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 patterns 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 framework 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 \( f: U \rightarrow V, \forall u \in U, a \in A, f(u, a) \in V_a \).

Definition 2 An information granule is expressed as \( IG = (\sigma, \psi(\sigma)) \), where
\(\sigma\), called the intension of information granule, is an abstract description of common attributes shared by elements in the extension, which is expressed as:

\[\psi(\sigma) = \{ u | f(u, a^{(i)}) = \sigma_{i}, f(u, a^{(j)}) = \sigma_{j}, \ldots, f(u, a^{(k)}) = \sigma_{k}, u \in U \}\]

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

\[\psi(\sigma) = \{ u | f(u, a^{(i)}) = \sigma_{i}, f(u, a^{(j)}) = \sigma_{j}, \ldots, f(u, a^{(k)}) = \sigma_{k}, u \in U \}\]

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

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

\(\omega\), called the intension of binary information granule, is a binary number, which is expressed as \((\omega_{1}, \omega_{2}, \ldots, \omega_{n})\), where

\[\forall u \in U, u \rightarrow \omega = (\omega_{1}, \omega_{2}, \ldots, \omega_{n})\]

\(\phi(\omega)\), called the extension of binary information granule, is a set of objects, which is expressed as \(\phi(\omega) = \{ u | \forall \omega_{i} = 1, f(u, a^{(i)}) = \omega_{i}, u \in U \}\).

Here, if \(\exists \omega_{i} = 1\) and \(\Delta = \omega_{i} \land \omega_{i} \neq \omega_{i} \land \omega_{i} \in \omega\) then \(BIG = (\omega, \phi(\omega))\) is called an atomic binary information granule \(ABIG = (\omega, \phi(\omega))\).

**Definition 4** An intersection of binary information granule is expressed as \(\bigotimes\), which is described as follows:

\[\exists BIG_{a} = (\omega_{a}, \phi(\omega_{a})), BIG_{b} = (\omega_{b}, \phi(\omega_{b})); \bigotimes BIG_{a} \ast BIG_{b} = (\omega_{a} \land \omega_{b}, \phi(\omega_{a}) \land \phi(\omega_{b})\), \& 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 Grane**

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 \(IS \leftarrow U, A, V, f \geq \);
(3) \(\bigcup_{k=1}^{2} S_{k} = \{ s = (\sigma, \psi(\sigma)), s \in AIG \} \); // get all atomic information granules
(4) For \(\forall s = (\sigma, \psi(\sigma)) \in S_{k}\) do begin
(5) If \(|\psi(\sigma)| < Min \sup\) then
(6) \(S_{k} = S_{k} - \{ s = (\sigma, \psi(\sigma)) \} \);
(7) End // get the intension of atomic information granule is frequent
(8) For \(\forall s = (\sigma, \psi(\sigma)) \in S_{k}\) do begin
(9) \(s = (\sigma, \psi(\sigma)) \rightarrow s^{*} = (\omega, \phi(\omega))\); // Definition 3
(10) \(S' = S \bigcup \{ s^{*} = (\omega, \phi(\omega)) \} \); \% \(ABIG\);
(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) \(A r s = a_{y} \ C a \alpha \ n \ d \ i \ d \ a(t e 1); // call the function\)
(15) \(Array \rightarrow \delta\); // turn into a binary number \(\delta\)
(16) If \(\forall \alpha \in NF \land (\alpha \in \delta \land \alpha \notin \alpha)\) then begin // delete the superset of non frequent itemsets
(17) \(F = \{ j : j \notin S \}; j++ \) do begin
If \( Array[j] = 1 \) then
\[
IG = (\omega, \varphi(\omega)) = IG \otimes S[j] ; \quad // \text{Definition 4}
\]
End
If \( |\varphi(\omega)| \geq M \text{ in-sup} \) then begin
\[
F = F - \{ \alpha | \alpha \& \omega = \alpha \} ; \quad // \text{delete all subset of frequent itemset for } \omega
\]
End
Else
\[
NF = NF \cup \{ \omega \} ; \quad // \text{write } \omega \text{ to non frequent set } NF
\]
End
End
\[
F \rightarrow \text{Frequent} ; \quad // \text{each binary number is turned into itemsets}
\]
\[
\text{Answer} = \text{Frequent} ;
\]

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 database, 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 timestamps; 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.

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


