Proposed Method for Modified Apriori Algorithm

Thanda Tin Yu, Khin Thidar Lynn
University of Computer Studies, Mandalay
Mandalay, Myanmar

Abstract - There are many algorithms in data mining. Apriori algorithm is the most important algorithm which is used to extract frequent itemsets from large database and which gets association rule. Firstly, we check if the items are greater than or equal to the minimum support and find the frequent itemsets respectively. Then, the minimum confidence is used to form association rule. This paper proposed the new algorithm based on Apriori algorithm. In this new algorithm, it can reduce the computational complexity than Apriori algorithm. So the processing time is faster. And it can be used in any dataset which is executable with Apriori algorithm.

Keywords: Data mining, Apriori algorithm, Frequent Pattern mining, Modified Apriori algorithm

1 Introduction

Mining Association rules can be divided into two phases: Phase one, developed by user in accordance with the minimum degree of support from the database to find a frequency greater than or equal to the minimum support of all frequent item sets.

The earliest in 1993, Agrawal, proposed AIS algorithm has too many sets of candidate projects, which results in the last mining association rules poor efficiency of the way, so in 1994, Agrawal, also filled a Apriori algorithm. Later many Apriori based on the improved algorithm have been proposed, such as Savasere, who proposed partition algorithm, Toivonen made sampling algorithm, Park who proposed the use of technology DHP hash algorithm in 1995, on the above-mentioned study, the mining association rules is to increase the efficiency of its method not only reducing the non-collection of related items. Data Mining has found its applications use classification, clustering, prediction, association rule mining, pattern recognition and pattern analysis. This research uses any dataset which can implement the Apriori algorithm. This research discuss the mechanism for WEKA to use Apriori algorithm. The new method approach is described as follows:

1.1 Modified Apriori algorithm

Algorithm: Find frequent itemsets using new approach based on term frequency matrix.

Input: \( D \), dataset \( \text{min}_\text{sup} \), the minimum support count threshold.

Output: frequent itemsets in \( D \).

Method:

1. Count numbers of distinct items, \( N \), in \( D \).
2. Create frequency matrix, \( M \), which is \( N \times N \) dimensions and initialize all elements with zero.
3. For each transaction \( t \) in \( D \)
   a. For each possible pair of items \( p \) in transaction \( t \)
   b. Increase frequency by 1 to corresponding element of frequency matrix, \( M \)
4. Generate infrequent itemsets from frequency matrix, \( M \)
5. For each row in frequency matrix, \( M \)
   a. Extract all possible frequent itemsets by matching rows and columns satisfying \( \text{min}_\text{sup} \)
6. Remove extracted frequent itemset which included infrequent subsets
7. Remove extracted frequent itemset which is not satisfied \( \text{min}_\text{sup} \)

1.2 Related works

Paula R.C.Silva et.al, 2015 expresses a novel approach to discover professional profile patterns from LinkedIn by using association rule mining to extract relevant patterns from the data warehouse, evaluate their approach academic activities and curricula in educational instructions[6]. P. Nancy et.al, 2013 discussed the facebook 100 universities data set in United States from which association rules are mined. Knowledge pattern regarding the association between the major(course) and gender were identify[11]. Trand et.al, 2010 examined the community structures of facebook networks whose links represent “friendship” between user pages within each of five American universities[2]. Ahmet Seiman Bozkir et.al, 2009 investigated demographic characteristics of facebook users and their frequency, time spent on facebook and membership in facebook group using association rules[3]. Xiao Cui et.al,2014 explored the relationship among different profile attributes in Sina Weibo using association rule mining to identified the dependency among the attributes [5]. Balaji Mahesh, VRK Rao G Subrahmany [14], they proposed adaptive implementation of Apriori algorithm for Retail Scenario in cloud environment which solves the time consuming problem for retail transactional databases. It aims to reduce the response time.
significantly by using the approach of the frequent itemsets. Wang Feng, Li Young-hua et al presented" An Improved Apriori Algorithm Based on the Matrix“ used the matrix effectively indicating the operations in the database and used the “AND operation” to deal with the matrix to generate the largest frequent itemsets. It is not needed to scan the database again and again to perform operations and therefore tasks less time and it also reduced the number of candidates of frequent itemsets greatly. Ke-Chung Lin et.al proposed new algorithm , LP tree (Linear Prefix-Tree) which is composed of array forms and minimize pointers between nodes . This algorithm requires minimum information required in mining process and linearly accesses corresponding nodes. This results in less usage of memory for building trees and it needs less time for traverse in a linear structure[17].

2 Background theory

2.1 States of problem

Apriori algorithm suffers from some weakness in spite of being clear and simple. Apriori will be very low and inefficient when memory capacity is limited with large number of transactions. The proposed approach method in this paper reduce the time spent for searching the database and performing transactions for frequent itemsets and also reduces the complexity computation with large number of data transaction which is described in the proposed method, modified Apriori algorithm.

Apriori Algorithm takes a lot of memory space and response time since it has exponential complexity. For example, there are 50 transactions so it will have $2^{50}$itemsets and it is also mining twice. We can reduce the itemsets by frequent pattern mining. It will reduce time taken, a lot of space and many transactions.

2.2 Association rule mining process

Association rule generation is usually split up into two split steps. First, minimum support is applied to find all frequent itemsets in a database. Second, these frequent itemsets and minimum confidence constraint are used to form rules.

While the second step is straight forward, the first step need more attention,

Finding all frequent itemset in a database is difficult since it involves searching all possible itemsets (item combination).

2.2.1 Pseudo code of Apriori algorithm

Ck: Candidate item set of size k
Lk: frequent item set of size k
L1= {frequent items};
for (k = 1; Lk!=∅; k++) do begin
Ck+1= candidates generated from Lk;
for each transaction t in database do
increment the count of all candidates in Ck+1 that are contained in t
Lk+1= candidates in Ck+1 with min_support
end
Return Uk Lk;

2.2.2 Apriori algorithm

The research work uses the classical Apriority algorithm for extracting the association rules. The problem of association rule mining is defined as:

Let I= {i1, i2, ......, in} be a set of n binary attributes called items.

Let D= {t1,t2,...., tm} be a set of transactions called the database.

Each transaction in D has a unique transaction ID and contains a subset of the items in I.

A rule is defined as an implication of the form X=>Y where X,Y C I and X∩Y=Ø.

The support $\text{supp}(X)$ of an itemset X is defined as the proportion in the data set which contain the itemset.

The confidence of a rule is defined $\text{conf}(X=>Y) = \text{supp}(X U Y) / \text{supp}(X)$.

The lift of a rule is defined as $\text{lift}(X=>Y) = \text{supp}(X U Y) / \text{supp}(Y) \times \text{supp}(X)$ that expected if X and Y were independent. Correlation analysis using lift that value is less than 1, there is a negative correlation between the occurrence.

2.3 Sample usages of Apriori algorithm

Transaction data for an All Electronics branch is followed.

Table 1: Transaction table

<table>
<thead>
<tr>
<th>TID</th>
<th>List of item_IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>I1,I2,15</td>
</tr>
</tbody>
</table>
The result of frequent itemsets may be \{I1, I2, I3\}, \{I1, I2, I5\}, for support count = 2 obtained by Apriori algorithm. This method takes a lot of time and steps to solve the problem by scanning database frequently. We will solve this problem by our proposed method as follows:

### 2.4 Example of our proposed algorithm

First, we construct the matrix the above example in table (1). Then the matrix may be following.

\[
\begin{array}{cccccc}
I1 & I2 & I3 & I4 & I5 \\
6 & 4 & 4 & 1 & 2 \\
4 & 7 & 4 & 2 & 2 \\
4 & 4 & 6 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
2 & 2 & 1 & 0 & 2
\end{array}
\]

**Figure 4.** Frequency matrix

**Step 1**

In support count = 2, extract all possible frequent itemset by matching rows and columns satisfying min-sup = 2.

Then we get \{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}

**Step 2**

1. Remove extracted frequent itemset which included infrequent subsets
2. Remove extracted frequent itemset which is not satisfied \text{min\_sup}

Then we get \{I1, I2, I3\}, \{I1, I2, I5\}, \{I2, I4\}

The result of frequent itemset may be \{I1, I2, I3\}, \{I1, I2, I5\}, \{I2, I4\}

The algorithm uses only 2 step for frequent itemset. Thus our proposed algorithm is less computational complexity than the Apriori algorithm.
3 Experimental results

Association rule in Apriori algorithm found in WEKA tools and Modified Apriori algorithm in Java with arff file. It can be found in time comparison and less than computational complexity. We can solve the problem in Table 2 as follows:

Min support = 0.2 (for 2 instances)
Total execution time = 18 ms (in Modified Apriori algorithm)

Table 2: result of rules in Apriori and Modified Apriori

<table>
<thead>
<tr>
<th>Usages</th>
<th>Conf</th>
<th>lift</th>
<th>Rule</th>
<th>Consequent</th>
<th>Antecedent</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEKA</td>
<td>1</td>
<td>1.29</td>
<td>R1</td>
<td>I4</td>
<td>I2</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>1</td>
<td>1.29</td>
<td>R1</td>
<td>I4</td>
<td>I2</td>
</tr>
<tr>
<td>WEKA</td>
<td>1</td>
<td>1.5</td>
<td>R2</td>
<td>I2, I5</td>
<td>I1</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>1</td>
<td>1.5</td>
<td>R2</td>
<td>I2, I5</td>
<td>I1</td>
</tr>
<tr>
<td>WEKA</td>
<td>1</td>
<td>1.29</td>
<td>R3</td>
<td>I1, I5</td>
<td>I2</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>1</td>
<td>1.29</td>
<td>R3</td>
<td>I1, I5</td>
<td>I2</td>
</tr>
<tr>
<td>WEKA</td>
<td>1</td>
<td>2.25</td>
<td>R4</td>
<td>I5</td>
<td>I1, I2</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>1</td>
<td>2.25</td>
<td>R4</td>
<td>I5</td>
<td>I1, I2</td>
</tr>
</tbody>
</table>

The result of two algorithm, data set is small, so execution time is not too much different. But frequent item set may differ. In section (2.3) the frequent item set may be \{I1,I2,I3\}, \{I1,I2, I5\}, \{I2, I4\} But in WEKA \{I1, I2, I5\}, \{I2, I4\}, \{I1, I5\}, \{I2, I5\}.

So our proposed algorithm is less computational complexity than the Apriori algorithm.

In 100 American university data set, we test 3000 instances and 8 attributes (such as ID, FacultyStatus, Gender, Major, SecondMajor, House, Year, HighSchool)

Minimum support: 0.1 (300 instances)

Minimum confidence: 0.9
Total execution time = 16 s (in WEKA Apriori algorithm)
Total execution time = 32 ms (in Modified Apriori algorithm)

Table 3: result of rules in Apriori and Modified Apriori in 100 American university data set

<table>
<thead>
<tr>
<th>Usages</th>
<th>Conf</th>
<th>lift</th>
<th>Rule</th>
<th>Consequent</th>
<th>Antecedent</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEKA</td>
<td>1</td>
<td>1.23</td>
<td>R1</td>
<td>Gender=1</td>
<td>Year=2009</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>1</td>
<td>1.23</td>
<td>R1</td>
<td>Gender=1</td>
<td>Year=2009</td>
</tr>
<tr>
<td>WEKA</td>
<td>0.99</td>
<td>1.22</td>
<td>R2</td>
<td>Gender=1</td>
<td>Year=2008</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>0.99</td>
<td>1.22</td>
<td>R2</td>
<td>Gender=1</td>
<td>Year=2008</td>
</tr>
<tr>
<td>WEKA</td>
<td>0.99</td>
<td>1.22</td>
<td>R3</td>
<td>House=89</td>
<td>FacultyStatus =1</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>0.99</td>
<td>1.22</td>
<td>R3</td>
<td>House=89</td>
<td>FacultyStatus =1</td>
</tr>
<tr>
<td>WEKA</td>
<td>0.99</td>
<td>2.31</td>
<td>R4</td>
<td>Major=0</td>
<td>SecondMajor =0</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>0.99</td>
<td>2.31</td>
<td>R4</td>
<td>Major=0</td>
<td>SecondMajor =0</td>
</tr>
<tr>
<td>WEKA</td>
<td>0.95</td>
<td>1.17</td>
<td>R5</td>
<td>Gender=1</td>
<td>Year=2007</td>
</tr>
<tr>
<td>Modified Apriori</td>
<td>0.95</td>
<td>1.17</td>
<td>R5</td>
<td>Gender=1</td>
<td>Year=2007</td>
</tr>
</tbody>
</table>

In other data sets, we test this two algorithm, association rules, confidence and lift are the same. But execution time is different the following 3 dataset .arff file.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Execution time (Weka)</th>
<th>Execution time (Modified Apriori)</th>
<th>Minimum support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabetes.arff</td>
<td>1s</td>
<td>0.021s</td>
<td>0.1(77 instances)</td>
</tr>
<tr>
<td>Segment.arff</td>
<td>9s</td>
<td>0.126s</td>
<td>0.15(122 instances)</td>
</tr>
<tr>
<td>Ame3000.arff</td>
<td>16s</td>
<td>0.032s</td>
<td>0.1 (300 instances)</td>
</tr>
</tbody>
</table>

Table 4. Time comparison in 3 data set of Apriori and Modified Apriori

We have performed experiments with multilevel association rules. Our testing result shows in Apriori and Modified Apriori method which is executed the frequent itemsets and association rules with minimum support, confidence and lift. According the above result, both of two are mostly the same. But our proposed algorithm has fewer steps to find the frequent item sets, so computational complexity is less than Apriori algorithm.

4 Conclusion

The association rules play a major role in many data mining applications, trying to find increasing patterns in databases. In order to obtain the frequent itemsets, new algorithm must be generated. In this paper, Apriori algorithm is based on the properties of cutting database. The typical Apriori algorithm has performance bottleneck in the massive data processing so that we need to optimize the algorithm in variety of methods. We proposed the new method in this paper not only optimizes the algorithm of reducing the size of the candidate set of k itemsets, but also reduce the computational complexity. The performance of Apriori algorithm is optimized so that we can mine association information from massive data better and faster. So there should be some approach which has less number of scans of database.

5 References


