Developing A-priori Algorithm for Fast Mining Association Rules[+]

Ghazi Johnny [*]

الخلاصه:

خوارزميه التنقيب عن القواعد العلائقيه تعتبر من اكثر الخوارزميات انتشارا في مجال التنقيب A-priori عن القواعد العلائقيه ،تحتاج هذه الخوارزميه الى معياريين مهمين هما المساند والموثوق لاستخلاص القواعد العلائقيه ،تقوم هذه الخوارزميه بفحص قاعده البيانات عده مرات لاكتشاف القواعد العلائقيه ،في هذا البحث اقترحنا خوارزميه تقوم بتقليل الفحص المتعدد لقاعده البيانات الى فحص واحد فقط لاستكشاف القواعد العلائقيه ذات الموثوقيه العاليه مما اعطت نفس النتائج وكذلك قللت من الوقت والجهد واضافت قدره تحليليه على فهم النتائج كونها تركز على المجاميع الصغيره .

Abstract:

The mining association algorithm well known as A-priori is one of the most popular data mining algorithms, The mining association algorithm requires two parameters support and confidence to derive rules, The A-priori algorithm scans the database multiple passes to discover the rules, In this research we proposed an algorithm to reduce the multiple passes to just one pass to derive rules with high confidence, which gives the same results and also reduces the time and the effort, the algorithm gives analytical ability for understanding the results because it focusing on small item sets.

Introduction:

Organizations have been actively implementing data warehousing technology, which facilitates enormous enterprise wide databases. As a result, with an enormous amount of data stored in databases and data warehouses, it is increasingly important to develop powerful tools for analysis of such data and mine interesting knowledge from it. Data mining is a process of inferring knowledge from such huge data.

There are many good definitions of the term "Data Mining" but the core concept behind all of them is the same. IBM defines Data Mining as," The process of extracting previously unknown, valid and actionable information to make crucial business decisions ".

The objective of data mining is to identify trends, similarities, and patterns to support managerial decision making by analyzing the data in large database. Data mining technologies generally use algorithms and advanced statistical models to analyze data according to rules set forth by the particular application at hand. Data mining consists of three major components to achieve its goal. They are: Clustering/Classification, Association Rules, and Sequence Analysis.

[+] **Received Date 26/1/2009 Acceptance Date 5/10/2009**

[*] **Lecturer / Staff Development Center**

An Association Rule is a rule, which implies certain association relationships among a set of objects in a database. In this process people discover a set of association rules at multiple levels of abstraction from the relevant set(s) of data in a database. For example, one may discover a set of symptoms often occurring together with certain kinds of diseases and further study the reasons behind them. Since finding interesting association rules in databases may disclose some useful patterns for decision support, selective marketing, financial forecast, medical diagnosis, and many other applications, it has attracted a lot of attention in recent data mining research.

Mining association rules may require iterative scanning of large transactions or relational databases, which is quite costly in terms of processing. Therefore, there is a need to study efficient mining of association rules in long transaction and /or relational databases. Or in other word, it is necessary to evaluate efficiency of association rule mining among different algorithms [1].

A-priori algorithm:

Agrawal et al. [1, 2] proposed the A-priori algorithm that is used to discover the association rules. The problem of discovering association rules can be decomposed into two sub problems: (1) Find the sets of items (itemsets) that have the supports above the minimum support. (2) Use the frequent itemsets to generate the desired association rules.

Definition 1 The *support* of an itemset I, sup(I), is defined as the number of transactions in the database containing I. The *minimum support*, min_sup, is a user predefined threshold. An itemset is *frequent* if its support is not less than the *min_sup*. An itemset with k items in called a *k*-itemset.

The A-priori algorithm scans the database multiple passes to discover frequent itemsets. The main steps of discovering frequent itemsets can be listed as follows:

(1) First of all, the supports of individual items are counted, and the items with the supports not less than the *min_sup* form frequent 1-itemsets.
(2) Next, let L_k be the set of all frequent *k*-itemsets and C_k be the set of candidate *k*-itemsets. We use L_k to generate C_{k+1} which is a superset of L_{k+1}. By scanning the databases once to check if c is frequent, for each $c \in C_{k+1}$, we can find L_{k+1}. This process repeats recursively until no frequent itemset is found.

Definition 2 Let D be a set of transactions and $I = \{i_1, i_2, \ldots, i_m\}$, an itemset is a subset of I. Given X and Y are itemsets, an *association rule* is of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \Phi$, where the $sup(X \cup Y) \geq min\_sup$, and the *confidence* of $X \Rightarrow Y$ is not less than a predefined threshold, *min_conf*, the *confidence* of $X \Rightarrow Y$ is $\frac{\sup(X \cup Y)}{\sup(X)}$.

After discovering all frequent itemsets, the algorithm of generating association rules uses the subsets of a frequent itemset as antecedents to generate the rules. For example, given a frequent itemset *ABCD*, we first consider the subset *ABC*, then *AB*,

etc. If the support of a subset υ of a frequent itemset l dividing that of l is less than the minimum confidence, the support of any subset of a dividing that of l is less than the minimum confidence too. That is because the support of any subset $\acute{\upsilon}$ of υ must be as great as the support of a. Therefore, the confidence of the rule $\acute{\upsilon} \rightarrow (l - \acute{\upsilon})$ cannot be greater than the confidence of $\upsilon \rightarrow (l - \upsilon)$. The detailed algorithm of generating association rules is showed below.

Algorithm: finding all association rules
Input: all frequent itemsets
Output: all association rules
For all frequent itemset l_k, $k \geq 2$ **do**
    Call genrules(l_k, l_k);

//The genrules generate all valid rules $\tilde{a} \rightarrow (l_k - \tilde{a})$, for all $\tilde{a} \subset a_m$
Procedure genrules(l_k: frequent k-itemset, a_m: frequent m-itemset)
1) A = {(m-1)-itemsets a_{m-1} | $a_{m-1} \subset a_m$ };
2) **For** all $a_{m-1} \in A$ **do**
3) $conf$ = support(l_k)/support(a_{m-1});
4) **If** ($conf \geq min\_conf$) **then**
5) **Output** the rules, with confidence = conf and support = support(l_k);
6) **If** (m-1 > 1) **then**
7) **Call** genrules(l_k, a_{m-1}); // to generate rules with subsets of a_{m-1} as the antecedents
8) **End**
9) **End**

Lets try an example based on the transactional data for All Electronic branch shown in Table[1] suppose the data contain the frequent itemset K = { I1, I2, I5} with minimum support count is 2.
What are the association rules that can be generated from K?

The nonempty subsets of K are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2} and {I5}.
The resulting association rules are as shown below, each listed with its confidence:

1. I1 I2 ➔ I5 CONFIDENCE = 2/4 =50%
2. I1 I5 ➔ I2 CONFIDENCE = 2/2 =100%
3. I2 I5 ➔ I1 CONFIDENCE = 2/2 =100%
4. I1 ➔ I2 I5 CONFIDENCE = 2/6 =33%
5. I2 ➔ I1 I5 CONFIDENCE = 2/7 =29%
6. I5 ➔ I1 I2 CONFIDENCE = 2/2 =100%

When choosing high confidence only the second ,third and the last are output, the A-priori passes three scans for the transactional database Table [2, 3, 4], [3].

The proposed Algorithm:

To accomplish our aim which is to reduce the number of scans to one, let us discuss the following idea, when choosing set of items contains A, B, C the A-priori scans the database three times one for the sets which contains one item like A, B, C and the second scan is for the sets which contains two items like {A,B} , {A,C} , {B,C}, the third scan is for the sets with three items like {A, B, C}.
Since the number of itemsets can be obtained as follows:
no of itemsets = 2^n -1 where n is the number of items .

And by using the permutation of the giving items which obtaining by concatenations programming technique we can determine firstly the itemsets and we can simply compute the frequent of each itemset in the database by just one main scan.

So the Algorithm as follows:

```
Algorithm: finding all association rules
Input: all frequent itemsets
Output: all association rules
Begin
      Select items of association rules.
      Generate K-itemset
         Input: Selected items.
         Output: K-itemset-table.
          Begin
                K= 1
                For J=1 to length (S) do
                Begin
                      For I= 1 to length (S) – (K- 1) do
                       Begin
                            Get Concatenates of substring (S, I, K);
                             Write K-itemset-table;
                       End;
                        K = K+1
                 End;
           End;
       Applying A-priori.
            Calculate frequency for each K-itemset.
            Calculate K-support to determine min-support.
            Calculate confidence.
            Determine final association rules.
END.
```

The following steps show the application of the proposed algorithm to the previous example of the transactional data for All Electronic branch:

1. By choosing I1=A, I2 =B, I5 = E Figure [1].
2. Find the proposed relations (nonempty itemsets) Figure [2].
3. Calculate the frequent of each itemsets Figure [3].
4. Calculate the support to determine min-support (min-sup = 2).
5. Calculate confidence and Determine final association rules Figure [4].

And so the results will be as follows:

	CONFIDENCE					
1.	I1➔I2	67%	I1 I2	I1	4	6
2.	I1➔I5	33%	I1 I5	I1	2	6
3.	I2➔I1	57%	I1 I2	I2	4	7
4.	I2➔I5	29%	I2 I5	I2	2	7
5.	I5➔I1	100%	I1 I5	I5	2	2
6.	I5➔I2	100%	I2 I5	I5	2	2
7.	I1 I2➔I5	50%	I1 I2 I5	I1I2	2	4
8.	I1 I5➔I2	100%	I1 I2 I5	I1I5	2	2
9.	I2 I5➔I1	100%	I1 I2 I5	I2I5	2	2

Conclusions and Recommendations:

The following conclusions can be staffed:

1. The proposed algorithm offers one main scan instead of multi-scanning process and also shows the importance of the smallest sets.
2. Steps of A-priori results (1, 2, and 3) match the results of the proposed algorithm (7, 8, and 9).
3. Step 4 of A-priori results matches the results of the proposed algorithm steps (1, 2) by the analytical conclusion (When I1 occurs together with I2 in confidence equal 67% and also occurs together with I5 in confidence equal 33% and I2 occurs with I5 in confidence equal 29%, so it is easy to conclude that I1 occurs with both of them, with the minimum confidence close to 33%) and as a result of that we can conclude that I5 has affecting the results to have low confidence which is not obvious in A-priori results.
4. Step 5 of A-priori results match the results of the proposed algorithm steps (3, 4) by the analytical conclusion (same as the previous step and also concludes the influence of I5.
5. Step 6 of A-priori results matches the results of the proposed algorithm steps (5, 6) by the analytical conclusion.
6. All the rules with high confidence matched in both algorithms (confidence 100%).
7. The proposed algorithm gives powerful and meaningful analysis when Choosing the proper values of support and confidence.
8. Applying the same idea to CHARM and FP-growth algoritms.
9. Build up an interactive system and applying the proposed algorithm to have fast mining association rules system.

D: Transactional Database.

TID	List of items
T100	I1 ,I2 ,I5
T200	I2 ,I4

T300	I2 ,I3
T400	I1 ,I2 ,I4
T500	I1 ,I3
T600	I2 ,I3
T700	I1 ,I3
T800	I1 ,I2 ,I3 ,I5
T900	I1 ,I2 ,I3

Table [1] All Electronics Branch Example.

C1

Itemsets	Sup-count
I1	6
I2	7
I3	6
I4	2
I5	2

➔

L1

Itemsets	Sup-count
I1	6
I2	7
I3	6
I4	2
I5	2

Table [2] The first scan of the A-priori (Scan for count of each candidate).

C2

Itemset	Sup-count
I1 ,I2	4
I1 ,I3	4
I1 ,I4	1
I1 ,I5	2
I2 ,I3	4
I2 ,I4	2
I2 ,I5	2
I3 ,I4	0
I3 ,I5	1
I4 ,I5	0

➔ Compare and prune

L2

Itemset	Sup-count
I1 ,I2	4
I1 ,I3	4
I1 ,I5	2
I2 ,I3	4
I2 ,I4	2
I2 ,I5	2

Table [3] The second scan of A-priori (Generate C2 and Scan D for count of each Candidate).

C3

Itemset	Sup-count
I1 ,I2 ,I3	2
I1 ,I2 ,I5	2

➔

L3

Itemset	Sup-count
I1 ,I2 ,I3	2
I1 ,I2 ,I5	2

Table [4]The third scan of A-priori (Generate C3 and Scan D for count of each Candidate).

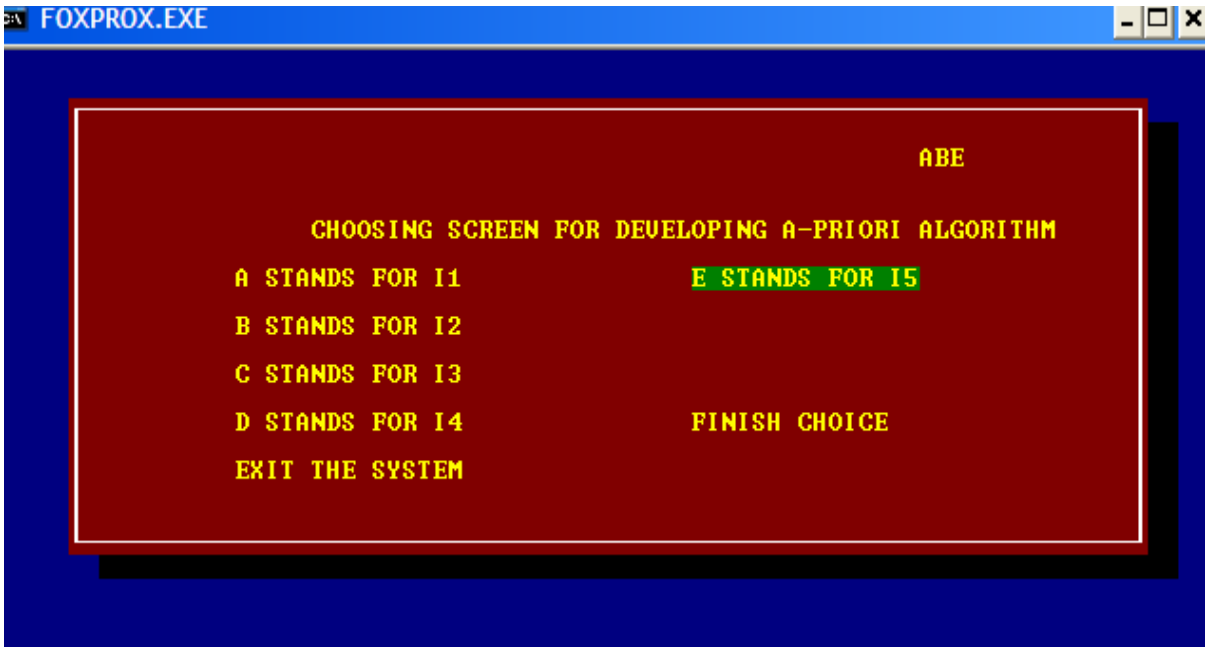

Figure [1] The choosing screen of the proposed Algorithm.

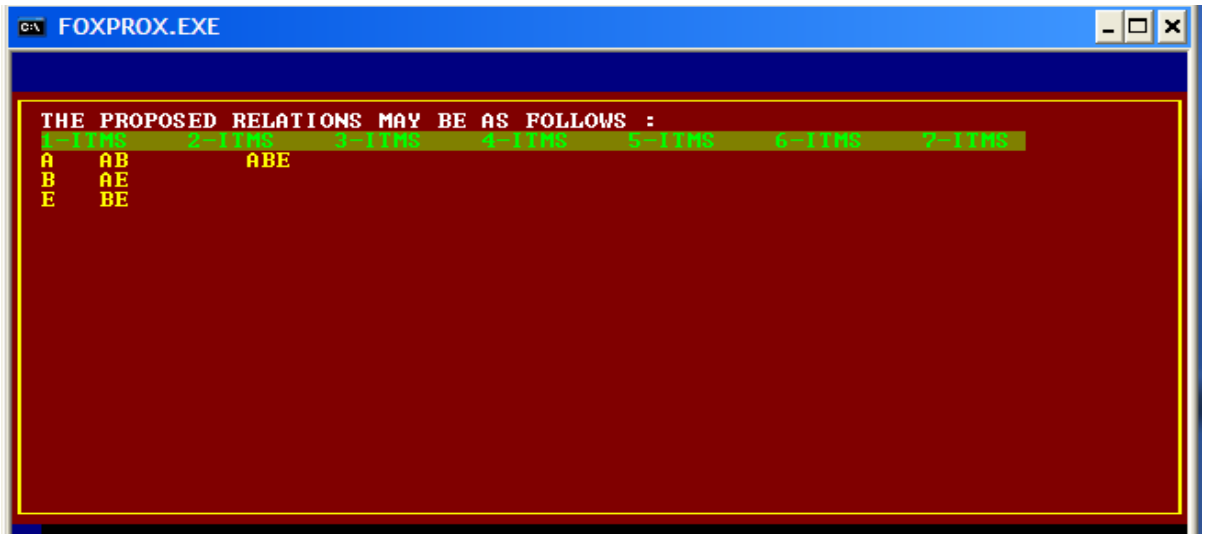

Figure [2] The proposed relations (non-empty itemsets).

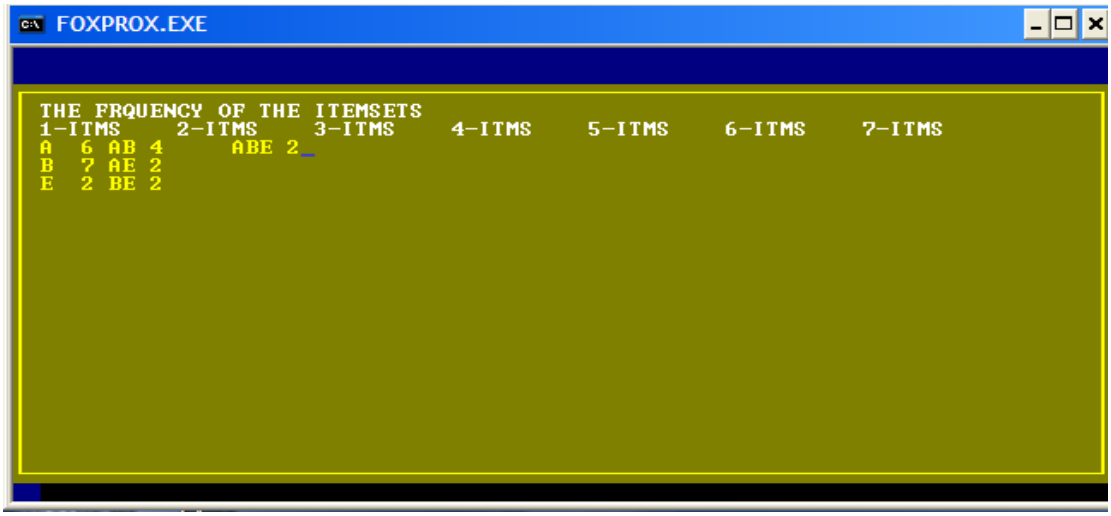

Figure [3] The frequent (occurrence) of each itemset.

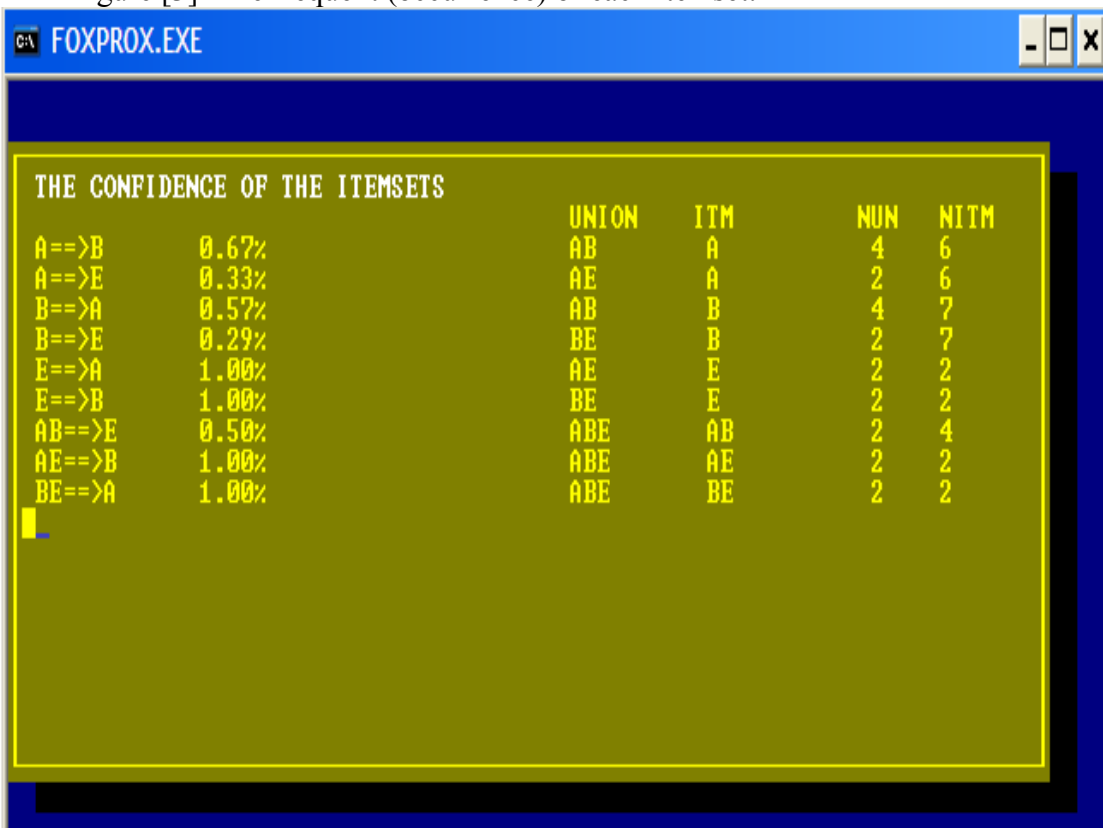

Figure (4) The Confidence and the Final Association Rules.

References:

[1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules”, In Proc. of Int. Conf. on Very Large Data Bases (VLDB’94), Santiago, Chile , pp. 487-499 , September 1994.

[2] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”, In Proc. 1993 ACM-SIGMOD Int. Conf. on Management of Data, Washington, D.C., pp. 207-216, May 1993.

[3] Jiawi Han , Micheline Kamber “Data Mining : Concepts and Techniques ”, Margon Kuffmann Publisher, 2001.