionally used in association rule mining to evaluate how interesting a rule is: the support and confidence. The support of an itemset $I' \subseteq I$ is the number of transactions that contains I'. The support of a rule $X \rightarrow Y$ is the support of $X \cup Y$, while its confidence is $support(X \cup Y)/support(X)$.

The confidence of an association rule is a measure of its strength. An association rule $X \rightarrow Y$ with a confidence of 80% means that 80% of the transactions that contain X also contain Y. The association rule mining problem consists of extracting all interesting (valid) rules from a transactional database. A rule is deemed interesting if its support and confidence are no less than some user-defined *MinSup* and *MinConf* [2] thresholds, respectively.