An Efficient Association Rules Algorithms for Medical Test Analysis

Dr. Ahmed Tariq Sadiq
Computer Science Department, University of Technology/Baghdad
Email: drahmaed_tark@yahoo.com

Alaa Sameer Ali
Computer Science Department, University of Technology/Baghdad
Email: Msc.alaa_sameer@yahoo.com

Received on: 14/9/2014 & Accepted on: 5/3/2015

ABSTRACT
Data Mining denotes mining knowledge from huge quantity of data. All algorithms of association rules mining include ‘first finding frequency of item sets, which accept a minimum support threshold, and then calculates confidence percentage for all k-item sets to construct robust association rules’. The trouble is there are some of algorithms that need more time for compute minimum support, minimum confidence and extraction larger item. In this paper one algorithm is proposed (enhanced reduces items Apriori algorithm) to reduce execute time. The proposed algorithm purpose to introduce algorithm to mine association rules to obtain fast algorithm by reducing execute time. Due to many experiments in (enhanced reduces items Apriori algorithm), this algorithm is very fast compared with (to pk-rules and to pk-non redundant rules) algorithms.

Keywords: Apriori Algorithm, Enhanced reduces Items A priori Algorithm, Experimental Results.

INTRODUCTION
Data mining is a process for ridding during lots of data to find beneficial information that helps in decision making. Preferable data means preferable results; there are bounded ways you can ameliorate your result from data mining, preferable data is one. Preferable data means you can construct more extensive and more precise models. Data mining needs lots of data and that means more sources of data but not all data is beneficial to extracting association rules. An association rule is a rule, which implies certain association connections among a set of objects in database. Extracting association rule is an main data-mining problem. Newly there has been huge research in designing fast algorithm for this mission, since the databases to mine are often very huge (measured in gigabyte and even terabytes) [1]. Data mining techniques which are exercised to medical data contain association rule mining for finding frequent types, prediction, classification and clustering [2]. Association rules are one of the promising aspects of data mining as knowledge discovery tool and have been widely explored to date, they allow capturing all possible rules that explain the presence of some attributes according to the presence of other attributes [3]. Traditionally data mining techniques were used in different fields. However, it is presented comparatively late into the Healthcare field. Nevertheless, as on today lot of research is existed in the literature. This has led to the evolution of smart systems and decision support systems in Healthcare field for precise diagnosis of diseases, predicting the strength different diseases, and remote health observation. Particular the data mining techniques are more helpful in predicting heart diseases, lung cancer, and breast cancer and so on. [2]. In this paper tried to produce new way to find association rules by using new technique may aid to reduce execution time and present a new method for extracting association rules. Section 2 describes the Related works. Section 3 introduce (Apriori algorithm concept). Section 4 introduce the (enhanced reduces items Apriori algorithm concept), Section 5 experimental results for the algorithm. Finally, Section 6 concludes the paper.
Related Works
The first manifestation of what is now called DATA MINING was in 1980s by research driven tools centered on lone mission. Through this time there was not much of a requirement to completely grasp multidimensional layers of data [4]. One dimensional analysis tools could solve these missions as constructing a classifier using decision tree or neural network tool, find cluster in data and data visualization of these tools were adequate for data analysis problem and to explain the results at that time, but with advancement of time using more than one of these tools became very uneasy for data analysis problems to solve users problems because the requires of users growth and users search for tools capable to drive largely by the investigation that the knowledge detection operation needs multiple kinds of data analysis [5].

In This paper presents some algorithms which have been proposed:

- In 2006 Ordonez, C., N. Ezquerra and C.A. Santana used constraint association rules in the medical area to minimize the number of detected types and introduced a greedy algorithm to calculate rule covers [6].
- In 2008 Razavi, A.R., H. Gill, H.Ahlfeldt and N. Shahsavar used decision tree convention to detect kinds of non-compliance with a post-mastectomy radiotherapy guideline, information which should help guideline writers and aid ameliorate the quality of on cological care [6].
- In 2009 Delen, D. used a diversity of data mining techniques, including artificial neural networks to determine factors affecting survivability from prostate cancer [6].
- In 2011 Abed, H. and L. Zaoui researched the trouble of mining interesting association rule from huge databases and proposed that background knowledge equips beneficial area information that may aid to subjectively estimate and explain the extracted rules. A framework was sophisticated for mining interesting rules from the medical area. Background knowledge aided to discriminate visible rules from bogus and to aid in rule translation. The framework was tested by using electronic medical data registries from the medical quality amelioration consortium [6].

Apriori Algorithm Concept

Apriori equips an iterative approach known as a level wise search, where k-item sets are used to discover (k+1)-item sets. First, the collection of frequent 1-itemsets is found by scanning the database to accumulate the count for each element, and collecting those elements that satisfy minimum support. The resulting collection is denoted L1. Next, L1 is used to find L2, the set of frequent 2-item sets, which is used to find L3, and so on, until no more frequent k-item sets can be found. The finding of each Lk needs one full scan of the database. To enhance the capability of the level-wise creation of frequent item sets, an important characteristic called the Apriori characteristic, presented is used to decrease the search space [7].

**Apriori characteristic** is based on:
If an item set I does not satisfy the minimum support threshold, min sup, then I is not frequent that is, P(I)<minsup. If an element A is inserted to the item set I, then the resulting item set cannot happen more frequently than I. Therefore, IUA is not frequent either. In public, to find Lk two step operation consisting of joins and prune actions:[8]

1- **the join step:**
(A) To find Lk , a collection of candidate k-item sets is created by joining Lk-1 with itself. This collection of candidates is denoted Ck. For the (k-1)-item set the elements are isolated such that li[1] < li[2] < . . . < li[k-1].
(B) The join , Lk-1 on Lk-1, is accomplished, where members of Lk-1 are combinable if their first (k-2) elements are in common. That is, members l1 and l2 of Lk-1 are combined if (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧ ... ∧ (l1[k-2] = l2[k-2]) ∧(l1[k-1] < l2[k-1]).

(C) The condition l1[k-1] <l2[k-1] simply includes that no duplicates are created. The resulting item set formed by combining l1 and l2 is l1[1], l1[2], . . . , l1[k-2], l1[k-1], l2[k-1].

2- **the prune step:**
(A) $C_k$ is a superset of $L_k$, that is, its members may or may not be frequent, but all of the frequent $k$-item sets are included in $C_k$.

(B) A scan of database to define the count of each candidate in $C_k$ would result in the definition of $L_k$.

Consequently by applying Apriori algorithm, and following the join and prune steps we lastly gain the frequent item sets.

**Algorithm:** Apriori Find frequent itemset using an iterative level-wise approach based on candidate generation [9].

**Input:** Database $D$, of transaction; minimum support; Threshold, min-sup in $D$.

**Output:** Association rules.

**Begin**

1. $L_1 = \text{find\_frequent\_1\_itemsets}(D)$;
2. For ($k = 2; L_{k-1} \neq \emptyset; k++$) {
   C$k$ = Apriori_gen($L_{k-1}$, min_sup);
3. For each transaction $t \in D$ {//scan D counts
   C$t$ = subset(C$k$, $t$); //get the subsets of $t$ that are candidates
4. For each candidate $c \in C_t$
   C$.count++;
5. $L_k = \{ c \in C_k | c$.count $\geq \text{min\_sup} \}$
6. return $L = \bigcup L_k$;

**End.**

**Figure (1-a) Apriori Algorithm**

**Procedure has\_infrequent\_subset (c: candidate k\_itemsets);**

// Use prior knowledge

**Begin**

1. For each $(k-1)$-subset $s$ of $c$
2. IF $s \notin L_{k-1}$ then return TRUE;
3. Return FALSE;

**End.**

**Figure (1-b) Apriori Algorithm Procedure has\_infrequent\_subset**

**Procedure Apriori\_gen(Lk-1:frequent(k-1)-itemsets; min\_sup: minimum support threshold)**

**Begin**

1. For each itemset $l_1 \in L_{k-1}$
2. For each itemset $l_2 \in L_{k-1}$
3. IF $(l_1[1] = l_2[1]) \land (l_1[2] = l_2[2]) \land \ldots \land (l_1[k-2] = l_2[k-2]) \land l_1[k-1] < l_2[k-1]$ then $c = l_1 \text{ join } l_2$; // join step: generate candidates
4. IF has\_infrequent\_subset($c,L_{k-1}$) then
   Delete $c$; // prune step: remove unfruitful candidate
5. Else add $c$ to $C_k$
6. Return $C_k$

**End.**

**Figure (1-c) Apriori Algorithm Procedure Apriori\_gen**

**Enhanced Reduces Items Apriori Algorithm (ERIA)**

The ERIA algorithm aims to produce an algorithm to mine association rules with aid of set theory and relationship concept to reduce execution time of mining association rules by extraction of a computation of items which contain the large rule with large confidence before computation of the support and the confidence for all rule. An association rule implies that there exists a relationship between all items which form it.
This proposal will introduce new algorithm for extraction of a collection of items which contain the large rule with large confidence. First the ERIA algorithm takes, (minimum support and minimum confidence equal to 0) and input collection of items with aggregate of each item must larger than minimum support and minimum confidence and then order these items descending, after that applies to the set from down to top, taking each two items from down if small item is equal to or larger than half large item then gather these two items else omit the small item and then continue this operation applying to all items in the (set1) until at the end two items remain. These two items are gather without action if small item is equal to or larger than half large item but if eventually one item remains this item is placed with nearest two items which are gather without gather with them and these two conditions are implemented only on (set1) after that (set2) results and then the same processes are applied to (set2) but without ordering descending for the (set2) and then the (set3) that is resulted from (set2).This is the final result set of items which contain the large rule with large confidence and then we are doing scan only to the final set for compute

\[ \text{Support} \ (X \Rightarrow Y) = \frac{\text{frequent} \ (X)}{\text{overall number of records in database}} \]

\[ \text{Confidence} \ (X \Rightarrow Y) = \frac{\text{frequent} \ (X \cup Y)}{\text{frequent} \ (X)} \]

for knowledge large item best than doing scan to all items for knowledge large item , The main procedure of ERIA :

ERIA Algorithm:

\[
\begin{align*}
\text{Input:} & \quad S=\sum A_i \text{, where } A_i \text{ is item}, T=A_i. \\
\text{Output:} & \quad \text{set of items contain large rule have large confidence.} \\
1. & \quad \text{Begin} \\
2. & \quad \text{For } i=\text{sizeof}(s) \text{ to } 0 \text{ decrement by } 2 \\
3. & \quad \text{Begin} \\
4. & \quad \text{If } (i=2) \text{ then goto step } 7 \\
5. & \quad \text{Else if } (i=1) \text{ then goto step } 15 \\
6. & \quad \text{Else if } (s(i-1)/2) \leq s(i) \text{ then} \\
7. & \quad \text{R(F)=s(i-1)+s(i)} \\
8. & \quad \text{L(F)=T(i-1) U T(i)} \\
9. & \quad F=F+1 \\
10. & \quad \text{Else Remove } s(i), T(i) \text{ from } (s) \text{ and } (T) \\
11. & \quad \text{End if} \\
12. & \quad \text{End if} \\
13. & \quad \text{End for} \\
14. & \quad L(F)=L(F) U T(i) \\
15. & \quad \text{For } j=0 \text{ to } F-1 \text{ increment by } 2 \\
16. & \quad \text{Begin} \\
17. & \quad \text{If } (j=1) \text{ then goto step } 28 \\
18. & \quad \text{Else if } (R(j+1)/2) \leq R(j) \text{ then} \\
19. & \quad \text{Res1(c)=R(j+1)+R(j)} \\
20. & \quad \text{Res2(c)=L(j+1) U L(j)} \\
21. & \quad C=C+1 \\
22. & \quad \text{Else Remove } R(j), L(j) \text{ from } (R) \text{ and } (L) \\
23. & \quad \text{End if} \\
24. & \quad \text{End for} \\
25. & \quad C=C-1 \\
26. & \quad \text{Res2(C)=Res2(C) U L(J)} \\
27. & \quad \text{Print the result } (\text{Res2}) \\
28. & \quad \text{End for} \\
29. & \quad C=C-1 \\
30. & \quad \text{Res2(C)=Res2(C) U L(J)} \\
31. & \quad \text{C=C+1} \\
32. & \quad \text{Else Remove } R(j), L(j) \text{ from } (R) \text{ and } (L) \\
33. & \quad \text{End if} \\
34. & \quad \text{End if} \\
35. & \quad \text{End for} \\
36. & \quad C=C-1 \\
37. & \quad \text{Res2(C)=Res2(C) U L(J)} \\
38. & \quad \text{Print the result } (\text{Res2})
\end{align*}
\]

Example of the ERIA Algorithm:-
1. Arrange these items in descending order, and then apply the set from down to top, take each two items from down if small item is equal to or larger than half large item then sum these two items else delete the small item.
2. The same operations that are used for calculating the (set 1) is done but without arranging in descending order for the (set 2) and then the (set 3).

### The Advantages and Disadvantages of the ERIA Algorithm:

#### The Advantages:-
1. Reducing the time and potential.
2. Obtaining the group containing large rule without need to compute the confidence and the support for all rules.
3. Delete some of the items not useful from start so there is no need to scan all items.

#### The Disadvantages:-
1. In this algorithm some items are not useful interference in final set but they are no more than three items.
2. If the comparison between sums of items is low, the small item will not be deleted.
Experimental Results
Medical data mining represents an main part in detecting diseases from ancient data of different areas. Data mining techniques have turn into a general research tool for medical researchers to determine and invest types and relations between a big number of variables, and make them able to portend the result of a disease using the ancient datasets. In this paper all algorithms are run on medical blood tests database which contains 10 items (blood sugar, cholesterol, triglyceride, creatinine, blood urea, potassium, sodium, calcium, alk-phosphatase, ferric) and 3120 patients were taken from (medical city hospital). In this work (top-k association rules, top-k non redundant rules, enhanced top-k association rules, enhanced top-k non redundant rules and enhanced reduces items Apriori algorithm) are run on the datasets while different the number of the parameter k from 1.000 to 10.000 with minconf=0 to evaluate its impact on the execution time of the algorithm. All experiments were performed on a computer with a core i5 processor running Windows 7 and 2 GB of free RAM. This paper is helpful to the doctors for auguring the relation between blood diseases and this operation is used in hospital. Then Table (1) to (4) show results of rules and show in other tables comparisons in time between all algorithms. And then show in Table (5) show comparisons in time between all algorithms.

Conf: 90%→80%

Table (1): Results of Rules have confidence between (90-80)

<table>
<thead>
<tr>
<th>No</th>
<th>Rules</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>K(low)→Na(low)</td>
<td>84.87%</td>
</tr>
</tbody>
</table>

Conf: 80%→70%

Table (2): Results of Rules have confidence between (80-70)

<table>
<thead>
<tr>
<th>No</th>
<th>Rules</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cr(high)→Bur(high)</td>
<td>79.34%</td>
</tr>
<tr>
<td>2</td>
<td>Bur(high)→Cr(high)</td>
<td>74.19%</td>
</tr>
<tr>
<td>3</td>
<td>Chol(high)→Tri(high)</td>
<td>72.09%</td>
</tr>
<tr>
<td>4</td>
<td>Cr(low)→Bur(low)</td>
<td>70.83%</td>
</tr>
</tbody>
</table>

Conf: 70%→60%

Table (3): Results of Rules have confidence between (70-60)

<table>
<thead>
<tr>
<th>No</th>
<th>Rules</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>K(high)→Bur(high)</td>
<td>67.3%</td>
</tr>
<tr>
<td>2</td>
<td>K(high)→Cr(high)</td>
<td>66.04%</td>
</tr>
<tr>
<td>3</td>
<td>Chol(high)→Glu(high)</td>
<td>65.89%</td>
</tr>
<tr>
<td>4</td>
<td>Tri(high)→Glu(high)</td>
<td>65.51%</td>
</tr>
<tr>
<td>5</td>
<td>Ca(low)→Na(low)</td>
<td>64.63%</td>
</tr>
</tbody>
</table>

Conf: 60%→50%

Table (4): Results of Rules have confidence between (60-50)

<table>
<thead>
<tr>
<th>No</th>
<th>Rules</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Na(low)→K(low)</td>
<td>59.43%</td>
</tr>
<tr>
<td>2</td>
<td>Tri(high)→Chol(high)</td>
<td>58.86%</td>
</tr>
<tr>
<td>3</td>
<td>Ca(low)→K(low)</td>
<td>53.38%</td>
</tr>
<tr>
<td>4</td>
<td>Alk. phosphate(low)→Cr(low)</td>
<td>50.36%</td>
</tr>
</tbody>
</table>
Table(5): Comparisons in time between algorithms

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K=1000</td>
</tr>
<tr>
<td>TopK-Rules Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>TopK-Non Redundant Rules Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>ERIA Algorithm</td>
<td>1</td>
</tr>
</tbody>
</table>

CONCLUSION

Based on the select of parameters, association rule mining algorithms can create an very big number of rules which drive algorithms to trouble from long execution time and large memory consuming. To manipulate this matter, we proposed (ERIA algorithm), an algorithm to discover the set of items contain The large rule have large confidence, where k is collection by the user. Experimental results show that ERIA algorithm has excellent accomplish, and that it is an advantageous instead to ancient association rule mining algorithms when the user obtain some known which rule is have large confidence.

REFERENCES