An Algorithm of Frequent Patterns Mining Based on Binary Information Granule

G. Fang
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Sichuan, China
School of Computer Science and Engineering
Chongqing Three Gorges University
Chongqing, China

Y. Wu
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Sichuan, China

***Abstract*--To get rid of these traditional frameworks for discovering frequent association patterns, this paper proposes an algorithm of frequent association patterns mining based on binary information granule, which is mainly different from the Apriori framework and the FP-growth framework. The algorithm generate candidate by Boolean complementation to avoid connecting candidate operation of the Apriori framework, and compute support by the intersection of binary information granules to avoid to repeatedly read the database; it also adopts a linear array to avoid using complex data structure similar to the FP-growth framework. Based on these comparisons of experiments, the results indicate that the proposed algorithm is better than the traditional mining frameworks, particularly, the Apriori framework and the FP-growth framework.**

Keywords-binary; frequent patterns; association rules; data mining; granular computing

I. INTRODUCTION

In recent years, data mining is a research hotspot in artificial intelligence. In data mining technologies, one of important methods is association rules, where discovering frequent association pattern is its key task. In 1993, Agrawal et al. firstly introduce the problem of discovering frequent association patterns over basket data, and then lots of algorithms for mining frequent association patterns are proposed. In these algorithms, according to the different ways of generating candidate frequent itemsets, we could divide them into two the following types:

The first type of discovering frequent patterns needs to generate candidates, where the Apriori algorithm is the most classic, which is proposed by Agrawal et al [1]. Afterward, scholars also proposed some extended algorithms based on the Apriori framework [2, 3, 4]. In these algorithms, repeatedly reading the database or generating redundant candidate itemsets had become the bottleneck of improved performance.

The other need not generate candidate in mining association rules, where FP-growth is a representative algorithm, which is proposed by Han et al in 2000 [5]. And then, some extended algorithms based on the FP-growth framework are proposed [6, 7, 8].The general idea of the FP-growth framework is divide and conquer, in the course of discovering frequent patterns, the FP-growth frame- work only needs to read the database twice to solve the bottleneck of the Apriori, but the memory issue still exists because of the application of highly condensed data structure.

To avoid these disadvantages existing in two types of algorithms for mining frequent association patterns from the view of formal and mathematical model of data mining, subsequently, granular computing (GrC) has been introduced in data mining [9, 10], especially, in mining frequent patterns, more algorithms based on GrC are proposed in recent years [11-14], where L. M. Tsai et al. construct a taxonomy tree structure, and extract generalized negative association rules via GrC [11], but it is also an extended algorithm based on the FP-growth algorithm, T. R. Qiu et al. propose a GrC approach based on the Apriori framework for mining frequent patterns[12].

In a word, for discovering frequent patterns, most of algorithms do not get rid of these traditional ideas, namely, the Apriori and FP-growth framework. Hence, in this paper, we propose an algorithm of frequent patterns mining based on binary information granule, which is different from them.

II. BASIC DEFINITIONS

For the rough set theory [15] and quotient space theory [16], in this paper, we use a partition model of granular computing to construct a granule [17].

Definition 1 An information system is a tetrad $IS = < U, A, V, f >$, where

U, called universe of discourse, consists of objects, each object is called an individual;

A, called set of attributes, each attribute is a Boolean attribute;

V, called set of range, for $\forall a \in A, V_a \in V$ is the range of attribute a;

f, called mapping function, namely $f: U \to V, \forall u \in U, a \in A, f(u,a) \in V_a$.

Definition 2 An information granule is expressed as $IG = (\varpi, \psi(\varpi))$, where

ϖ, called the intension of information granule, is an abstract description of common attributes shared by elements in the extension, which is expressed as:

$\varpi = (\varpi_1, \varpi_2, ..., \varpi_k)(\{\varpi \mid V_{a^{\theta(1)}} \times V_{a^{\theta(2)}} \times ... \times V_{a^{\theta(k)}}, a^{\theta(i)} \in A, i = 1,2,...,k\}, k \leq |A|)$

$\psi(\varpi)$, called the extension of information granule, it is the set of objects which information granule applies, which is expressed as follows:

$\psi(\varpi) = \{u \mid f(u, a^{\theta(1)}) = \varpi_1 \wedge f(u, a^{\theta(2)}) = \varpi_2 \wedge ... \wedge f(u, a^{\theta(k)}) = \varpi_k, u \in U\}$

Here, for the information granule, if $|\varpi| = 1$, and then it is called an atomic information granule $AIG = (\varpi, \psi(\varpi))$.

Definition 3 A binary information granule is expressed as $BIG = (\omega, \varphi(\omega))$, where

ω, called the intension of binary information granule, is a binary number, which is expressed as $(\omega_{|A^*|}\omega_{|A^*|-1}...\omega_1)_2 (\omega_i \in \{0,1\}, i = 1,2,...,|A^*|, A^* \subseteq A)$, where

$\forall u \in U, u \to \omega = (\omega_{|A^*|}\omega_{|A^*|-1}...\omega_1)_2$

$(\omega_i = \xi(u, a^{\theta(i)}) = \begin{cases} 1 & f(u, a^{\theta(i)}) \in u \\ 0 & f(u, a^{\theta(i)}) \notin u \end{cases}, a^{\theta(i)} \in A^*, i = 1,2,...,|A^*|)$;

$\varphi(\omega)$, called the extension of binary information granule, is a set of objects, which is expressed as $\varphi(\omega) = \{u \mid \forall \omega_i = 1, f(u, a^{\theta(i)}) \in u, i = 1,2,...,|A^*|, u \in U\}$.

Here, if $\exists(\omega_k = 1 \wedge \omega_k \in \omega), \forall(\omega_i = 0 \wedge i \neq k \wedge \omega_i \in \omega)$, then $BIG = (\omega, \varphi(\omega))$ is called an atomic binary information granule $ABIG = (\omega, \varphi(\omega))$.

Definition 4 An intersection of binary information granule is expressed as $\otimes$, which is described as follows:

$\exists BIG_\alpha = (\omega_\alpha, \varphi(\omega_\alpha)), BIG_\beta = (\omega_\beta, \varphi(\omega_\beta))$;

$BIG_\alpha \otimes BIG_\beta = (\omega_\alpha \& \omega_\beta, \varphi(\omega_\alpha) \cap \varphi(\omega_\beta))$, & is the logical and operation.

III. An Algorithm of Frequent Association Patterns Mining

In this paper, we propose an algorithm of frequent patterns mining based on binary information granule, called *BFPM*, which need to generate candidate by the simple data structure similar to the Apriori framework, but it need not repeatedly read database to compute support similar to the FP-growth framework.

A. Candidate Generator

Here, we use a linear array to save a binary number, and generate candidate via Boolean complementation. The candidate helps to construct a binary information granule, which is regarded as the intension of binary information granule, and use the intension to get the extension of binary information granule through the intersection operation. The algorithm discovers frequent patterns via the binary information granule. The course of generating candidate is expressed as follows:

Candidate-generator(*Array*: *binary*) :

Input: binary, is a binary number saved by a linear array;

Output: next-binary, is also a binary number saved by a linear array;

(1) Array = binary; // Array [0] save the lowest bit of binary
(2) Count = 0;
(3) No-Exit = true;
(4) While (No-Exit && Count < |Array|) do begin
(5) Array [Count] = ~ Array [Count]; // ~ is Boolean complementation
(6) If (Array [Count] == 0) then
(7) No-Exit = false; //search the first 0 of binary number
(8) Else
(9) Count++;
(10) End; // execute Boolean complementation for the first 0 bit and its front
(11) Return next-binary = Array;

B. A Mining Algorithm Based on Binary Information Granule

In this part, we use the pseudo code to introduce the algorithm *BFPM* as follows:

Input: A tradition database is D;
A parameter given by user is *Min-sup* ;

Output: A set of frequent patterns is *Frequent* ;

(1) Read D ; // scan database once
(2) Construct $IS =< U, A, V, f >$;
(3) $S_A = \{s \mid s = (\varpi, \psi(\varpi)), s \in AIG\}$; //get all atomic information granules
(4) For $(\forall s = (\varpi, \psi(\varpi)) \in S_A)$ do begin
(5) If $(|\psi(\varpi)| < Min\text{-}sup)$ then
(6) $S_A = S_A - \{s = (\varpi, \psi(\varpi))\}$;
(7) End //get the intension of atomic information granule is frequent
(8) For $(\forall s = (\varpi, \psi(\varpi)) \in S_A)$ do begin
(9) $s = (\varpi, \psi(\varpi)) \to s^* = (\omega, \varphi(\omega))$; // **Definition 3**
(10) $S = S \bigcup \{s^* = (\omega, \varphi(\omega))\}$ $(s^* \in ABIG;$
(11) End //compute all atomic binary information granules *ABIG*
(12) $Array[|S|] = (0,0,...,0)$; //there is $|S|$ bit for a binary saved by the array
(13) For $(i = 0; i < 2^{|S|} - 1; i++)$ do begin
(14) $Array = Candidate(Array)$; //call the function
(15) $Array \to \delta$; //turn into a binary number δ
(16) If $(\forall \alpha \in NF \wedge (\alpha \& \delta \neq \alpha))$ then begin //delete the superset of non frequent itemsets
(17) For $(j = 0; j < |S|; j++)$ do begin

(18) If ($Array[j]=1$) then

(19) $IG=(\omega,\varphi(\omega))=IG\otimes S[j]$; // **Definition 4**

(20) End

(21) If ($|\varphi(\omega)|\geq M\ in\text{-}sup$) then begin

(22) $F=F-\{\alpha\,|\,\alpha\&\omega=\alpha\}$; // delete all subset of frequent itemset for ω

(23) $F=F\bigcup\{\omega\}$; // write ω to frequent set F

(24) End

(25) Else

(26) $NF=NF\bigcup\{\omega\}$; // write ω to non frequent set NF

(27) End

(28) End

(29) $F\rightarrow Frequent$; //each binary number is turned into itemsets

(30) $Answer=Frequent$;

IV. EXPERIMENTAL RESULTS

In this section, we compare the proposed algorithm *BFPM* with these traditional frequent patterns mining algorithms, particularly, Apriori and FP-growth.

The experimental data source is from the Food Mart 2000 retail data set, which comes from SQL Server 2000. It contains 164558 records in 1998. From these records, we take items purchased by the same customer at the same time as a basket. Because the supports of the bottom items are small, we generalize the bottom items to the product department. Finally, we obtain 34015 transactions with time-stamps; the number of product department is 22.

All the experiments are performed on the following environment:

Hardware environment is an Intel (R) Core (TM) 2 Duo CPU (T6570 @) 2.10 GHz 1.19GHz) PC with 1.99 GB main memory.

Software environment is C# with Microsoft Visual Studio 2008.

OS is Microsoft Window XP Professional.

We design three groups of experiments as follows:

Firstly, we test the performance of three algorithms as the support is added. The comparison result is expressed as figure 1.

Secondly, we test their performance as the number of frequent patterns is added. The comparison result is expressed as figure 2.

Finally, we test their performance as the maximal length of frequent pattern is added. The comparison result is expressed as figure 3.

According to these comparisons, we can find that the algorithm *BFPM* is better than the traditional mining framework.

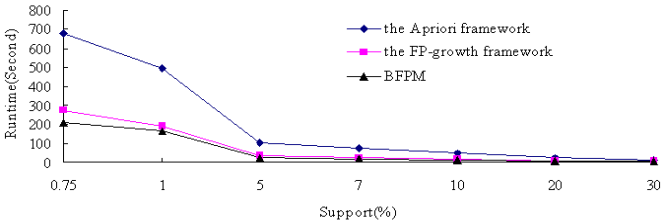

FIGURE I. THE COMPARISON WITH SUPPORTS.

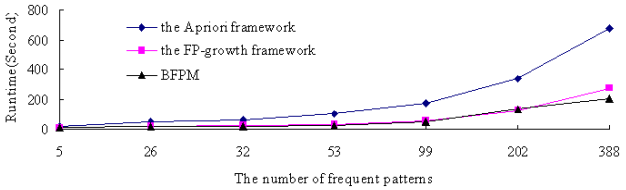

FIGURE II. THE COMPARISON WITH FREQUENT PATTERNS.

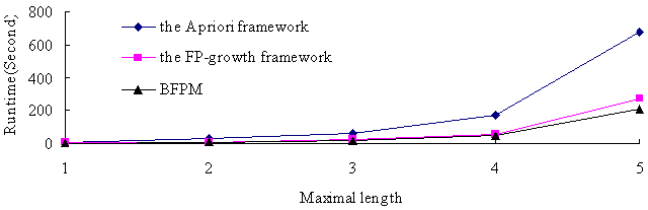

FIGURE III. THE COMPARISON WITH MAXIMAL LENGTH.

V. CONCLUSIONS

The algorithm need not repeated read the database to compute support, and it also need not use complex data structure to save information, which get ride of these traditional ideas, namely, the Apriori and FP-growth framework. It is also suitable for dynamic data mining because of binary information granule. In the future, we will discuss its application on cloud environments.

ACKNOWLEDGEMENTS

This work was a project supported by Chongqing Cutting-edge and Applied Foundation Research Program (Grant No. cstc2014jcyjA40035). And it was also supported by Scientific and Technological Research Program of Chongqing Three Gorges University (Grant No.13ZD20).

REFERENCES

[1] R. Agrawal and R. Srikant, Fast algorithms for mining association rules. *Proc. Of the 20th VLDB Conf*, Santiago, Chile, pp. 487–499, 1994.

[2] B. Boutsinas, A new biclustering algorithm based on association rule mining. *International Journal on Artificial Intelligence Tools*, 22(3), pp. 1350017-1-13, 2013.

[3] Wang Le, Feng Lin, and Zhang Jing et al. An efficient algorithm of frequent itemsets mining based on MapReduce. *Journal of Information and Computational Science*, 11(8), pp.2809-2816, 2014.

[4] Lin Ke-Chung, Liao I-En, and Chang Tsui-Ping et al. A frequent itemset mining algorithm based on the Principle of Inclusion-Exclusion and transaction mapping. *Information Sciences*, 276(4), pp. 278-289, 2014.

[5] J. Han, J. Pei, and Y. Yin, Mining frequent patterns without candidate generation. *Proc. of 2000 ACM SIGMOD Conf*, New York, USA, June, pp. 1–12, 2000.

[6] Silva Andreia1, Antunes Cláudia. Mining stars with FP-growth: Case study on bibliographic data. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, 19(SUPPL. 1), pp.65-91, 2011.

[7] Arour Khedija1, Belkahla Amani. Frequent pattern-growth algorithm on multi-core CPU and GPU processors. *Journal of Computing and Information Technology*, 22(3), pp.159-169, 2014.

[8] Chen Xing-Shu, Zhang Shuai1, and Tong Hao1 et al. FP-growth algorithm based on Boolean matrix and MapReduce. *Journal of South China University of Technology (Natural Science)*, 42(1), pp.135-141, 2014.

[9] Y.Y. Yao and N. Zhong. Potential applications of granular computing in knowledge discovery and data mining. *Proc. of World Multi conference on Systemics, Cybernetics and Informatics*, pp. 573–580, 1999.

[10] W. Z. Wu, Y. Leung, and J. S. Mi, Granular Computing and Knowledge Reduction in Formal Contexts. *IEEE Transactions on Knowledge and Data Engineering*, 21(10), pp. 1461–1474, 2009.

[11] L. M. Tsai, S. J. Lin, and D. L. Yang, Efficient Mining of Generalized Negative Association Rules. *Proc. of 2010 IEEE Int. Conf. on Granular Computing*, San Jose, CA, pp. 471–476, 2010.

[12] T. R. Qiu, X. Q. Chen, and Q. Liu et al. Granular Computing Approach to Finding Association Rules in Relational Database. *International Journal of intelligent systems*, 25, pp. 165–179, 2010.

[13] Chen Hui, Lin Tsau Young, and Zhang Zhibing et al. Parallel mining frequent patterns over big transactional data in extended MapReduce. *Proc. of 2013 IEEE International Conference on Granular Computing*, Beijing, China. pp. 43–48, 2013.

[14] Fang Gang, Wu Yue. Frequent spatiotemporal association patterns mining based on granular computing. *Informatica*, 37(4), pp.443–453, 2013.

[15] Z. Pawlak. Granularity of knowledge, indiscernibility and rough sets. *Proc. of 1998 IEEE Int. Conf. on Fuzzy Systems*, Anchorage, AK, pp.106–110, 1998.

[16] L. Zhang and B. Zhang. The quotient space theory of problem solving, *Lecture Notes in Computer Science*, 2639, pp.11–15, 2003.

[17] Y.Y. Yao, A partition model of granular computing, *Lecture Notes in Computer Science Transactions on Rough Sets*, 3100, 232–253, 2004.