Hindawi
Scientific Programming
Volume 2022, Article ID 4684461, 10 pages
https://doi.org/10.1155/2022/4684461

Research Article

Research on the Characteristic Model of Learners in Modern Distance Music Classroom Based on Big Data

Yushan Wang[1] and Lianhong Liu[2]

[1]*Sichuan Vocational College of Finance and Economics, chengdu, Sichuan, Chengdu 610101, China*
[2]*Chengdu Institute of Physical Education, Sichuan, Chengdu 610041, China*

Correspondence should be addressed to Lianhong Liu; 100345@cdsu.edu.cn

Received 14 March 2022; Revised 28 March 2022; Accepted 8 April 2022; Published 14 May 2022

Academic Editor: Jie Liu

This paper makes in-depth research on data mining, especially association rule mining, improves the FP-tree algorithm in both the algorithm itself and the data source, and finds out a mining algorithm suitable for learner characteristics. Association rule algorithm for actor feature model mining. By establishing the characteristic model of learners in modern distance music classroom, simulation experiments are carried out on FP-tree and three improved algorithms. This paper improves the FP-tree algorithm. Firstly, we improve the algorithm itself; aiming at the problem of too many frequent itemsets, an improved key item extraction algorithm KEFP-growth based on FP-growth is proposed, which ignores the frequent itemsets that are not concerned in the analysis. Then, improvements were made in terms of data sources. In view of the problem that the data source is too large, the mining efficiency is low, and the FP-tree cannot be loaded in memory, this paper proposes a data projection algorithm, which adopts the idea of divide and conquer, divides the frequent 1-itemsets of the database into database subsets of each frequent 1-itemsets, and then mines the database subsets separately and then merges them. Finally, the KEFP-growth algorithm and the projection algorithm are combined, which can not only eliminate the mining of meaningless frequent items but also divide the data when there is a large amount of data. This paper also compares the performance of the three improved algorithms and the original FP-tree algorithm through experiments. The experiments show that the combination of the KEFP-growth algorithm and the database projection algorithm is the most suitable one for the learner feature mining of the adaptive learning system. (1) The KEFP-growth algorithm reduces the number of frequent items output by the original FP-tree algorithm by about 50%, and the mining time is reduced by 50%. (2) The data projection algorithm is more suitable for data mining with less support. When the support is 10%, the mining time of the data projection algorithm is reduced by 80% compared with the FP-tree algorithm. (3) When the support degree is 10%, the running time of the hybrid algorithm is reduced by 10% compared with the KEFP-growth algorithm and the data projection algorithm.

1. Introduction

Learner characteristics refer to the psychological, physical, and social characteristics that have an impact on the learner's learning, that is, the learner's personality factors. Wenger believes that the student model represents all the behavior and knowledge about the student [1]. Subsequently, many scholars have defined learner models based on different learning environments. Vanlehn believes that "a learner model is a data structure that represents the student's current state of knowledge" [2]. Asarta and Schmidt believe that "the learner model can be represented as a quadruple (P, C, T, H), where P represents procedural knowledge; C represents conceptual knowledge; T represents individual characteristics (traits) of individual; and H represents the learning history [3]. Holt et al. believe that the learner model is an abstract representation of the learner's belief by the computer system, that is, the learner model represents the system's cognition of the student [4]. Later, many scholars define learner models based on different learning environments. Currently, the representative learner characteristic systems are as follows: (1) classical learning characteristic analysis system [5] and (2) distance learning model theoretical analysis system for educating students [6] (the

theoretical analysis system has seven dimensions (general data, demographic data, sociological data, geographic data, situational state data, motivational data, and opinion and evaluation data)).

The modern distance music education classroom learner characteristic model is a guiding form that provides personalized learning paths and learning resources for learners according to their characteristics and behavioral tendencies in the distance learning environment. It is ultimately to achieve the purpose of teaching students in accordance with their aptitude.

In the era of big data, technology transforms all behaviors in the educational process into educational data, which helps to observe the performance of each student, promotes educational research from macro-groups to micro-individuals, and is conducive to the "tailoring" of teaching and the realization of data-driven personalization.

In 2012, the United States pointed out in the report "Promoting Teaching and Learning through Educational Data Mining and Learning Analysis" that "with the support of big data and cloud computing, the core trend of international information education technology development is personalized learning" [7]. The "2016 National Educational Technology Plan" in the United States, "Learning for the Future: Reshaping the Role of Technology in Education," also emphasizes the development of personalized learning based on big data analysis.

China's "Ten-Year Development Plan for Education Informatization (2011–2020)" clearly pointed out that "it is necessary to build an information-based environment and provide individualized learning services for each student." The Ministry of Education's "Thirteenth Five-Year Plan for Educational Informatization" also proposed that "the construction of online learning space should meet the needs of personalized learning and realize "one space for every life, and every life has its own characteristics."" The Horizon Report of the American New Media Alliance has also pointed out many times that adaptive learning is the trend of information technology development in higher education. According to the 2016 Key Issues in Teaching and Learning Report released by the Higher Education Information Association (EDUCAUSE), the focus of teaching and learning is not advanced technology, but the learners themselves and how to use technology to provide a personalized learning experience. Adaptive learning technology can provide learners with personalized learning services [8]. Zhu and Shen believe that with the rapid development of educational big data and data science, personalized adaptive learning will become an important part of the new paradigm of educational technology and smart learning environment, and it is necessary to carry out systematic and in-depth research on it [9].

2. Related Theories

2.1. Data Mining. Data mining is the non-trivial process of obtaining valid, novel, potentially useful, and ultimately understandable patterns from large amounts of data.

2.1.1. Data Mining Process. Data mining is a process of discovering various models, summaries, and derived values from known datasets. A general experimental procedure suitable for data mining problems includes the following steps [10].

(1) State the problem and clarify the hypothesis.

(2) *Data Collection*. There are usually two methods: "design experiment method" and "observation method."

(3) *Data Preprocessing*. It usually includes at least two common tasks: (1) outlier detection (and removal); (2) scaling, encoding, and feature selection.

(4) *Model Evaluation*. Selecting and implementing appropriate data mining techniques is the main task of this stage.

(5) Explain the model and draw conclusions.

2.1.2. Data Mining Algorithms. Data mining tasks include concept description, association analysis, classification analysis, cluster analysis, outlier analysis, evolution analysis, etc. Among them, association rule mining is the most active and deeply researched field.

(1) Association Rule Mining Algorithm. Association rule mining is to search for valuable associations between data items from a given dataset. Association rule mining in a transactional database can be described as follows [11].

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items, the transaction database $D = \{t\_1, t\_2, \ldots, t\_n\}$ is composed of a series of transactions with a unique identifier TID, and each transaction $t_i\,(i = 1, 2, \ldots, n)$ corresponds to a subset on I.

Definition 1. Let I_1⊆I, and the support of itemset I_1 on dataset D is the percentage of transaction D containing I_1:

$$\text{support}(I_1) = \frac{\left\|\{t \in D | I_1 \subseteq t\}\right\|}{\|\|D\|\|}. \tag{1}$$

Definition 2. For itemset I and transaction database D, all itemsets in T that satisfy the minimum support specified by the user are called frequent itemsets or maximum itemsets. Picking out all frequent itemsets that are not contained by other elements in the frequent itemsets is called maximal frequent itemsets or maximal large itemsets.

Definition 3. An association rule of the form $I_1 \Rightarrow I_2$ defined on I and D is given by satisfying a certain degree of credibility, trust, or confidence. The so-called credibility of the rule refers to the ratio of the number of transactions including I_1 and I_2:

$$\text{Confidence}(I_1 \Rightarrow I_2) = \frac{\text{support}(I_1 \cup I_2)}{\text{support}(I_1)}, \tag{2}$$

where $I_1, I_2 \subseteq I, I_1 \cap I_2 = \varnothing$.

Table 1: FP-tree algorithm.

Algorithm: use the FP-tree algorithm to mine frequent patterns through pattern segment growth
Input: transaction database-D: minimum support threshold min_sup
Output: the complete set of frequent patterns.
Method:
(1) Follow the steps to construct FP-tree;
(a) Scan D once. Collect a set F of frequent items and their support. Sort F in descending order of support, and the result is the frequent item table L.
(b) Create the root node of the FP-tree, marking it with "null." For each transaction in D, execute:
Select the frequent items in transaction and sort them by the order in L. Let the sorted frequent item table be $[p|P]$, where p is the first element and P is the list of remaining elements. Call insert_tree$([p|P], T)$.
The process is performed as follows. If T has child N such that N.item − name = p.item − name, then N's count is incremented by 1; otherwise, a new node N is created with its count set to 1, linked to its parent node T, and it is linked to nodes with the same item − name through a node chain structure. If $P \neq \varnothing$, insert_tree(P, N) is called recursively.
(2) The mining of FP-tree is realized by calling FP_growth(FP_tree, α). The process is implemented as follows:
Procedure FP_growth(Tree, α)
(1) If tree contains a single path P then
(2) For each combination of nodes in the path P (denoted by β)
(3) Generate a pattern $\beta \cup \alpha$, whose support = support$(\beta)_{\text{minimum}}$;
(4) Else for each α_i at the head of tree {
(5) Generate a pattern $\beta = \alpha_i \cup \alpha$, and its support degree support = support (α_i);
(6) Construct the conditional pattern basis of β, and then construct the conditional FP-tree β of β;

Definition 4. The association criterion that satisfies the minimum support and the minimum trust degree on I is called the strong association criterion.

At present, there are many algorithms for mining association rules, which are mainly divided into two categories: generating candidate sets and not generating candidate sets. In order to improve the efficiency of the algorithm, this paper adopts the association rule algorithm that does not generate candidate sets. Han et al. proposed FP-tree algorithm [12], and Liu proposed the Relim algorithm [13]. Among them, the FP-tree algorithm is the most typical one. Compared with the FP-tree algorithm, the Relim algorithm has a simple structure, high space utilization, and runs faster when mining datasets with high minimum support or few frequent rules. The tree algorithm is effective and scalable for mining long and short frequent patterns, while the FIMA algorithm occupies less memory and has high algorithm execution efficiency, but it can only mine Boolean association rules, not quantitative association rules. The FP-tree algorithm only performs 2 database scans. It does not use candidate sets, directly compresses the database into a frequent pattern tree, and finally generates association rules through this tree. The specific mining process is shown in Table 1.

2.2. Learner Characteristics. Learner characteristics refer to the psychological, physical, and social characteristics that have an impact on the learner's learning, that is, the learner's personality factors.

The characteristics of learners involve many aspects, but the characteristics that have an important impact on learning mainly involve two aspects: intellectual factors and non-intellectual factors. The characteristics related to intellectual factors mainly include the general characteristics of learners, knowledge base, cognitive ability and cognitive structure variables, etc., and the characteristics related to non-intellectual factors include interest, motivation, emotion, learning style, anxiety level, will and personality, learner's cultural and religious background, etc.

Through the analysis of specific learner characteristic systems, we find that they have their own characteristics. For example, Ding Xingfu's theoretical analysis system for distance education students is relatively comprehensive, covering a wide range of aspects, covering almost all the characteristics of learners. However, it is too large and lacks follow-up research to combine learner characteristics with specific applications, and the operability is not very strong. Considering the implementation problem, this paper intends to use the learner characteristic analysis system in network distance education in Tempelaar's "Construction of learner characteristic analysis system in network distance education and the design of student model" [14] as a prototype. For research, the feature analysis system mainly includes six major items, each of which includes many minor items, a total of 52 minor items:

(a) *Intelligence*. Intelligence refers to the ability of people to recognize and understand objective things and use knowledge and experience to solve problems, such as memory, observation, imagination, thinking, and judgment. Intelligence mainly includes intelligence and five components of Professor Hanbidge [15] (observation, memory, imagination, thinking, and attention). Intelligence includes language intelligence, mathematical logic intelligence, etc.

(b) *Learning Style*. Learning style is a continuous and consistent way of learning with individual characteristics, and it is a combination of learning strategies and learning tendencies. Learning styles are mainly divided into physiological elements, cognitive

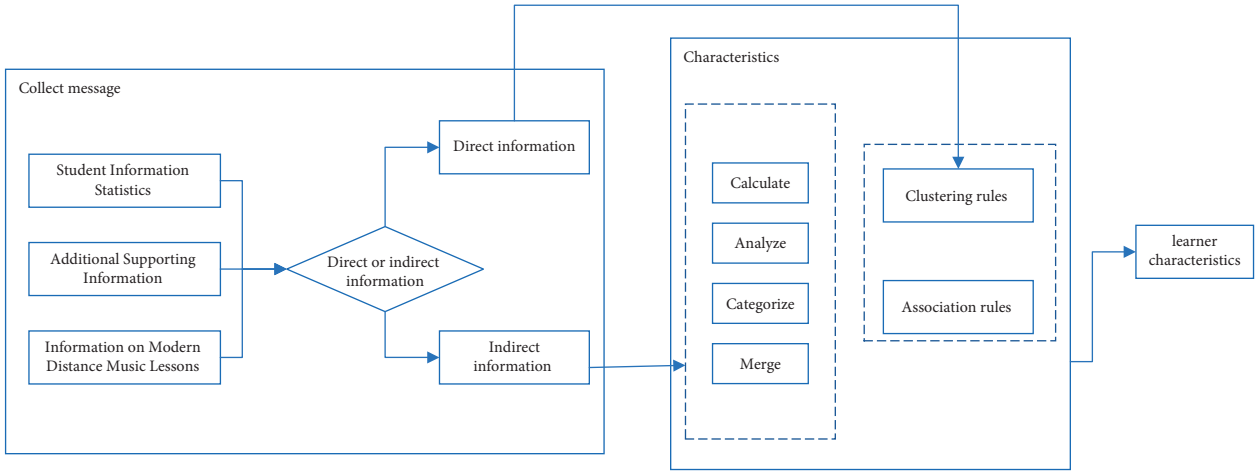

Figure 1: Framework diagram for the analysis of learners' characteristics in modern distance music classrooms.

elements, emotional elements, brain function, and personality. Physiological factors include time preference, perceptual response, and sound preference. Cognitive elements include perceptual style (verbal-spatial preference, discriminating skills, etc.), thinking style (analytical and non-analytical, scattered/concentrated, etc.) confidence processing style, memory style, and problem-solving style.

(c) *Study Preparation*. Learning readiness refers to the adaptability of students' original knowledge level or psychological development level to new learning. Learning preparation includes motivation, cognitive structure, and learning attitude.

(d) *Web Learner Characteristics*. We take this class into account since learners need to use computers and networks for adaptive learning. This category includes four subitems: technical level, information literacy, online learning adaptability, and online psychology.

(e) General information includes demographic and sociological data and opinion and evaluation data.

(f) *Others*. Since there are many repetitions of non-intellectual factors and learning styles, non-intellectual factors (excluding items that overlap with learning styles) are classified into this category, including knowledge interest, learning enthusiasm, learning responsibility, dominance, competitiveness, and self-confidence.

3. Analysis of Characteristics and Model Assumptions of Learners in Modern Distance Music Classrooms

3.1. Analysis Framework. Collect learners' behaviors from different perspectives and explore the deep reasons behind the behaviors, as a basis for formulating teaching procedures. Combining with domestic and foreign learning feature analysis models, combined with the basic elements of learner characteristics in the modern distance environment, this research proposes an analysis framework for learner characteristics in the modern distance environment, as shown in Figure 1.

3.2. Learner Feature Model. Using appropriate technology to build a personality feature model can obtain the learning status of learners in real time and effectively support personalized learning. The construction methods of personality feature models include coverage model, lead model, perturbation model, machine learning technology, model based on cognitive theory, constraint-based model, fuzzy logic technology, Bayesian network, and semantic web ontology model. Among them, the cover model and lead plate model are the most common modeling techniques.

The coverage model, proposed by Jaques et al., is a method commonly used to describe the user's level of knowledge about each concept. When using the coverage method to construct the knowledge level model of the learner, the domain knowledge model represents the expert level knowledge of a certain subject, and the learner model is regarded as a subset of the domain knowledge model. The lead plate model was introduced into the GRUNDY system by Rich to build a user feature model [16]. The core idea is to group and cluster all potential users in ALS according to specific characteristics, and each group is a user lead plate. Both perturbation and constraint-based models are modeled based on learner errors/misunderstandings. The perturbation model, also known as the deviation model, is an extension of the coverage model. The researchers of this model believe that the knowledge of the learner includes not only the partial knowledge possessed by the domain experts but also the wrong knowledge that the learner may generate.

The ultimate goal of the development of educational big data is to return to the essence of education and realize "teaching according to aptitude." The biggest drawback of the "one-size-fits-all" unified teaching model is that it ignores individual differences of learners and analyzes learners' personality characteristics including knowledge level, errors/misunderstandings, emotional characteristics,

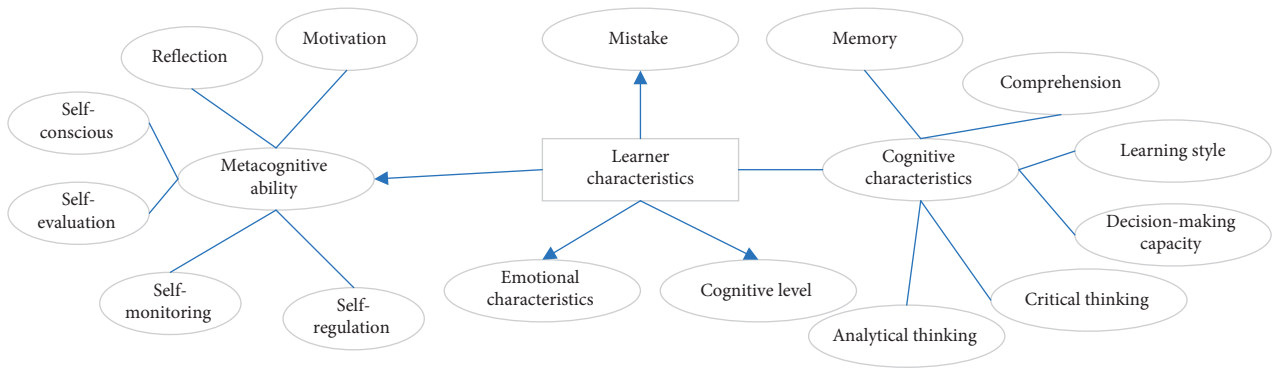

Figure 2: Learner characteristics.

Table 2: KEFP-growth algorithm.

Input: transaction database D; minimum support threshold min_sup
Output: frequent itemsets including key items
(1) Construct FP-tree according to the following steps
(i) Scan the D once. Collect a set F of frequent items and their support. Sort F in descending order of support, and the result is a frequent 1-item set L
(ii) Create the root node of FP-tree and mark it with null. For each transaction trans in D, execute:
Select frequent items in trans and sort by the order in L. Let the sorted frequent item table be $[p|P]$,
Where, p is the first element, and P is the list of remaining elements. Call insert_tree$([p|P], T)$. Execution of the process as follows. If T has child N such that N.item_name = p.item_name, then increment N's count by 1; otherwise create a new node N with its count set to 1, linked to its parent node T, and it is linked to nodes with the same item-name through a node chain structure. If $P \neq \varnothing$, insert_tree(P, N) is called recursively.
(2) Select key items and divide the header table into key item table $L1$ and non-key item table $L2$
(3) The mining of FP-tree is realized by calling KEFP_growth(Tree, α). The process is realized as follows:
Procedure KEFP_growth(Tree, α)
(1) If (tree contains a single path P)
(2) {
(3) If (α contains key items)//check whether the frequent items in α appear in $L1$
(4) Output the union of each node in the tree and α;
(5) Else
(6) { //All items in the key item node are key items, all items in the non-key item node are non-key items key item
(7) Tree nodes are divided into key item node $N1$, non-key item node $N2$
(8) Output the combination of any item in $N1$ and the union of α;
(9) Output the union between the connection of the combination item of $H1$ and the combination item of $H2$ and α;
(10) }
(11)}
(12) Else if (key items are included in α)
(13) Output the union of each frequent item and α in the tree header table;//output the frequent item set
(14) Else
(15) { //All items in the key item header table are key items, and all items in the non-key item header table are non-key items
(16) The header table of tree is divided into key item header table $N1$ and non-key item header table $N2$;
(17) Output the union of frequent items and α in $H1$;//only output frequent itemsets containing key items
(18)}

cognition and metacognitive abilities, etc., as shown in Figure 2. It is the precondition to realize self-adaptive learning. Applying appropriate technology to build a personality trait model will help ALS provide accurate and personalized learning services.

3.3. Improvement of FP-Tree Algorithm

3.3.1. KEFP-Tree Algorithm. FP-tree algorithm mining will generate a large number of frequent itemsets, but although some frequent itemsets satisfy the minimum support, it affects the user's analysis and judgment and reduces the efficiency of mining. The purpose of this paper is to discover the relationship between learner behavior and learner characteristics in distance music classrooms and to determine learner characteristics based on some learning behaviors of learners, that is, to find out "learner behavior => learner characteristics" such as association rules. A meaningful frequent itemset should contain both learner behavior attribute items and learning strategy attribute items. When mining frequent itemsets, frequent itemsets that only contain learner feature items or only learning strategy items can be ignored [17]. The specific implementation steps are shown in Table 2.

Table 3: Database projection algorithm.

Input: transaction database D; minimum support threshold
Output: complete set of frequent patterns
Algorithm:
(1) Scan the D for the first time, get the items that satisfy the minimum support degree and arrange them in descending order, get the set L of candidate 1-itemsets, delete the items whose support degree is less than the minimum support degree in L, and get the set L of frequent 1-itemsets.
(2) Scan the source data for the second time, and rearrange the items in each transaction according to the support counts of the frequent items contained in each transaction, and get the database as D'.
(3) According to the support count of the items in the frequent 1-item set L, build the database subsets of items from small to large according to the following rules, and use the FP − growth algorithm to mine frequent items: for each term in L $I_i\,(i = m, m-1, \ldots, 1)$
(a) Scan the database D', extract all transactions containing item I_i from it, then delete the items whose support degree is less than the support degree of this item in these transactions, and the resulting transaction set is the database subset D_i of item I_i.
(b) For the database subset D_i, use the FP_growth (Tree, α) algorithm to mine frequent itemsets containing item I_i. The mining process is as follows:
(i) If tree contains a single path P then
(ii) For each combination of nodes in path P (denoted as β)
(iii) Generate the pattern $\beta \cup \alpha$, whose support degree support = support $(\beta)_{\text{minimum}}$;
(iv) Else for each α_i at the head of tree {
(v) Generate a pattern $\beta = \alpha_i \cup \alpha$ with support = αi.support ;
(vi) Construct the conditional pattern basis of β , and then construct the conditional FP-tree β of β;
(vii) If Tree$\beta \neq \varnothing$ then
(viii) Call FP_growth (Treeβ, β);}
(4) After the frequent items of all items in L are mined in turn, and these frequent itemsets are merged, all the frequent itemsets in the database D can be obtained.

3.3.2. Database Projection Algorithm. The performance study of the FP-tree algorithm shows that under normal circumstances, the FP-tree algorithm is about an order of magnitude faster than the Apriori algorithm. However, the FP-tree algorithm has a disadvantage, that is, it needs to occupy a lot of memory during tree building and mining. When the database is very large or the support is very small (the total number of generated 1-frequent itemsets is too large), it needs to be loaded into the memory. A large amount of data will cause the running speed to become very slow, and it is even impossible to build a memory-based FP-tree. A good solution is to first partition the database into subsets of the database and then construct an FP-tree on each database subset and mine it and finally combining these frequent itemsets can get the frequent itemsets of the entire database.

The database projection algorithm first scans the database for the first time and finds the 1-frequent itemsets that satisfy the minimum support degree. Each database subset uses the FP-growth algorithm to mine frequent items to obtain the frequent itemsets of each database subset. Finally, these frequent itemsets are combined to obtain the final frequent itemsets. The specific implementation process adopts the algorithm proposed in [18], and the specific algorithm is shown in Table 3.

3.3.3. Hybrid Algorithm. Combining the KEFP-growth algorithm with the database projection algorithm can not only eliminate the mining of meaningless frequent itemsets but also project the data when there is a large amount of data. The idea of combining the two improved algorithms is very simple, because the mining of FP-tree is realized through FP-growth, and the KEFP-growth algorithm is based on FP-growth plus keyword judgment, so only the projection algorithm needs to be used. By replacing the FP-growth algorithm with the KEFP-growth algorithm, the two algorithms can be combined. In addition, before dividing the database into subsets, the 1-frequent itemsets obtained from the first scan of the database should be counted [18]. If the value is greater than the given 1-frequent itemsets total number threshold, the database needs to be divided into subsets. The specific implementation process is shown in Table 4.

4. Simulation Results

4.1. Experimental Data. In order to collect real data on the characteristics of learners in modern distance music education classrooms, this study selected learners who participated in H University's online music elective courses as the research objects. H University's online music elective course conforms to the general characteristics of modern distance education teaching and can reflect the characteristics of learners in the modern distance environment to a certain extent.

In this study, the characteristics of learners in a modern remote environment are analyzed with the help of the online compulsory course "Music History" of H University's online learning on the Chaoxing Xuetong platform. The distance music learning course consists of three parts: courses, resources, and micro-applications. The course module includes multiple learning sections, and the course teaching form covered by each learning week includes three steps: learning guidance, learning content, and learning activities, that is, complete teaching, learning, practice, and examination. The main learning methods of distance learners are the combination of sight, hearing, and practice, and they can

Table 4: Hybrid algorithm.

Input: transaction database D; minimum support threshold; 1—threshold of total items of frequent items
Output: complete set of frequent patterns
(1) Scan the D for the first time, get the items that satisfy the minimum support degree and arrange them in descending order, get the set L of candidate 1-itemsets, delete the items whose support degree is less than the minimum support degree in L, and get the set L of frequent 1-itemsets, and the value C is obtained by counting the set L at the same time.
(2) If ($C > 1$-threshold of the total number of items of frequent items)
{
(i) Scan the source data for the second time, rearrange the frequent items in each transaction according to the support count of each item, and rearrange the items in each transaction, and get the database as D'
(ii) According to the support count of the items in the frequent 1-item set L, build the database subsets of items from small to large according to the following rules, and use the FP-growth algorithm to mine frequent items:
For each term in L $I_i\,(i = m, m-1, \ldots, 1)$
(a) Scan the database D', extract all transactions containing item iii from it, then delete the items whose support degree is less than the support degree of this item in these transactions, and the resulting transaction set is the database subset di of item Ii.
(b) For database subset Di, use KEFP_growth (tree, α) algorithm to mine frequent itemsets containing item Ii
(iii) After the frequent items of all items in L are excavated in turn, these frequent itemsets are merged to get all frequent itemsets in database D
}
Else
{
Create the root node of the FP-tree, marking it with "null." For each transaction trans in D, do: select the frequent items in trans and sort by the order in L. Let the sorted list of frequent items be $[p|P]$,
Where p is the first element and P is the list of remaining elements. Call insert_tree $([p|P], T)$. The process is performed as follows. If T has child N such that N.item − name = p.item − nam, then N's increment the count by 1; otherwise create a new node N, set its count to 1, link to its parent node T, and link it to a node with the same item-name through a node chain structure. If $P \neq \varnothing$, insert_tree (P, N) is called recursively.
(3) Mining FP-tree by calling KEFP_growth (Tree, α)
}

interact and ask questions according to their learning needs. The course has a total of 20 study sections. After the course is over, the academic performance of distance learners will be assessed according to a percentile system, which consists of two parts: the formative assessment and the final assessment. Formative grades refer to the usual grades, which are mainly composed of the learners' online answers and the interaction of the number of posts, accounting for 30%; the final examination assessment refers to the learners' final grades, which are mainly obtained from the learners' final assessment results. According to an online test, it accounts for 70%.

4.2. Experimental Design

(1) Data source is Superstar learning platform background data; number of project transactions is 20000; dataset is 60 (1.3M); algorithms are FP-tree algorithm and KEFP-tree algorithm; results are running time and total number of frequent itemsets.

(2) Using the background data of the Superstar Learning Platform, select a dataset (2.2 M) with 40,000 itemset transactions and 60 different items, using the original FP-tree algorithm and database projection algorithm under different minimum support degrees. The dataset is mined and the time of its run is recorded separately.

(3) Using the background data of the Superstar Learning Platform, select the dataset (1.8 M) with 30,000 itemset transactions and 60 different items (1.8 M), and use the KEFP-growth algorithm, data projection algorithm, and combination algorithm to achieve different minimum support degrees. The dataset is mined below, and its running time is recorded separately.

4.3. Experimental Results. The experimental results of the original FP-tree algorithm mining on the 1.3 M dataset and the 2.2 M dataset are shown in Table 5:

When the minimum support degree is high, that is, when the number of generated 1-frequent itemsets is small, the original FP-tree algorithm runs fast and has high mining efficiency. However, when the minimum support is low, the running speed will be greatly reduced due to the large number of 1-frequent itemsets generated. The number of sets is very large, which will cause the running speed to be greatly reduced and even the situation that the memory-based FP-tree cannot be built.

The experimental comparison results of the FP-tree algorithm and the KEFP-growth algorithm mining on the 1.3 M dataset are shown in Table 6.

It can be seen from the table that under different support degrees, the KEFP-growth algorithm reduces the number of frequent items output by the original FP-tree algorithm by about 50%, and the mining time is reduced by 50%. This is because the output of irrelevant frequent itemsets can be reduced by using the KEFP-growth algorithm, thereby shortening the time spent in selecting useful items from

Table 5: The experimental results of the original FP-tree algorithm.

Support (%)	1.3 M		2.2 M	
	Number of frequent itemsets	Operation hours	Number of frequent itemsets	Operation hours
50	34	10	57	35
40	55	20	124	64
30	309	55	880	145
20	1350	255	2842	588
10	5599	808	9815	1535
1	13442	6741	25467	—

Table 6: The experimental comparison results of the FP-tree algorithm and the KEFP-growth algorithm.

Support (%)	Number of frequent itemsets		Operation hours	
	FP-tree algorithm	KEFP-growth algorithm	FP-tree algorithm	KEFP-growth algorithm
50	34	19	35	9
40	55	31	64	35
30	309	157	145	77
20	1350	679	588	259
10	5599	2562	1535	1006
1	13442	6458	—	—

Table 7: The experimental comparison results of the original FP-tree algorithm and the database projection algorithm.

Support (%)	Number of frequent itemsets	Operation hours	
	FP-tree algorithm	FP-tree algorithm	Database projection algorithm
50	57	35	63
40	124	64	89
30	880	145	163
20	2842	588	574
10	9815	1535	1366
1	25467	—	5786

useless frequent items, significantly improving the efficiency of mining, and meeting the original intention of mining.

The experimental comparison results of the original FP-tree algorithm and the database projection algorithm on the 2.2 M dataset are shown in Table 7.

Experiments show that when the minimum support is reduced to a certain threshold or the amount of data to be mined is greater than a certain level (the total number of items in the frequent 1-item set is too large), the original FP-tree algorithm cannot build a memory-based FP-tree, which will cause the excavation to fail. The data projection algorithm can divide the frequent 1-itemsets of a large database into several database subsets and then mine the database subsets, respectively. Because each database subset occupies a small memory, it can overcome the data of 1-itemsets. The problem that the amount is too large makes the memory unable to load the FP-tree, so that the data mining can proceed smoothly. In addition, experiments also show that the data projection algorithm is more suitable for data mining with less support. When the support is 10%, the mining time of the data projection algorithm is reduced by 80% compared with the FP-tree algorithm. When the minimum support is large or the amount of data to be mined is small (that is, the total number of frequent 1-itemsets is small), the running speed of the new algorithm is not as fast as that of the original FP-tree algorithm. The reason for this phenomenon is that the cost of the original FP-tree algorithm lies in tree building and frequent itemset mining, while the data projection algorithm has the cost of dividing the database subset in addition to the cost of tree building and frequent itemset mining. When the database support degree is large, the total number of 1-frequent itemsets generated by the database is small, so the two algorithms have little difference in the time cost of building trees and mining frequent itemsets, but the database projection algorithm divides the database subsets. There is a time overhead in the aspect of FP-tree, so the mining speed of the original FP-tree algorithm is faster than that of the database projection algorithm. With the increase of the minimum support, the memory and time overhead of the original FP-tree algorithm increase, resulting in slower and slower mining speed, while the database projection algorithm also increases the cost of generating database subsets. However, the memory overhead and time overhead of building a database and mining frequent itemsets are less, so the overall running speed is faster than the original FP-tree algorithm.

The experimental comparison results of the three improved algorithms' mining on the 1.8 M dataset are shown in Table 8.

Table 8: The experimental comparison results of the three improved algorithms.

Support (%)	Number of frequent itemsets	Operation hours		
		KEFP-growth algorithm	Database projection algorithm	Hybrid algorithm
50	46	12	38	15
40	90	27	65	29
30	545	64	90	65
20	2351	167	157	146
10	7648	998	985	833
1	1987	—	5786	3756

Experiments show that the mining efficiency of the hybrid algorithm and the FEFP-growth algorithm is not much different when the minimum support is large, but when the minimum support is small to a certain extent, the mining efficiency of the hybrid algorithm is higher than that of the KEFP-growth algorithm. When the support degree is 10%, the running time of the hybrid algorithm is reduced by 10% compared with the KEFP-growth algorithm and the data projection algorithm. When the support degree is large, the KEFP-growth algorithm branch is combined with the algorithm, and when the support degree is small (the total number of 1-frequent itemsets generated is large), the database projection algorithm combined with the KEFP-growth algorithm is used. In addition, it can be seen from the experimental results that the hybrid algorithm is better than the database projection algorithm when the support is small or large. This is because the KEFP-growth algorithm used in the hybrid algorithm removes many irrelevant frequent itemsets, thereby reducing the amount of excavation. Comprehensive analysis shows that the hybrid algorithm is better than the first two improved algorithms.

5. Conclusion

This paper makes in-depth research on data mining, especially association rule mining, improves the FP-tree algorithm in both the algorithm itself and the data source, and finds out a mining algorithm suitable for learner characteristics. By establishing the characteristic model of learners in modern distance music classroom, simulation experiments are carried out on FP-tree and three improved algorithms.

Compared with the original FP-tree algorithm, the number of frequent items output by the KEFP-growth algorithm is much less, and the mining time is also significantly reduced.

(1) The KEFP-growth algorithm reduces the number of frequent items output by the original FP-tree algorithm by about 50%, and the mining time is reduced by 50%.

(2) The data projection algorithm is more suitable for data mining with less support. When the support is 10%, the mining time of the data projection algorithm is reduced by 80% compared with the FP-tree algorithm.

The mining efficiency of the hybrid algorithm and the KEFP-growth algorithm is not much different when the minimum support is large. However, when the minimum support is small to a certain extent, the mining efficiency of the hybrid algorithm is higher than that of the KEFP-growth algorithm. When the support degree is 10%, the running time of the hybrid algorithm is reduced by 10% compared with the KEFP-growth algorithm and the data projection algorithm.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] E. Wenger, *Artificial Intelligence and Tutoring Systems*, 486 pages, Morgan Kaufmann Publishers Inc, San Francisco CA USA, 1987.

[2] K. Vanlehn, "Student modeling," in *Foundations of Intelligent Tutoring Systems,*, M. Polson and J. Richardson, Eds., pp. 55–78, Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.

[3] C. J. Asarta and J. R. Schmidt, "Access patterns of online materials in a blended course," *Decision Sciences Journal of Innovative Education*, vol. 11, no. 1, pp. 107–123, 2013.

[4] P. Holt, S. Dubs, M. Jones, and J. Greer, "The state of student modeling," in *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, J. E. Greer and G. I. Mc Calla, Eds., 38 pages, Springer-Verlag, Berlin/Heidelberg Germany, 1994.

[5] D. D. Prior, J. Mazanov, D. Meacheam, G. Heaslip, and J. Hanson, "Attitude digital literacy and self efficacy: flow-on effects for online learning behavior," *The Internet and Higher Education*, vol. 29, pp. 91–97, 2016.

[6] E. Gaudioso, M. Montero, and F. H. D. Olmo, "Supporting teachers in adaptive educational systems through predictive models: a proof of concept:A Proof of Concept," *Expert Systems with Applications*, vol. 39, no. 1, pp. 621–625, 2012.

[7] Educause, "Key Issues in Teaching and Learning," 2016, http://www.educause.edu/eli/initiatives/key -issues -in -teach-ing-and-learning.

[8] G. Goguadze, S. A. Sosnovsky, S. Isotani, and B McLaren, "Evaluating a bayesian student model of decimal misconceptions," in *Proceedings of the 4th International Conference on Educational Data Mining*, pp. 31–40, Eindhoven, The Netherlands, June 2011.

[9] Z. Zhu and D. Shen, "A new paradigm of educational technology research based on big data," *Research on Electronic Education study*, vol. 15, no. 1, pp. 21–25, 2012.

[10] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large database," in *Proceedings of the ACM SIGMOD Conference On Management of Data*, Washington, D.C., USA, May 1993.

[11] M. Guojun, *Research on Data Mining Technology and Association Rules Mining Algorithm*, Tsinghua University Press, Beijing, China, 2003.

[12] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate generation," in *Proceedings of the 2000 ACM—SIGMOD Int Conf Management of Data F SIGMOD 2000)*, ACM Press, Dellas, TX, USA, May 2000.

[13] X. Liu, "A new algorithm for mining association rules that does not require candidate sets—Relim algorithm research," *Computing Technology and Automation*, vol. 12, pp. 162–168, 2006.

[14] D. T. Tempelaar, B. Rienties, and B. Giesbers, "In search for the most informative data for feedback generation: learning analytics in a data-rich context," *Computers in Human Behavior*, vol. 47, pp. 157–167, 2015.

[15] A. S. Hanbidge, T. Tin, and N. Sanderson, "Student Learner Characteristics and Adoption of M-Learning: Are We Effectively Supporting Students?" *Mobile and Ubiquitous Learning*, Vol. 8, Springer, Berlin, Germany, 2018.

[16] N. Jaques, C. Conati, J. M. Harley, and R. Azevedo, "Predicting affect from gaze data during interaction with an intelligent tutoring System,Martha crosby," in *Proceedings of the 12th International Conference,Honolulu Intelligent Tutoring Systems*, pp. 29–38, Honolulu, HI, USA, June2014.

[17] S. S. Liaw and H. M. Huang, "Perceived Satisfaction Perceived Usefulness and Interactive Learning Environments as Predictors to Self-Regulation in E - Learning Environments," *Computers & Education*, vol. 60, no. 1, pp. 14–24, 2013.

[18] S. Alkhuraiji and B. Cheetham, O. Bamasak, Dynamic Adaptive Mechanism in Learning Management System Based on Learning Styles," in *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies (ICALT)*, pp. 215–217, IEEE, Athens, GA, USA, July2011.