An Effective Algorithm for Mining Indirect Association Rules

Duan Qiaoling*
*School of Computer Science, Wuhan Qingchuan University, No.9, Yuping Avenue, Wuhan 430204, China

*490539618@qq.com

Abstract. In order to solve the problem that it is necessary to scan the database many times and produce unnecessary frequent itemsets in the process of mining indirect association rules, a new algorithm FPI-mine based on FP-Tree is designed to mine the indirect association rules in transaction database. Firstly, FP-Tree is constructed, and then the indirect item pairs and intermediate support sets of all frequent items are found. Finally, all the indirect association rules are obtained by mining algorithm. It can directly mine indirect association rules without generating all frequent itemsets. Finally, the effectiveness of the algorithm is verified by experiments.

Keywords: Frequent Itemsets, Indirect Association Rules, FPI-mine, Transaction Database.

1. Introduction

Association rules, first proposed by R.Argawal in 1993, are a very important field in data mining[1]. In general, We mine frequent itemsets to determine strong association rules, but for some infrequent itemsets, interesting information between them is often overlooked. Considering data item pair (a,b), they rarely appear in the same data transaction, but data item pairs (a,M) and (b,M) appear very frequently, and a and b respectively meet the dependent conditions, then it is said that (a,b) indirectly associates through intermediate set M[2]. Rules that are indirectly associated with data item pairs through intermediate sets are called indirect association rules and are expressed as <a,b,M>. Indirect association rules provide a very valuable perspective, which can be applied in many aspects, such as product competition analysis, marketing, network recommendation, and biomedical analysis[3].

Tan et al. proposed indirect association rules mining algorithm, namely indirect-a[4]. The algorithm is mainly divided into two stages. Firstly, use Apriori algorithm to mine all frequent itemsets. Secondly, candidate indirect rules are generated from frequent itemsets, and all indirect association rules are obtained by pruning indirect rules[5]. The algorithm needs to scan the database
many times and produce many unnecessary frequent itemsets. This paper proposes an indirect association rules mining algorithm FPI-Mine based on FP-Tree in transaction database. The algorithm firstly constructs FP-tree, then finds out the indirect item pair set and intermediate support set of all frequent items, and finally gets all indirect association rules through FPI-mine algorithm. Through this method, indirect association rules can be mined directly without generating all frequent itemsets. Finally, the effectiveness of this algorithm is verified by experiments.

2. Relevant concepts
Suppose M is a set of large transaction databases, and \( x_s \), \( x_f \) and \( x_d \) are respectively represented as the item pair support threshold[6], intermediate support threshold and intermediate dependency threshold. We define the indirect item pair set and the intermediate support set as below:

**Definition one:** Suppose \( I \) is an itemset of database M, and the indirect item pair set is defined as:
\[
IIS(M) = \{ (a, b) \mid a \in L_1 \land b \in L_1 \land \sup(\{a\}) \geq x_f \land \sup(\{b\}) \geq x_f \land \sup(\{a, b\}) < x_s \}
\]

**Definition two:** Suppose L is all itemsets in database M and \( L_1 \) is an itemset in database M. The intermediate support set of \( a \) is defined as follows:
\[
MSS(a) = \{ M \mid M \in L \land \sup(M \cup \{a\}) \geq x_f \land \dep(M, \{a\}) \geq x_d \}
\]

Based on the above definitions, we can give the following theorems:

**Theorem one:** If \( <a, b, \{M\}> \) is an indirect association rule in data M, then:

1. \( <a, b > \in IIS(M) \);
2. \( M \in MSS(a) \) and \( M \in MSS(b) \);

**Theorem two:** suppose that \( \{a\}, \{b\} \) and M are three item sets in database M. If \( <a, b, \{M\}> \) satisfies the following conditions, then it is an indirect association rule:

1. \( <a, b > \in IIS(M) \);
2. \( M \in MSS(a) \) and \( M \in MSS(b) \);

**Theorem 3:** Suppose \( L_1 \) is a frequent item set in database M with a support threshold of \( x_f \). Through cross MSS(a) and MSS(b) \( <a, b > \in IIS(M) \), completely indirect association rules in database M are generated from all MSS(a) and IIS (M).

3. Indirect Association rules mining algorithm based on FP-Tree
FPI-mine algorithm is divided into two steps. Step one, FP-Tree is constructed, and indirect item pair support set IIS and intermediate support collection MSS are generated. Step two, the complete indirect association rules are mined through IIS and MSS. The algorithm is described as below:

**Algorithm:** FPI-mine;

**Input:** transaction database M; \( x_s \) : item pair support threshold; \( x_f \) : intermediate support threshold; \( x_d \) : intermediate dependency threshold;

**Output:** Completely indirect association rules;

1. Scan the database, construct L in descending order, and initialize FP-tree according to L;

2. Search FP-tree from bottom to top for each node

begin
Construct a prefix path and condition tree ending with node i

call fpi_mine(Fi)
end

(3) if IIS(M) ≠ ∅ then
for all itempair <a, b> in IIS(M) do
MS ← MSS(a) ∩ MSS(b)
if MS ≠ ∅ then
for all mediator M in MS do
output <a, b, M>

The fpi_mine(Fi) function is defined as follows:

(1) Start with the condition tree Fi for each node j
(2) if the size of I is 1 and j’s count < minimum itempair support count then add <i, j> to IIS (D)
   else if j’s count > minimum itempair count then
   if IS(j, i) > t_d then
   add m to MSS(j)
end if
(3) Create the conditional tree Fij
(4) call fpi_mine(Fij)

Suppose the L = {a_1, a_2, ..., a_n} is an set of frequent itemsets arranged in descending order of support from high to low in database M. Firstly, construct initialized FP-tree according to L. Then, for each item in L according to the support degree from low to high, the fpi_mine function is called to find the IIS and MSS set containing the corresponding item.

The function of the subfunction fpi_mine(a_n) is to determine whether to incorporate into the intermediate support set by checking (a_n). Moreover, it recursively mines (a_n) when the item is locally frequent.

After mining the indirect item pair set IIS(D) and MSS containing item a_n, delete all a_i nodes in FPI. Then, mine the set that contains the item a_{n-1} but does not contain a_n.

Similarly, if we can find all IIS(M) and MSS(a), we can generate all indirect association rules in the database according to theorem three.

4. Experiment and analysis

In order to prove the effectiveness of fpi_mine algorithm, we compare the experimental results with the two versions of the indirect algorithm. They use Apriori algorithm and FP-growth algorithm to mine frequent itemsets, namely Indirect-A and Indirect-F. By using Visual Studio 2019, the experiment was carried out on a machine with 1.6ghz and memory of 16G.

The test data we use comes from a synthetic data set to imitate the market basket data. In the experiment, we use two synthetic data. The first is T10.I5.D20K, which includes 250 items and 20,000 transactions, and the second is T10.I5.D50K, which includes 250 items and 50000 transactions. Of the two data sets the average transaction length is 10, and the average minimum frequent item set length is Fig.1 and 2 show the experiment results. They depicts a comparison of the running time of three algorithms namely, FPI-mine, Indirect-A and Indirect-F, in different intermediate set support
thresholds. In the experiment, the item pair support threshold and the intermediate set support threshold were set as the same, and the dependency threshold was set as 0.1. From the data analysis shows that the FPI-Mine algorithm is better than the other two algorithms. In high support threshold, the performance of FPI-mine and indirect-F is not significantly different, and both are superior to indirect-A. As the support threshold gets lower and lower, the gap between Indirect-F and FPI-mine and the gap between FPI-mine and Indirect-A becomes larger. Moreover, from the figure, we can see that the data curve of FPI-mine is quite flat, indicating that the running time of algorithm increases slowly with the decrease of support threshold.

![Graph showing running time comparison](image1)

**Fig.1** Comparison of running time of synthetic data set T10.I5.D20K

![Graph showing running time comparison](image2)

**Fig.2** Comparison of running time of synthetic data set T10.I5.D50K

5. Conclusion
We propose an indirect association rule mining algorithm FPI-mine based on FP-tree in transaction database. The algorithm first constructs FP-tree, then finds out all the indirect item pair sets and intermediate support sets of all frequent items, and finally obtains all indirect association rules through mining algorithms. Through it, indirect association rules can be mined directly without generating all frequent itemsets. Finally, the effectiveness of the algorithm is verified by experiments.

In the next work, further in-depth research and improvement will be made on the related problems in this article:

- (1) The issue of scalability. Current mining algorithms mine indirect rules by compressing the database. If the data is very large, we need to scan the tree multiple times into memory. The future work is how to reduce disk input and output when the data in the database is very large[7].

- (2) Interest measurement problem. In this article, we use interest to measure the dependence of two itemsets. Future work will look for more interesting metrics to determine the importance of indirect association rules, and give appropriate evaluation measures to determine what is the correct measurement. The choice of threshold is also a problem and requires further investigation.

Acknowledgments
This work was supported by Science and Technology Research Project of Hubei Provincial Department of Education in 2019 no. B2019356.

References
[1] Agrawal R, Imielinski T and Swami A 1993 Mining association rules between sets of items in large databases Acm.Sigmod.Rec 22 pp207-216
[2] Wan Q and An A 2006 An efficient approach to mining indirect associations J. Intell. INF.SYST 27 pp135-158
[3] Tsuruoka Y, Miwa M and Hamamoto K 2011 Discovering and visualizing indirect associations between biomedical concepts Bioinformatics 27 pp111-119
[4] Pangning T, Kumar V and Srivastava J 2000 4th European Conf. on Principles and Practice of Knowledge Discovery in Databases(Lyon) 1910(Berlin:Springer Verlag) pp632-637
[5] Przemyslaw K 2009 Mining Indirect Association Rules for Web Recommendation Int.J. Appl.Math.C.Sci 19 pp165-186
[6] Kalpana W and J. W. Bakal 2014 Generation of Direct and Indirect Association Rule from Web Log Data Int.J.Inno.Res.Dev 3 pp193
[7] Wenyang L, Youen W and Chunhao C 2017 GIAMS: A generic approach for mining indirect association rules in data streams Intel.Data.Anal 21 pp177-194