An Efficient Association Rule Mining Method for Personalized Recommendation in Mobile E-commerce

Xiaoyi Deng[1] Chun Jin[1] Yoshiyuki Higuchi[2] C. Jim Han[3]

[1]Institute of Systems Engineering, Dalian University of Technology, Dalian, China
[2]Faculty of Symbiotic Systems Science, Fukushima University, Fukushima, Japan
[3]Department of Information Technology and Operations Management, Florida Atlantic University, Boca Raton, USA

Abstract

The association rule mining (ARM) is an important method to solve personalized recommendation problem in e-commerce. However, when applied in personalized recommendation system in mobile e-commerce(MEC), traditional ARMs are with low mining efficiency and accuracy. To enhance the efficiency in obtaining frequent itemsets and the accuracy of rules mining, this paper proposes an algorithm based on matrix and interestingness, named MIbARM, which only scans the database once, can deletes infrequent items in the mining process to compressing searching space. Finally, experiments among Apriori, CBAR and BitTableFI with two synthetic datasets and 64 different parameter combinations were carried out to verify MIbARM. The results show that the MIbARM succeed to avoid redundant candidate itemsets and significantly reduce the number of redundant rules, and it is efficient and effective for personalized recommendation in MEC.

Keywords: Association rules mining, Transaction matrix, Interestingness, Personalized recommendation, Mobile e-commerce

1. Introduction

Nowadays, the personalized recommendation has became an significant part of for mobile e-commerce(MEC) services, and the association rules mining, ARM technology is the core and key method for personalized recommendation. And the effect of personalized recommendation directly depends on the quality and quantity of association rules[1]. In MEC, the customers' interests and demands change along with the time and the number of association rules increases exponentially in pace with the growing number of customers and size business database. That is, customers are sensitive to the time and quality of personalized recommending services. Thus, how to obtain customers' interests dynamically in the shortest time has become a key issue[2].

Currently, ARM algorithms mainly base on the Apriori algorithm proposed by Agrawal in 1993, which has a large number of applications in various types of e-commerce recommendation system, such as Amazon, eBay and Taobao. However, these Apriori based algorithms still have some disadvantages[3]:

(1) In the process of searching frequent pattern, it requires multiple database scans, as many as the longest candidate itemsets C_k, to generate frequent itemsets L_k from C_k. Meanwhile, it produces lots of redundant candidate itemsets. These cause the algorithm low efficiency, and

reduce the efficiency of recommending services, which is incompatible with requirements of personalized recommendation services in MEC.

(2) When mining association rules, it is based on the support and confidence, which produces dozens of redundant association rules that leads low rule generation accuracy. This will reduce the QoS of personalized recommendation.

In order to improve the Apriori algorithm, this paper presents a novel ARM based on transaction matrix and interestingness, and it has some significant differences from Apriori based algorithms. And it is verified and validated to be more capable for personalized recommendation service in MEC.

2. Analysis of Apriori Algorithm

2.1. Apriori Algorithm

The Apriori algorithm is one of the most classic and popular frequent itemsets mining algorithms, which many other algorithms base on. Let $I=\{I_j|j=1,2,\ldots,n\}$ be a set of n distinct items. Let $D=\{t_i|i=1,2,\ldots,m\}$ be a set of m transactions including I, where each transaction t_i is a set of items such that $t_i \subseteq I$. Relevant concepts in Apriori algorithm can be stated as follows:

Support, *Sup*: If a transaction t_i contains all items of I_r, i.e., $I_r \subseteq t_i$, it is said t_i supports an itemset $I_r \subseteq I$. And the support of an itemset I_r in D is the ratio of the number of transactions support I_r and the number of all transactions in Eq(1).

$$Sup(I_r) = \left\|\{t_i | I_r \subseteq t_i, i=1,2,\ldots,m\}\right\| / \|D\| \quad (1)$$

Confidence, *Conf*: The confidence of an rule $R{:}A \rightarrow B$ is the support of the set of all items in rule R divided by the support of the former rule, shown in Eq.(2).

$$Conf(R: A \rightarrow B) = Sup(A,B) / Sup(A) \quad (2)$$

Frequent itemset, *FI*: An *FI* is such an itemset that its support is greater than or equal to a specified minimum support threshold, denoted Sup_{min}.

The Apriori algorithm consists of two main steps:

Step1: Finding frequent itemsets L_k in candidate itemsets C_k, whose occurences exceed a predefined minimum support threshold.

Step2: Generating association rules from frequent itemsets found in Step1 with constrains of minimum confidence threshold.

The Apriori algorithm is very simple and easy to be implemented. However, there are serval problems in Apriori algorithm:

(1) The algorithm needs multiple database scanning operation to construct candidate itemsets. In the scanning process, if L_k is generated, databases scan need to be performed k times, by which the performance is dramatically affected. And the generation process of candidate itemsets C_k need to compare all items in each two itemset to check if they have the same first k-items and it employs lots of CPU time.

(2) When generating association rules, it is based on the support and confidence, and these is no any pertinent optimization according to characteristics of the problem of personalized recommendation, which produces a mass of redundant association rules that not only cost lots of computing time, but also leads to low service quality.

2.2. Related Works

To improve the efficiency of traditional mining algorithm, Tsay[4] proposed an efficient method called the FIUT, which enhance the efficiency in mining frequent itemsets. Dong[5] designed a BitTableFI algorithm to compress database for quick candidate itemsets generation. After that, the Index-BitTableFI algorithm is proposed by Song[6], to avoid the redundant operations on frequent itemsets checking.

On the other hand, in order to improve the quality of e-commerce recommending services, Enrique[7] introduced a new method based on the users' behaviors in web-based information systems towards automatic personalized recommendation, Then, Wang[8] proposed a method that can automatically detect hidden profit building opportunities through discovering soft rules from transactions. To make best meet customers' needs and interests, a hybrid algorithm was proposed by Forsati[9] to solve the web page recommendation problem.

Previous Researches above had made several improvements and extensions of the Apriori algorithm, which mostly improved efficiency of mining frequent 1-Itemsets and frequent 2-Itemsets. However, those algorithms remain powerless when dealing with the growing mass of business information. Especially when mining frequent *k*-Itemsets (*k*>2), BitTableFI and other Apriori liked algorithms are still with low efficiency; In the other respect, towards reducing the number of redundant frequent itemsets, some researches suggested using different interestingness measures for determination of association rules[10], but they were not is suitable for personalized recommendation [11].

3. MIbARM Descriptions

To improve the algorithm's efficiency and towards the high quality of personalized recommendation, this paper proposes a matrix-and-interestingness-based ARM algorithm, named MIbARM, which only scans the database once, deletes non-frequent items in the mining process to compressing searching space in order to avoid redundant candidate itemsets, and improves the traditional Piatetsky Shapiro, PS formula[13] to avoid redundant rules. The Design of MIbARM is shown in two phases: frequent itemsets mining and association rules generation.

3.1. Frequent itemsets mining based on transaction matrix

In order to overcome the drawback that Apriori scans database repeatedly to create candidate itemsets, this paper introduces a process based on transaction matrix, which only scans the database once, deletes non-frequent items in the mining process to compressing searching space in order to avoid redundant candidate items. And this process is specifically optimized to deal with frequent *k*-Itemsets when *k*>2.Relevant concepts and process of MIbARM are stated as follows.

Definition.1 *Transaction Matrix*, *TM*: The database with m transactions is denoted by D, and $I=\{I_j|j=1,2,\ldots,n\}$ is a set of n distinct items in D. According to the database D, a two-dimensional matrix $T=(T_1,T_2,\ldots,T_m)'=(t_{ij})_{m\times n}$, is built, whose rows stand for transactions in D. If the i-th transaction T_i includes the item I_j, the t_{ij} =1; else, t_{ij}=0.

Definition.2 *Absolute Support*, *SupA*: In T, the absolute support of item I_j is the sum of all elements of j-th colmun, shown in Eq.(3).

$$SupA\left(I_j\right)=\sum_{i=1}^{m} t_{ij}\ ,\ \ i=1,2,\ldots,m \qquad (3)$$

Definition.3 *Support*, *Sup*: In T, the support of item I_j is is the ratio of its absolute support and the number of all transactions m, shown in Eq.(4),

$$Sup\left(I_j\right)=SupA\left(I_j\right)\Big/m=\sum_{i=1}^{m} t_{ij}\Big/m \qquad (4)$$

Definition.4 *Mining Frequent 1-Itemsets*: It's assumed the minimum support is Sup_{min}, and the candidate 1-Itemsets C_k is equal to set I. The frequent 1-Itemsets L_1 can be obtained by Eq(5).

$$L_1 = \left\{ I_j \middle| Sup\left(I_j\right) \geq Sup_{\min}, j = 1,2,...,n \right\} \quad (5)$$

Then, remove columns including item I_j', whose support is smaller than the minimum support Sup_{min}. Finally, remove those rows in T that $\sum_{j=1}^{n} t_{ij} < 2$. The number of rows and columns is denoted by m_2 and n_2 ($m_2 \leqslant m, n_2 \leqslant n$).

Definition.5 *Mining Frequent 2-Itemsets*: At first, a m_2-order Hermite matrix is constructed according to T:

$$H = T'T = \begin{pmatrix} \sum_{i=1}^{m_2} t_{i1}t_{i1} & \sum_{i=1}^{m_2} t_{i1}t_{i2} & \cdots & \sum_{i=1}^{m_2} t_{i1}t_{in_2} \\ \sum_{i=1}^{m_2} t_{i2}t_{i1} & \sum_{i=1}^{m_2} t_{i2}t_{i2} & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ \sum_{i=1}^{m_2} t_{in_2}t_{i1} & \cdots & \cdots & \sum_{i=1}^{m_2} t_{in_2}t_{in_2} \end{pmatrix} \quad (6)$$

In the matrix H, diagonal elements are equal to the absolute support of items in L_1 i.e. $SupA(L_1)$. And non-diagonal elements equal to the absolute support of member in candidate 2-Itemsets C_2, i.e. $SupA(C_2)$; Then, a symmetric matrix $G=(g_{ij})$ is established in accord with the Hermite matrix H.

Where, $g_{ij} = g_{ji} = \begin{cases} 1, & h_{ij} \geq mSup_{\min} \quad i \neq j \\ 0, & h_{ij} < mSup_{\min} \quad i \neq j \\ 0, & \qquad\qquad\qquad i = j \end{cases}$.

Obviously, the frequent 2-Itemsets L_2 can be gotten from the matrix G, where $L_2=\{C_2|g_{ij}=1, i,j=1,2,\ldots,P_1,\ i\neq j\}$. At last, those rows in T that $\sum_{j=1}^{n} t_{ij} < 3$ should be removed and the number of rows and columns is denoted by m_3 and n_3.

Definition.6 *Mining Frequent k-Itemsets*: First of all, a matrix U^k is built by zero vector and candidate k-Itemsets C_k, which is generated from frequent (k-1)-Itemsets.

$$U^k = \left(U_1^k, U_2^k, ..., U_m^k\right)' = \left(C_k, C_k, ..., C_k, 0, ..., 0\right)'$$

Set a matrix $W=(w_{ij})_{m\times m}=U^kT_k'$, define a special function $F(w_{ij})$, shown in Eq.(7). For the matrix W, $F(W)$ is shown as Eq.(8).

$$F\left(w_{ij}\right) = \begin{cases} 1, & w_{ij} = k \\ 0, & w_{ij} < k \end{cases}, \ i,j = 1,2,\ldots,m \quad (7)$$

$$F\left(W\right) = \begin{pmatrix} F\left(\sum_{i=1}^{n} u_{1i}t_{1i}\right) & \cdots & F\left(\sum_{i=1}^{n} u_{1i}t_{mi}\right) \\ \vdots & \ddots & \vdots \\ F\left(\sum_{i=1}^{n} u_{mi}t_{1i}\right) & \cdots & F\left(\sum_{i=1}^{n} u_{mi}t_{mi}\right) \end{pmatrix} \quad (8)$$

After that, a m-dimensional column vector v is constructed by the matrix W, shown in Eq.(9).

$$v = \left(v_i\right)_{m\times 1} = \left(\sum_{i=1}^{m} f\left(w_{1i}\right), \ldots, \sum_{i=1}^{m} f\left(w_{mi}\right)\right)' \quad (9)$$

In the vector v, every v_i (v_i>0) equals to the absolute support of candidate k-Itemsets, $SupA(C_k)$, and $Sup(C_k)$ = v/m. Thus, the frequent k-Itemsets L_k can be figured out, and the member number of L_k is denoted by P_k. In the end, remove rows in T that $\sum_{j=1}^{n} t_{ij} < k+1$.

When k=k+1, if the the member number of frequent (k-1)-Itemsets L_{k-1} is less than k, the candidate k-Itemsets generated by the L_{k-1} will be null. Thus, the mining process of frequent k-Itemsets stops; else, go on mining frequent (k+1)-Itemsets.

3.2. Interestingness based association rules generation

On the other side, this paper modifies the traditional interestingness measure method, PS formula, to avoid generating redundant association rules in order to improve the quality of personalized recommending services. The modified PS formula is stated as follows.

Definition.7 *Interestingness*: For a random rule like R: $A \rightarrow B$ in D, the interestingness of R is defined in Eq.(10),

$$I(R) = \left(P(A,B) - P(A)P(B)\right)\left|V(R)\right| \quad (10)$$

where, the function $V(R)$ is the validity[10], $I \in [-0.25, 0.25]$. For any rule like R: $A \rightarrow B$, its validity is the difference between two probabilities, shown in Eq.(11).

$$V(R) = P(A,B) - P(\bar{A},B) \qquad (11)$$

From Eq.(10) and Eq.(11), it can be known that the modified PS formula considers the rules' validity, and it is still consistent with three principles of the defining interestingness measures proposed by P. Shapiro[12].

3.3. Procedure of MIbARM

```
Input a database $D$ of $m$ transactions, the set of $n$ distinct items $I$, $Sup_{min}$ and $I_{min}$
  Set candidate itemsets $C_k=\Phi$, Frequent itemset $L_k=\Phi$, association rules set $R=\Phi$
  Transform database $D$ to matrix $T$
  Begin
  Remove columns that $\sum_{i=1}^{m} t_{ij} < m \cdot Sup_{\min}$
  Let $C_1$=I, $L_1$=Φ
  For $j=1; j \leqslant n; j{+}{+}$
     Calculate the $Sup(I_j)$
  if  $Sup(I_j) < Sup_{min}$
     Delete $I_j$ from $C_1$
  endif
  Set $L_1= C_1$, denote the number of rows and columns by $m_1$, $n_1$. Remove rows that $\sum_{j=1}^{n} t_{ij} < 2$
  Let $C_2=\{I_iI_j|i,j=1,2,\ldots,m_1\}$, $L_2=\Phi$
  do{Construct Hermite matrix $H$
        if  $Sup(I_iI_j) < Sup_{min}$
           Delete $I_iI_j$ from $C_2$
        endif
  }while(Length of $C_2$ doesn't change)
  Let $L_2=C_2$, denote the number of rows and columns by $m_2$, $n_2$. Remove rows that $\sum_{j=1}^{n} t_{ij} < 3$
  Let $C_k=\{c_l|l=1,2,\ldots,m_2\}$, $L_k=\Phi$
  do{do{ Construct matrix $W$
           Calculate the $Sup(C_k)$
        if  $Sup(c_l) < Sup_{min}$
           Delete $c_l$ from $C_k$
        endif
  }while(Length of $C_k$ doesn't change)
  Set $L_k=C_k$, denote the number of rows and columns by $m_k$, $n_k$, Remove rows that $\sum_{j=1}^{n} t_{ij} < k+1$
  }while($C_k=\Phi$)
  Generate rules $R_c$ by $L_2$, $L_3$,…, $L_k$
  do{ Calculate the $I(r_i)$
  if  $I(r_i) > I_{min}$
        Add $r_i$ into $R$
  endif
  }while($R_c \neq \Phi$)
  Output $R$
  End
```

4. Performance Evaluation

To evaluate the efficiency of this proposed algorithm, experiments are carried out among Apriori, CBAR[4], BitTableFI[6] and MIbARM with two synthetic datasets, D10K.T10.I5.N5 and D50K.T20.I10.N5, which are provided by IBM Almaden Quest research group [3,5]

The running time of those four algorithms on two synthetic datasets is shown in Fig.1, with the setting of $Conf$=0.2, I_{min}=0.15, and Sup_{min}={7.5%, 5%, 2.5%, 1%, 0.75%}.

The Fig.1 consists of two charts which are runtime comparison results of Apriori, CBAR, BitTableFI and MIbARM on two datasets under different minimum support threshold. The experimental results show that running time increases rapidly with the decreasing of the Sup_{min} in both Apriori and CBAR. In BitTableFI, the growth of computation time is relatively slow, while there is barely increasing in MIbARM. The average running time of MIbARM is only 1.2% of Apriori, 1.4% of CBAR and 8.5% of BitTableFI, and it increases nearly linearly. Thus, MIbARM outperforms other methods in efficiency.

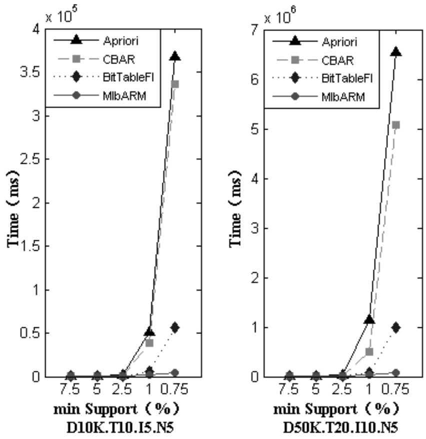

Fig. 1: The comparison of time of algorithms on two datasets.

Because Apriori, CBAR and BitTableFI are all based on support and confidence, only Apriori and MIbARM are chosen to mining rules on two data sets. In order to test the influences of support, confidence and interestingness, experiments include four levels: 0.75%, 0.5%, 0.25% and 0.1% for Sup_{min}, confidence is 0.4, 0.3, 0.2 and 0.1 and I_{min}={0.2, 0.15, 0.1, 0.05}. The experiment scale is 2×2×4×4=64. Comparisons of rules' number of Apriori and MIbARM on two data sets are shown in Fig.2 and Fig.3

From Fig.2 and Fig.3, the rule number of interestingness-based MIbARM is less than that of Apriori based on support and confidence, under 64 different conditions of parameter combinations. In Apriori, a mass of redundant rules are conducted with rapid decreasing of the Sup_{min}, which makes the number of association rules increase exponentially. Compared with Apriori, the number of rules of MIbARM grows relatively slowly; in the meanwhile, the average number of rules in MIbARM is only 31% of that of Apriori. Clearly, the MIbARM has the effect of reducing the number of redundant rules.

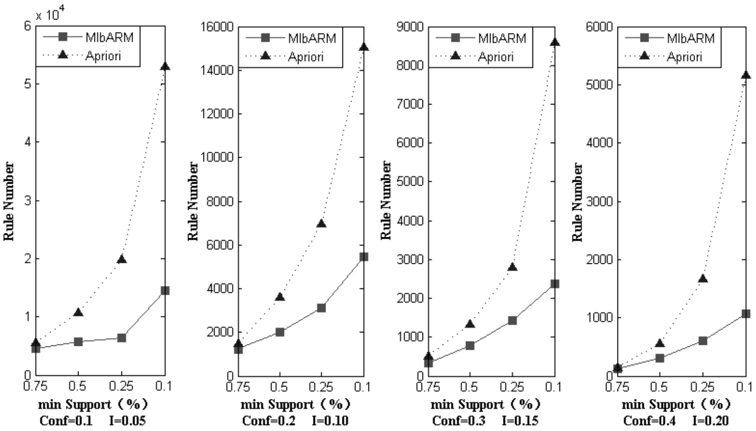

Fig. 2: Comparison of rules' number of Apriori and MIbARM on D10K.T10.I5.N5.

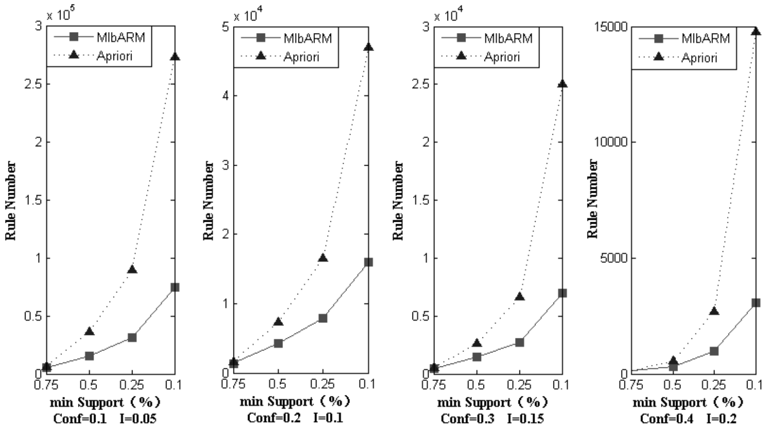

Fig. 3: Comparison of rules' number of Apriori and MIbARM on D50K.T20.I10.N5.

5. Conclusions

This paper proposes a novel algorithm, named MIbARM for solving the frequent itemsets generation and redundant association rules problems. MIbARM scans the database only once and uses simple matrix operations to minimize the searching space of candidate itemsets in order to avoid redundant candidate item. And it includes a modified mechanism of the rule' interestingness measure, which can avoid mining redundant rules and improve the service quality of personalized recommending.

The proposed algorithm was tested with two synthetic datasets comparing with Apriori, CBAR and BitTableFI and results show that MIbARM succeeds to avoid redundant candidate itemset and significantly reduces the number of redundant rules. Compared with Apriori, CBAR and BitTableFI, the MIbARM is the most efficient and effective one for personalized recommendation in MEC.

Acknowledgment

This work is sponsored by the Major Program of the National Natural Science Foundation of China (70890080, 70890083).

References

[1] S.H. Liao, C.M. Chen. Mining information users' knowledge for one-to-one marketing on information appliance[J]. Expert Systems with Applications, 2009, 36(3): 4967-4979.

[2] Y.Y. Zhang, J.X. Jiao. An associative classification-based recommendation system for personalization in B2C e-commerce applications[J]. Expert Systems with Applications, 2007, 33(2):357-367.

[3] J.W. Han. Frequent pattern mining: current status and future directions[J]. Data Mining and Knowledge Discovery, 2007, 15(1):55-86.

[4] T.J. Tsay. FIUT: A new method for mining frequent itemsets[J]. Information Sciences, 2009, 179(11):1724-1737.

[5] J. Dong, M. Han. BitTableFI: An efficient mining frequent itemsets algorithm[J]. Knowledge-Based Systems, 2007, 20(4):329-335.

[6] W. Song, B.R. Yang, Z.Y. Xu. Index-MaxMiner: A new maximal frequent item set mining algorithm[J]. Interna-

tional Journal on Artificial Intelligence Tools, 2008, 17(2):303-320.

[7] L. Enrique. Towards personalized recommendation by two-step modified apriori data mining algorithm[J]. Expert Systems with Applications, 2008, 35(3):1422-1429.

[8] F.H. Wang. On discovery of soft associations with most fuzzy quantifier for item promotion applications[J]. Information Sciences, 2008, 178(7): 1848-1876.

[9] R. Forsati, M.R. Meybodi. Effective page recommendation algorithms based on distributed learning automata and weighted association rules[J]. Expert Systems with Applications, 2010, 37(2):1316-1330.

[10] M. Ohsaki, H. Abe. Evaluation of rule interestingness measures in medical knowledge discovery in databases [J]. Artificial Intelligence in Medicine, 2007, 41(3): 177-196.

[11] P. Lenca, P. Meyer. On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid[J]. European Journal of Operational Research, 2008, 184(2):610-626.