Research Article

Designing the Education Resource Management System Using Apriori Algorithm

Xinlei Yuan
School of Education Science, Mudanjiang Normal University, Mudanjiang 157011, China

Correspondence should be addressed to Xinlei Yuan; yuanxinlei@mdjnu.edu.cn

Received 17 February 2022; Revised 20 March 2022; Accepted 24 March 2022; Published 8 April 2022

Academic Editor: Mian Ahmad Jan

Copyright © 2022 Xinlei Yuan. 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.

Standardization of educational resources and advanced use of appropriate computer knowledge to construct an efficient educational management system is integral for the modern system of education in the age of digitalized education. In the context of this need, this study is aimed at analyzing the Apriori algorithm of education resource management system and its implementation. The Apriori algorithm is analyzed and studied in this paper. Firstly, the background and function of the Apriori algorithm are summarized. Secondly, the content of the Apriori algorithm is introduced in detail. The educational resources managed by this system all follow the Apriori algorithm. It is the most widely used distance education standard at present. In addition, the idea and method of making educational resources in the specification are also expounded. The educational resources made by using this method have higher sharing and reuse. The article analyzes the present situation of the education resource management system, for the insufficiency in its function. The solution for different shortcomings has been dealt with one by one and presented as a grid with an Apriori algorithm for technical support. Moreover, various heterogeneous education resources have been integrated to a unified standardization conforming to the Apriori algorithm. It has been scientifically designed to provide an application platform that can store and manage educational resources efficiently and query educational resources conveniently. The universities can not only effectively manage their own educational resources but also use the resources of other schools, so as to avoid the repeated construction of educational resources. This enables them to achieve the whole Apriori within the scope of resource sharing and reuse. On this basis, this paper gives the system architecture diagram and topological structure diagram of educational resource management system on the Apriori algorithm based, puts forward the design scheme of the system, and describes the system's technical architecture and functional architecture, respectively.

1. Introduction

With the development of society and economy, the enrollment scale of higher education in China is increasing, and the teaching resources of colleges and universities are gradually becoming scarce resources. It is crucial to figure out how to quickly assess and process college students’ teaching data, as well as how to boost the management efficiency of college teaching affairs. In light of this issue, the research focuses on how to create and execute a universal educational administration system using data mining technologies. At present, a lot of research has been going on the teaching resource management, and many foreign academics are currently working on data mining technology research. Kochetov was one of them, and he highlighted the recent growth of data mining technology, which is quite significant [1]. By using cloud platform technology, Marozzo et al. proposed an improved data mining and analysis technology [2] to promote the development of this research direction. In addition, many domestic scholars have gradually applied data mining technology to the educational administration management system of colleges and universities [3–5]. However, its practicability is high, but the theory itself is low. For example, Kuppuraju and Kumar applied DM technology
into the educational administration management system, and Li et al. applied the Apriori algorithm into the study of educational administration management [6, 7].

Studying data mining methods, extracting useful information, determining a reasonable human resource structure based on college and university development goals, and determining the purpose of talent introduction and training are all important practical considerations [8]. A university’s human resource database contains huge information. This information comprises basic information about faculty and staff, such as name, gender, age, date of birth, and place of birth, as well as staff details and information about staff rewards and punishments, work experience, published papers, marital status, and other personal and professional data. The author employs ASP technology, JavaScript language, and the Apriori algorithm to build and implement a university human resource data mining system based on the relationship between instructors’ gender, age, academic degree, professional title, and published articles [9].

Digital campus platform of education resource management system is a new subject and is the basic education information management system. Its execution is the key to education information sharing, which necessitates more time and effort from the employees involved in the development of computer education software. A realistic information management system for secondary school education resource management was designed based on the characteristics of middle school new curriculum educational reform design [10]. This topic is brought up in response to teacher, parent, and student demand for teacher education resources. Because of the research on the present situation of education resource management, a reasonable plan could be found for resource sharing. In order to achieve the education e-government platform function, consider the interaction between parents and teachers, and open up the teacher BBS zones for better service to the education. In this context, the research on this topic will help all kinds of middle schools and other network education service institutions in the current nine-year compulsory education stage to improve the quality of education, expand resources, and promote the sharing and interaction of educational resources [11].

The rest of this paper is organized as follows. The related work of the research paper has been discussed in Section 2. Different algorithms have been discussed that have been used in several previous studies. After the related work, the algorithm idea is discussed in Section 3, which lay a solid foundation for the proposed work of Section 4. Next, the experimental results are presented in Section 5, which is followed by conclusion of this research paper in Section 6.

2. Related Work

In this section of the research paper, the work related to the article has been discussed. Different algorithms are discussed for this purpose, e.g., partition, DPH, and DIC algorithms, which are all relevant to this study. Lastly, the idea and design of the algorithm are discussed. On a broader scale of study and practice, many researchers are working on the algorithm for the prototype to decrease the number of scans each time based on the condition of the system and introduction of the memory to reduce the number of readings of original database [12]. As the amount of data processed grows, the Apriori algorithm is proposed.

2.1. Partition Algorithm. The overall database is divided into independent data blocks, and then, in each data block, Apriori algorithm is applied to generate the user-required association rules; the method is called the Partition algorithm. The biggest advantage of this algorithm is that it does
not need to scan the database every time a layer of candidate itemset is generated. The algorithm needs to go through two steps to generate the corresponding frequent itemset:

The first step of the algorithm is to determine a segmentation principle according to the nature of mining and then divide the whole database into several small data blocks. The data blocks after classification shall have two basic conditions: First of all, the separated data blocks should be relatively independent, and there is no shared part of the data. Second, according to the hardware system’s needs and to increase the system’s execution efficiency, the segmented data blocks should be put into memory all at once, allowing to process data in high-speed storage devices. After the segmentation, each part can be treated as an independent database, and the support degree of itemsets can be calculated separately to find out the frequent itemsets in each section. If the data generated in the segmentation stage is indeed independent of each other, then the final association rules
are included in these segments of frequent itemsets. Therefore, we need to record all the generated regional itemsets and determine whether they are frequent itemsets of the whole database in the following [13].

On the other hand, the frequent itemsets are not necessarily the final result; according to the nature of Apriori, to find frequent itemsets is the only section of this process and is not necessarily frequent in the whole database; at this point, we need various sections of frequent itemsets, and a full scan of the database is done again. The confidence and support of all the regional frequent itemsets are calculated to find the final frequent itemsets.

Under the condition of reasonable segmentation rules, scan of database is needed of this algorithm when the regional frequent itemset is segmented and finally screened, which greatly reduces the number of data exchanges between the system and external storage devices and improves the algorithm efficiency. However, this algorithm has some unavoidable defects: In the process of generating frequent itemsets, as the generation scope is only the current segment, many frequent itemsets may be frequent only for the current segment. Therefore, many nontrue intermediate results may be generated in the regional calculation, resulting in too many frequent itemsets to be determined, increasing the burden of determination. Moreover, in order to avoid the repetition of frequent itemsets in each section and in order to be able to further use and estimate itemsets in the mining process, on the whole, a database system for mining must be ordered and well maintained by the original database. As the growth of the number of data increases in a geometric pattern, therefore, when the amount of data is relatively large, the time spent on sorting is considerable, which greatly limits the efficiency of the estimation method. Due to the need to carry out a complete data mining process for each small section, the calculation time in the estimate method will be much larger than the Apriori algorithm.

2.2. DPH (Direct Hashing and Pruning) Algorithm. The results are compared with the minimum support defined frequent itemsets, using the DPH algorithm of the hash

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Num</td>
<td>Name</td>
<td>Ceb</td>
<td>Aml</td>
<td>La</td>
</tr>
<tr>
<td>2</td>
<td>4140101</td>
<td>User 1</td>
<td>01111</td>
<td>01</td>
<td>01111</td>
</tr>
<tr>
<td>3</td>
<td>4140102</td>
<td>User 2</td>
<td>01111</td>
<td>01</td>
<td>01111</td>
</tr>
<tr>
<td>4</td>
<td>4140104</td>
<td>User 3</td>
<td>10111</td>
<td>11101</td>
<td>01111</td>
</tr>
<tr>
<td>5</td>
<td>4140104</td>
<td>User 4</td>
<td>10111</td>
<td>11011</td>
<td>01</td>
</tr>
<tr>
<td>6</td>
<td>4140105</td>
<td>User 5</td>
<td>0111</td>
<td>01</td>
<td>11011</td>
</tr>
<tr>
<td>7</td>
<td>4140106</td>
<td>User 6</td>
<td>01111</td>
<td>10111</td>
<td>10111</td>
</tr>
<tr>
<td>8</td>
<td>4140107</td>
<td>User 7</td>
<td>01111</td>
<td>01</td>
<td>10111</td>
</tr>
<tr>
<td>9</td>
<td>4140108</td>
<td>User 8</td>
<td>10111</td>
<td>10111</td>
<td>11101</td>
</tr>
<tr>
<td>10</td>
<td>4140109</td>
<td>User 9</td>
<td>01111</td>
<td>10111</td>
<td>01111</td>
</tr>
<tr>
<td>11</td>
<td>4140110</td>
<td>User 10</td>
<td>01111</td>
<td>01111</td>
<td>01</td>
</tr>
<tr>
<td>12</td>
<td>4140111</td>
<td>User 11</td>
<td>01111</td>
<td>01</td>
<td>01111</td>
</tr>
<tr>
<td>13</td>
<td>4140112 / User 12</td>
<td>01111</td>
<td>01111</td>
<td>10111</td>
<td>1011</td>
</tr>
<tr>
<td>14</td>
<td>4140114</td>
<td>User 13</td>
<td>01111</td>
<td>10111</td>
<td>01111</td>
</tr>
<tr>
<td>15</td>
<td>4140115</td>
<td>User 14</td>
<td>01111</td>
<td>01111</td>
<td>01111</td>
</tr>
<tr>
<td>16</td>
<td>4140116</td>
<td>User 15</td>
<td>01111</td>
<td>11101</td>
<td>11011</td>
</tr>
<tr>
<td>17</td>
<td>4140117</td>
<td>User 16</td>
<td>11101</td>
<td>01</td>
<td>11101</td>
</tr>
<tr>
<td>18</td>
<td>4140118</td>
<td>User 17</td>
<td>011</td>
<td>11110</td>
<td>11110</td>
</tr>
<tr>
<td>19</td>
<td>4140119</td>
<td>User 18</td>
<td>11011</td>
<td>0</td>
<td>11110</td>
</tr>
<tr>
<td>20</td>
<td>4140120</td>
<td>User 19</td>
<td>11011</td>
<td>11110</td>
<td>01</td>
</tr>
<tr>
<td>21</td>
<td>4140121</td>
<td>User 20</td>
<td>11011</td>
<td>11110</td>
<td>11110</td>
</tr>
<tr>
<td>22</td>
<td>4140122</td>
<td>User 21</td>
<td>11011</td>
<td>01</td>
<td>011</td>
</tr>
</tbody>
</table>
function to filter all the frequent itemsets, thus reducing the number of the collection of the next need to scan the data [14]. When filtering frequent itemsets, the DPH algorithm mainly uses hash attributes and deletes unnecessary candidate itemsets according to bucket values in the hash table. In this way, the algorithm will delete more and more itemsets as the number of rounds increases, resulting in less and less data being scanned at the end of each round. Thus, the execution efficiency of the Apriori algorithm is greatly improved.

2.3. DIC Algorithm (Dynamic Data Item Calculation Method). Since the Apriori algorithm generates candidate itemset by scanning layer by layer, the whole database needs to be scanned for frequent itemsets of each layer. After that, the current frequent itemset is connected to generate candidate itemsets of the next layer until the frequent itemsets can no longer be connected to generate candidate itemsets. The core of the DIC algorithm [15] is to divide the database into multiple sections. Secondly, it connects the frequent itemsets generated in the subsections after processing one section. The above algorithms have improved the execution efficiency of the Apriori algorithm to a certain extent but have not completely solved the inherent defects of the Apriori algorithm. For that purpose, we need to explore a more effective algorithm in the field of data mining to improve the efficiency of mining.

3. Algorithm Idea

Basically, data mining is an operation of excavating effective data from random data. The technical organization is shown in Figure 1. The proposed structure lay a foundation for this research study, which will be presented in Section 4.

According to the technical structure, compared with traditional database technology, data mining deals with a larger scale of data and can find more potential effective rules, and the execution speed is faster. This method has the advantages listed above and provides a theoretical foundation for its use in the field of association rules. Association rules mean that from a certain dimension, any two things in the world have some kind of correlation, which is particularly evident between data. Data association is an important law existing in massive data. The algorithm consists of two steps.

3.1. Mining Frequent Itemsets in a Large Amount of Data

3.1.1. Summarize Potential Association Rules by Using Known Frequent Itemsets. Apriori algorithm is usually efficient and accurate because of its simple structure and derivation. However, this algorithm is difficult to mine objects with low support, multifrequency patterns, and complex structures, mainly because:

When the Apriori algorithm processes candidate sets, if the data volume of frequent 1-itemsets is large, it will be more difficult for the algorithm to process candidate 2-itemsets, leading to a sharp decrease in algorithm’s efficiency.

The Apriori algorithm must constantly scan the database, which gets a lot of computing resources, thus reducing the pattern matching efficiency of the algorithm.

4. Algorithmic Design of the Proposed Work

Apriori algorithm is composed of items, association rules, transactions, frequency, and support. The details are as follows:

Hash Technology. When the algorithm scans transactions, frequent 1-itemset L1 is generated by candidate 1-itemset belonging to candidate itemset Cl. In this case, the hash technology can be used to map the two itemsets to different buckets in the hash table to increase the number of buckets. Therefore, if the bucket count in the hash table is lower than the candidate 2-itemset, the candidate itemset can be directly deleted to achieve the purpose of compressing the candidate 2-itemset.

Transaction Compression Techniques. According to the characteristics of the Apriori algorithm itself, if a transaction...
does not include any frequent \( k \)-itemset, it also does not contain any frequent \( k + 1 \)-itemset. Therefore, when generating frequent \( j \)-itemsets \( (j > k) \), the algorithm no longer needs to scan the database. Thus, the algorithm can cut the number of database scanning and improve the execution speed of the algorithm.

Partition Technology. When partitioning is applied, the number of database scans can be directly reduced to 2. In the first scan, the transaction set is divided into multiple logical subsets, and the local frequent itemsets of each subset are searched. During the second scan, the support degree of all local candidate itemsets is calculated, and the global frequent itemsets are determined by dividing the database.

Sampling Techniques. When the sampling technology is applied, the algorithm needs to select the random sample \( S \) in database \( D \) and directly search the frequent itemset in \( S \). Instead of scanning the entire database \( D \), this algorithm scans only one transaction in \( S \). In addition, the sampling technique improves the execution efficiency by reducing the accuracy of the algorithm.

Using the above technologies, this paper improves the Apriori algorithm, so as to improve the efficiency of the algorithm. The improved algorithm flow is shown in Figure 2.

The functional modules can be realized in this paper, and a detailed introduction is as follows:

1. Collection and Pretreatment. A variety of data has been collected which includes the data from different databases. This system makes use of a SQL Server database, which could be used in a variety of ways, for example, Oracle, MySQL, or other types of databases. In order to avoid the mismatch caused by a variety of databases, the system uses Open Database Connectivity (ODBC) to communicate with each other to achieve access to different databases.

2. Data Sorting. After data extraction, the system needs to filter the data and delete redundant data and data with no obvious features. In this process, the system also needs to classify valid data, that is, cleanse data.

3. Data Analysis. The data analysis module is the core module of the educational administration management system. It mainly uses the improved Apriori algorithm to analyze and process the effective data of the intermediate database and obtain the potential association rules between these data, so as to provide the basis for system users to make decisions.

4. Data Revision. Because the system cannot predict the final results of data analysis, users cannot determine the significance of the analysis results. In order to avoid the situation, the system needs to set up a data correction module to revise and analyze the results of data analysis. This technique will judge whether the final results have practical significance or have no practical meaning or fuzzy meaning.

5. Present the Results. Data mining results need to be presented to system users, so the system should improve the readability of analysis results as much as possible on the premise of maintaining the accuracy of analysis results. This also means that the module will most likely present the system’s analytical results in the form of charts.

The pseudo-code of the improved Apriori algorithm is also given in this paper, which is described as follows:

Initialize array \( n = \text{Max. Weight} \). Using the simple matching algorithm, the similarity between the retrieval results and each user keyword is obtained, and the weight \( (I) \) is updated. Calculate the total similarity between each mobile learning resource and \( W \) and record. According to the formula:

\[
\text{sim}_{\text{total}} = L(k) \times \sum_{j=1}^{n} \sum_{i=1}^{m} \text{sim}_{m}(w_{j}, K_{i}). \tag{1}
\]

Calculate the total similarity between each mobile learning resource and \( W \) and record.

According to the method of feature extraction, the feature vector of each learning resource is obtained. The corresponding feature scale is generated through calculation. The calculation method of the weight value of the feature vector of each text in a certain dimension is as follows: Take the \( i \)-th feature word and the quantity of documents in which the feature word appears from the feature vector. To calculate the feature word’s frequency appearing in the text, we use the formula:

\[
W_{i}(d) = \log \left( \frac{N}{n_{i}} \right). \tag{2}
\]

Calculate its weight value, until the weight calculation of feature words in the feature vector is finished; finally, the vector is normalized.

It is often not possible for a resource node to deploy only one grid service. To make full use of its resources, a resource node may deploy several different grid services. When the number of external services provided by resource nodes increases, resources (such as CPU and memory) are limited. The usage of the marginal principle has been discussed in this section to analyze and adjust the price of each service under the condition of multiple services and single resource constraints, so as to maximize the income of this node.

This problem needs to be solved by the Lagrange multiplier method, which transforms it into the objective function containing constraint conditions is max \( E = e(P_{1}, P_{2}, \ldots, P_{n}) + \lambda [R - g(P_{1}, P_{2}, \ldots, P_{n})] \), and \( \lambda \) in the equation is an undetermined coefficient. Thus, the first-order condition for \( E \) to take the extreme value is as follows:

\[
E_{\lambda} = \frac{\partial E}{\partial \lambda} = R - g(P_{1}, P_{2}, \ldots, P_{n}), \tag{3}
\]

\[
E_{P_{i}} = \frac{\partial E}{\partial P_{i}} = e_{P_{i}} - \lambda g_{P_{i}} = 0. \tag{4}
\]

In economics, \( e_{P_{i}} = \partial e(P_{1}, P_{2}, \ldots, P_{n})/\partial P_{i} \),

\( g_{P_{i}} = \partial g(P_{1}, P_{2}, \ldots, P_{n})/\partial P_{i} \) is the price of the resource, that is the amount of resource occupied by this service. Simultaneous formulas (3) and (4) can be obtained as...
follows:

\[
\frac{\partial e_{P_i}}{g_{P_i}} = \frac{\partial e_{P_{i+1}}}{g_{P_{i+1}}} = \ldots = \frac{\partial e_{P_n}}{g_{P_n}} = \lambda. \tag{5}
\]

In this way, the marginal equation for the constrained optimization problem is obtained. It shows that when the marginal revenue of each service provided by the resource node is equal to the ratio of the resource consumed, the revenue \( E \) will be maximized. The user demand function \( D = f(P_i) = (a_i - P_i)/b_i \) can be substituted into the above equation.

\[
\frac{2P_1 - a_1}{r_1} = \frac{2P_2 - a_2}{r_2} = \ldots = \frac{2P_n - a_n}{r_n} = \lambda. \tag{6}
\]

Formula (5) is simultaneous with constraint condition \( g(P_1, P_2, \ldots, P_n) = R \); assuming that the value of \( A \) is known, the target value can be obtained.

That is, subject and learning sections are introduced as background constraints in the model, and interest weight function based on temporal changes is introduced in vector space.

At a certain moment \( T \), the user interest model is expressed as follows:

\[
K = \{(k_1, \omega(T_1)), (k_2, \omega(T_2)), \ldots, (k_n, \omega(T_n))\}. \tag{7}
\]

The calculation is based on the time window mechanism; that is, in a certain time window, if the keyword is submitted, the weight is increased; otherwise, the weight is attenuated. Then, at a certain time \( t \), the interest weight function of keyword \( KN \) is expressed as follows:

\[
\omega(T_n) = \left\{ \begin{array}{ll}
\sum_{i=1}^{f(t, \alpha)} & (f(t_n, i-1) \cdot \Delta t) \cdot a - c \cdot b) \\
0 & \end{array} \right., \tag{8}
\]

According to the design scheme, this paper gives a concrete implementation scheme, the deployment of the system, the production of educational resources, the realization of each function module, and other contents.

5. Experimental Results and Analysis

It should be noted that the test data came from the scores of 840 students in several courses in a university. Partial test numbers are shown in Table 1.

A key method of performance evaluation is using concurrent testing. When simulating the operation of a software system, it is necessary to estimate the number of users under normal and extreme conditions, so as to test the performance of core functions and important services under different numbers of users. The test is run in user accumulation mode, which means that the number of users is increased to raise the system’s load. Until the system meets undesirable performance points or bottlenecks, the test data must be real service data.

This system carries on a user concurrent test to the resource download service, the user logs in the system, uploads a file, then adds news, views a resource, views a bulletin, and then exits the system. Log in 100 users in total, start with 10 users, and then add 5 users every 2 seconds. The test results are shown in Figures 3 and 4.

After numerous users enter the system, the concurrent execution of distinct functional modules and operations can more closely imitate the actual circumstances of a multi-user operating system and carry out the combinatorial test; for example, when using educational resources, it is necessary to test the combination of upload, download, search, view, and management of resources.

According to the above combined performance test principle and in the situation of big capacity resource operation, this article simulates the maximum number of users in a certain period of time, conducts the concurrent execution of the key modules of the system, and focuses on assessing the system network throughput and system response time. Through comprehensive analysis of execution indicators and resource monitoring to determine the performance of the system when handling the maximum traffic, the test results are shown in Figure 5.

Teaching resource library management system in the design process involved in the business process of the actual life process as the standard, so that users can be more convenient to use the system. After the functional test of the final design effect of the system and the expected effect is consistent, prove that the development of the system function in line with the requirements of users. The test tool’s performance test of the system reveals that the system can maintain an average response time of 0.005 seconds when 100 users operate concurrently, basically meeting the needs of users.

The Apriori algorithm was applied to mining association rules for the preprocessed data. Taking “air conditioning power consumption per unit period” as the rule post, three strong association rules were obtained, as shown in Table 2.

Through the experimental mining results and related facts, we can find some rules from the experimental results: Association rule 1 indicates that there are many cases of no one in the room from 12:00 to 12:30 because 11:30 to 12:30 is the lunchtime. The air conditioner energy-saving controller delays the automatic shutdown of the air conditioner for 20 minutes while no one is in the room. Using this association rule, we can set the delay of the air conditioner automatic shutdown when no one is in the room to 5 minutes during lunch (generally 11:30 to 12:30) to reduce air conditioner energy consumption.

6. Conclusion

This paper mainly focuses on how to dig out the connections between the real user data in the huge database system and the mining of association rules in data mining. The algorithm’s vulnerabilities have been discovered. In order to address these flaws, a new better method is developed, which
is primarily improved from the old algorithm’s requirement to scan the database numerous times, resulting in a significant increase in the Apriori algorithm’s running performance. The system, which focuses on the five themes of rule mining, curriculum association, curriculum category association, student basic information association, curriculum, basic information association, and teaching mode association, can help with teaching management. The practice proves that the application of the Apriori algorithm in student information management systems has certain practical value. The pertinence and accuracy of data mining make the association rule algorithm plays the function of course construction and improving teaching levels in student information management systems. Data mining technology is not widely used in university student management systems. With the continued promotion of association rule algorithms, their unique superiority is causing data mining technology to gain traction. Student information management systems based on the Apriori algorithm will also be important for future development in the field of university student management.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that he has no conflict of interest.

Acknowledgments

The paper was supported by the 2021 Education Project supported by the National Social Science Fund: A Research on the Retention Mechanism of Rural Teachers (BHA210137).

References