MATEC Web of Conferences **22**, 01060 (2015)
DOI: 10.1051/matecconf/20152201060

Research of the Occupational Psychological Impact Factors Based on the Frequent Item Mining of the Transactional Database

Dongmei Cheng
The Office of Moral Education Teaching-research, Department of Public Elementary Courses, Langfang Health Vocational College, Langfang, Hebei, China

Xuejun Zuo & Zhaohua Liu
Langfang Health Vocational College, Langfang, Hebei, China

ABSTRACT: Based on the massive reading of data mining and association rules mining documents, this paper will start from compressing transactional database and propose the frequent complementary item storage structure of the transactional database. According to the previous analysis, this paper will also study the association rules mining algorithm based on the frequent complementary item storage structure of the transactional database. At last, this paper will apply this mining algorithm in the test results analysis module of team psychological health assessment system, and will extract the relationship between each psychological impact factor, so as to provide certain guidance for psychologists in their mental illness treatment.

Keywords: transactional database; frequent item mining; psychological impact factor analysis

1 INTRODUCTION

There will never be an end for data structure optimization and algorithm efficiency improvement. To a certain degree, the frequent complementary storage structure can compress the transactional data structure. However, when the scale of the former item sets becomes large in the later period of mining algorithm, the transactions which can support the current item sets will become less and cause a decline in the data storage of the supporting transaction list with the same length. Moreover, the application of more pruning methods can reduce unnecessary calculation and help control the performance period. Therefore, this paper will do a research on how to further compress the data structure and improve the algorithm efficiency.

2 CLASSICAL ALGORITHM OF ASSOCIATION RULES MINING

2.1 *Apriori algorithm*

The Apriori algorithm is a classical algorithm of association rules mining. Its main idea is to apply the support degree and the confidence coefficient between the frequent item sets to conduct effective pruning on the support degree and the confidence coefficient, so as to greatly reduce the computing cost. According to the antagonistic nature of the theorem, it can be concluded that if one item set is non-frequent, its all supersets must be non-frequent accordingly. The Apriori algorithm applies this nature to do effective pruning to the search space and systematically controls the exponential growth of the candidate item sets. This algorithm uses serial layer scanning method. It firstly scans the item sets with only one item, and then it scans the item sets with two items. The rest can be done in the same manner. During the scanning, if one item set is not a frequent item set which means its support degree cannot meet the given value, its supersets are all non-frequent item sets. As a result, pruning can be conducted to all of its supersets in order to effectively reduce the calculating quantity.

In the generation process of the frequent item sets in the Apriori algorithm, q refers to the candidate k-item sets and f refers to the frequent k-item sets. There're two important features in the generation part of the frequent item sets in the Apriori algorithm: first, it is a layer-by-layer (breadth first) algorithm which means that it traverses a layer of each item set during the process from the frequent 1-item sets to the longest frequent item set. Secondly, it applies the Generate-and-Test strategy to find the frequent item sets. In each iteration, the new candidate item sets generate from the frequent item sets found in the previous iteration. And then, the algorithm counts the support degree of each candidate item set and compares it with the minimum support degree. The Apriori algorithm applies this feature to manage pruning to the association rules. The algorithm uses a layer-by-layer method to generate association rules of which each layer corresponds with the item number of the rule consequent. In the beginning, the extracted rule consequent only includes all high confidence coefficient rules of one item. And then, these rules will be used to generate new candidate rules. During the generation process of rules, if one rule fails to reach the closed value of the

Algorithm	The Generation of the Frequent Item Sets of the Apriori Algorithm
1: k=1	
2: Fk={i/i∈f∧({i})≥NxminsuP}	Find all frequent 1-item sets.
3: repeat	
4: k=k+1	
5: C_k=apriori-gen(F_k-1)	
6: for each transaction t∈T do	
7: C^1=subset(Ck_1t) {Identify all candidate items belonging to t }	
8: For each candidate item set C∈C_1 do	
9: end for	
10: end foe	
11: untilF_k==∉	
12: Result=Ufl	

Figure 1. The Generation of the Frequent Item Sets of the Aprioro Algorithm

confidence coefficient, all the rules of which the consequents are the consequent supersets of this rule cannot reach the threshold value of the confidence coefficient by any means. In consequence, pruning can be managed to these rules, so as to effectively reduce the computing cost.

Generation of the Rules in the Apriori Algorithm
1: for each frequent K-item set f_k,k≥2 do
2: H_1={i/i∈f_k} the 1-items consequent of the rule
3: call ap=genrules(fkHl)
4: end for

Figure 2. Generation of the Rules in the Apriori Algorithm

Process of Ap-genrules(fk,H)
1: k=/fk/
2: m=/Hm/
3: if k＞m+1 then
4: Hm+1=apriori=gen(Hm)
5: for each hm+1∈Hx+1 do
6: Conf=(fk)/o(fk-hm+1)
7: If conf≥minconf then
8: Output: rule(fk-hm+1)-hm+1
9: Else
10: From Hm+1delete hm+1
11: End if

Figure 3. Process of Ap-genrules

Although the Candidate Generate-and-Test method of the Apriori algorithm can reduce the quantity of the candidate frequent item sets to some extent and bring good performance, the three kinds of cost coming from the algorithm are definitely not insignificant. The three kinds of cost are as follows:

(1) The possibility of massive candidate item set generation. When there're 10000 frequent item sets with length of 1, the quantity of the candidate item sets with length of 2 generated by the Apiori algorithm can be 10M. If a long rule needs to be generated, the number of the middle elements going to be generated will be huge.

(2) Failure to make analysis of rare information. As minsup is applied in the frequent item sets, no analysis can be made to the events which are smaller than minsup. If the minsup is set as a low value, algorithm efficiency will be a problem difficult to deal with.

(3) Possibility of repeated database scanning. The algorithm may need to scan the databases repeatedly while checking a huge candidate item set by pattern matching. It is especially true in the mining of long-mode item sets.

2.2 *SETM algorithm*

The SETM algorithm appeared in 1993. In fact, it is a variant of the Apriori Algorithm. The original target of SETM was to conduct JOIN operation by standard SQL language. It can separate the generation and the accumulation of the candidate sets. All candidate sets and the whole generation process of the frequent item sets with corresponding transaction serial numbers are stored in a self-defined linear memory structure. The generation of candidate item sets, counting of support degree and the pruning on non-frequent item sets of the SETM algorithm can all be completed by standard SQL operation. See the figure given below for the SETM algorithm in which Ck refers to candidate item sets and FIS belong to frequent item sets.

Description of the SETMAlgorithm
1: Sort D by Itom
2: Divide C1 into groups by Item, and calculate the frequent set FIS1.
3: Manage ascending sort to FIS1 according to Tid
4: foe K=2 La-1≠0 do
5: Connect FISk-1and calculate Ck
6: Calculate the assembly of C1 by Item to get the frequent set FISK
7: Manage ascending sort to FISk according to Tid
8: FIS=FIS-FSIk
9: End for

Figure 4. Description of the SETM Algorithm

Establishment of the Frequent Complementary Storage Structure of the Transactional Database
1: for each item i∈I do 2: if s({i})is equal or greater than minsuP, then 3: Add item I and its TdiList to Array A 4: end if 5: end for 6: for=0to Asize-1 do 7: form=1 to A size do 8: if s({A[l],A[m]})≥minsuP, then 9: Add A[m] to the Frecomiten List of A[l] 10: End if 11: End for 12: End for

Figure 5. Establishment of the Frequent Complementary Storage Structure of the Transactional Database

Layer-by-layer Iteration Frequent Item Set Mining Algorithm
1: forl=0 toAsize-1 do 2: Add 1item of frequent item set A[l] to the frequent item set FIS 3: end for 4: forl=0 to Asize-2 do 5: form=0 to A[l]FreComltemList size-1 do 6: Add 2 items of frequent item set {A[l],A{l}.FreComltemList{m} to FIS 7: Add {A[l],A{l}.FreComltemList{m} to KFIS 8: end of 9: end for 10: K=2 11: repeat 12: for each K item of the frequent item set X of KFIS do 13: for each frequent complementary item f of the last item I of the item set i do 14: if the item f is the frequent complementary item in the other items of the item set X then 15: if S(X{f})≥minsuP then 16: Add(X{f})to FIS 17: Add(X{f})to KFIS 18: end if 19: end if 20 end if 21: Delete all the K frequent item sets of KFIS 22: K=K+1

Figure 6. Layer-by-layer Iteration Frequent Item Set Mining Algorithm

3 RESEARCH OF THE ASSOCIATION RULES MINING BASED ON FREQUENT COMPLEMENTARY ITEMS

3.1 *The Frequent Complementary Item Storage Structure of Transactional Database*

Set the minimum support degree count minsup is 6 and the steps for building its frequent complementary storage structure is as follows:

Find the 1-item frequent item sets. Manage descending sort to the 1-item sets of which all support degree counts are equal to or greater than the minimum support degree count 6 by support degree count in the transactional database and store the order in the array. The array elements are {D,B,A,E,c,G}. For each array element, store the integer corresponding to its support transaction list in the meantime. The support transaction lists corresponding to the items {D,B,A,E,c,G} should be {60285,53949,41595,16172,53429,23856} respectively. Search the frequent complementary item of each item in the array. For each item of the array {D,B,A,E,c,G}, calculate its complementary item respectively. Firstly, start from item {D} because the integers corresponding to the support transaction lists of items {D,B} are $(60285)_{10}$ & & $(53949)_{l0}=(49725)_{10}=(1100001000111101)_{2.}$ So, s({D,B})=8>6 and {D, B}is a 2-item frequent item set. As a result, B is the frequent complementary item of item D. Similarly, items A, E and C are the frequent complementary items of item D while items A and C are the frequent complementary items of item B. Generally, for a transactional database of vertical data structure which is expressed in binary form through the conversion from the original transactional database, the algorithm to build the frequent complementary

item storage structure of its transactional database is as follows. The algorithm lists the process of obtaining the frequent complementary item storage structure from the original transactional database.

3.2 *Frequent Item Set Mining*

For a given transactional database, the association rules mining can extract the mutual relation and interrelation between each data, and present the relations in the form of interference rules. The current association rules mining are usually performed through the following two steps:

(1) Generation of frequent item sets: the task of this step is to generate all item sets which can meet the standard of the closed value of the minimum support degree. These item sets are called frequent item sets.

(2) Generation of rules: the task of this step is to extract all rules with high confidence coefficient from the frequent item sets generated in the previous step. Compared with the generation of frequent item sets, the generation of rules is simple and intuitive. In general, the computing cost required for the generation of frequent item sets is far more than that of the generation of rules. At present, the research of association rules mining mainly focuses on improving the efficiency of the frequent item sets generation. The work done in this paper is mainly about the first step of the association rules mining procedures which is the mining of frequent item sets. After storing the initial transactional database in the form of frequent complementary item storage structure, all 1-item frequent item sets and 2-item frequent item sets in the transactional database and their supporting transaction lists are saved. According to the priori principle, if an item set is a frequent one, its all subsets must be frequent ones. It means that the frequent item sets can only be obtained by 1-item frequent item sets. A frequent item set will never include any 1-item non-frequent item sets. Therefore, we respectively combine the frequent item sets of the frequent complementary storage structure and their supporting transaction lists; and extract all frequent item sets from them. The main idea of the layer-by-layer iteration frequent item set mining algorithm is to obtain the (k+1) –item candidate item sets by pruning on the basis of having obtained all k-item frequent item sets. And then calculate the supporting count of all (k+1)-item candidate item sets in order to get the (k+1)-item frequent item sets. As this algorithm is operated through the layer-by-layer calculation of the item set scale, we call this algorithm as the layer-by-layer iteration mining algorithm.

The receiving of the transactional database and the threshold of the minimum support degree are the inputs of the layer-by-layer iteration frequent item set mining algorithm. The algorithm will firstly scan the arrays and obtains the 1-item frequent item sets and the 2-item frequent item sets. And then, it will calculate the intersection of the frequent complementary items in each 2-item frequent item set; integrate the public frequent complementary items of each 2-item frequent item set and the 2-item frequent item sets to obtain the 3-item candidate item sets; and calculate the supporting count of the 3-item candidate item sets in order to judge whether they are 3-item frequent item sets. Similarly, suppose that all k-item frequent item sets have been obtained. Generate (k+1)-item candidate item sets based on the assumption firstly, and then calculate the supporting count of the (k+1)-item candidate item sets, so as to get the (k+1)-item frequent item sets. When the generated (k+1)-item frequent item sets are empty, the calculation should end. The algorithm will output all extracted frequent item sets. The procedures for the frequent item sets obtained during the storage mining process and the K-item frequent item sets obtained in KFIS (K-FSI) storage mining process with the mining steps of the layer-by-layer iteration frequent item sets are as follows. First, scan the arrays and add all primary colors into FIS. After the first step, FIS will include all 1-item frequent item sets. And then, scan the arrays; add the primary colors and the item sets which are formed by the corresponding frequent complementary items to FIS and KFIS; and set the value of k as 2. After this step, the FIS will include all 1-item frequent item sets and 2-item frequent item sets while the KFIS will include all 2-item frequent item sets in the mining process. If KFIS=$\varnothing$ exists, the algorithm should end. However, if these two are not equal to each other, then the KFIS needs to be scanned. For the k-item frequent item sets in KFIS, judge whether the final item i-frequent item sets of x belong to all frequent complementary item of item set x or not. If P belongs to x and j is not a p-frequent complementary item, x can be considered to be unfit for frequent item sets. Then the j step should be skipped and the frequent complementary items of i should be judged. If j belongs to all item frequent complementary items of x, it is possible that s≥minsup. If it is impossible to have s≥minsup, x doesn't belong to frequent item set. Then j should be skipped and the follow-up frequent complementary items of i should be judged. If s≥minsup exits, x belongs to frequent item set and the (k+1)-item frequent item set x can be added to FIS and KFIS. Delete the k-item frequent item sets k++ in KFIS and return to the previous step. The process of the layer-by-layer iteration frequent item set mining algorithm is shown in Figure 6.

4 APPLICATION OF ASSOCIATION MINING IN PSYCHOLOGICAL HEALTH ASSESSMENT

This part mainly analyzes the association rule mining problems with high degree of difficulty. After obtaining the initial transactional database by data pre-processing, the association rules mining of the initial transactional database can be started. After the data pre-processing, the data size of the initial transactional database will become much larger. This paper

applies the conversion tool of Microsoft Office Access to converse the database tables into text files. And then, this paper uses the mining algorithm to read the text files for mining. In this way, the mining efficiency can be improved. This paper applies the commonly used two-step association rules mining method: the minimum closed value of the support degree minsup and the minimum closed value of the confidence coefficient min confident must be given respectively before the mining generated by frequent item sets and before the mining generated by strong rules. In some actual mining, when minsup is given as 50% and min confident is given as 50% , minute change in the support degree can cause the quantity of the association rules to vary in two magnitude orders for the same given database. This paper conducts the association rules mining to the transactional database; makes analysis of the mining results; and reaches the final conclusion as follows.

5 CONCLUSION

This paper studies the one branch-association rules mining of data mining. Besides the association rules mining, data mining includes many other aspects, including artificial neural network, decision-making tree and support vector machine. Research of these fields should be continued from now on, so as to get better control of data mining and apply it in other applications. The final goal of scientific research is application in production. There's extensive application prospect in data rules mining. This paper only applies data rules mining in the team psychological health assessment system. However, the data analysis module has certain adaptation to many other application backgrounds. More efforts should be made in the universality of the data analysis module in the future. It is also feasible to separate the data analysis module from the system and build a universal data analysis system.

REFERENCES

[1] Zhong J.P.2007. *The Applied Research of Association Rules Mining in Psychological Analysis*. Zhejiang University.

[2] Cheng Y. 2010. *The Applied Research of Association Rules Mining in Disease Data Processing*. Chongqing Medical University.

[3] Kou Y.2010. *The Applied Research of Association Rules Mining in the Cross-selling of Telecommunication Products*. Harbin Institute of Technology.

[4] Lin J.X.2011. *The Applied Research of Association Rules Mining in E-Commerce Recommendation Systems*. Ji'nan University.

[5] Huang W.J.2014. *The Applied Research of Association Rules Mining Based on Ant Colony Algorithm in Marketing*. Jiangxi Normal University.

[6] Wang X.2013. *The Research of Incremental Updating Association Rules Mining and Its Application in Mobile Phone Virus Detection*. Beijing University of Posts and Telecommunications.