A Filtering Approach for Mining Frequent Itemsets

Jen-Peng Huang* and Huang-Cheng Kuo

Abstract. Many efficient association rule mining algorithms have been proposed in the literature. In this paper, we propose an algorithm FRM (Mining Frequent Itemsets by Frequent-Related Mechanism). Most of the studies adopt an Apriori-like candidate set generation-and-test approach. However, candidate generation is still costly when there exist a large number of long patterns. FRM scans database only four times and it does not adopt the Apriori-like approach in mining process. It uses the frequent-related mechanism to generate the itemsets which are the most possible to be frequent and it eliminates a great number of infrequent itemsets. So FRM is very suitable to mine the databases whose record length is very long.

1 Introduction

Association rule mining finds interesting association or correlation relationships among a large set of data items. With massive amounts of data continuously being collected and stored in databases, many industries are becoming interested in mining association rules from their databases. For example, the discovery of interesting association relationships among huge amounts of business transaction records can help catalog design, cross-marketing and other business decision making processes. Market basket analysis is one of the powerful methods which aim at finding regularities in the shopping behaviors of customers of supermarkets, mail-order companies, and on-line shops. This process analyzes customer buying habits by finding associations between the different items that customers place in their "shopping baskets." The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers. Such information can lead to increased sales by helping retailers to do selective marketing and plan their shelf space.

Jen-Peng Huang
Department of Information Management
Southern Taiwan University

Huang-Cheng Kuo
Department of Computer Science and Information Engineering
National Chiayi University

* Corresponding author.

B.-C. Chien and T.-P. Hong (Eds.): Opportunities and Challenges, SCI 214, pp. 59–63.
springerlink.com

Frequent-pattern mining plays an essential role in mining association rules [1, 2, 3, 5, 6, 7, 9], correlations[5], bag database[4], and many other important data mining tasks.

Definition 1. We call the number of items in an itemset its size, and call an itemset of size k a k-itemset.

Definition 2. A k-subset is a sub-itemset of size k.

2 FRM: Mining Frequent Itemsets by Frequent-Related Mechanism

We propose a new algorithm FRM (Mining Frequent Itemsets by Frequent-Related Mechanism) which uses the frequent-related mechanism to generate those itemsets which are very possible to be frequent. Besides, FRM uses a hash based technique, Hash MAP, which is similar to Hash Table to store the sub-itemsets in order to increase the access efficiency.

Definition 3. Given a *k*-itemset *X* and a set of *k*-itemset $X_s = \{S_1, S_2, \ldots, S_m\}$ where S_i, S_j are the sub-itemsets of *X*, $S_i \neq S_j$, $1 \leqq i < j \leqq m \leqq 2^k-1$ then X_s is called a *decompositional set* of *X*.

Lemma 1. If *X* is a *k*-itemset then there are 2^k-1 distinct subsets of *X*.

Definition 4. Given *k*-itemset *X* and $X_C = \{S_1, S_2, \ldots, S_m\}$ where X_C is a decompositional set of *X*, $S_i \neq S_j$, $1 \leqq i < j \leqq m$, if $m=2^k-1$ then X_C is called the *complete decompositional set* of *X*.

The *complete decompositional sets* of all transaction records are very huge, so it is almost impossible to use them as the candidate itemsets for mining frequent itemsets is nearly impossible. Instead of generating *complete decompositional set*, FRM uses the frequent-related mechanism to reduce the number of candidate itemsets.

Definition 5. Given two items *x* and *y*, if the combined itemset *xy* is a frequent 2-itemsets then *x* and *y* are *frequent-related*. Otherwise, *x* and *y* are *infrequent-related*.

Definition 6. Given an itemset *X* and an item *y*, if every item of *X* is *frequent-related* to y then *X* and *y* are *frequent-related*. Otherwise, *X* and *y* are *infrequent-related*.

By the Apriori property we know that any superset of an infrequent itemset is not frequent, so we can use the frequent 1-itmesets to trim database. The pseudo-code of FRM is shown in Fig. 1.

Lemma 2. If an itemset X and an item y are not frequent-related, then the combined itemsets Xy is not frequent.

```
Algorithm FRM        // FRM algorithm
Input: DB_data  // DB_data: Database
       min_sup  // min_sup: minimum support threshold
        CMap    // a hash map to store the candidate itemsets
        LMap    // a hash map to store the frequent itemsets
Output:  the frequent itemset

(1)  CMap = Null;  LMap= NULL; //initialize CMap and LMap
(2)  scan DBdata to get the frequent 1-itemsets, and save in Ln[1];
(3)  Use the frequent 1-itemsets to shorten the DB_data and save in NewDB;
(4)  scan DBdata to get the frequent 2-itemsets, and save in Ln[2];
(5)  NewDB=TrimRecord(NewDB, Ln[2]);  //Use Apriori property to trim the DB
(6)  While (there is any record r in NewDB){
(7)       For (i=1; i<=Length(r);i++) {
(8)            For (every itemset X in  CMap]) do  {
(9)               if (r[i] is frequent-related to X) then
(10)                   Append itemset (x r[i]) to CMap; //Append the new combined itemset to CMap
(11)              }
(12)         }
(13)        Append r[i] to CMap; //Append the item r[i] itself to CMap
(14)      }
(15) }
(16)  For (every itemset X in CMap) do {
(17)       if Support(X) > min_sup then  //check the support of itemset X
(18)          Append itemset X to LMap;  //Append the frequent itemset to LMap
(19) }
(20)  return(LMap);  //return the frequent itemsets
```

Fig. 1 The pseudo-code of FRM

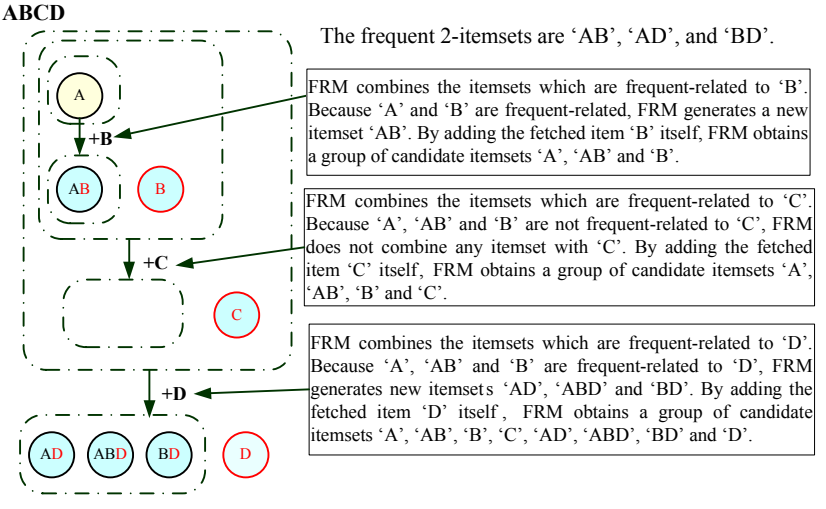

Fig. 2 The processes of generating itemsets by frequent-related mechanism

Instead of generating *complete decompositional sets* of all transaction records, the frequent-related mechanism of FRM uses Lemma 2 to reduce the number of candidate itemsets. For each transaction record of database FRM goes through the items of the record sequentially and uses the frequent-related mechanism to generate a group of candidate itemsets which are very possible to be frequent.

Here we use an example which is shown in Fig. 2 to explain the processes of generating itemsets with frequent-related mechanism.

3 Experimental Results

This section compares the experimental performance of FRM, FP-growth[3] and OP[8]. The three algorithms are implemented with Java language running under J2SDK1.4.2 environment. All experiments are performed on Intel Pentium IV 2.8GHz PC machine with 512MB memory. The operating system is Windows 2000 Server. Synthetic datasets are generated using publicly available synthetic data generation program of IBM Quest data mining project at http://www.almaden.ibm.com/cs/quest/, which has been used in most association rules mining studies. Besides, we use the BMS-POS dataset, and the Retail market basket dataset which are publicly available at http://fimi.cs.helsinki.fi/ as the other test datasets. The experimental results are shown in Fig. 3.

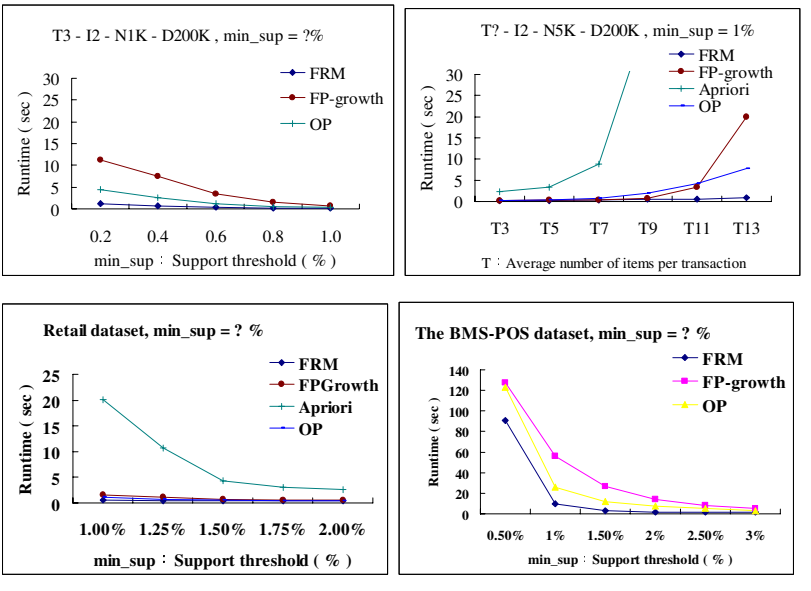

Fig. 3 Experimental result

4 Conclusion

FRM mainly uses the frequent-related mechanism to increase performance of discovering all of the frequent itemsets. In the mining processes, it can avoid generating a great number of candidate itemsets via the frequent-related mechanism, and then increases the efficiency and the utility rate of memory.

The advantages of FRM are as follows:

(1) FRM scans the database only four times to finish the mining task. Hence, FRM can avoid wasting a good deal of unnecessary I/O time, and then increase the efficiency.

(2) In mining frequent itemsets, FRM does not adopt the Apriori-like candidate set generation-and-test approach. It can use the frequent-related mechanism to filter out a huge number of candidate itemsets so it can save a great deal of time of generation of candidate itemsets, and then increase the efficiency.

(3) The framework of FRM is easy to be implemented.

Acknowledgments. This work was partially supported by grant NSC 92-2213-E-218 -020 and NSC 93-2213-E-218-012 from the National Science Council, Taiwan.

References

1. Agrawal, R., Imieliski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216. ACM Press, Washington (1993)
2. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the 1997 ACM SIGMOD International Conference on Management of data, pp. 255–264. ACM Press, Tucson (1997)
3. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining Knowledge Discovery 8, 53–87 (2004)
4. Hsu, P.-Y., Chen, Y.-L., Ling, C.-C.: Algorithms for mining association rules in bag databases. Information Sciences 166, 31–47 (2004)
5. Huang, J.-P., Chen, S.-J., Kuo, H.-C.: An efficient incremental mining algorithm-QSD. Intelligent Data Analysis 11(3), 265–278 (2007)
6. Huang, J.-P., Lan, G.-C., Kuo, H.-C., Hong, T.-P.: A decomposition approach for mining frequent itemsets. In: Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2007), vol. 2, pp. 605–608 (2007)
7. Lin, D.I., Kedem, Z.: Pincer-Search: A new algorithm for discovering the maximum frequent set. In: Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology, pp. 105–119 (1998)
8. Liu, J., Pan, Y., Wang, K., Han, J.: Mining frequent itemsets by opportunistic projection. In: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 229–238. ACM Press, Edmonton (2002)
9. Park, J.S., Chen, M.-S., Yu, P.S.: Using a hash-based method with transaction trimming for mining association rules. IEEE Transactions on Knowledge and Data Engineering 9, 813–825 (1997)