EXTRACTING ASSOCIATION RULES FROM DISTRIBUTED ASSOCIATION RULES

Emad Kadum Jabbar Alfatly and Rawia Tahrir Saleh Kadoori*

Department of Computer Science, University of Technology. Baghdad-Iraq.

**Department of Computer Science, College of Education/Ibn Al-Haitham, University of Baghdad. Baghdad-Iraq.*

Abstract

Mining for associations rules between items in large transactional distributed databases is a central problem in the field of knowledge discovery. When distributed databases are merged at single machine to mining knowledge it will required large capacity of storage, long execution time in addition to that; transferring a huge volume of data over network might take extremely much time and also require an unbearable financial cost.

In this paper proposed algorithm is presented toward saving communication cost over the network, central storage cost requirements, and accelerating required execution time. The algorithm consist of two parts: ***Part one:*** **E**xtracting **A**ssociation **R**ules **for D**istributed **A**ssociation **R**ules (**EAR4DAR**) Algorithm; aims to extract association rules for distributed association rules instead of extracting the association rules from a huge quantity of distributed data located at several sites. This is done by collecting the local association rules from each site and storing them in a file. These Local Association Rules turn in series of operations to produce association rules over the whole distributed systems. ***Part two:*** **A**ssociation **R**ules_map (**AR_map**) algorithm aims to get association rules by using AND logic operation which is suitable for representing association relations between items, since it gives indication for finding a relation or not. Additionally, this algorithm uses Karnough_map (K_map) propriety to reduce the duplicate and to generate accurate and logical results with saving time and storage space.

الخلاصة

أن التنقيب عن العلاقات الترابطية بين العناصر في حجم كبير من الحركات لقواعد بيانات موزعة يعتبر مشكلة رئيسية في مجال الأستكشاف المعرفي. وعندما تدمج هذه القواعد البيانية الموزعة في جهاز واحد للتنقيب عن المعرفة سوف يحتاج ذلك الى مساحة خزنية كبيرة ووقت تنفيذي كبير بالأضافة حجم البيانات المتناقلة عبر الشبكة والتي تحتاج الى وقت و تكلفة عالية جدا

في هذا البحث سيتم أقتراح خوارزمية التي ستوفر كثيرا في التكلفة اللازمة لعملية الأتصال عبر الشبكة و تكلفة متطلبات الخزن المركزي بالأضافة الى الخفض المعدل الزمني اللازم للتنفيذ. و أن هذه الخوارمية تتكون من جزئين:

الجزء الأول EAR4DAR :

تهدف هذه الخوارزمية الى أستخلاص علاقات ترابطية من علاقات ترابطية موزعة بدلا من أستخدام بيانات المراكز ذاتها, و يتم هذا من خلال تجميع العلاقات الترابطية المحلية(LAR) من كل مركز و تخزينها في ملف ومن ثم تمر هذه العلاقات الترابطية المحلية المجمعة بسلسلة من العمليات لتنتج علاقات ترابطية للنظام الموزع.

الجزء الثاني AR_map :

تهدف هذه الخوارزمية الى الحصول على علاقات ترابطية بستخدام الأداة المنطقية (AND) والتي تمثل أداة مناسبة للتعبير عن العلاقات الترابطية بين العناصر فهي تعطي مؤشر واضح و سريع الى وجود أو عدم وجود علاقة ترابطية بالأضافة الى أستخدام الكارنوف ماب (karnough-map) الذي ساعد من خلال خصائصه على أختزال التكرار و توليد علاقات أكثر دقة مع توفير الوقت و المساحة الخزنية.

Introduction

With the time, larger and larger amounts of data are collected and stored in databases, increasing the need for efficient and effective analysis methods to make use of the information contained implicitly in the data. The extraction of such potentially useful information is called data mining. Actually, many of these data sets are in real world, geographically distributed across multiple sites. To mine in such large distributed data sets, it is important to investigate efficient distributed algorithm to reduce the communication overhead, central storage requirements, and computation time.

Most of working fields are mining knowledge over centralized data set or partitioning it in many locations for improving computation processing; in this paper two proposed algorithms are introduced which focus on the principle of mining knowledge over geographical distributed systems which are: *Main Proposed Algorithm:* **E**xtracting **A**ssociation **R**ules **for D**istributed **A**ssociation **R**ules (EAR4DAR) Algorithm and *Secondary Proposed Algorithm:* **A**ssociation **R**ules_map (AR_map) algorithm.

Data mining

The extraction of useful and non-trivial information from the huge amount of data that is possible to collect in many and diverse fields of science, business and engineering, is called Data Mining (DM). DM is part of a bigger framework, referred to as Knowledge Discovery in Databases (KDD), which covers a complex process, from data preparation to knowledge modeling. Within this process, DM techniques and algorithms are the actual tools that analysts have at their disposal to find unknown patterns and correlation in the data. Typical DM tasks are classification, clustering or association rules and others [1].

DM is now bringing important contributions in crucial fields of investigations. Among the traditional sciences we mention astronomy, high energy physics, biology and medicine that have always provided a rich source of applications to data miners. An important field of application for data mining techniques is also the World Wide Web. The Web provides the ability to access one of the largest data repositories, which in most cases still remains to be analyzed and understood. Recently, Data Mining techniques are also being applied to social sciences [2].

Association Rules

Association rules are one of the promising aspects of data mining as knowledge discovery tool and have been widely explored to date, they allow to capture all possible rules that explain the presence of some attributes according to the presence of other attributes [3].

An association rule is a rule, which implies certain association relationships among a set of objects, 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 $X \rightarrow Y$, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions of the database, which contain X, tend to contain Y [1]. Association rules identify relationships between attributes and items in database such as the presence or absence of one pattern implies the presence or absence of another pattern. An association rule is an expression $X \rightarrow Y$ where $X= \{x_1, x_2 \ldots x_n\}$ and $Y= \{y_1, y_2 \ldots y_n\}$ are set of items with left hand side (LHS) and right hand side (RHS). The meaning of such rules is quite intuitive: given database (D) of transactions (T) where each transaction $T \in D$ is a set of items, $X \rightarrow Y$ which expresses that whenever a transaction T contains X, the T probably contains Y. Also the probability of rule strength is defined as the percentage of transactions containing Y in addition to X. The prevalence of rule is the percentage of transactions that hold all the items in the union. If prevalence is low, it implies that there is no overwhelming evidence that

items in X ∪ Y occur together. The rule X→Y has support(S) in (D) if the fractions of the transactions in (D) contain (X ∪ Y). The problem of mining association rules is to generate all association rules that have certain user-specified minimum support called (min-sup) and confidence (called min_conf) [4]. The important measures for association rules, support (S) and confidence (C) can be defined as: The support (S) of an association rule is the ratio (in percent) of the records that contain (X ∪ Y) to the total number of records in database[5].

Support(X→Y) =P(X∪Y)……2-1

Support(X→Y) =frequent(X∪Y)/total number of records in database..2-2

For given number of records, confidence (C) is the ratio (in percent) of the numbers of records that contain (X ∪ Y), to the number of records that contain X. thus, if we say that a rule has a confidence of 85% it means that 85% of the records containing X also contain Y. The confidence of rule indicates the degree of correlation in the database between X and Y. Confidence (X→Y =frequent(X∪Y)/frequent.2-3.

Confidence is also a measure of rules strength [6].

Proposed Algorithm

Part One

Extracting **A**ssociation **R**ules **for D**istributed **A**ssociation **R**ules **(EAR4DAR).**

EAR4DAR attempts to get association rules for distributed association rules. The basic behind this proposed algorithm depends on extracting association rules from a distributed association rules instead of extracting association rules from a large distributed data located at several sites. In other words; each site has responsibility to extract its own i.e. local association rules, and then EAR4DAR collects these association rules in a controller site to find out the global association rules, which are more accurate than those mined from all raw data located at distributed sites when they are collected together. EAR4DAR could play a significant task in distributed data mining since it works with many records (which represent association rules records of sites) instead of huge quantity of data records.

The EAR4DAR depends on simple fact that its member is part of a group. Since raw data for each site represents member and it is part of all data sites; all data sites represent a group and it consists of these members. Consider a system which consists of S = (S_1, S_2, S_3…) sites, S_1 has N_1 data records, S_2 has N_2 data records, S_3 has N_3 data records and so on. Finding association rules (A_1, A_2, A_3…) of sites (S_1, S_2, S_3…) requires mining in (N_1, N_2, N_3…) data records respectively. To extract the global association rules for the whole system requires (N_1+N_2+N_3+….) data records collected together at once; need large space of memory, long execution time and may cause losing some of association rules which it's important in its site. On the other hand, EAR4DAR requires (A_1+A_2+A_3+…) records to extract association rules from distributed association rules which are actually fewer and more accurate than (N_1+N_2+N_3+….) data records to find association rules. So EAR4DAR introduces good advantage to distributed database by isolating local analysis at each site from other for finding local association rules (LAR); and then local association rules for each site can be used as inputs to EAR4DAR to find global association rules. This implies reduction the communication overhead, central storage requirements, and computation times.

EAR4DAR Algorithm Steps:

Step 1: Collecting the local association rules from each site, and storing them in database file named **C**ollection of **A**ssociation **R**ules **D**ata**B**ase (CARDB) which has structure as shown in Table (1) where Association_From represents the right side of the association rule A→B and Association_To represents the left side of the association rule A→B.

Step2: Coding CARDB file by converting association rules of (CARDB) to binary representation and storing them in file named Binary Database File (BDB) as in following table (2).

Step3: Apply AR_map algorithm.

Part Two

Association **R**ules_**map (AR_map)** Algorithm.

Basically association rules algorithms apply to mine knowledge about relation between items; and this knowledge in statistical methods like A-priori is obtained by computing frequency of each K-item-sets with huge quantity of data records; but in real world the knowledge comes from one's information and repeating same information doesn't give new knowledge and will be just frequent without any useful.

Moreover EAR4DAR uses local association rules as input and these association rules are little so they are not valuable within A-priori. So AR_map algorithm has good new idea to get association rules by using AND logic operation which is suitable for representing association relation.

If A exists and B exists then association relation is true. If any of them does not exist then association relation is false. This idea makes use advantage of Karnough_map (K_map) for getting association rules without duplicate. So using AND logic operation and reducing table like Karnough_map (K_map) to shrink relations will decrease storage size, execution time and provide high performance.

This is done as follows: Let (AE) and (ABE) are two relations that are represented in binary (10001) and (11001) respectively and have the same item group (AE) and it's easy to get this result by using AR_map algorithm; (10001)AND(11001)= (10001) which equivalent (AE).

AR_map Algorithm Steps:

1. Construct binary table depending on number of items (instance as shown in Table (3) for 5 items).
2. Sign local association rules collected from all sites in BDB on table A at its cell position as shown in Table (3).
3. Apply all K_map propriety and ability to reduce cells by using AND logic operation.
4. Sign the result of AND logic operation in step 3 in table B as shown in Table (4).
5. Analyze table B and extracts new association rules through crossing rows with columns.

An Applicable Example

Example: Let's look at a system which has three branches distributed at three different sites S_1,S_2, S_3, and S_4 is company center for controlling the three sites and giving reports to the higher management to make decisions by extracting global association rules from distributed association rules. Each site (S_1, S_2, and S_3) has private database file which represents site transaction for five types of items as set below:

- Transaction Database file for $site_1$ (S_1) is shown in Table (5) and Association Rules file for sit_1 (S_1) is shown in Table (8).
- Transaction Database file for $site_2$ (S_2) is shown in Table (6) and Association Rules file for sit_2 (S_2) is shown in Table (9).
- Transaction Database file for $site_3$ (S_3) is shown in Table (7) and Association Rules file for sit_3 (S_3) is shown in Table (10)

Solution

Now to get Association Rules at control site (S_4), there are two techniques:

1 Traditional techniques (all sites transactions).
2 EAR4DAR Technique.

1- Traditional techniques (all sites transactions)

This technique uses Apriori algorithm [7] to compute global association rules from raw data for all sites and the results are shown in Table (11).

Note the results have only one association rule from all sites transactions while in actuality:
$Site_1$ has (14) association rules,
$Site_2$ has (2) association rules, and
$Site_3$ has (4) association rules.

2- EAR4DAR Technique

In this example EAR4DAR is used to find Association Rule from three Local Association Rules (LAR) A_1, A_2, and A_3 which are ***computed in parallel*** and independent at sites S_1, S_2, and S_3, as shown in Tables (8), (9), and (10).

Now EAR4DAR is used to extract association rules:

Step1: Merge A_1, A_2, and A_3 on Collect Association Rules DataBase file (CARDB) presented by Table (12).

Step2: Convert CARDB to binary layout and save it on Binary DataBase file (BDB) as shown in Table (13)

Step3: Implement AR_map algorithm with Binary DataBase file BDB to mine association rules; as following:

3-1 Use number of items to determine dimensions of Table A as shown in Table (14) and Table B as shown in Table (15). In this example number of items are five.

3-2 Sign cells of table A by association rules of BDB file on Table A as shown in Table (14).

3-3 Use AND logic operation with K_map ability to reduce cells as follows:

At the beginning search for each 8 sign-cells neighbored, if no then search for each 4 sign-cells neighbored if no then search for each 2 sign-cells neighbored in this example there are just 2 sign-cells neighbored, as:

1. Cell 24 and cell 25 which is a form closer to (24, 25) then 11000AND11001=11000 so the result is 24.
2. Cell 9 and cell 25 which is a form closer to (17, 25) then 01001AND11001=01001 so the result is 9.
3. Cell 17 and cell 25 which is a form closer to (17, 25) then 10001AND11001=10001 so the result is 17.
4. Cell 26 and cell 30 which is a form closer to (26, 30) then 11010AND11110=11010 so the result is 26.
5. Cell 14 and cell 30 which is a form closer to (14, 30) then 01110AND11110=01110 so the result is 14.
6. Cell 22 and cell 30 which is a form closer to (22, 30) then 10110AND11110=10110 so the result is 22.
7. Cell 28 and cell 30 which is a form closer to (28, 30) then 11100AND11110=11100 so the result is 28.
8. Cell 6 and cell 14 which is a form closer to (6, 14) then 00110AND01110=00110 so the result is 6.
9. Cell 12 and cell 28 which is a form closer to (12, 28) then 01100AND11100=01100 so the result is 12.

And then sign the results on Table B as shown in Table (15).

3-4 Analyze table B and extract association rules through crossing rows with columns as shown in Table (16), as:

- Cell 24 with 11000 binary representations has only AB relation leads to $A \rightarrow B$.
- Cell 9 with 01001 binary representation has only BE relation leads to $E \rightarrow B$.
- Cell 17 with 10001 binary representation has only AE relation leads to $E \rightarrow A$.
- Cell 6 with 00110 binary representation has only CD relation implies to $C \rightarrow D$.
- Cell 12 with 01100 binary representation has only BC relation leads to $C \rightarrow B$.
- Cell 14 with 01110 binary representation has only relation leads to $BC \rightarrow D$, $BD \rightarrow C$, $C \rightarrow B$, $D \rightarrow B$, $CD \rightarrow B$.
- Cell 26 with 11010 binary representation has only ABD relation leads to $AD \rightarrow B$, $BD \rightarrow A$, $D \rightarrow B$, $D \rightarrow A$.
- Cell 22 with 10110 binary representation has only ACD relation leads to $AC \rightarrow D$, $AD \rightarrow C$, $C \rightarrow A$, $D \rightarrow A$, $CD \rightarrow A$.
- Cell 28 with 11100 binary representation has only ABC relation leads to $AC \rightarrow B$, $BC \rightarrow A$, $C \rightarrow A$, $C \rightarrow B$.

Note the results are 18 association rules while in actuality:

$Site_1$ has (14) association rules,
$Site_2$ has (2) association rules, and
$Site_3$ has (4) association rules.

Comparison between EAR4DAR and Traditional Techniques

This section makes a comparison between EAR4DAR and traditional techniques to measure efficiency of the proposed algorithm comparison vectors given in Table (17). They are implemented in $site_4$ (the controller site) and applied directly to 1500.000 transactions or indirectly through the association rules of sites and then execution time charts are compared as shown in Figure (1) and Figure (3) and Association Rules results are compared as shown in Figure (2).

Table (17) gives details about the power of proposed algorithm when compared with that of traditional technique in its two approaches (on all raw data and on all association rules from each site) covering number of transactions works, execution time and storage required space, and finally number of association rules results which are named or called the **A**ssociation **R**ules (AR), as well as other vectors.

Accuracy of Results

The EAR4DAR was implemented to find the global association rules which are more accurate than the global association rules which were found from all of the raw data by using traditional technique is shown in table (17) and figure (2), since EAR4DAR guarantees correct and independent local analysis for each site. That is because it's keeping the private data at each site and works its association rules which are computed locally at own site, and then the global association rules mining from it.

Storage Cost

EAR4DAR works with many records (which are basically association rules records of sites) instead of huge quantity of records. Therefore, EAR4DAR will reduce required storage sized; as shown in table (17)

Communication Cost

Transferring a huge volume of data over network might take extremely much time and also require an unbearable financial cost. EAR4DAR saves time and money needed because it works on the distributed association rules of each site instead of using the raw data of all sites; as shown in table (17)

Execution Time

EAR4DAR needs less execution time because it works with association rules instead of all raw data. In other words EAR4DAR works with many records (which are basically association rules records of sites) instead of huge quantity of records as show in figures (1 and 3).

Conclusions

The conclusions which are drawn from implementing the proposed algorithm in real world and comparing its results with those that are obtained from the most famous traditional technique i.e. A-priori, are:

1. Applying proposed algorithm doesn't require huge quantity of data and that will reduce size of storage in controller site.
2. High performance in extracting association rules is carried out through reducing execution time and storage space.
3. Statistical methods aren't used to discover association rules.
4. Using AND logic operation makes it convinced to get 100% of relation out of the relation ratio that is required to compute the confidence, and that of course will avoid using mathematical operations to mine association rules.
5. Using AR_map participates in reducing some relations that are included indirectly in other relations and does not given new knowledge.
6. Extracting association rules from association rules gives the optimal case of the relations between sites.
7. A-priori fails to extract association rules from association rules over all sites, compared with efficient and powerful proposed algorithm.
8. Also threshold isn't required with proposed algorithm.

Table (1): Collect Association Rules DataBase (CARDB) file structured

Association_From	**Association_To**
A	B
E	B
E	A
AE	B
BD	A

Table (2): Binary Database File (BDB) structured

A	B	C	D	**E**
1	1	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	0	1
1	1	0	1	0

Table (3): Table A[4 x 8] K_map for 5 items structured

	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	✓9	11	10	14	15	13	12
11	✓24	✓25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

Table (4): Table B[4 x 8] K_map for 5 items structured

Table (5): Transactional data of site$_1$ (S_1)

TID	List of item IDs
T1	ABE
T2	D
T3	ABCDE
T4	BC
T5	ABD
T6	C
T7	BC
T8	C
T9	ABCD

Table (6): Transactional data of site$_2$ (S_2)

TID	List of item IDs
T1	ABC
T2	ABE
T3	D
T4	BC
T5	ABD
T6	BC

Table (7): Transactional data of site$_3$ (S_3)

TID	List of item IDs
T1	AE
T2	AE
T3	BCD
T4	ABE
T5	CD
T6	CD

Transaction data of Sites

Association Rules of Sites

Table (8): Association Rules of site$_1$ (S_1)

No	Association_Rules
1	A → B
2	E → B
3	E → A
4	AE → B
5	BE → A
6	AC → B
7	AC → D
8	AD → B
9	BD → A
10	CD → A
11	CD → B
12	ABC → D
13	ACD → B
14	BCD → A

Table (9): Association Rules of site$_2$ (S_2)

No	Association_Rules
1	A→ B
2	C → B

Table (10): Association Rules of site$_3$ (S_3)

No	Association_Rules
1	A→ E
2	E → A
3	D → C
4	C → D

Table (11): Association Rules by Apriori

Association Rules
E → A

Table (12): Collect Association Rules Database file (CARDB)

Branch_no	AR_NO	Item_From	Item_To
1	1	A	B
1	2	E	B
1	3	E	A
1	4	AC	B
1	5	AC	D
1	6	AD	B
1	7	AE	B
1	8	BD	A
1	9	BE	A
1	10	CD	A
1	11	CD	B
1	12	ABC	D
1	13	ACD	B
1	14	BCD	A
2	15	A	B
2	16	C	B
3	17	A	E
3	18	E	A
3	19	C	D
3	20	D	C

Table (13): Binary DataBase file (BDB)

A	B	C	D	E
1	1	0	0	0
0	1	0	0	1
1	0	0	0	1
1	0	1	1	0
1	1	1	0	0
1	1	0	1	0
1	1	0	0	1
1	1	0	1	0
1	1	0	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	0
1	1	1	1	0
1	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	0	0	1
0	0	1	1	0
0	0	1	1	0

Table (14): Table A [4 x 8] for 5 items

	000	001	011	010	110	111	101	100
00	0	1	3	2	✓6	7	5	4
01	8	✓9	11	10	✓14	15	13	✓12
11	✓24	✓25	27	✓26	✓30	31	29	✓28
10	16	✓17	19	18	✓22	23	21	20

Table (15): Table B[4 x 8] for 5 items

	000	001	011	010	110	111	101	100
00	0	1	3	2	✓6	7	5	4
01	8	✓9	11	10	✓14	15	13	✓12
11	✓24	25	27	✓26	30	31	29	✓28
10	16	✓17	19	18	✓22	23	21	20

Table (16): Association Rules by EAR4DAR with AR_map

Table (16): Continued

13	C → A
14	AC → D
15	AD → C
16	CD → A
17	AC → B
18	BC → A

No	Global Association Rules
1	A → B
2	E → A
3	E → B
4	C → D
5	BC → D
6	BD → C
7	CD → B
8	C → B
9	D → B
10	AD → B
11	BD → A
12	D → A

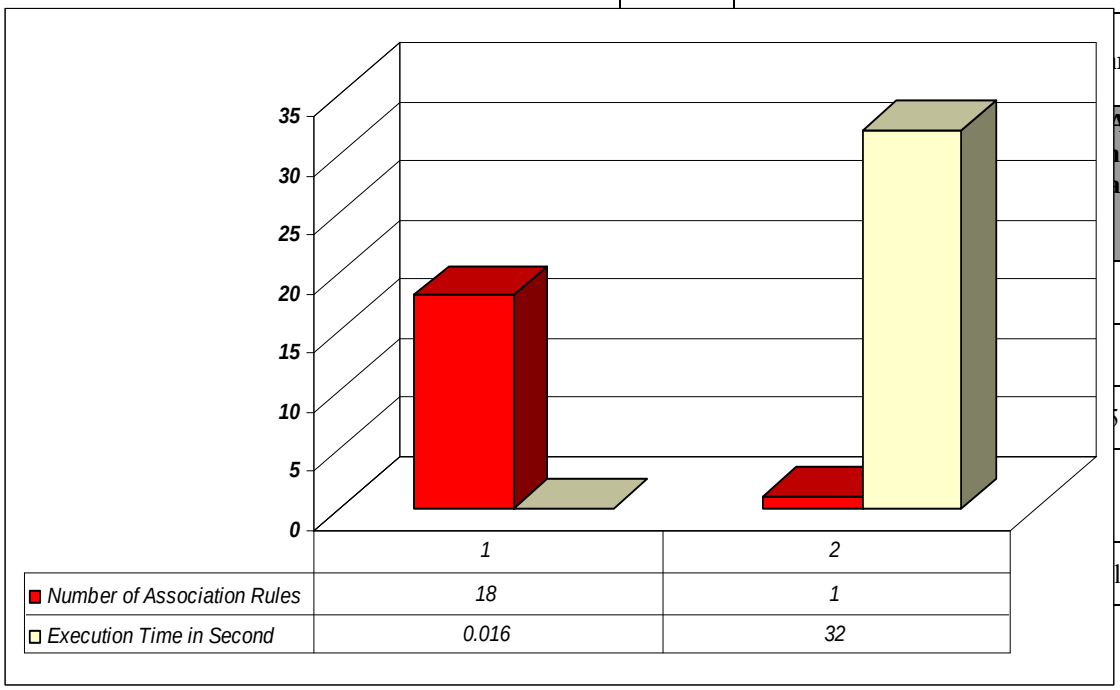


Figure (1): Comparison between execution time and results of the two implemented methods

References

1. Pieter Adriaans, Dolf Zantinge, **1998,** "*Data Mining*", Addision Wesley.
2. Jiawei Han, Micheline Kanmber, **2001,** "*Data Mining Concepts and Techniques*", Academic press, USA.
3. Charu C. Agrawal and Philip S. Ya, March **1998,** "*Mining large itemsets for association*

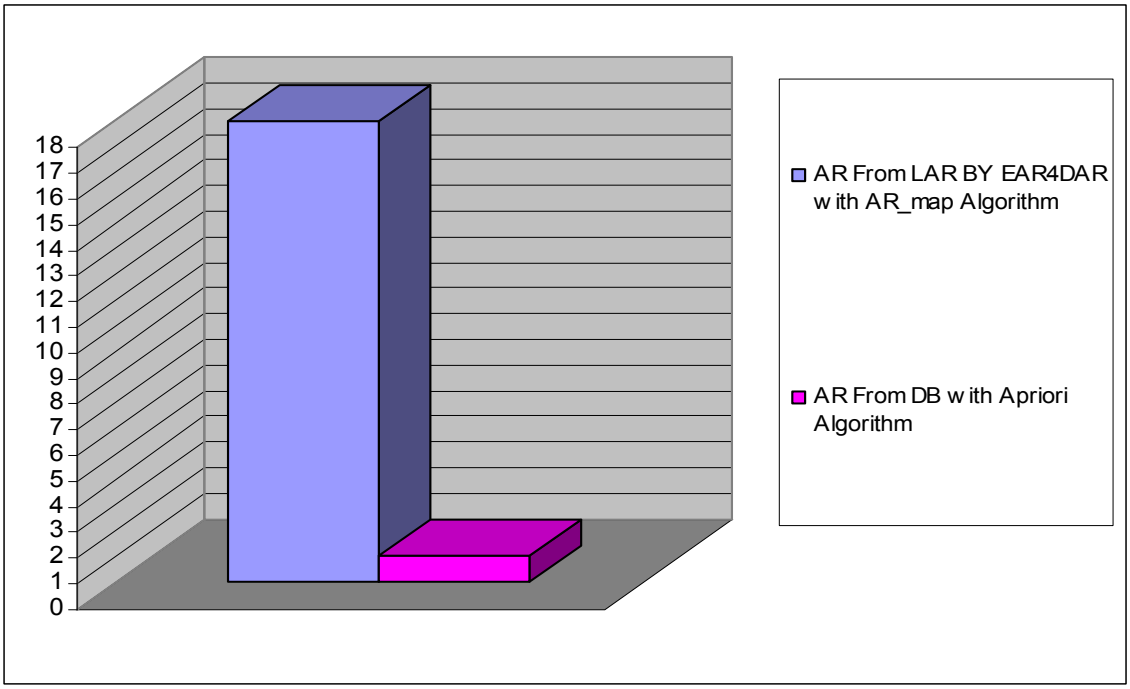
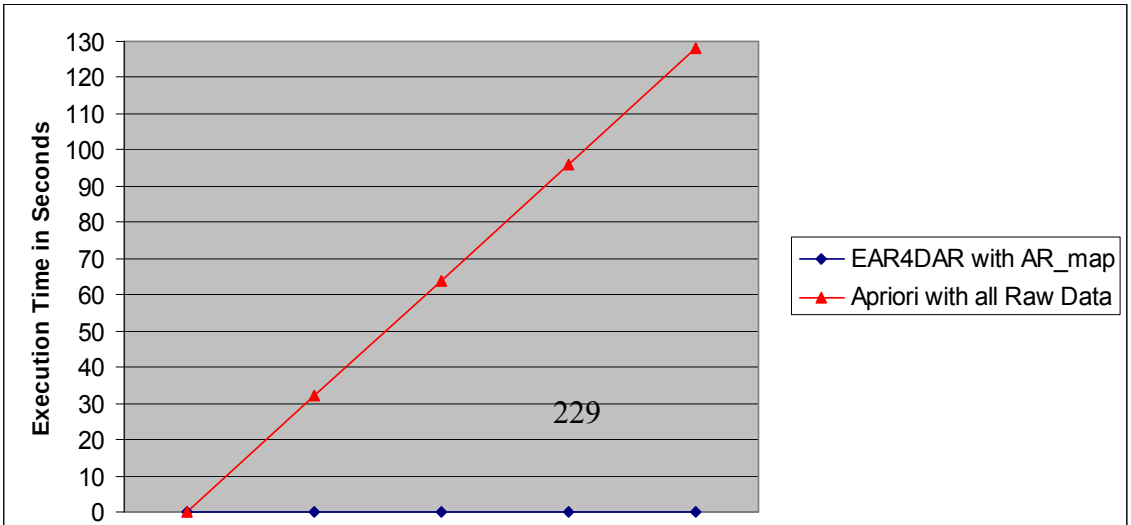


Figure (2): Comparison between Association Rules gets from two implemented methods



rules", Bulletin of the IEEE computer society Technical Committee on Data Engineering, 21(1).

4. Rakesh Agrawal, Heikki Mannila, Srikant R., Hannu Toivenen and A. Inkeri Verkamo, **1996,** "*Fast discovery of association rules*", Springer publisher, Santiago de Chile.
5. Sergy Brin, Rajeev Motwani, Jeffery D. Ullman and Sergy Tsur, May **1997,** "*Dynamic itemset counting and implication rules for market basket data*", proceeding of data (SGMOD97) Tucson, Arizona USA.
6. Chen M. S., J Han and P.S. YU, **1996,** "*Data mining an overview from a database perspective*", IEEE trans, knowledge and data Engineering.
7. Maimon Oded, Rokach Lior, **2005, "***The Data Mining and Knowledge Discovery Handbook***"**, Springer, USA.