Distributed BitTable Multi-Agent Association Rules Mining Algorithm

Walid Adly Atteya¹, Keshav Dahal¹, and M. Alamgir Hossain²

¹ School of Computing, Informatics and Media, Bradford University, United Kingdom
{waaabdo, k.p.dahal}@bradford.ac.uk

² School of Computing, Engineering and Information Sciences, Northumbria University, United Kingdom
alamgir.hossain@northumbria.ac.uk

Abstract. Many algorithms have been proposed for the discovery of association rules. The efficiency of these algorithms needs to be improved to handle real-world large datasets. This efficiency can be determined mainly by three factors. The way candidates are generated, the way their supports are counted and the data structure used. Most papers focus on the first and the second factors while few focus on the underlying data structures. In this paper, we present a distributed Multi-Agent based algorithm for mining association rules in distributed environments. The distributed MAS algorithm uses Bit vector data structure that was proved to have better performance in centralized environments. The algorithm is implemented in the context of Multi-Agent systems and complies with global communication standard Foundation for Intelligent Physical Agents (FIPA). The distributed Multi-Agent based algorithm with its new data structure improves implementations reported in the literature that were based on Apriori. The algorithm has better performance over Apriori-like algorithms.

Keywords: Multi-Agent Systems, Distributed Data Mining, Association Rules.

1 Introduction

Finding frequent itemsets is one of the most important data mining research fields. The problem was first presented in [1] with another extension in [2]. Its main algorithm, Apriori, had an impact on other data mining techniques as well. Association rules and frequent itemsets mining became a widely research area, and hence, most researchers have tried to present faster algorithms. Many of these algorithms were Apriori-based or Apriori extensions. Most association rule algorithms use hash-trees extensively to speed up the search for itemsets. Those who adopted Apriori strategy tended to adopt the whole set of procedures and data structures as well.

Recently, algorithms have been proposed to increase the efficiency of these algorithms to improve real-world large datasets. Some algorithms focused on the way candidates are generated. Others focused on the way their supports are counted. Few researchers have focused on the underlying data structure used which was a hash-tree in case of Apriori-based algorithms.

Park et al. [9] has invented a well known technique called DHP (Direct Hashing and Pruning) and was enhanced in [10]. DHP uses a hash technique that makes it very...
efficient for the generation of candidate itemsets, in particular for the large two-itemsets and employs effective pruning techniques. The reduction of the number of generated candidates greatly improves the performance of the whole process. However, Park used this hashing technique to mine association rules in centralized database. Bodon \cite{6} has demonstrated that a Trie data structure outperforms hash-trees. Tries appeared to offer simpler and scalable algorithms which turned out to be faster. Bodon has implemented Apriori association rule mining algorithm using Trie data structure rather than Hash Tree. Further publication \cite{3} proved that the data structure Trie appeared to be faster than the original algorithm. Bodon has extended his implementation for mining itemset sequences in \cite{5}. Other researchers have adopted the Trie structure to mine association rules on centralized databases \cite{4}.

Recently, a novel approach by Dong has presented a very effective algorithm named as BitTableFI \cite{7}. The algorithm uses a special data structure BitTable horizontally and vertically to compress database for quick candidate itemsets generation and support count, respectively. Dong has proven that this data structure is faster than the hash tree used by Apriori. Results were obtained by applying the BitTable data structure on two synthetic centralized datasets. Song et al. \cite{11} is one of the extensions that is based on this technique.

In this paper, we present an efficient distributed MAS algorithm. The efficiency of the algorithm is obtained by modifying the data structure used and the way candidates are generated and counted. The rest of the paper is organized as follows. The next section describes the proposed distributed BitTable Multi-Agent based algorithm. Section 3 describes the model experiments and evaluation. The last section presents the conclusion and the future work.

2 Distributed BitTable Multi-Agent Association Rules Algorithm

In earlier work, we have presented an enhancement for Apriori algorithm using a simpler data structure \cite{8}. The algorithm was implemented on centralized database. Previous work has extended the basic concepts of Apriori like algorithms to work in distributed environments using cooperative Multi-Agents \cite{12}. The parallelism of the candidate generation and the support count processes among these distributed agents helped in decreasing the time needed for the whole mining process. The previously proposed algorithm was implemented on distributed medical databases \cite{13} for patient diagnostic system regarding Inflammation of urinary bladder and Nephritis of renal pelvis origin diseases. The proposed model improved the diagnostic knowledge and discovered the diseases based on the minimum number of effective tests, thus, provided accurate medical decisions based on cost effective treatments. The constructed Knowledge base could predict the existence or the absence of the diseases, thus improving the medical service for the patients.

In this section, we present the distributed BMAS algorithm which combines the best of different association rules algorithms and techniques in order to achieve the best performance and execution time. The proposed algorithm combines the association rules as a data mining technique, the BitTable data structure that was proved to be a very efficient data structure for mining frequent itemsets \cite{11} \cite{14} and the Multi-Agents technique to decrease the time needed for the candidate generation and the support count.