A Developed Algorithm of Apriori Based on Association Analysis

LI Pingxiang  CHEN Jiangping  BIAN Fuling

ABSTRACT A method for mining frequent itemsets by evaluating their probability of supports based on association analysis is presented. This paper obtains the probability of every 1-itemset by scanning the database, then evaluates the probability of every 2-itemset, every 3-itemset, every k-itemset from the frequent 1-itemsets and gains all the candidate frequent itemsets. This paper also scans the database for verifying the support of the candidate frequent itemsets. Last, the frequent itemsets are mined. The method reduces a lot of time of scanning database and shortens the computation time of the algorithm.

KEYWORDS association rule; algorithm apriori; frequent itemset; association analysis

CLC NUMBER TP312

Introduction

In the algorithms of association rules mining, apriori is the ancestor. The main idea of the apriori is scanning the database repeatedly. The most important step in mining association is the generation of frequent itemsets. In algorithm apriori, most time is consumed for scanning the database repeatedly.

1 Related concepts

We present the definitions of the concepts that are used to describe the improved algorithm. Let us start from the following definitions for association rules.

Let \( I = \{ i_1, i_2, \ldots, i_n \} \) be a set of all items, where an item is an object with some predefined attributes (e.g. price, weight, etc.). A transaction \( T = \langle \text{tid}, I, \rangle \) is a tuple, where \( \text{tid} \) is the identifier of the transaction and \( I, \subseteq I \). A transaction database \( T \) consists of a set of transactions. An itemset is a subset of the set of items. A \( k \)-itemset is an itemset of size \( k \). We write itemsets as \( S = i_1, i_2, \ldots, i_k, S \subseteq I, \) omitting set brackets.

Definition 1. An association rule takes the form \( X \Rightarrow Y \) where \( X \subseteq I, Y \subseteq I, \) and \( X \cap Y = \emptyset \). The support of the rule \( X \Rightarrow Y \) in the transaction database is:

\[
\text{support}(X \Rightarrow Y) = \frac{| \{ T : X \cup Y \cup T, T \in D \} |}{| D |}
\]

marked by \( \text{support}(X \Rightarrow Y) \).

Definition 2. The confidence of the rule \( X \Rightarrow Y \) in transaction database is:

\[
\text{confidence}(X \Rightarrow Y) = \frac{| \{ T : X \cup Y \cup T, T \in D \} |}{| \{ T : X \subseteq T, T \in D \} |}
\]

marked by \( \text{confidence}(X \Rightarrow Y) \).

The mining association rules is generating association rules with itemsets which have equal or larger support and confidence.

Definition 3: Let \( A = A_1, A_2, \ldots, A_n \) be a set of
all events. If there is
\[ P(A_1 A_2 \cdots A_i) = P(A_i) \prod_{j=1}^{i} P(A_j) \cdots P(A_i), \]
(1)
and it is right for any \( k(1 \leq k \leq n) \) and any \( 1 \leq i_1 < i_2 < \cdots < i_k \leq n \), then \( A_1, A_2, \cdots, A_k \) are the inter-independent events.

The candidate frequent itemsets refer to the itemsets whose probabilities are larger than the user sets.

## 2 Improved algorithm for apriori

### 2.1 Main idea of the improved algorithm

Let \( P_1, P_2, \cdots, P_n \) be the independent probability of every item \( A_1, A_2, \cdots, A_n \), then the probability for any two item \( A_k, A_m \) both appearing in one transaction is \( P_k P_m \).

If \( A_k \) and \( A_m \) are total non-correlation, from Definition 4 it can be concluded that \( P_{k,m} = P_k P_m \), if \( A_k \) and \( A_m \) are total correlation, then \( P_{k,m} \) is the minimum of the \( P_k \) and \( P_m \), so, \( P_k P_m \leq P_{k,m} \leq P_k \).

Now the problem is: given \( P_k \) and \( P_m \), and \( P_k P_m \leq P_{k,m} \leq P_k \), how to evaluate \( P_{k,m} \)? The problem can not be solved with the conditions in mathematics. But in fact, there is a lot of information without accurate mathematic formula which is omitted. This paper offers a method for confirming the formula by association analysis.

Let parameter \( a \) be the probability for which \( A_k \) and \( A_m \) are total correlation, and parameter \( b \) for total non-correlation, and \( a + b = 1 \), \( 0 < a, b < 1 \), then \( P_{k,m} \) can be defined as:

\[ P_{k,m} = a \cdot P_k + b \cdot P_m \]  
(2)

There are a lot methods for proving the values of parameters \( a \) and \( b \). This paper provides a way to define \( a \) and \( b \) by association analysis.

### 2.2 Proving \( a \) and \( b \) by association analysis

There is a series of criterion about environment. The most ingredients of the pollution can be obtained from the source. So we can consider the criterion as a reference list and there is a need to find the correlation as a comparison list. Thus we will get the correlation coefficient which is the parameter \( a \) in Eq. (2), and \( b = 1 - a \). The details are described as follows.

Let \( S = (S_1, S_2, \cdots, S_m) \) be the value list of item \( A_n \), where \( S_1, S_2, \cdots, S_m \) are the samples extracted from the database and \( X = (X_1, X_2, \cdots, X_m) \) be the value list for item \( A_n \), where \( X_1, X_2, \cdots, X_m \) are the sample extracted from the database.

\[ a_{lm} = \frac{\min \Delta_i(k) + \rho \max \Delta_i(k)}{\max \Delta_i(k) + \rho \max \Delta_i(k)} \]  
(3)

where \( a_{lm} \) is the correlation coefficient of item \( A_m \) and \( A_l \); \( \Delta_i(k) = |S_i - X_i|; \rho \) is the distinguished coefficient which is set by users, and usually \( \rho \in (0, 1) \).

We can use Eq. (3) to calculate the correlation coefficient of any item.

### 2.3 Description of algorithm

1) Create a new array \( PFA[n] \), and the original value for each element is 0; scan the database and calculate the probability of each itemset \( A_1, A_2, \cdots, A_n \), respectively, marked by \( P_1, P_2, \cdots, P_n \) respectively. Let each element of the array \( PFA[1], PFA[2], \cdots, PFA[n] \) be \( P_1, P_2, \cdots, P_n \).

2) Set a minimum value \( V \) for the probability of \( A_i \) appearing, If the probability of \( A_i \) appearing \( PFA[i] \) is larger than \( V \), then the itemset \( A_i \) is a frequent 1-itemset. So, you get some frequent 1-itemset appearing.

3) From the probability of 1-itemset appearing \( PFA[1], PFA[j], \cdots, PFA[m] \), and according
to Eq. (2), the probability of any two itemsets appearing in one recorder can be evaluated. Set a minimum value \( V_2 \) for the probability of \( A_i \) and \( A_i \) appearing synchronously in one record. If the probability is larger than \( V_2 \), then the itemset \( A_i \), \( A_i \) is a candidate frequent 2-itemset. Otherwise, set the value of the probability to be zero to simplify the later calculation. Let the element of array \( PFA_{k-1}[j] \) record the value of candidate frequent 2-itemsets.

Let \( V_i \) be the minimum probability for candidate frequent 2-itemset, and \( V_1 \) for candidate frequent 3-itemset, \( V_{k-1} \) for candidate frequent \((k-1)\)-itemset.

Set the minimum probability \( V_{k-1} \):

\[
V_{k-1} = a \cdot \min(PFA_{k-1}[1], PFA_{k-1}[2], \ldots, PFA_{k-1}[m]) + b \cdot \min(PFA_{k-1}[1], PFA_{k-1}[2], \ldots, PFA_{k-1}[m]) \cdot \max(PFA_{k-1}[1], PFA_{k-1}[2], \ldots, PFA_{k-1}[m])
\]

4) From \( k = 2 \) to \( n \), repeat the above steps 1), 2), 3) to calculate the probability of \( k \)-itemsets \( A_1, A_2, \ldots, A_k \) appearing in one recorder;

5) Scan the database again to calculate the support of the candidate frequent itemsets which is the result of step 4), and the steps are as follows:

1) Create a new array \( DMA[m] \) with zero as the original value of each element, where \( m \) is the number of candidate frequent itemsets;

2) Read and get the recorder of the database until the end of the database;

3) If there are itemsets \( A_1, A_2, \ldots, A_k \) in any recorder synchronously and \( A_i \neq 0, A_i \neq 0, \ldots, A_i \neq 0 \), then the support for \( A_1, A_2, \ldots, A_k \) \( DMA[k] = DMA[k]+1 \).

Repeat the above steps 2), 3) to calculate the actual support of every candidate frequent itemsets until the end of the database.

6) Find out the frequent itemsets from the candidate frequent itemsets. If \( DMA[k] \) is larger than the minimum support which is set by the users, then output the frequent itemsets.

Steps 5), 6) are used to confirm the probability and support of the candidate frequent itemsets.

7) Output the association rule from the result of step 6).

### 2.4 Explanation and simulation of the algorithm

The transaction database of some suboffice in AllElectronics is chosen to compare the efficiency of our algorithm with Apriori. The detailed data is given in Table 1. In the algorithm given by this paper the process of finding frequent itemsets includes three steps:

1) Scan the database to obtain the probability of frequent 1-itemsets;

2) Evaluate the probability of candidate frequent 2-itemsets, 3-itemsets, \( \ldots, m \)-itemsets, on the basis of the probability of frequent 1-itemsets;

3) Scan the database again to confirm the frequent itemsets from the candidate frequent itemsets.

There is a database with nine transactions and five itemsets. And \( a, b \) are \( 5/9 \) and \( 4/9 \), respectively, in the simulation of the algorithm.

| TID  | Itemsets |
|------|----------|
| T100 | I1, I2, I5 |
| T200 | I2, I4    |
| T300 | I2, I3    |
| T400 | I1, I2, I4|
| T500 | I1, I3    |
| T600 | I2, I5    |
| T700 | I1, I3    |
| T800 | I1, I2, I3, I5 |
| T900 | I1, I2, I3 |

Table 2 Frequent C1

| Itemsets | Probability |
|----------|-------------|
| I1       | 6/9         |
| I2       | 7/9         |
| I3       | 3/9         |
| I4       | 2/9         |
| I5       | 3/9         |

Calculate the probability of each itemset to find out that the frequent 1-itemsets at the support is \( 2/9 \).

Set \( a = 5/9, b = 4/9 \), then the threshold \( V_2 = (5 \times (2/9) + 4 \times (2/9) \times (7/9)) / 9 = 146 / 729 \).

From frequent C1 (Table 2), evaluate the probability of candidate frequent 2-itemsets (Table 3).
Table 3  Probability of candidate frequent 2-itemsets

| Itemsets    | Probability |
|-------------|-------------|
| \{I_1, I_2\} | 438/729     |
| \{I_1, I_3\} | 345/729     |
| \{I_1, I_4\} | 138/729     |
| \{I_1, I_5\} | 207/729     |
| \{I_2, I_3\} | 365/729     |
| \{I_2, I_4\} | 146/729     |
| \{I_2, I_5\} | 219/729     |
| \{I_3, I_4\} | 195/729     |
| \{I_3, I_5\} | 114/729     |

Set \( a = 5/9, b = 4/9 \), the threshold \( V_1 = (5 \times (146/729) + 4 \times (146/729) \times (7/9))/9 = 73 \times 146/59 049 \).

Obtain the candidate frequent 2-itemsets (Table 4).

Table 4  Frequent C2

| 2-itemsets | Probability |
|------------|-------------|
| \{I_1, I_2\} | 438/729     |
| \{I_1, I_3\} | 345/729     |
| \{I_1, I_5\} | 207/729     |
| \{I_2, I_3\} | 365/729     |
| \{I_2, I_4\} | 146/729     |
| \{I_2, I_5\} | 219/729     |
| \{I_3, I_5\} | 195/729     |

From Table 4, evaluate the probability of candidate frequent 3-itemsets (Table 5).

Table 5  Probability of candidate frequent 3-itemsets

| Itemset       | Probability |
|---------------|-------------|
| \{I_1, I_2, I_3\} | 69 \times 345/59 049 |
| \{I_1, I_2, I_4\} | 73 \times 138/59 049 |
| \{I_1, I_2, I_5\} | 73 \times 207/59 049 |
| \{I_1, I_3, I_4\} | 69 \times 130/59 049 |
| \{I_1, I_3, I_5\} | 69 \times 195/59 049 |
| \{I_1, I_4, I_5\} | 69 \times 114/59 049 |
| \{I_2, I_3, I_4\} | 73 \times 130/59 049 |
| \{I_2, I_3, I_5\} | 65 \times 114/59 049 |
| \{I_2, I_4, I_5\} | 73 \times 114/59 049 |
| \{I_2, I_3, I_5\} | 73 \times 195/59 049 |

Obtain the candidate frequent 3-itemsets (Table 6).

Table 6  Candidate frequent 3-itemsets

| 3-itemsets      | Probability |
|-----------------|-------------|
| \{I_1, I_2, I_3\} | 69 \times 345/59 049 |
| \{I_1, I_2, I_5\} | 73 \times 138/59 049 |
| \{I_2, I_3, I_5\} | 73 \times 195/59 049 |

3  Comparison between the algorithms

The number of the candidate frequent itemsets and times of scanning the database are compared between this algorithm and apriori, which are the linchpin of the efficiency in an algorithm. The algorithm in this paper is better than apriori in reducing the time for scanning the database. The database will be scanned \( k \) times to find out frequent \( k \)-itemsets in apriori while only two times in our algorithm by putting forward a concept of candidate frequent itemset.

In Table 7 the frequent 1-itemsets, 2-itemsets and 3-itemsets for the two algorithms are listed, respectively.

From Table 7, we know that there are more candidate frequent itemsets in this algorithm given by this paper than those in the algorithm apriori.

4  Experiment and analysis of the results

To validate the algorithm presented in this paper, a database for recording the update of the spatial database are used in the experiment. There are 10 items in the database, i.e. 11 for mender, 12 for DEPT of the mender, 13 for layer of the tower, 14 for layer of the road, 15 for layer of the polluter area, 16 for layer of line, 17 for layer of water, 18 for layer of building, 19 for layer of thunder and 110 for layer plant. The
The purpose of the experiment was mining the association rules between the mender and other spatial layers. All experiments are conducted on a PC with an Intel Pentium III 800MHz CPU and 128M main memory, running Microsoft Windows2000. All programs are coded in Microsoft VC++ 6.0. The experiments are conducted on real data sets by use of the algorithm described in this paper and apriori. In Table 8 the candidate frequent 1-itemsets, 2-itemsets, 3-itemsets of this algorithm and frequent 1-itemsets, 2-itemsets, 3-itemsets of algorithm apriori are listed. We report here only some results of some typical data sets, with 10 itemsets and in 5 000 tuples.

Table 8 Comparison between the frequent itemsets in the experiments

| Frequent itemsets for association rules | Algorithm apriori | Algorithm given by this paper |
|----------------------------------------|-------------------|------------------------------|
| 1-itemsets                             | (I1, I2, I4, I5, I7, I8, I9, I10) | (I1, I2, I4, I5, I7, I8, I9, I10) |
| 2-itemsets                             | (I1, I2), (I1, I5), (I1, I8), (I1, I9), (I1, I10), (I5, I8), (I5, I10) | (I1, I2, I4, I5, I7, I8, I9, I10) |
| 3-itemsets                             | (I1, I2, I5), (I1, I2, I8), (I1, I2, I10), (I1, I5, I8), (I1, I5, I10), (I5, I8, I10) | (I1, I2, I4, I5, I7, I8, I9, I10) |
| 4-itemsets                             | (I1, I2, I5, I8) | (I1, I2, I5, I8) |

The first data set used by us has 1 000 tuples, and then increases 1 000 tuples every time. We test the execution time of the algorithms with respect to number of tuples and itemsets. Fig. 1 shows that, with tuples from 0 to 5 000, when the number of tuples is small, both algorithms have similar performance; however, as the number of tuples grows, the algorithm proposed in this paper takes more effect. It keeps the runtime short. In contrast, the algorithm apriori does not scale well under large number of tuples. Fig. 1 also shows the execution time of both algorithms with respect to the itemsets increasing. As the number of itemsets goes up, the runtime of both algorithms increases and that of the algorithm given by this paper grows slower than algorithm apriori.

5 Conclusions

On the basis of the previous studies on algorithm apriori, we proposed a method for mining association rules in large databases with association analysis and probability evaluation. This method explores efficient mining of association rules by probability evaluation. First it scans the database to filter frequent 1-itemsets and gets their probabilities. Then it obtains the candidate frequent 2-itemset, 3-itemsets up to n-itemset by evaluating their probabilities in Eq. (2) and the result of the first step. Last it scans the database again to refine the candidate frequent itemsets to the frequent itemsets.

This paper also presents an efficient method for confirming the parameters $a$, $b$ in Eq. (2) by association analysis. If the parameters $a$, $b$ are seemless, it will omit some association rules which users are interested in. The solution for this problem is for parameter $a$ and $b$, then choose the best. The method of grey relational analysis is widely used in the association analysis of environment databases.

REFERENCES

1 Agarwal S, Agrawal R, Deshpande P M (1996) On the computation of multidimensional aggregates. VLDB*96, Bombay, India.
2 Aumann Y, Lindell Y (1999) A statistical theory for quantitative association rules. KDD*99, San Diego, CA.

(Continued on Page 116)