Research on Improved Apriori Algorithm Based on Simplified Boolean Matrix

Jingpeng Ruan [1], and Gang Fang [2]

[1] *College of Electronic & Information Engineering, Chongqing Three Gorges University, Chongqing, China*
[2] *College of Computer Science & Technology Chongqing Three Gorges University, Chongqing, China*

Abstract
The execution of the Apriori algorithm requires multiple scans of the database and produces a large number of unnecessary frequent itemsets. To improve the efficiency of the Apriori algorithm, an improved Apriori algorithm based on the simplified Boolean matrix is proposed in this paper. This work reduces the number of database scans by introducing a Boolean matrix, reduces the generation of frequent itemsets by simplifying the matrix, and introduces weight vectors to simplify the support of the Apriori algorithm. Experimental results show that the running time of this algorithm is significantly reduced compared with the apriori algorithm, which greatly improves the operational efficiency of the apriori algorithm.

Keywords
Apriori algorithm; Weight vector; Boolean matrix; Matrix simplification

1. Introduction

Because of its simple principle and easy implementation, Apriori algorithm is very suitable for database Association Rule Mining. However, the conventional Apriori algorithm will produce a large number of frequent itemsets in the operation process, which affects the efficiency of the execution process. In addition, the algorithm will scan the database many times in the execution process, which will also cause the low efficiency of the algorithm[1-3].

In order to solve these problems, researchers have proposed many schemes to improve the algorithm, such as a constrained association rule algorithm to reduce the generation of frequent itemsets by adding user interest items[4], and some researchers have proposed an incremental association rule algorithm to limit transaction cardinality[5]. However, the above improved Apriori algorithm does not have advantages in the evaluation of support degree[6].Therefore, some researchers proposed a matrix-based association rule generation algorithm, which can not only ensure the support of the algorithm, but also reduce the number of database scans[7, 8].

Boolean matrix is a matrix in which all elements are not "0" or "1". The Boolean matrix is introduced to process the dataset. By scanning the database once, the Boolean matrix corresponding to the database is generated, which reduces the scanning time of the database and improves the operational efficiency. In addition, on the basis of introducing Boolean matrix, the matrix is further simplified, and then the itemsets is generated from the simplified matrix, which can greatly reduce the generation of frequent itemsets and further improve the efficiency of operation[9].

In summary, an improved Apriori algorithm based on simplified Boolean matrix is proposed. The Boolean matrix is introduced, and the transaction is regarded as the row of the matrix, and the item is regarded as the column of the matrix. If the transaction contains the item, it is denoted as "1", otherwise, it is denoted as "0". In addition, the weight vector is introduced to simplify the calculation of support degree.And add a column at the end of the matrix, which is used to record the number of "1" occurrences in each row of the matrix. The next step is to simplify the matrix. When calculating the support degree in the previous step, if the support degree of a frequent itemset is not satisfied, the corresponding column

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China
EMAIL: ry99997@163.com (Jingpeng Ruan); 350874385@qq.com (Gang Fang);

of the item in the matrix can be directly deleted, so as to realize the simplification of the matrix,when generating frequent k-itemsets from frequent (k-1)-itemsets, If a line n<k,Then, the row can be directly deleted in the matrix, so as to simplify the matrix and reduce the generation of candidate itemsets. In addition,in the process of generating candidate k-itemsets from frequent (k-1) -itemsets,a new combination method is adopted to combine single items that do not repeat, such as $A_1A_2A_3$ and $A_1A_2A_4$,only one item, A_4, is different and can be combined. However, if $A_1A_2A_3$ and $A_1A_4A_5$, are different beyond a single project, they can be directly skipped without combining, which greatly saves the execution time of the algorithm. This method is used to generate candidate k-item itemsets, which further reduces the generation of candidate itemsets. This scheme not only reduces the number of database scans, but also reduces the generated frequent itemsets, simplifies the calculation of support degree, and then improves the efficiency of the Apriori algorithm.

2. Improvement principle of Apriori algorithm

2.1. Scheme of algorithm improvement

A Boolean matrix is a matrix whose entries take only "1" or "0", so it is also called a 0-1 matrix. We treat database transactions as the rows of the matrix, items as the columns of the matrix, and transactions that contain the item are denoted as “1”, and transactions that do not contain the item are denoted as “0”. Add a column “n” at the end of the matrix to record the number of "1" in each row. In addition, a $1\times p$ weight vector w= (1, 1, 1...) is introduced. To simplify the calculation of support, where p= the number of rows of the matrix. The multiplication of this vector with each column of the matrix results in the support number of the corresponding entries in that column. Support number and support degree can be converted into each other.

2.2. Algorithm execution steps

After mapping the original transaction database to the matrix, if there are repeated transactions, that is, there are identical rows in the matrix, the identical rows can be merged, and only one row of all the same rows is retained, and the merged row is added to the corresponding position of the row of the weight vector. For example, for a 5×5 matrix, its corresponding weight vector should be w= (1, 1, 1, 1). When row 1, row 3 and row 4 are exactly the same, these three rows can be merged, and the merged matrix will be a 3×5 matrix, and the corresponding weight vector will be w= (3, 1, 1).

Then multiply the weight vector with each column of the matrix to calculate the support degree of each item and compare it with the minimum support degree. If not, delete the column directly, and recalculate the number of "1" in each row of the matrix, and then update the last column “n”. The resulting matrix is the matrix corresponding to the frequent 1- itemset.

In order to generate frequent k-itemsets from frequent (k-1) -itemsets, it is necessary to simplify the matrix. In order to generate frequent k-itemsets, it is necessary to combine any two items in frequent (k-1) -itemsets. This operation in the matrix is the logical "and" operation on any two columns of the matrix. In this case, you can tell how many "1" there are in each row of the matrix based on the last column of the matrix "n". If n<k, it is easy to know according to the nature of Apriori algorithm, so this line can be deleted directly. Therefore, we obtain a method to simplify the matrix: when we generate the frequent k-itemset from the frequent (k-1) itemset, if in a row, the number of "1" n<k, the row can be directly deleted to simplify the matrix.

After deleting the nonconforming rows, multiply the weight vector with each column of the matrix, and calculate the support of each item again. If there are unqualified rows, delete the column directly, then recalculate the number of "1" in each row and update the last column n of the matrix, and judge whether n is less than k again. If n<k delete the row and repeat the above operation until each column meets the minimum support degree and n in each row is not less than k, at which point the matrix is simplified.

Logic "and" operation by the minimalist matrix composite matrix is generated by the k - frequent itemsets matrix, puts forward a new method of combined here, if you take any two column logic "and" the method of operation, will produce a lot of unnecessary combination, affect the efficiency of algorithm, the new method will be only a single item don't repeat the two columns of logic "and" operation, for example, $A_1A_2A_3$ and $A_1A_2A_4$ are different only in A_4 and can be combined. However, if there is more than one difference between $A_1A_2A_3$ and $A_1A_4A_5$, the combination can be directly skipped, which greatly saves the execution time of the algorithm. The resulting matrix is the frequent k-itemset matrix.

The above operation is repeated until no more frequent k-item sets can generate frequent (k+1) - itemsets, and the algorithm ends.

3. Experiment and simulation

3.1. Case Analysis

Table 1
Transaction information

orders	goods
1	ABCD
2	ABCE
3	BDEF
4	BCDE
5	ACDF
6	ABC
7	ABE

A shopping list of items is shown in Table 1, with a given minimum support of 0.3.We map Table 1 into Boolean matrix form and obtain the matrix M_1:

$$M_1=\begin{bmatrix} A & B & C & D & E & F & n \\ 1 & 1 & 1 & 1 & 0 & 0 & 4 \\ 1 & 1 & 1 & 0 & 1 & 0 & 4 \\ 0 & 1 & 0 & 1 & 1 & 1 & 4 \\ 0 & 1 & 1 & 1 & 1 & 0 & 4 \\ 1 & 0 & 1 & 1 & 0 & 1 & 4 \\ 1 & 1 & 1 & 0 & 0 & 0 & 3 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

If you buy this item, call it “1”;No purchase of the item, denoted as “0”.

The rows of the matrix represent transactions, and the columns of the matrix represent items.

At this time, the weight row vector w= (1, 1, 1, 1, 1, 1) is introduced and multiplied with each column of the matrix to obtain the support degree of each item. At the end of the matrix, a column “n” is added to represent the number of items in each transaction, that is, the number of "1" in each row.

And then we can simplify the matrix. According to the calculation of support degree, $w\times F=2<7\times 0.3=2.1$, does not meet the minimum support, so direct pruning, that is, delete the column in the matrix, to obtain a new matrix M_2:

$$M_2=\begin{bmatrix} A & B & C & D & E & n \\ 1 & 1 & 1 & 1 & 0 & 4 \\ 1 & 1 & 1 & 0 & 1 & 4 \\ 0 & 1 & 0 & 1 & 1 & 3 \\ 0 & 1 & 1 & 1 & 1 & 4 \\ 1 & 0 & 1 & 1 & 0 & 3 \\ 1 & 1 & 1 & 0 & 0 & 3 \\ 1 & 1 & 0 & 0 & 1 & 3 \end{bmatrix} \quad (2)$$

This matrix is the matrix corresponding to the frequent 1- itemset. Next, perform logical "and" operation on any two columns of the matrix to generate the matrix of frequent 2-itemset, as shown in the matrix M_3:

$$M_3=\begin{bmatrix} AB & AC & AD & AE & BC & BD & BE & CD & CE & DE & n \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 6 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 6 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 3 \end{bmatrix} \quad (3)$$

According to the support degree calculation, w×AD, w×AE, w×CE, w×DE all less than 2.1.That is, AD,AE,CE, and DE do not meet the minimum support, so all the branches are pruned. The simplified matrix is shown in M_4:

$$M_4=\begin{bmatrix} AB & AC & BC & BD & BE & CD & n \\ 1 & 1 & 1 & 1 & 0 & 1 & 5 \\ 1 & 1 & 1 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 & 1 & 4 \\ 0 & 1 & 0 & 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 0 & 0 & 0 & 3 \\ 1 & 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} \quad (4)$$

The matrix is a frequent 2- itemset matrix.

Next, the frequent 3- itemset is generated by the frequent 2- itemset matrix.

In this step, we can apply the pruning property of Apriori algorithm in advance to further simplify the matrix. For example, the combination of AB and BD will generate ABD, but in the previous step, we have already known that AD is infrequent, and ABD contains AD. According to the property of Apriori algorithm, ABD must be infrequent.In the process of generation, we can directly calculate the support degree without performing logical "and" operation on the two columns after generation.

The frequent 3-itemset we want to generate requires that each row must have at least three items. If there are less than three items, the final result will be infrequent no matter how the logical "and" operation is carried out. For example, the candidate 3-item set is generated from the frequent 2-item set, and A row only contains AB and AC, $n=2 < 3$. At this time, it can be known that in the initial matrix, the row must contain three items, A, B and C, and there are only two cases of BC. The BC is deleted or marked as "1" in the row because the support degree is not satisfied. There is no BC in the row marked as "0". The row contains only AB and AC, and the BC column is deleted because it does not meet the support. BC is a subset of ABC, while ABC is the item generated by the logical "and" operation between AB and AC. According to the nature of Apriori algorithm, ABC will not meet the support degree.

Therefore, when the number of entries is less than 3, the row can be directly deleted to simplify the matrix.The simplified matrix is shown in M_5:

$$M_5=\begin{bmatrix} AB & AC & BC & BD & BE & CD & n \\ 1 & 1 & 1 & 1 & 0 & 1 & 5 \\ 1 & 1 & 1 & 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 1 & 1 & 1 & 4 \\ 1 & 1 & 1 & 0 & 0 & 0 & 3 \end{bmatrix} \tag{5}$$

In this matrix, the support degrees of BD,BE and CD are all 2<2.1, so it can be deleted directly.The further simplified matrix obtained is shown in M_6:

$$M_6=\begin{bmatrix} AB & AC & BC & n \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 3 \end{bmatrix} \tag{6}$$

The number of entries in the third row is less than 3, and the simplified matrix obtained by rounding up again is shown in M_7:

$$M_7=\begin{bmatrix} AB & AC & BC & n \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \end{bmatrix} \tag{7}$$

The three rows are exactly the same, and the matrix obtained by merging is shown in M_8. This matrix cannot be reduced any more, so the matrix has been reduced to the simplest.

$$M_8=\begin{bmatrix} AB & AC & BC & n \\ 1 & 1 & 1 & 3 \end{bmatrix} \tag{8}$$

In this case, the weight vector changes from w= (1,1,1) to w= (3).

Finally, the candidate 3-item set matrix generated by logical "and" operation of this simplest matrix is shown in M_9:

$$M_9=\begin{bmatrix} ABC & n \\ 1 & 1 \end{bmatrix} \tag{9}$$

Calculated from the support degree, w×ABC=3>2.1. Therefore, the frequent 3-itemset is the frequent 3-itemset. However, the number of items in this matrix is only 1, and the frequent 4-itemset cannot be generated from the frequent 3-itemset, so the final frequent 3-itemset is {ABC}, and the algorithm ends.

3.2. Experimental Verification

In order to verify that the improved algorithm really improves the time performance of Apriori algorithm, movielens dataset is selected, different support degrees are set, and Python language is used for implementation. The line chart of the experimental results is shown in Figure 1:

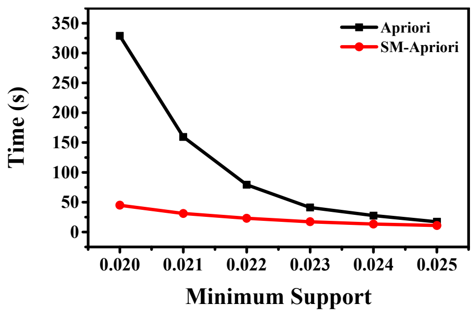

Figure 1 Time comparison of the two algorithms under different support degrees

According to the simulation results, it can be seen that when the minimum support degree increases, that is, when the item is more frequent and the correlation is stronger, the time of the two algorithms begin to approach. However, when the value of the minimum support is small, that is, the correlation is weak, the time required by the improved algorithm is much lower than that of the classical Apriori algorithm. Therefore, the improvement can effectively improve the efficiency of the algorithm.

4. Conclusion

Aiming at the problem that the classical Apriori algorithm overscans and produces a large number of unnecessary candidate item sets, an improved Apriori algorithm based on reduced Boolean matrix is proposed, and the weight vector is introduced to simplify the calculation of support degree. The improved algorithm can only scan the database once to form the corresponding Boolean matrix, and the following steps of the algorithm are carried out in this matrix. At the same time, a new method to generate frequent k-itemsets from frequent (k-1)-itemsets is also proposed. The experimental comparison shows that the improved algorithm can significantly reduce the scanning time and time required by the algorithm on the premise of ensuring the correct results, so as to improve the efficiency of data mining.By introducing mathematical knowledge such as matrix and vector to improve algorithm efficiency, it is expected to become a conventional method to improve algorithm efficiency in the future.

5. References

[1] action dataset, 2013. URL: http: //www.iesl.cs.umass.edu/data/data-umasscitationfield.
[2] The Operation Optimization of Small Index Method Based on Improved Apriori Algorithm [J]. Mechatronics, 2017.
[3] CHEN W, ZHANG X, MANAGEMENT D O. Excavation of key risk factors based on Apriori algorithm for fire and explosion in storage and transportation of hazardous chemicals [J]. Fire Safety Science, 2017.
[4] CHENG Z. Application of Data Mining in Power Network Safety Evaluation [J]. Electrical Engineering, 2010.
[5] CHENGYU C, YING X. Research and Improvement of Apriori Algorithm for Association Rules [J]. Physical Review A, 2016: 1-4.
[6] GONG Q, LIU X, ZENG Y, et al. An energy efficiency solution based on time series data mining algorithm on elementary school building [J]. International Journal of Low-Carbon Technologies, 2022.
[7] LIN W, WU Z, ZHOU H, et al. Research on mining association between weather and power grid based on Apriori algorithm [J]. Electrical Engineering, 2019.
[8] PRASAD G D, HANMANDLU M, SAHA T K. Real-Time Decoupled Equivalent Models for Static Security Analysis [J]. IEEE Transactions on Power Systems, 1.
[9] QIAO H. Research On QAR Data Mining Method Based On Improved Association Rule [J]. Physics Procedia, 2012.
[10] ZHU W, GUO Q. Study on Association Rules Mining Algorithm for Micro-grid Fault Detection [J]. Electrical Engineering, 2015.