Eng. & Tech. Journal, Vol. 29, No.11, 2011

Proposed Algorithm for Extracting Association Rule Depend on Closed Frequent Itemset (EACFI)

Dr. Emad k. Jbbar* & Yaser Munther*

Received on: 6/4/2011
Accepted on: 20/6/2011

Abstract

Association rules are important one of data mining activities. All algorithms of association rule mining consist of finding frequency of itemsets, which satisfy a minimum support threshold, and then compute confidence percentage for each k-itemsets to construct strong association rules'. Some of these association rules are not important because the extracted knowledge from their is found in other. Hence we proposed algorithm to construct only important association rules by depend on closed frequent itemset. An itemset is closed if one of its immediate supersets has the same support as the itemset. Finding these closed frequent itemsets can be of a great help to purge a lot of itemsets that are not needed to find association rules. So that aid to reduce execution time and work space of algorithm and end the algorithm at any level of k-itemset, without construct all K-itemset.

خوارزمية مقترحه لاستخلاص العلاقات الترابطية بالاعتماد على التكرار المنغلق

الخلاصة

ان العلاقات الترابطية هي احدى الانشطة المهمة في تنقيب البيانات وكل خوارزميات العلاقات الترابطية تحتوي على البحث عن التكرار للـ itemsets والتي تحقق حد عتبه معين لاصغر اسناد ومن ثم حساب نسبة الوثوقية لكل k-itemsets لبناء العلاقات الترابطية. ان بعض هذه العلاقات الترابطية ليست مهمه لان المعرفة المستخلصة منها موجوده في علاقات ترابطية اخرى ووجودها مجرد تكرار للمعرفة. الخوارزمية المقترحة في هذا البحث هي لبناء العلاقات الترابطية المهمه فقط بالاعتماد على تكرارية closed itemset حيث ان العناصر تكون مغلقه اذا كان احد الـ supersets له نفس الاسناد في الـ itemset وان ايجاد هذه العناصر له اهمية كبيره في التخلي عن او ابعاد عدد من الـ itemset التي ليس لها حاجة في ايجاد العلاقات الترابطية وهذا يساعد على التقليل من زمن التنفيذ والمساحة المطلوبه لعمل الخوارزمية وانهاء عمل الخوارزميه في اي مستوي من الـ k-itemset بدون بناء البقية.

1. Introduction

During the second half of the eighties digital information technology completed its victory in our modern world. Today nearly everything is "digitized". This development is not restricted to the obvious domains, like the Internet, common database applications, or electronic commerce. Even traditional domains of our

Everyday life increasingly depend on modern information technology.. As a result gathering data that mirrors our world has become fairly easy and rather inexpensive. Consequently during the last ten years specialized techniques have been developed that

***Computer Sciences Department, University of Technology/ Baghdad**

https://doi.org/10.30684/etj.29.11.9
2412-0758/University of Technology-Iraq, Baghdad, Iraq

can be subsumed under the term data mining. The main goal behind these methods is to allow the efficient analysis of even very large datasets.[1] Association rule mining has received a great deal of attention. Today the generation of association rules is one of the most popular data mining methods. The idea of mining association rules originates from the analysis of market-basket data. Their direct applicability to business problems together with their inherent understandability – even for non data mining experts – made association rules such a popular mining method. Moreover it became clear that association rules are not restricted to dependency analysis in the context of retail applications but are successfully applicable to a wide range of business problems. [2]

2. Association Rules

Association rule discovery, a successful and important mining task, aims at uncovering all frequent patterns among transactions composed of data attributes or items. Results are presented in the form of rules between different sets of items, along with metrics like the joint and conditional probabilities of the antecedent and consequent, to judge a rule's importance. It is widely recognized that the set of association rules can rapidly grow to be unwieldy, especially as we lower the frequency requirements. [3]The larger the set of frequent itemsets the more the number of rules presented to the user, many of which are redundant. This is true even for sparse datasets, but for dense datasets it is simply not feasible to mine all possible frequent itemsets, Experiments using several "hard" or dense, as well as sparse databases confirm the utility of our framework in terms of reduction in the number of rules presented to the user, and in terms of time. We show that closed itemsets can be found in a fraction of the time it takes to mine all frequent itemsets (with improvements of more than 100 times), and the number of rules returned to the user can be smaller by a factor of 3000 or more! (the gap widens for lower frequency values). As mentioned, association rules are a popular mining method for dependency analysis. An association rule is a rule, which implies certain association relationships among a set of objects (such as occur together or one implies the other"), in a database. Given a set of transaction, where each transaction is a set of literal (called items), an association rule is an expression of the form XY, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions of the database, which contains X, tends to contain Y. [4]

3. Frequent Itemsets [5]

A frequent itemset is an itemset that occurrs frequently. But "How frequent is enough frequent?" and How do we know what "frequent" means? 10 occurrences? 20? 100? Well, actually this is parameters that we, the miners, have to set. If only 1 customer bought S1, S2, S3, S4 this fact isn't worth any consideration. We call it not frequent. The parameter that we have to decide upon is called support of an itemset. So we need to concentrate on the problem of finding all itemsets which have the support that we previously set up. We call such itemsets as frequent itemsets. Let's say we are only interested in the items that have a 40% minimum support - in our example that means these itemsets (or

combinations of riders) should have been purchased by at least 4 customers out of 10. Only in this case an itemset becomes frequent. So the correct definition says that a frequent itemset is one that occurs in at least a user-specific percentage of the database. Now, when we know what a frequent itemset is, let's list down 2 major properties that will help us later on in defining algorithms to find the frequent itemsets:

- Every subset of a frequent itemset is also frequent. Also known as Apriori Property or Downward Closure Property, this rule essentially says that we don't need to find the count of an itemset, if all its subsets are not frequent. This is made possible because of the anti-monotone property of support measure - the support for an itemset never exceeds the support for its subsets. Stay tuned for this.
- If we divide the entire database in several partitions, then an itemset can be frequent only if it is frequent in at least one partition. Bear in mind that the support of an itemset is actually a percentage and if this minimum percentage requirement is not met for at least one individual partitions, it will not be met for the whole database. This property enables us to apply divide and conquer type of algorithms. Again, stay tuned for this too.

4. Maximal Frequent Itemsets

Well, setting up a support percentage for an itermset, solved only a part of the problem. Now at least we know what we want. We know how frequent an itemset should be to become worth considering it. But the toughest part is still unsolved. In order to find a frequent itemset we have to go through all the sub-itemsets which themselves are frequent due to the Downward Closure Property. Some of itemset have large frequency but it haven't confidence relations with other items So we unavoidably generate an exponential number of subpatterns that we might not really need. The definition says that an itemset is maximal frequent if none of its immediate supersets is frequent. The only one downside of a maximal frequent itemset is that, even though we know that all the sub-itemsets are frequent, we don't know the actual support of those sub-itemsets.[6]

5. Closed itemsets

An itemsets C subset I from D is a closed itemset iff h(C) = C. the smallest closed itemset containing an itemset I is obtained by applying h to I . we call h(I) the closure of I. The set of frequent closed itemsets uniquely determines the exact frequency of all itemsets, yet it can be orders of magnitude smaller than the set of all frequent itemsets. an itemset is closed if one of its immediate supersets has the same support as the itemset. Finding these closed frequent itemsets can be of a great help to purge a lot of itemsets that are not needed and to find, as I said above, the right associations rules. [5]

Example

Let us see table (1) as example of Closed and Maximal Frequent Itemsets. As you see in the below graph fig (1) which explain closed k-itemsets, all individual items ABCDE are frequent itemsets because their support in greater than 2 *(our minimum support)*. But only 3 of them are closed because E has the superset (BE), D has the superset (BD) and AE has the superset (ABE) having the

same support. The itemset (ABC) are frequent because it is presented in at least 2 of the contracts, but it hasn't confidence relations with other items. Now we can determine the following terms as show in fig (2) which explain condition of each k-itemsets:

Non frequent items: all itemset have frequent is zero like DE,DC,CDE,

Less than minsupport: all itemset have frequent less than minsupport (which is 2 in our example) like CE, BCE, ACE, ABD, and ABCE.

Non close frequent: all items have frequent equal or greater than minsupport but it haven't any confidence relations with other k-1 itemset (when $k \geq 2$) like ABC.

Close frequent: all itemset have frequent equal or greater than minsupport and have confidence relation with other k-1 items like BD with D and AE with E.

6. Proposed algorithm

All existence association rules algorithms have two sub problems: Find all sets of items (itemsets) whose support is greater than the user specified minimum support, and generate the desired rules by computing the Confidence for each itemsets. These two processes require many scanning for huge database, and that need too long time. So that may cause many difficulties to extraction association rules. In our proposal we reduce scan number and discover important association rules only depended on closed itemsets frequency between items. Each k_itemsets and k-1_itemset (*k >1*) which have equal frequencies and k-1_itemset is subset of k_itemsets can use to extraction association rules. These closed itemsets never can generate other k-1_itemsets because their frequency is used to generate k_itemsets so there is no other itemsets include this closed itemset.

7. Algorithm steps

Step 1: Prepare Database transaction table (1).

Step 2: Scan Database table (1) to find the number of occurrence (frequency) of each 1_itemset as in table (2) .

Step 3: Eliminate each 1-itemset that have frequent less than the minimum support (minsup = 2 / 9 = 22 % in our example and no itemsets eliminate)

Step 4: Generate 2-itemset from table (2) for reminder 1-itemset which have minsup ≥ 2 as in table (3).

Step 5:

5-1 Search for frequency in table (3) equal frequency in table (2) and 1_itemset $\subseteq$ 2_itemset such as in table (4).

5-2 Construct association rules

BD/D $\Rightarrow$ D $\rightarrow$ B and BE/E $\Rightarrow$ E $\rightarrow$ B and AE/E $\Rightarrow$ E $\rightarrow$ A

Step 6: Eliminate upper used 2_itemsets from table (3) and put the reminder in table (5).

6-1 Generate 3-itemset from table (5) for reminder 2-itemset as in table (6).

6-2 Eliminate each 3-itemset that have frequent less than the minimum support as shadow row in table(6)

Step 7: Repeated step 5 by using table (5) and table (6), put the result in table (7) and Construct association rules. ABE/AE $\Rightarrow$ AE $\rightarrow$ B

Step 8: Repeated step 6 with 3_itemset and put the reminder in table (8).

Now look to table (8) there is only one row of 3_itemset so cannot

generate 4_itemsets and that stop the algorithm.

8. Discussion

Now we discuss the results of our proposed algorithm compare with the results of A-priori algorithm by using the same data transaction in table (1).

The extracted association rules by using proposed algorithm (EACF) are:

D	⟶	B
E	⟶	B
AE	⟶	B
E	⟶	A
BE	⟶	A

Look table (4) you found superset (D and E) and itemsets (BD and BE) have frequency = 2 that mean D and E never can generate other itemsets because their frequent closed to BD and BE and construct association rules. So it can remove from table (2) because there is no other transaction has it in table (1). And then get the reminder to generate 3-itemset and so on as in table (7). Closed itemsets frequent aid to reduce the process of generates other k-itemsets and that reduces the execution time as well as reduces work space. The algorithm extracted association rules as in other algorithm like A-priori.

The extracted association rules by using A-priori with same data transaction are:

D	⟶	B
E	⟶	B
AE	⟶	B
E	⟶	A
BE	⟶	A

Look there are 100% of association rules are exactly same as in (EACF) and the A-priori.

9. Conclusions

Let's go back to our proposed algorithm (EACFI) and extract many notes such as:

1. The proposed algorithm extracted only important association rules.
2. The proposed algorithm doesn't need to construct all itemset because each closed itemset haven't another relation, so it not needs other itemset.
3. The execution time and work space area are reduced because we didn't need to generate all itemsets.
4. The proposed algorithm can stop at any k-1 itemset level if there is no close frequent.
5. For each k-1-itemset frequent equal to k-itemset frequent and k-1-itemset is subset of k-itemset , their existence association rules.

References

[1] Larose, Daniel T. **"Discovering Knowledge in Data: An Introduction to Data Mining".** John Wiley & Sons, Inc., Hoboken, New Jersey. 2005.

[2] Han, Jiawei , and Micheline Kamber. **"Data Mining: Concepts and Techniques"**. Second Edition. Morgan Kaufmann publications. 2006.

[3] Rakesh Agrawal , Heikki Mannila, R. Srikant, Hannu Toivenen and A. Inkeri verkamo**,"Fast discovery of association rules"**, Santiago de Chile, 1996.

[4] Rakesh Agrawal, Tomasz Imielinski and Arun Swami**."Mining association rules between sets of items in large database"**, Washington, USA, May 1994.

[5] J. Lloyd and V. Dahl and U. Furbach and M. Kerber, **"Mining inimal Non-**

Redundant Association Rules Using Frequent Closed Itemsets", year = 2000,

[6] Jianyong Wang and Jiawei Han and Ying Lu and Petre Tzvetkov, **"An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets",** IEEE Trans. Knowl. Data Eng, volume = 17, year = 2005

Table (1) Transactional data

Itemsets	Frequent
AB	4
AC	4
AE	2
BC	4
BD	2
BE	2

Table (2) 1_itemset frequent

TID	List of item_IDs
T1	A,B,E
T2	B,D
T3	B,C
T4	A,B,D
T5	A,C
T6	B,C
T7	A,C
T8	A,B,C,E
T9	A,B,C

Table (3) 2_itemset frequent

k-1_ Itemset	k_ Itemset	Freq	Associ Rule
D	BD	2	D B
E	BE	2	E B
E	AE	2	E → A

Table (4)

Itemsets	Frequent
AB	4
AC	4
AE	2
BC	4
BE	2

Table (5)

Itemsets	Frequent
A	6
B	7
C	6
D	2
E	2
Itemsets	**Frequent**
A	6
B	7
C	6
D	2
E	2

Table (6)

Itemsets	Frequent
ABC	2
ABE	2
ACE	1
BCE	1

Table (7)

k-1_ Itemset	k_ Itemset	Freq	Association Rule
AE	ABE	2	AE B

BE →	ABE	2	BE	A
BE →	ABE	2	BE	A

Table (8)

Itemsets	Frequent
ABC	2

ABCDE 0									
ABCD 0	ABCE 1	ABDE 0	ACDE 0	BCDE 0					
ABC 2	ABD 1	**ABE 2**	ACD 0	ACE 1	ADE 0	BCD 0	BCE 1	BDE 0	CDE 0
AB 4	AC 4	AD 1	**AE 2**	**BC 4**	**BD 2**	**BE 2**	CD 0	CE 1	DE 0
A 6	B 7	C 6	**D 2**	**E 2**					

Figure (1) Closed k-itemsets

Non Frequent Itemset	Non close Frequent	**ABE** Close Frequent to **AE**
Less Than Minsuport	**BE** Close Frequent to **E**	**BD** Close Frequent to **D**

Figure (2) Explain condition of each k-itemset