Computers in Human Behavior 68 (2017) 244–253

Contents lists available at ScienceDirect

Computers in Human Behavior

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

Full length article

Frequent itemset mining using cellular learning automata

Mohammad Karim Sohrabi*, Reza Roshani

Department of Computer Engineering, Semnan Branch, Islamic Azad University, Semnan, Iran

ARTICLE INFO

Article history:
Received 21 January 2016
Received in revised form
14 October 2016
Accepted 20 November 2016

Keywords:
Frequent itemset mining
Cellular automata
Data mining
Association rules
Parallel frequent itemset mining

ABSTRACT

A core issue of the association rule extracting process in the data mining field is to find the frequent patterns in the database of operational transactions. If these patterns discovered, the decision making process and determining strategies in organizations will be accomplished with greater precision. Frequent pattern is a pattern seen in a significant number of transactions. Due to the properties of these data models which are unlimited and high-speed production, these data could not be stored in memory and for this reason it is necessary to develop techniques that enable them to be processed online and find repetitive patterns. Several mining methods have been proposed in the literature which attempt to efficiently extract a complete or a closed set of different types of frequent patterns from a dataset. In this paper, a method underpinned upon Cellular Learning Automata (CLA) is presented for mining frequent itemsets. The proposed method is compared with Apriori, FP-Growth and BitTable methods and it is ultimately concluded that the frequent itemset mining could be achieved in less running time. The experiments are conducted on several experimental data sets with different amounts of minsup for all the algorithms as well as the presented method individually. Eventually the results prod to the effectiveness of the proposed method.

1. Introduction

Frequent patterns are one of the major issues in the field of data analysis. Many books and articles have been published in this regard and significant progresses were made. Frequent patterns are essentially itemsets, sequences or infrastructures which are repeated in a data set with a frequency greater than or equal to a threshold determined by the user. In this paper, frequent patterns and frequent itemsets will be used interchangeably. Since the frequency of customer transactions tend to be enormous in the shops, hence they may not necessarily fit in memory. Besides, the potential number of frequent itemsets can vary from item numbers, Even though the actual number of frequent items could be less than that. Therefore calls for some scalable algorithms which are originally comparable with highly frequent and less frequent itemsets are heard. Many algorithms can be availed for this task, among which some act based on candidate creation and then investigation while others attempt to create a Tree without candidate generation and then find the frequent items by scanning on the Tree. Some other algorithms try to perform this task through the BitTable. The researchers first consider a number of algorithms in this field, and then present a cellular learning automata-based approach for frequent itemsets mining.

1.1. Statement of the problem

Suppose that E is an itemset. A transaction on E is $T = (t_{id}, E)$, where T_{id} is the transaction ID and E is an itemset. Also, a database of DB on E is: consists mainly of transaction sets in a way that each transaction enjoys a unique ID. The researchers claim that a transaction $T = (t_{id}, E)$ supports itemsets A if $A \subseteq E$. The cover of A set in DB is composed of a set of transactions that support A $cover(A, DB) = \{t_{id} | (t_{id}, E) \in DB, A \subseteq E\}$ Besides, the support of A set in DB denotes the number of transactions already affiliated to A cover in DB $support(A, DB) = |cover(A, DB)|$. It is interpreted that an itemset is frequent if its support is more than $threshold support(A, DB) \geq min\_sup$, which min_sup is the minimum threshold defined by the user.

1.2. Cellular learning automata

Cellular Learning Automata (CLA) is better known as a modified model of Cellular Automata. In Cellular Automata, the space is

* Corresponding author.
E-mail addresses: Amir_sohraby@aut.ac.ir (M.K. Sohrabi), r.roshany@gmail.com (R. Roshani).

http://dx.doi.org/10.1016/j.chb.2016.11.036
0747-5632/

defined as a network in which each part is called a cell. CLA is mostly resorted for the systems which comprise simple components and their behavior is determined and modified based on their neighbors' behavior as well as past experiences. Fig. 1 displays the examples of well-known neighborhood in Cellular automata.

Simple components of this model could exhibit complex behavior by means of interaction with one another. Each CLA interacts with an environment and is composed of a CA as well as Learning automata. Fig. 2 puts the relationship between the automata and the environment on display.

Taking the learning rules in CLA and neighbors' modes into account, each new transaction in dataset will result in either a reward or a penalty. The reward or penalty updates the structure of CLA respectively. The reward and penalty in Cellular Automata are considered in the following 3 modes:

1. Linear Reward Penalty (LRP): the amount of reward and penalty are the same in this mode.
2. Linear Reward Epsilon Penalty (LRεP): the amount of reward is manifold to penalty in this mode.
3. Linear Reward Inaction (LRI): In this mode, rewards are given without any penalties.

Based on their properties, the automata come into different classifications. A Cellular automaton is called regular if the neighborhood in cells already possesses a sorted regularity such as the neighborhood pattern of Von Neumann or Moore. In some applications, the need for an unlimited model like subsequence mining is from one neighborhood is felt, that is, the groups are completely random in these networks consequently it is not possible to come up with a sorted structure for them. This type of automata is labeled as Irregular Cellular Learning Automata (ICLA) (FathiNavid & Aghababa, 2012). Besides, a Cellular Learning Automata is called uniform if all cells in CLA have the same neighborhood function, rules and learning. Otherwise, it is named non-uniform (Esnaashari & Meybodi, 2007).

The rest of paper includes the following sections. In the second section, the researchers describe the main approaches to solve the frequent itemset mining problem. It is then followed by the explanation of the new method. Next comes the experimental results and it is concluded in the fifth section.

2. Related works

Data mining has numerous applications, including analysis of customer purchase patterns (Hu & Yeh, 2014), analysis of web access patterns (Ristoski & Paulheim, 2016), the investigation of scientific or medical processes (Wang, Davis, & Ren, 2016), and several types of prediction I social networks (Sohrabi & Akbari, 2016). There exist several types of data mining techniques. Association rule mining and frequent pattern extraction are two of the most

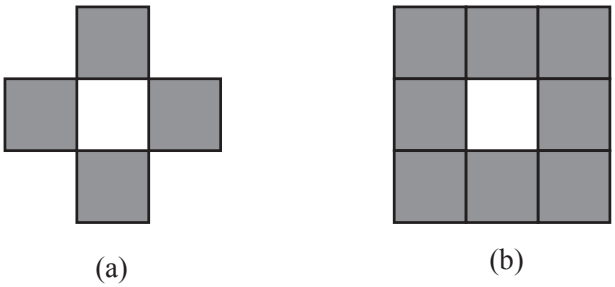

Fig. 1. Neighborhoods (a) Von Neumann, (b) Moore.

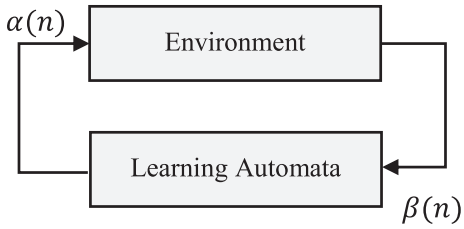

Fig. 2. The interaction of learning automata and the environment.

important data mining techniques which have been introduced in 1993 for the first time. There are two main approaches to find and extract frequent patterns from databases, efficiently. Apriori-like algorithms mine data using the 'candidate generation and test' method to find the frequent patterns in a breadth first search manner. Depth first mining algorithms use the second approach which usually compress the dataset in a tree structure and mine that compressed tree. Itemsets, Sequences, and graphs are three of most using types of patterns which have been mined and extracted from tremendous volumes of data by different pattern mining techniques. Since this work is a bit wise parallel mining algorithm based on cellular learning automata, the literature review is organized as follows in this section. First of all, some of the most important traditional itemset mining algorithms will be discussed. Then, several mining methods will be described which used bit-wise approach as their improvement technique. Finally, different existing distributed and parallel mining approaches will be explained.

Agarwal, Imielinski, and Swami (1993) for the first time presented an interesting property called apriori in the form of association rules. Thanks to this property, a k-itemset can only be frequent when all its subsets are frequent. The result that could be retrieved from this property is that the super itemset of a non-frequent itemset are essentially non-frequent. Furthermore, this property will enable the apriori-based algorithm to piece together an itemsets of non-frequent k-items in itemsets mining of a (k+1)-item. Thus, in order to come up with a complete set of all existing frequent itemsets in a transaction database, firstly the transaction database should be scanned once thoroughly to find all the existing frequent 1-item itemsets and then generate all frequent 2-items itemsets (which can be created by these frequent items) namely as a frequent candidate. Since each of these 2-itemsets consists of only frequent items, they are potentially frequent within the transaction base. To ensure the frequency, it is required to scan the transaction database afresh and identify the frequent and non-frequent 2-items candidates. When the frequent 2-items itemsets are already identified, we take them identically to create 3-items candidates and to test their frequency or non-frequency through rescanning the transaction database. This process of using frequent k-item itemsets in generating (k+1)-items candidates and testing their frequency or non-frequency through a complete scan of transaction, continues until all the existing frequent itemsets in transaction database is mined. After the apriori method was presented, extensive studies were carried out to improve its efficiency and application. In 1995, Park attempted to use Hashing Techniques to improve its efficiency (Park, Chen, & Yu, 1995). Savasere, Omiecinski, and Navathe (1995) applied the Partitioning Method to improve the algorithm. In this method, by identifying a specific number of large frequent itemsets through a two-step process of transaction database scan -based on the large itemsets mined- the transaction database will be divided into the overlapping partitions and then the results of each separate partition mining will be subsequently merged. Toivonen (1996) used the Sampling Method