INTRODUCTION

With progress of times, the comprehensiveness of knowledge has been enhanced and students are required to master knowledge better by the society accordingly. As a source of knowledge for teachers and students, school library is playing a decisive function. College teachers’ and students’ directions of study and research can be directly influenced by category and layout of library collections. With new purchase of books every year, a library will have its collections increased continually. It is very difficult for teachers and students to find what they need among numerous books. Therefore, reasonably adjusting library book collections and their layout to promptly meet college teachers’ and students’ needs and make book borrowing more convenient is of great significance.

As a new technology of demand analysis, data mining can well analyze reader’s demand. Firstly, data mining establishes structure of collected books according to reader’s behavioral characteristics and reading habits. It analyzes and finds advantages and disadvantages of collection structure. After subdividing reader groups, data mining then conducts specific research and analysis activities. It finds the internal mode of reader’s demand from massive data; tracks, re-tracks, and re-predicts a mass of readers and their reading behaviors. Lastly, data mining assesses the effect of collection structure so as to make corresponding adjustment.

As book circulation data is updated each year. With history data becoming more and more massive, related history circulation data association rule mining and update of association rules among books need to be continually proceeded. Repeated mining of old circulation data certainly will turn into a significant burden for data mining work. Hence, this thesis aims to improve Apriori mining algorithm and thus to reduce data mining workload which is an important method to improve mining efficiency.

RELATED WORK

There are many mining algorithms for association
rules, such as Apriori algorithm, FP-Growth algorithm\cite{11}, Max. Miner algorithm \cite{9}, and MFP-Miner algorithm among which Apriori algorithm is the most well-known and the most classic one. Apriori algorithm is also the most fundamental algorithm for association rule mining. It has left great influence on association rule mining research since its emergence.

Traditional Apriori algorithm realizes mining of frequent item sets through iterative method of layer-by-layer search. Frequent item set mining is a key part of association rule mining and sequence pattern mining. Since Dr. Agrawal et al. from IBM proposed Apriori algorithm in 1993, many researchers have conducted a number of studies of frequent item set mining \cite{3, 4, 7}. In Apriori algorithm, the whole transactional database needs to be scanned during process of finding each frequent item set. While doing pattern matching check on candidate item sets, efficiency of the algorithm will be much lower if the database content is large or the pattern needs to be matched is very long. Besides, Apriori algorithm needs to generate a large number of candidate item sets during computation. In this way, the number of candidate item sets which need to be generated will become much bigger if there are many frequent item sets or the frequency pattern is long. As a result, efficiency of the algorithm can be dramatically decreased \cite{8}.

The FP-Growth algorithm proposed by Han JiaWei et al. can compress and store frequent items contained in each transactional database into FP-Tree \cite{9} with a descending order of support degree by scanning transactional databases twice. During the process of searching frequent item sets later, no transactional database needs to be scanned again and the item sets can be found within FP-Tree. Moreover, no candidate item set needs to be generated during the whole searching process as recursive call is applied to FP-Growth algorithm and thus can directly generate frequent item sets \cite{10}. Compared with Apriori algorithm, FP-Growth algorithm has made much improvement on execution efficiency.

However, information technology is in rapid development while database content is in dramatic expansion today, either Apriori algorithm or FP-Growth algorithm can be widely used to face huge databases. Therefore, we must improve them or find a new and more effective mining method \cite{11}.

Searching maximal frequent item sets is an important research direction in association rule mining. Nevertheless, no sufficient research has been completed in this area. Pincer-Search algorithm proposed by Lin et al. \cite{11}, Max-Mille algorithm proposed by Bayardo et al. \cite{13} and Mafia algorithm proposed by Burdick et al. \cite{14} are the main algorithms overseas.

In this study, we will apply MFP-Miner algorithm to conduct data mining on library collection association rules according to characteristics of circulation data.

3 AN IMPROVED MFP-MINER MINING ALGORITHM

In practical application, we have found that databases need to be mined become continuously bigger with significant increase of circulation records in history and adjustable minimal support degree. While database changes, minimal support degree may also need to be adjusted accordingly so as to find new association rules. It will cost lots of time and information previously found will be wasted if we use algorithms for maximal frequent item set mining (such as MFP-Miner and Mafia) to re-mine data.

In order to improve mining efficiency, it is in sore need of exploring a new algorithm to realize synthesized update mining to satisfy the requirement. According to the need and based on MFP-Miner algorithm, we will present preliminary discussion on how to realize synthesized update mining and propose IM-Miner algorithm which can be used for synthesized update mining of maximal frequent item sets.

3.1 Definition and hypothesis

Set database before change as D and after change as D’. Set increased or reduced data set as d. IM_{D} and IM_{d} refer to collection of maximal frequent item sets under D and d. minSUP and minSUP’ refer to previous and new minimal support degrees respectively. Set Q_{D} and IM_{D} as frequent item list and maximal frequent item set collection corresponding to D and minSUP. Set Q_{d} and IM_{d} as frequent item list and maximal frequent item set collection corresponding to D’ and minSUP’.

Character 1: If an item set is non-frequent in both D and d when minimal support degree remains the same, it must be non-frequent in D ∪ d.

Character 2: If an item set is frequent in both D and d when minimal support degree remains the same, it must be frequent in D ∪ d.

Character 3: Set minimal support degree in constant status, frequent item (or item set) in D or d may turn into non-frequent item (or item set) in D ∪ d while non-frequent item (or item set) in D or d may turn into frequent item (or item set) in D ∪ d.

Character 4: When minimal support degree remains the same, maximal frequent item set contained in updated database D ∪ d must be subset of some maximal frequent item set in IM_{D} or IM_{d}; or may be subset of some maximal frequent item set in IM_{D} and IM_{d} at the same time.

3.2 IM-Miner and its description

From the above characters, we can see that it is almost impossible to manage transition from FP-Tree T in D to FP-Tree T’ in Dt when database is in change. In order to realize continuous update mining, we recon-
struct $T'$ in $D'$ by which means we only need to adjust $IM_D$ without any adjustment on FP-Tree.

Therefore, the specific steps of conducting synthesized update mining on maximal frequent item set are as follows: 1) reconstruct $T'$ in $D'$ according to minSUP'; 2) adjust $IM_D$; 3) update mining of maximal frequent item set of minSUP' in $T'$ after adjusting $IM_D$.

Step 1: Reconstruct $T'$. Scan $D'$ according to the minimal support degree minSUP' after change. Find the frequent item set $Q_d$ which can satisfy the requirement of minSUP', and sort it by descending order of support degree to construct FP-Tree.

Step 2: Adjust $IM_D$. Delete all the non-frequent items which belong to minSUP' in $Q_d$ from each item set of $IM_D$. Then, delete redundant item sets generated as a result from MFSD. Sort all items of each item set in $IM_D$ according to the sequence in $Q_d$.

Step 3: Search maximal frequent item sets with suffix of each item from $Q_d$ ($Q_d=Q_p-Q_d$ is that is collection formed by frequent item sets newly generated by $D'$ in minSUP'). Then, apply the rule—“If the count of a node is smaller than the minimal support degree in FP-Tree, the pattern (item set) formed by the node and nodes in the path with prefix of the node must be a frequent pattern”.

Construct FP-Tree by arranging each item of item set $m \in IM_d$ via the structure condition given below, and find the new maximal frequent item set with certain suffix.

$$c = \{m_{ij} | (k = i = j) & (nodecount < |D| \times \text{min\,SUP'}) & \& (nodecount = |D| \times \text{min\,SUP'})\}$$

4 TESTING AND DISCUSSION

To verify the feasibility of the algorithm proposed in this thesis, we have realized the algorithm by editing codes. The running environment for this algorithm is: dual-path quad-core 2.4GHz CPU, 4G memory, and 1TB hard disk. Mysql database is used to preprocess data. Mining algorithm of association rules can be realized in VS2008.

4.1 Book-Crossing Dataset

We compared the efficiency of IM-Miner algorithm with those of the most commonly used Mafia algorithm and FP-Growth algorithm in Book-Crossing Dataset. One of the characteristics of Book-Crossing Dataset is its maximal frequent item sets are in comparatively symmetrical distribution and most maximal frequent item sets have low dimensions. With an average length of about 10, data of Chess database shares similar features with circulation data of library. See Figure 1 for the experimental results.

![Figure 1. Comparison of IM-Miner algorithm, FP-Growth algorithm and Mafia.](image)

From the above figures, we can see that, compared with Mafia algorithm, IM-Miner algorithm has much higher efficiency. When minimal support degree is higher than 20%, execution efficiency of MFP-Miner can be twice or three times higher than that of Mafia algorithm. However, performance of IM-Miner algorithm gets declined when minimal support degree is lower than 50% and has a higher declining rate when minimal support degree is lower than 30%. The cause for this phenomenon is that IM-Miner algorithm uses FP-Tree to compress and store transaction in database. Data mining of IM-Miner algorithm takes full advantages of FP-Tree and uses it as the basis. It doesn’t need to generate candidate item set during mining. As a result, the mining process can receive higher efficiency. By the same token, execution efficiency of IM-Miner algorithm can be twice of that of FP-Growth algorithm, showing that IM-Miner algorithm can bring higher efficiency of association rule mining to circulation data.

4.2 IM-Miner Dataset performance testing

To test the performance of IM-Miner algorithm, we selected T1014D100k as the testing database which has larger data content. By using this database, we are able to test the performance of IM-Miner algorithm in databases with different distribution features.

From Figure 2, we can see that IM-Miner algorithm is more efficient in synthesized update than Mafia algorithm as the latter one needs to re-mine data. We can see that the synthesized updating efficiency of IM-Miner algorithm in T1014D100k database is twice or three times higher than that of FP-Growth algorithm.

According to the testing results of IM-Miner algorithm in different databases, it is easy for us to find IM-Miner algorithm has improved running efficiency when original MFS is not too big. Moreover, as
IM-Miner algorithm contains great advantage in mining high-dimensional maximal frequent item sets, it also has higher efficiency in conducting update mining on high-dimensional maximal frequent item sets.

Figure 2. Testing comparison in T10I4D100k database.

5 SUMMARY

Based on practical problems, this thesis establishes a rapid mining method of data association rules for library circulation data. It can execute mining through the association rules of circulation data stored in internal storage when database is too big and can also find relations among the data. IM-Miner algorithm is an algorithm improvement based on the difficulty of data mining when database is in continuous increase. By taking the maximal frequent item sets mined by MFP-Miner algorithm as the basis, we only need to scan newly increased data and search MFS according to the latest minimal support degree. In this way, overall mining efficiency can be significantly improved.

In future research, there are two directions for further exploration.

How to improve efficiency of association rule mining in large-scale databases is an important research direction in the future. In general, efficiency refers to run time and result usability. Existing algorithm rules are too big in quantity, making it very inconvenient for users to apply.

How to present mining results in a friendly way to end users is also very important. Currently, most algorithms sort results by certain support degrees (e.g. support degree, trust degree) for users to browse. Some others visualize rules in graphs for users. However, as quantity of rules in large-scale databases is normally very big, all the above methods have certain limitations. How to post process mined rules and submit them to users in a visualized way is another direction for future research.

REFERENCES