A Data Mining Algorithm for Association Rules with Chronic Disease Constraints

YanRong Liu, LiJun Wang, Rong Miao, and HengNi Ren

College of Information Engineering, Shaanxi Institute of International Trade & Commerce, Xi’an 712046, China

Correspondence should be addressed to YanRong Liu; 20051022@csiec.edu.cn

Received 29 June 2022; Accepted 23 July 2022; Published 23 August 2022

Academic Editor: D. Plewczynski

Copyright © 2022 YanRong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Apriori algorithm in association rules is the main algorithm used in the treatment and prevention of chronic diseases in data mining, and the algorithm in the current stage of China’s medical field of association between chronic diseases has some problems, such as the need to scan the transaction database of cases several times, producing a large data set and more redundant rules. To address the above problems, a data mining algorithm of association rules combining clustering matrix and pruning strategy is proposed, which improves the algorithm by using the clustering matrix method to compress the stored transaction database and introducing the prepruning and postpruning strategy methods on the basis of adding constraint conditions. The experimental results show that the optimization algorithm has unique advantages in reducing the number of database scans and the number of candidate item sets generated and ultimately greatly reduces the running time and I/O load of the algorithm, and the running efficiency of the algorithm is greatly improved.

1. Introduction

With the development of digital technology today, behind it is the support of powerful computer technology. Powerful computer technology must rely on massive databases. Computer database application technology has penetrated many aspects of people’s lives, such as online shopping, smart home appliances, pathological analysis, prevention, etc. Extensive application is bound to produce a large amount of data. How to quickly find potentially useful information in this massive data has been a major problem that people are facing. Data mining technology can help people discover the knowledge hidden behind massive data. Driven by various medical and health modes, data mining technology has shown good application value and attracted more and more experts and scholars’ attention. The improvement of information technology has also brought about the wide application of various medical systems. In the medical field, big data has also begun to be applied, and the data collected in various medical systems are of great research value for medical research and the diagnosis and treatment of chronic diseases. However, at present, all hospitals only collect and store patients’ medical data without in-depth analysis and effective use, let alone using these data to analyze and mine the knowledge behind them. Using data mining technology to discover the hidden relationship from these medical data and provide services for doctors’ clinical diagnoses has become a problem that researchers pay much more attention to.

In medicine, there is a lot of unknown information hidden behind the sign data [1] of patients with chronic diseases, which can provide a correct prediction for early diagnosis through data mining. Therefore, data mining technology has become an indispensable part of the hospital evaluation system and medical research. Methods of data mining techniques such as association Rules Mining, rough set theory [2], machine learning, and neural network [3] have their own advantages and disadvantages in medical data mining [4]. Among the many application fields of data mining, a promising research direction is the data mining algorithm of association rules which has been widely applied to e-commerce, prevention of chronic diseases [5], telecommunications industry, and insurance industry. Whether the association rule algorithm is good or bad in practice is
shown by the final result of any system. Therefore, the study of the association rule algorithm is to study whether the association rules generated by it have greater relevance and whether they can help human intelligent life.

In the medical data analysis of chronic diseases, many scholars have applied the data mining technology of association rules to it and have achieved remarkable experimental results. Xu and Xu [6] used the characteristics of a numerical attribute of the aggregation algorithm to discrete the data set and divided it into several optimized data sets, mining useful association rules in tumor diagnosis data and providing important reference value for clinical diagnosis of chronic cancer. Serbanati [7] established a prediction model for chronic disease patients using labor neural network and Logistic regression analysis technology, which provided a good prediction effect for chronic diseases such as hypertension and diabetes. Zhang et al. [8] used the classification algorithm to analyze the data of type I diabetes patients and found that the generated classification association rules were highly consistent with the results of medical research. The classification association rule technique has a good theoretical basis in the study of chronic diabetes. Zhou et al. [9] summarized the traditional Apriori algorithm and found that its biggest defect was that frequent item sets could be obtained only after repeated scanning of the database, which would inevitably affect the efficiency of data mining and occupy a large amount of memory. Wen et al. [10] put forward a kind of an association rule data mining algorithm combining matrix and index sort is proposed, the algorithm can delete useless transaction data and item sets in a transaction database, then the set of a frequent binomial is obtained by the product of matrix. Finally, the frequent item sets are obtained, the algorithm can directly find the frequent item sets are the biggest advantage of scanning the database. Although the efficiency of data mining has been improved, the database needs to be updated continuously during the mining process, which increases the time overhead of I/O. To sum up, although the theoretical research on the mining technology of association rules has been applied in many aspects, there is still a problem that effective association rules can not be quickly obtained during mining, which will have a great impact on the accuracy of the system association. Therefore, the research data mining algorithm of association rules is of great significance to people’s convenient life.

2. Association Rule

2.1. Knowledge about Association Rules. The relevant definitions are as follows.

**Definition 1.** Suppose there is a set of items as follows: $A = \{a_1, a_2, a_3, \ldots, a_n \}$, $D$ is the transaction database collection. Both $M$ and $N$ are nonempty subsets and both belong to $A$, and $M \rightarrow N$ is an expression in the form of an association rule. In the expression, $M \subseteq A$, $N \subseteq A$, and the intersection of $M$ and $N$ is not equal to the empty set. If $M$ contains $k$ number of sets, it is called $k$-item sets.

**Definition 2** Suppose $(M \rightarrow N)$ is the ratio of the number of tuples containing $M$ to the number of tuples containing $N$, and the result is expressed as a percentage [11], as shown in the following formula:

$$S(M \rightarrow N) = \frac{\text{count}(M \cap N)}{\text{count}(D)} \times 100\%.$$  

**Definition 3** (confident). refers to the percentage that the database contains transactions $N$ even if it contains one transaction $M$, namely, the trustworthiness of the value [11]. The definition pattern of (confident) $(M \rightarrow N)$ is (confident) $(M \rightarrow N)$ represents the percentage of tuples containing $M$ and $N$ in tuple $M$, as shown in the following formula:

$$C(M \rightarrow N) = \frac{S(M \cap N)}{S(N)} \times 100\%.$$  

**Definition 4.** Let $\minsup$ be the minimum support threshold [12] and $\minconf$ the minimum confidence threshold [13]. Assuming that there is an association rule in $D$ that satisfies formulas (3) and (4), it is a strong association rule in $D$ [14, 15]. The set that satisfies formula (3) is called frequent item sets, and the set that does not satisfy formula (3) is called infrequent item sets [16].

$$S(M \rightarrow N) \geq \minsup,$$  

$$C(M \rightarrow N) \geq \minconf.$$  

2.2. Apriori Algorithm. Apriori algorithm is one of the most widely used algorithms in association rule data mining. Such as medical system, commercial system, and the association rules generated by this algorithm are single-level, Boolean type [17] association rules in terms of classification. Apriori association rules algorithm scans the transaction database by a layer-by-layer iterative method and then obtains frequent item sets. The main idea of generating frequent item sets is as follows [18, 19]: first, scan the transaction database multiple times to generate candidate item sets of length $k + 1$ from $k$ frequent item sets; then, perform mining and judgment on $k + 1$ candidate item sets to obtain $k + 1$ Frequent item sets. According to the same method, until no more $k$-item sets are found, the algorithm execution ends. Table 1 is the definition symbols in the algorithm.

The first stage of Apriori algorithm execution is to scan the transaction database to find out the support of each element in the candidate item sets $C_k$ to determine whether the element joins $L_k$ or not. If the generated frequent item set is a large item set containing 12 items, the minimum number of times to scan the transaction database is 12. When the larger the transaction database is, the larger the I/O load is; in the second stage of algorithm execution, the frequent item sets $L_{k-1}$ generating the candidate item sets $C_k$ is exponentially growing, and if the length of the frequent pattern is 200, the minimum number of postoption
item set is $2^{100}$. When the candidate 2-item sets are larger, the final result is very considerable. When there is a large amount of data in the transaction database, the min\textunderscore supp threshold and the min\textunderscore conf threshold are set too small, and a large number of redundant rules will be generated [20], and with the increase of data in the transaction database, the generated association rules and he numbers grow exponentially. In order to improve the execution efficiency of the algorithm, this paper proposes an algorithm that combines a clustering matrix and pruning strategy under the condition of setting project constraints. The algorithm is optimized on the basis of the traditional Apriori algorithm, which will greatly improve the execution efficiency of the algorithm.

3. A Data Mining Algorithm with Item Constraints

The knowledge discovery of traditional association rule algorithm is achieved by setting min\textunderscore supp and min\textunderscore conf. A single setting of min\textunderscore supp and min\textunderscore conf results in more association rules, and a large part of them are redundant rules, and fewer rules meet the conditions. If the user can guide and control the mining process, the number of valid association rules can be increased, and the efficiency of mining can be ameliorated. Setting appropriate item constraints in the mining process can achieve better results and thus obtain effective association rules. In the research of this paper, first stage, preconstraints (hypertension, diabetes, etc.) are added to the database before generating frequent item sets to constrain the database; second stage, fundamentally reduce the frequency of database scans; the association rules that do not meet the set conditions are dropped. The above idea of applying preconstraints and post-constraints is applied to the generated Apriori algorithm, and Figure 1 shows the detailed execution flow of the algorithm.

3.1. The Main Idea of an Apriori Algorithm with Item Constraints. In the analysis of the previous section, the existing problems of the traditional Apriori have been clarified. The research in this paper is to add constraints on the basis of the traditional algorithm, use the cluster matrix compression storage method to store the transaction database, and obtain the candidate items by operating the matrix vector. Then, before the frequent item sets are self-connected, the pruning method is used to delete the item sets that do not meet the conditions. At this time, the redundant candidate item sets generated by the pruning strategy will be greatly reduced, thereby improving the data mining efficiency.

3.1.1. Matrix Vector Strategy. The following definitions are made in the transaction database $D$.

**Definition 5.** The vector of $nj$ is denoted as $D_j$, As shown in the following formula:

$$D_j = \begin{bmatrix} d_{1j} \\ d_{2j} \\ \vdots \\ d_{dj} \end{bmatrix},$$

where $d_{nj} = \begin{cases} 0, n j \notin D, \\ 1, n j \in D. \end{cases}$

The support frequency of $nj$ is as follows:

$$\text{support}_c(nj) = \sum_{m=1}^{m} d_{nj}.$$

**Definition 6.** The vector of the 2-term set $\{i, i_j\}$ is denoted as $D_{ij}$:

$$D_{ij} = \begin{bmatrix} d_{ij} \land d_{1j} \\ d_{ij} \land d_{2j} \\ \vdots \\ d_{ij} \land d_{nj} \end{bmatrix},$$

where, the symbol “$\land$” indicates the “AND” operation.

The support frequency of the 2-item set is as follows:

$$\text{support}_c(\{i, i_j\}) = \sum_{r=1}^{m} (d_{ij} \land d_{1j}).$$

**Definition 7.** The vector of the $k$-term set is denoted as $D_{1 \ldots k}$.

The support frequency of the $k$-term set is as follows:

$$\% \text{support}_c(\{i_1, i_2, \ldots, i_k\}) = \frac{\sum_{r=1}^{n} (d_{i_1} \land d_{i_2} \land \ldots \land d_{i(k-1)} \land d_{i_k})}{n}.$$

According to the above definition, the matrix vector strategy execution process is as follows.

1. In this paper, the strategy of clustering matrix is applied to the Apriori algorithm. First, the transaction database $D$ is scanned and a clustering matrix $A$ is generated for it. If this number of items in this transaction database is $k$, then it is clustered into the matrix $A(k)$. Each column of the clustering matrix $A(k)$ represents a different item in the column, and each row represents a detailed convert record containing item sets, and the Boolean quantities 0 and 1 in the matrix indicate whether the item has a record in the matrix or not, respectively, with a record of 1 and no record of 0. The column vectors in the matrix are calculated using the logical operation “AND.”

2. $L_{k-1}$ ($k \geq 2$) self-join produces the set of the candidate item set $C_k$. 

### Table 1: Define symbols.

<table>
<thead>
<tr>
<th>$k$-item</th>
<th>The set of $k$ items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_k$</td>
<td>min\textunderscore supp item Sets</td>
</tr>
<tr>
<td>$C_k$</td>
<td>Candidate item sets</td>
</tr>
</tbody>
</table>
(3) Sum the column vectors of the cluster matrix using the “AND” operation. If the support obtained after the “AND” operation of the cluster matrix meets the set threshold value, the item set is directly put into the set \( L_k \) of frequent-item set \( k \)-item. Otherwise, the corresponding column vector “AND” operation is performed on the cluster matrix \( A(k + 1) \), and the result is accumulated with the previous support frequency. When the set \( \text{min supp} \) threshold is not greater than the accumulated support, the subsequent clustering matrix continues to be scanned until the accumulated support is greater than the set \( \text{min supp} \) threshold, and the subsequent matrix ends scanning.

3.1.2. Pruning Strategy. The Apriori algorithm uses \( L_{k-1} \) self-join to generate a set \( C_k \) of candidate item sets. In the course of self-join, a large number of infrequent item sets are inevitably generated, and if the algorithm can determine that some items are infrequent item sets according to some rules before generation of \( C_k \), it can cut out this part of item sets in advance, which saves their self-join operation and reduces the time to calculate their support frequency. When scanning the database, the pruning strategy reduces the number of frequent item sets generated and reduces the time complexity of the algorithm execution. The pruning strategy is suitable for frequent item sets with the following properties.

**Property 1.** If all subsets contained in \( A \) set are frequent item sets, then \( A \) set must be frequent item sets.

**Property 2.** Among \( k \)-itemset \( A = \{a_1, a_2, \ldots, a_k\} \), if there are items \( i \in A \) and \( |L_{k-1}(i)| < i - 1 \), then \( |L_{k-1}(i)| \) represents the number of items \( i \) contained in the set \( L_{k-1} \).

Based on the above properties, the pruning strategy implementation process is as follows:

1. Calculate \( |L_{k-1}(i)| \)
2. Calculate the frequencies of all items and record the items with frequencies less than \( k - 1 \), noted as \( A = \{i \mid |L_{k-1}(i)| < i - 1\} \)
3. Delete all frequent item sets containing \( A \) any of the elements in \( L_{k-1} \) and record as \( L_{k-1}' \)
4. Candidate set \( C_k' \) is obtained by self-joining of set \( L_{k-1}' \)

During the implementation of the pruning strategy, the pruning strategy with the Apriori algorithm can be used after the self-join \( L_{k-1}' \) to further reduce the number of candidate item sets generated again. The time required for frequent item sets when joined is reduced by the prepruning strategy, and the workload is also reduced for the implementation of postpruning. After implementing the prepruning and postpruning strategies, candidate item sets are reduced. This to some extent reduces the time overhead during the database scanning process when calculating the support.

3.2. Algorithm Flow and Examples

3.2.1. Execution Process. The algorithm proposed in this paper adds project constraints, matrix-vector strategy, and pruning strategy to the traditional Apriori algorithm. The specific implementation process is as follows:
Step 1: by scanning the transaction database $D$, the number of items in $D$ is obtained, and the matrix is clustered to obtain the $A$ matrix. If the matrix contains $k$ items, the clustering matrix $A(k)$ is obtained.

Step 2: add and sum the vector values of each column in the matrix, calculate the support frequency of the items, and add the items that meet the minimum support threshold value to $L_1$ and produce a frequent 1-item set.

Step 3: perform operations on $L_{k-1}$ with the help of the pre-pruning strategy, record the number of occurrences of the obtained items in the itemset, delete all infrequent item sets whose support frequency is less than the item $k-1$, and record the obtained set as $L_{k-1}$.

Step 4: a self-connection operation is performed on $L_{k-1}$, and the connection conditions can be appropriately added to the set constraint items (hypertension, diabetes, etc.), and a candidate $k$-item set $C_k^1$ is generated at this time.

Step 5: after pruning $C_k^1$ with the properties of the traditional Apriori algorithm to generate $C_k$, the irrelevant items are reduced.

Step 6: the length of each item set in the generated itemset $C_k$ is recorded as $k$, so there is no need to consider the clustering matrix whose length is smaller than $k$. The column vectors in the clustering matrix $A(k)$ are subjected to “AND” operation to obtain the corresponding column vectors for each candidate item set, and the sum of the column elements in the clustering matrix is counted to support the calculation of the corresponding candidate item set. If the minimum support threshold value is less than the support of the candidate items, the set will be added to $L_k$. If the minimum support threshold value is greater than the support of the candidate items, the column vectors in the $A(k+1)$ will be processed to find the support frequency of the item set and added to the count until the support is not less than the set minimum support threshold value, or all the items in the cluster matrix are scanned.

Step 7: repeat execution of step 3 to step 6 until $L_k$ is empty to determine the frequent item set $C_k$.

3.2.2. Algorithm Implementation Process. In Table 2, database $D$ exists and $\text{min \_supp} = 2$. The generation process of frequent item sets is as follows.

1. Scan the transaction database and generate three clustering matrices based on the number of items contained, denoted by $A(1)$, $A(2)$, and $A(3)$, respectively, and the matrix for each item expressed as a Boolean value is as follows:

$$A(1) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$A(2) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$A(3) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The support of the item $a_1$ is as follows:

$$\text{support}_c(a_1) = \sum_{n=1}^{5} d_{2n1} = 2 \geq 2$$

where the $d_{2nj}$ denotes the Boolean value of the $k$ th row and $j$ th column of the matrix $A_4$. Similarly, the support of $a_2$ and $a_3$ can be calculated as 3 and 4, and both are greater than the minimum support threshold value. The support of the item $a_4$ is as follows:

$$\text{support}_c(a_4) = \sum_{n=1}^{5} d_{2n4} = 1 \leq 2$$

According to the same method, the support of $a_5$ can be found as 2. The final 1-item set $L_1$ is obtained as follows: $\{a_1\}$, $\{a_2\}$, $\{a_3\}$, $\{a_4\}$ and $\{a_5\}$.

2. After self-join, $C_k$ are as follows: $\{a_1,a_2\}$, $\{a_1,a_3\}$, $\{a_1,a_4\}$, $\{a_1,a_5\}$, $\{a_2,a_3\}$, $\{a_2,a_4\}$, $\{a_2,a_5\}$, $\{a_3,a_4\}$, $\{a_3,a_5\}$ and $\{a_4,a_5\}$.

$$\text{support}_c(\{a_1,a_2\}) = \sum_{n=1}^{m} (d_{2n1} \land d_{2n2}) + \sum_{n=1}^{m} (d_{3n1} \land d_{3n4}) + \sum_{n=1}^{m} (d_{4n1} \land d_{4n3}) = 0 + 1 + 0 = 1 \leq 2,$$

$$\text{support}_c(\{a_1,a_3\}) = \sum_{n=1}^{m} (d_{2n1} \land d_{2n3}) + \sum_{n=1}^{m} (d_{3n1} \land d_{3n3}) + \sum_{n=1}^{m} (d_{4n1} \land d_{4n4}) = 0 + 0 + 0 = 0 \leq 2,$$

$$\text{support}_c(\{a_1,a_5\}) = \sum_{n=1}^{m} (d_{2n1} \land d_{2n5}) + \sum_{n=1}^{m} (d_{3n1} \land d_{3n5}) + \sum_{n=1}^{m} (d_{4n1} \land d_{4n5}) = 0 + 1 + 0 = 1 \leq 2,$$

$$\text{support}_c(\{a_2,a_3\}) = \sum_{n=1}^{m} (d_{2n2} \land d_{2n3}) + \sum_{n=1}^{m} (d_{3n2} \land d_{3n3}) + \sum_{n=1}^{m} (d_{4n2} \land d_{4n4}) = 0 + 1 + 0 = 1 \leq 2,$$

And you can see that in the calculation, $\{a_3,a_5\}$ and $\{a_4,a_5\}$ are not satisfied with the set threshold value.

The support of the candidate item sets $\{a_1,a_2\}$, $\{a_1,a_3\}$, $\{a_2,a_3\}$, $\{a_2,a_4\}$, $\{a_2,a_5\}$ and $\{a_1,a_5\}$ are
both greater than 2, satisfying the set condition. Thus, the generated frequent 2-item set \( L_2 \) is obtained as follows: \([a_1, a_2], [a_1, a_3], [a_2, a_3], [a_2, a_4], [a_1, a_3] \) and \([a_2, a_5] \).

(3) Use the prepruning strategy, the number of each item in \( L_3 \) is calculated as follows: \(|L_2(a_1)| = 3, |L_2(a_2)| = 4, |L_2(a_3)| = 2, |L_2(a_4)| = 1\) and \(|L_2(a_5)| = 3\). In the frequent 2-item set, the occurrences number of \( a_i \) is not greater than 2, so \( a_3 \) will be deleted and \( L_3' \) are obtained as follows: \([a_1, a_2], [a_1, a_3], [a_1, a_4] \) and \([a_2, a_3] \). Perform self-join for \( L_3' \), the candidate item set \( C_3 \) are obtained as follows: \([a_1, a_2, a_3], [a_1, a_2, a_4], [a_1, a_3, a_3], [a_2, a_3, a_3] \) and \([a_2, a_3, a_3] \) if the prepruning strategy is not used, the candidate 3-item set is obtained as follows: \([a_1, a_2, a_3], [a_1, a_2, a_4], [a_1, a_3, a_3], [a_1, a_3, a_4], [a_2, a_3, a_3], [a_2, a_4, a_5] \). According to the above calculation, 6 candidate item sets are generated before using the former pruning strategy, and 4 candidate 3-item sets are obtained after using the former pruning strategy. In a database with only 9 items, 2 candidate item sets are reduced by using the one-time pruning strategy. When the transaction database is larger, the performance of using the pruning strategy is more obvious strategy. Perform postpruning for the candidate 3-item set, then the candidate item set \( C_3' \) is obtained as follows: \([a_1, a_2, a_3] \) and \([a_1, a_2, a_5] \).

(4) Calculate the support of the candidate item set \( C_3 \) because the number of transactions of the candidate item set is 3, so the frequent item set only needs to be calculated from the clustering matrix \( A(3) \). Because the support of \([a_1, a_2, a_3] \) is 2 and the support of \([a_1, a_3, a_3] \) is 2, both satisfy the set minimum support threshold value, so \( L_3 \) is as follows: \([a_1, a_2, a_3] \) and \([a_1, a_2, a_3] \).

(5) Perform prepruning strategy for \( L_3 \), \( L_3 \) is empty and the algorithm ends. The set of all frequent item sets is as follows: \([a_1, a_2, a_3] \) and \([a_1, a_2, a_3] \).

It can be seen from the above example of algorithm execution that when the traditional Apriori algorithm generates a frequent item set \( L_3 \), it needs to scan the transaction database three times, and the algorithm, after the pruning strategy, only scans the database once. From the perspective of time consumption, it takes less time, and from the perspective of data mining efficiency, the efficiency is improved.

### 4. Experiments

From the content of the previous section, it can be seen that the algorithm has good advancements, and the advantages of the algorithm will be verified below. The environment of this experiment uses a Pentium(R) 2.40 GHz/3.0 GB microcomputer (operating system is Win 7), the simulation environment uses MATLABR2012a, and the data set of the experiment uses Mushroom of the UCI standard test data set, which contains 8, 124 records, each record has 23 attributes, and each attribute has 12 enumeration values.

<table>
<thead>
<tr>
<th>Item</th>
<th>Item sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>(a_1, a_2, a_5)</td>
</tr>
<tr>
<td>A2</td>
<td>(a_2, a_5)</td>
</tr>
<tr>
<td>A3</td>
<td>(a_3, a_2)</td>
</tr>
<tr>
<td>A4</td>
<td>(a_1, a_2, a_4)</td>
</tr>
<tr>
<td>A5</td>
<td>(a_1, a_3)</td>
</tr>
<tr>
<td>A6</td>
<td>(a_2, a_5)</td>
</tr>
<tr>
<td>A7</td>
<td>(a_3, a_3)</td>
</tr>
<tr>
<td>A8</td>
<td>(a_1, a_3, a_3, a_5)</td>
</tr>
<tr>
<td>A9</td>
<td>(a_1, a_2, a_3)</td>
</tr>
</tbody>
</table>

(1) The experiments of setting the two algorithms are carried out under the premise of the same data set and different support thresholds, and the number of generated candidate item sets is shown in Figure 2. Observing Figure 2, we can see that when the \( \text{min_supp} \) threshold increases from 10 to 30, the number of candidate item sets for the optimization algorithm is reduced from 670 to 256, and the number of candidate item sets for the traditional Apriori algorithm is reduced from 820 to 385. The number of sets is decreasing. However, under the same \( \text{min_supp} \), the number generated by the optimization algorithm is significantly lower than that of the traditional Apriori algorithm. Therefore, the optimization algorithm has obvious advantages.

(2) The experiment of setting the two algorithms is carried out under the premise of the same data set and different support thresholds. The execution time of the two algorithms is shown in Figure 3. Observing Figure 3, we can see that the \( \text{min_supp} \) threshold increased from 10 to 40, the execution time of the optimized algorithm decreased from 112 to 40, and the execution time of the traditional algorithm decreased from 175 to 75. The execution time of both algorithms is decreasing. However, under the same \( \text{min_supp} \), the execution time of the optimization
algorithm is significantly lower than that of the traditional Apriori algorithm, so the optimization algorithm has obvious advantages.

(3) Under the premise of \( \min \text{ supp} = 0.3 \) and different numbers of datasets, the execution time of the two algorithms is shown in Figure 4.

Observing Figure 4, we can see that the number of datasets increased from 2000 to 8000, the execution time of the optimized algorithm increased from 25 to 55, the execution time of the traditional algorithm increased from 32 to 81, and the execution time of both algorithms increased. However, under the same dataset, the execution time of the optimization algorithm is significantly lower than that of the traditional Apriori algorithm, so the optimization algorithm has obvious advantages.

5. Conclusions

This paper proposes a project under the constraint condition of clustering matrix and the pruning strategy of combining the association rule data mining algorithm, the algorithm is applied in setting project constraints under the premise of clustering matrix and the method of pruning strategy largely reduced the number of algorithms to generate candidate item sets, avoid the I/O overhead of the problems of multiple scanning database, and improve the execution efficiency of the algorithm. The simulation experiment is verified from three aspects. It can be seen from the experimental results that the number of candidate sets generated by the optimized Apriori algorithm under different support threshold conditions is less than that of the traditional algorithm, and the execution time of the algorithm is greatly reduced, thereby improving the execution efficiency of the algorithm. At present, the association rules generated by the algorithm in this paper are given in a formal formula, which is inconvenient for users to understand. In future work, the visualization of the data mining process will be studied.

Data Availability

The data used in the experiment are obtained from the UCI standard test dataset of the Mushroom data.

Conflicts of Interest

YanRong Liu optimized the algorithm in this paper and compared it with the original algorithm to verify the characteristics of the optimized algorithm. Lilun Wang And Rong Miu cleaned the data, obtained the valid data and trained the data model. HengNi Ren mainly completed the collation of experimental data and the examination of the first draft of the paper. All the authors have read the manuscript, agreed to its publication. The authors declare that there are no conflicts of interest.

Acknowledgments

This research was funded by Key Research Project of Shaanxi Province, grant number 2022NY-207; Xianyang Science and Technology Bureau Project, grant number 2021ADYF-SF-0028; and Special Project of Shaanxi Institute of International Trade & Commerce School, grant number SMZX202145.

References

8 Computational Intelligence and Neuroscience


