A Survey on Techniques For Speeding Up Association Rule Mining

Vinayak Suresh Shukla¹, Prof. Dr. S.A. Itkar²

Computer Department, P. E. S. Modern College of Engineering, Shivajinagar, Pune-05

Abstract- The interest in digital data storage is increased in recent years due to ease of use and low cost storage Medias. This, in parallel, has affected the knowledge mining process from large volumes of data in distributed as well as heterogeneous data. This led to evolution of association mining algorithms like Apriori and FP-Growth, to meet up time and memory requirements. Anyhow, these algorithms underperform in case of extremely large number of stored data in order to mine. Hence, instead of proposing a better algorithms or modifying the existing ones, the mining performance can be effectively improved by defining a New Data Structure on which the existing algorithms are applied. So, here we studied the technique to speed up the association rule mining with traditional algorithms and new data structures, in order to meet up time and memory requirements in efficient manner.

General Terms- Data Compression, Data Structure, Knowledge Discovery.

Keywords- Association Rule Mining (ARM), Index Compression.

I. INTRODUCTION

In recent years, production of digital data in every field is increased [1], so the storage of data. This made the knowledge discovery much interesting but time consuming. Association rule mining, the well-known and researched mining technique lags behind in situations where the volume of data is too extreme.

An association rule is defined as an implication, X → Y, where X and Y be set of items from a data set. I = {i₁,i₂,...,in} being the set of all the items comprised in a data set, and T = {t₁,t₂,...,tn} being the set of all transactions within that data set. Then, X and Y are subsets of I, i.e., X ⊂ I and Y ⊂ I. An association rule X → Y determines that if X is satisfied, then it is highly likely that Y is also satisfied, X and Y being the antecedent and consequent, respectively.

Many efficient algorithms, like e.g ARM [2], have been proposed for mining data sets with the reduction of number of candidate sets, number of transactions and comparison of both. But it can be found that these algorithms are time consuming and hard processes when the data set sizes are extremely high.

So as a solution, instead of modifying original functioning of mining algorithms, a new compressed data structure [3] is to be defined which will be used by algorithms for mining. That will help in better handling, speeding up the access to data and fast computing when further used by any existing mining algorithm.

The paper is organized into four sections: Section II gives brief review of the speeding up the rule mining. Section III describes the performance parameters considered to compare these approaches and finally, Section IV summarizes and presents the conclusions.

II. RELATED WORK

Association rule mining [2] is the most important technique used for mining the rules. It aims to extract strong and relative items among the given data set. Basically it was designed for market basket analysis which would help shopkeepers arranging the sale items in order to grow the business.

Apriori [4] is the first algorithm for association rule mining. It is based on comprehensive search. It works in two steps as I. finding all frequent item sets from data set and II. Deriving association rules
from those frequent item sets. If data set has k single items, and N number of transactions, then \(2^{k-1}\) item sets can be generated. And of the data set size is huge, the processing is hard, because the complexity of above operation will be \(O(N \times M \times k)\) just to obtain candidate sets.

The frequent pattern FP-growth algorithm [5] was introduced to reduce the number of transactions and related comparisons. Data needs to be scanned only once since FP-growth stored frequent items in a tree structure. But it suffers with large number if I/O and large number of memory required storing all sets. To overcome the existing drawback, like large memory requirements and long computational times, many evolutionary association rule mining algorithms [6],[7],[8],[9], were proposed. But, even after considering mining rules from different views, working with high-dimensional data was hard.

So as a solution, it was found that creating a new data structure[3] in order to reduce the original data set size would be advantageous, and that can be used for speeding up the mining, with current algorithms without changing their original functioning.

The data structure proposed in the paper was meant to reduce the data set size and produce faster data access. Despite this guarantees the data validity and doesn’t change the original data values. The conversion of data into new data structure comprises of three steps viz. 1. Shuffling, 2. Inverted Index Mapping and finally and 3. Data Compression [3].

1. **Shuffling**: The shuffling strategy makes the use of data set property, i.e. similarity among records and transactions within same or specific data sets. It aims to cluster the records or transactions with the short Hamming Distance. Instead of storing all the consecutive records, it stores the distance of consecutive records after the initial record in each cluster/group. This strategy has complexity of \(O(N)\) for \(N\) number of records.

2. **Inverted Index Mapping**: On a resulting data, this uses mapping strategy to index the attribute values according to the transactions satisfied. The Inverted Index Mapping is the most important step of building the proposed data structure, and it is responsible to construct an index-based structure. This step takes sorted data as an input. Following the inverted index strategy, it assigns key-value pair to each list of attributes. Complexity of this step is \(O(P \times Q)\) where \(P\) is number of unique values and \(Q\) is number of instances.

3. **Data Compression**: It uses procedure similar to Run Length Encoding (RLE). The consecutive indices for each attribute can be grouped together in two-tuple i.e. starting index and total number of consecutive records/transaction. Complexity of this step is \(O(P \times Q)\) where \(P\) is number of unique values and \(Q\) is number of instances.

### III. PERFORMANCE PARAMETERS

1. **Data Size**, 2. **Processing Time**, and 3. **Memory Requirements**:

The traditional data set are structured in 2-D tables [10] which stores the continues data, and sometimes duplicate data. The New Data Structure compresses the structure using above mentioned techniques and the resulting structure is much less than the traditional one. This leads to reduce loading time and memory requirements.

### IV. CONCLUSION

With the reduced main memory requirements and less data access time, the proposed data structure allows the arm algorithms to work efficiently with larger volumes of data. The described data structure overcomes the drawbacks of larger memory requirements and long data access time of arm algorithms, without changing their original schema. Use of this data structure resulted in improving performance capabilities of existing algorithms. The experimental result showed that the proposed data structure ensures data reduction (loss-less), reducing main memory requirements and shorter data access time at high levels. Hence, it can be used where efficient data access time is required with the large data sets.
REFERENCES