World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:7, 2007

Post Mining- Discovering Valid Rules from Different Sized Data Sources

R.Nedunchezhian and K.Anbumani

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/10650/pdf

***Abstract*—**A big organization may have multiple branches spread across different locations. Processing of data from these branches becomes a huge task when innumerable transactions take place. Also, branches may be reluctant to forward their data for centralized processing but are ready to pass their association rules. Local mining may also generate a large amount of rules. Further, it is not practically possible for all local data sources to be of the same size. A model is proposed for discovering valid rules from different sized data sources where the valid rules are high weighted rules. These rules can be obtained from the high frequency rules generated from each of the data sources. A data source selection procedure is considered in order to efficiently synthesize rules. Support Equalization is another method proposed which focuses on eliminating low frequency rules at the local sites itself thus reducing the rules by a significant amount.

***Keywords*—**Association rules, multiple data stores, synthesizing, valid rules

I. Introduction

AUTOMATED data collection tools and mature database technology lead to tremendous amounts of data stored in databases and other information repositories [6]. There is huge amount of data but dearth of knowledge. Data mining offers a solution to this by extracting interesting information or patterns from the data in large databases [1][3][7]. In accordance to this, research has been done on mining these useful patterns.

Existing data mining techniques are not efficient enough to mine databases present in multiple branches where the data sources are of different sizes. Furthermore, little research has been done on post mining that involves gathering, analyzing and maintaining the mined rules.

A big organization may have multiple data sources, such as different branches. When all these data are forwarded for centralized processing, it might amass a huge database. So rather than forwarding the raw data, the association rules are forwarded to the centralized headquarter. However, the number of association rules might also be large for processing. Also association analysis or mining association rules at individual data sources is necessary.

Manuscript received January, 10, 2006.

R. Nedunchezhian is now heading the Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India. (Email: rachezhian@yahoo.co.in , Phone No. +919842523005).
Dr. K. Anbumani is the Director, School of Computer Science and Technology, Karunya Deemed University, Coimbatore-641114,India. Email:anbumani_k@yahoo.co.uk

The current synthesizing [12] model available focuses on similar sized data sources. When numerous data sources are considered, it is practically impossible to have similar sized data stores. To process data sources of different sizes, merging or splitting of the data sources can be done to make them of the same size. But these types of operations involve time complexity, huge manual work, cost consumption and may not be supported by all the data sources involved. Also this type of merging or splitting may involve security problems and sharing violation. Also identification of local rules for each data store may not be precise.

This paper proposes a new approach to discover the high frequency valid association rules from data sources containing different amounts of data. This approach is novel in that weights are assigned for each of the data sources based on their size and the rules occurring in the data sources. Specifically it focuses on how the valid rules are discovered in the union of all these different sized data sources. For dredging out the data sources which are below the threshold specified, a data source selection procedure is used. The threshold is calculated based on the size of the data source.

The approach proposed in this paper is different from the previously mentioned schemas. It is based entirely on the idea of heuristics involved in weighting. Because many factors reflecting properties of the data source size can be fused into the weighted model, previous model [12] is only a special case of the proposed model.

The second approach discussed in this paper aims at elimination of low frequency rules at the local sites itself. This is done by the equalization of the minimum support mentioned by the user for all the different sized data sources. This reduces the total number of rules passed to the centralized system and thereby reduces time and cost constraints involved to a certain degree.

A performance study is conducted that shows the proposed methods are efficient and effective. Furthermore, by considering the trends one can obtain more useful association rules.

The paper is organized as follows. Chapter II discusses the problem description and related work. Chapter III and IV elaborate the proposed works. Performance comparison is done in chapter V. Chapter VI explains the experimental setup. Finally chapter VII concludes the paper.

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/10650/pdf

II. Problem Description and Related Existing Work

Discovering rules from different sized data sources: The problem:

Mining rules focuses on the generation of association rules from different data sources. But all of the rules mined may not be of equal importance. Some rules may not be used frequently and are of little importance, such rules need not be considered.

Forwarding the rules for centralized processing will also lead to huge collection of rules. This massive amount of rules occupy ample amount of space and also involves lot of time for processing.

The number of rules passed from individual branches to the centralized system is therefore reduced by passing only the high frequency rules from individual sites. The high frequency rules are obtained by scouring out the infrequent rules. This is because more frequent rules have the larger chances to become valid during the association of all data sources.

The synthesizing of similar sized data sources may not be always possible. Because in reality all of the data sources may not contain equal amount of data. Splitting or merging of these different sizes to make similar size is likewise a difficult task.

So determination of valid rules from different sized data sources is highly important without altering the size of data sources.

While considering the similar sized data sources the supports and confidences of the rules are in the same range, but this is not the case with different sized data sources. The overall supports and confidences of data sources will in turn be based on the weight of the data source calculated from their corresponding sizes.

The number of data sources considered for discovery of valid rules may itself become huge and there is a need to limit the data sources sent for centralized processing.

This can be achieved by the Data Source Selection procedure.

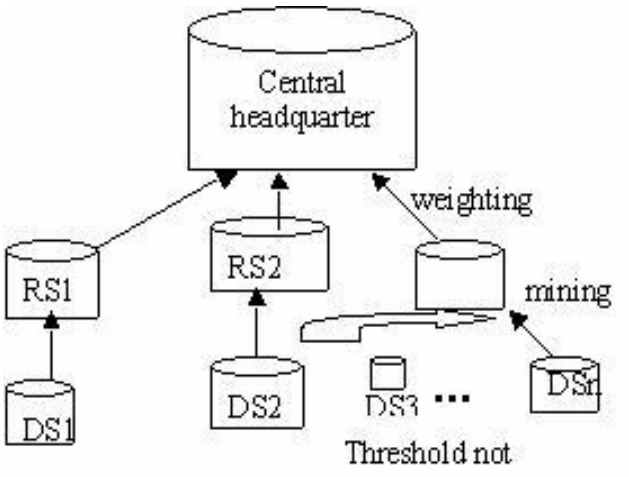


Fig. 1 Discovering valid rules from different sized data sources

Fig 1 shows the block diagram of the proposed model. As per the study and analysis stated above, the problem can be designed as follows:

Given n data sources of a big organization that are of different sizes, it focuses on 1) association analysis at the local sites and passing the high frequency rules mined for centralized processing 2) determining these high frequency rules to find valid rules considering the size of the data source 3) dredging out the low weighted data sources with their rules.

The second approach insists on filtering out the rules which are not satisfying the user specified minimum threshold by equalizing the supports of all the sites and then applying weighting model.

There are various data mining algorithms available for mining rules at local instances. Some of the popular algorithms used are [2] [5][10][11].

A. Our Approach

In many research applications, to gather, analyze, store and generate information, weighting is an efficient approach [12]. To mine fashionable rules, the proposed work constructs a weight model to highlight the novelty of data and its size by weighting. Gathering all rules together from different data sources of a big company might also amass a huge rule set. A procedure for eliminating rules is designed which employs a voting degree based on the number of data sources and the occurrence of the rules.

The objective is to focus on different sized data sources of a big company. The frequency of a rule is how much transactions support that particular rule out of total transactions of all the data stores of different sizes. When a rule is supported or voted by most of the transactions, it becomes a high frequency rule or relevant rule. The other low frequency rules are considered irrelevant. Valid rules are the high frequency rules having the more weight. The less weight rules are dropped as invalid rules.

It is important to maintain the size of the data sources unaltered because the local rules determined at each data source may be needed for the individual branch for its operations. This can be achieved by a weighting model involving certain mathematical formulae and calculation.

In the case of similar sized data sources, local rules can be dealt without considering the minimum support and minimum confidence as each data source has equal power to recommend its rules. But this is not the case with different sized data sources where the supports and confidences have different powers depending on their sizes. So the overall support and confidence of a rule is highly based on the weight of data source which in turn is calculated from its size.

A data source selection procedure is followed for filtering out the data sources below the threshold specified hence making the model more efficient.

B. Existing Related Work

Data mining also known as data archaeology focuses on the study of the data obtained from different sources by examining the various factors and discovering relevant information from them leading to knowledge [1]-[9][11].The

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/10650/pdf

relevant information may be the frequently occurring patterns, rules which can be used to support various intelligent activities such as planning, problem solving and decision making [11].

Let $I= \{i_1,i_2,...,i_n\}$ be a set of N distinct literals called items. An association rule focuses on the relationship between items. It is an implication of the form A→B where A, B ⊂ I, and A∩B = ∅. A is called the antecedent of the rule and B is called the consequent of the rule.

In general, a set of items (such as the antecedent or consequent of a rule) is called an itemset. Each itemset has an associated statistical measure called the support which is the number of instances which the rule applies to within the data set. Confidence is the accuracy with which the rule predicts correctly.

The problem of association rule mining and discovering of valid rules is to generate all rules A→B that have both support and confidence greater than or equal to some minimum threshold specified called minimum support (minsupp) and minimum confidence(minconf).

Support (A∪B) ≥ minsupp

Confidence (A→B)= Support(A∪B)/Support(A) ≥ minconf

Let $D_1,D_2,...D_m$ be m different data sources from the branches of a large company of similar size and S_i be the set of association rules from D_i, $(i=1,2,...m)$.

Let $S=(S_1,S_2....S_m)$ and $R_1,R_2,....R_n$ be all rules in S. Then the weight of R_i is defined as [12]

$$w_{Ri}=Num(R_i) / \sum^{n}_{j=1} Num\ (R_j) \quad (1)$$

Where i=1, 2,..n; Num(R) is the number of data sources that contain rule R, or the frequency of R in S.

The weight of D_i is defined as follows [12]

$$w_{Di}=\sum_{Rk\ \epsilon Si} Num\ (R_k)*w_{Rk}/\sum^{m}_{j=1}\sum_{Rh\epsilon Sj} Num(R_h)*w_{Rk} \quad (2)$$

where i = 1,2…m.

For a given rule $X\rightarrow Y$, suppose $w_1,w_2,w_3....w_m$ are the weights of $D_1,D_2,...D_m$ respectively, the model defined is as follows [12]

$$supp_w(XUY)=w_1*supp_1(XUY)+w_2*supp_2(XUY)+.....+w_m*supp_m(XUY) \quad (3)$$

$$conf_w(X\rightarrow Y)=w_1*conf_1(X\rightarrow Y)+w_2*conf_2(X\rightarrow Y)+.....+w_m*conf_m(X\rightarrow Y) \quad (4)$$

where $supp_w(R)$ is the support of R after synthesizing. $conf_w(R)$ is the confidence of R after synthesizing. $supp_i(R)$ is the support of R in D_i, and $conf_i(R)$ is the confidence of R in D_i, $(i=1,2...m)$.

III. Discovering of Valid rules by Weighting - proposed work Solving size of Data sources

To determine the valid rules from different branches of a big organization, we need to consider the size of those data sources. Because all the data sources may not be of equal size always and also merging or splitting of the data sources to make them of similar size is also a difficult task. So we go in for a method without altering the data source size.

The weighting method can be applied to the data sources sizes to find the size weight of each data source.

The size weight of each of the data source can be calculated based on the ratio of the size of that particular data source to the sum of sizes of all available data sources.

A. Finding valid rules

The weighting model is applied to find out the rules' weight and also the data sources' weight. The weight of the data source will be based both on the size of the data source and also the rules it supports. Thus we can assign a high weight to a data source that supports/ votes more high frequency valid rules and having a higher size.

Let $D_1,D_2....D_m$ be m different data sources in the branches of a company, S_i be the set of association rules from D_i, $(i=1,2....m)$, $S=(S_1,S_2....S_m)$ and $R_1,R_2....R_n$ be all rules in S. Then the weight of D_i is defined as follows:

$$w_{Di}=\sum_{Rk\ \epsilon Si} Num(R_k)*w_{Rk}/\sum^{m}_{j=1}\sum_{Rh\epsilon Sj} Num(R_h)*w_{Rk}$$

where i = 1,2…m.

This gives weight of the data source by calculating weights of the rules as mentioned in the above sections.

A second weight for the data source is calculated using the below mentioned method:

$$W_{Di}=size(D_i)/\sum^{m}_{i=1} size(D_i) \quad (5)$$

The net weight of the data source can be calculated as follows:

$$Net\ weight= (w_{Di}+W_{Di})/2 \quad (6)$$

The net weight calculated here will be efficient for data sources of different sizes. This weight can help synthesize high frequency rules in the above specified methods.

A more efficient method for synthesizing the rules can be by using the data source selection. Here a weight is calculated for each of the data sources and the user is prompted to specify a threshold required.

The threshold specified by the user takes only the data sources that have higher weights. Data sources having higher weights have larger possibilities of having valid rules, hence those data sources are only selected.

We now illustrate the above idea by an example [12].

Let minsupp=0.2,minconf=0.3, and the following rules be mined from 3 data sources.

S_1 the set of association rules from data sources D_1:A,B→C with supp=0.4;conf=0.72;A→D with supp=0.3, conf=0.8; B→E with supp=0.34,conf=0.04;

S_2 the set of association rules from data source D_2:B→C with supp=0.45, conf=0.87; A→D with supp=0.36, conf=0.7; B→E with supp=0.4, conf=0.6.

S_3, the set of association rules from data sources D_3:A,B→C with supp=0.5,conf=0.82;A→D with supp=0.25,conf=0.62.

Let us consider three data sources, D_1, D_2 and D_3 of sizes 10k, 20k and 30k respectively.

Assume $Ś=(S_1,S_2,S_3)$. Then there are a total of four rules in Ś:

R_1 A,B→C
R_2 A→D
R_3 B→E
R_4 B→C

From the above rules mined from different data sources,

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/10650/pdf

there are two data sources that support /vote rule R_1,three data sources that support/vote rule R_2, two data sources that support/vote rule R_3, and one data source that supports/votes rule R_4.

Following Good's weight of evidence, the frequency of a rule in Ś can be used to assign it a weight. After normalization, the weights are assigned as follows:

w_{R1}= 2/2+3+2+1=0.25
w_{R2}=3/2+3+2+1=0.375
w_{R3}=2/2+3+2+1=0.25
w_{R4}=1/2+3+2+1=0.125

The two weights of the data source as mentioned in the equations are calculated for all data sources.

$W_{Di}=size(D_i)/\sum^{m}_{i=1} size(D_i)$
W_{D1} = 10/(10+20+30) = 0.167
W_{D2} = 20/(10+20+30) = 0.33
W_{D3} = 30/(10+20+30) = 0.5
$w_{Di}=\sum_{Rk \in Si} Num(R_k)*w_{Rk}/\sum^{m}_{j=1}\sum_{Rh \in Sj} Num(R_h)*w_{Rk}$
w_{D1}=2.125/2.125+2+1.625=0.3695
w_{D2}=2/2.125+2+1.625=0.348
w_{D3}=1.625/2.125+2+1.625=0.2825

The net weight is calculated as per *Net weight*= $(w_{Di}+W_{Di})/2$

Weight[D_1] = (0.167+0.3695)/2 = 0.26825
Weight[D_2] = (0.348+0.33)/2 = 0.339
Weight[D_3] = (0.5+0.2825)/2 = 0.39125

Supp(A∪D)=Weight[D_1] *supp$_1$(A∪D) + Weight[D_2] *supp$_2$(A∪D) + Weight[D_3] *supp$_3$(A∪D)
=0.26825*0.3 + 0.339*0.36 + 0.39125*0.25
= 0.3003275

Conf(A→D) =Weight[D_1] *conf$_1$(A→D) + Weight[D_2] *conf$_2$(A→D) + Weight[D_3] *conf$_3$(A→D)
= 0.651555

Supp(A∪B∪C) = 0.26825*0.4 + 0.39125*0.5= 0.302925
Conf(A, B→C) = 0.26825*0.72 + 0.39125*0.82= 0.523965
Supp (B∪E)= 0.26825*0.34 + 0.339*0.4 = 0.226805
Conf (B→E)= 0.26825*0.7 + 0.339*0.6 =0.391175
Supp(B∪C) = 0.339*0.45 = 0.15255
Conf(B→C)= 0.339*0.87 =0.29493

The synthesized rules for different sized data sources will hence be:

Rule	Support	Confidence
A→D	0.3003275	0.651555
A, B→C	0.302925	0.513965
B→E	0.226805	0.391175
B→C	0.15255	0.29493

B. Algorithm Design:

Algorithm1: Discovering valid rules from different sized data sources

Input: $S_1,S_2,\ldots S_m$: rulesets, minsupp, minconf: threshold values, n-number of data sources, γ-minimum voting degree

Output: X→Y: discovered valid association rules;

1. let S←{S1,S2,..,Sm}
2. for each rule R in S do
let Num(R) ←the number of data sources that contain rule R in S
3. Calculate voting degree by,
If(Num(Ri)/n)> γ
S←S-{R_i};
let w_R=Num(R)/$\sum_{R \in S}$Num (R$^{|}$)
4. For i=1 to m do
Let
$W_{Di}=size(D_i)/\sum^{m}_{i=1} size(D_i)$
$w_{Di} \leftarrow \sum_{Rk \in Si} Num(R_{K)*}w_{Rk}/\sum^{m}_{j=1}\sum_{Rk \in Sj} Num(Rh)*w_{Rk}$
$w_i \leftarrow (w_{Di}+W_{Di})/2$
5. For each rule X→Y ϵS do
let $Supp_w \leftarrow w_1*supp_1+w_2*supp_2+\ldots..+w_m*supp_m$;
let $conf_w \leftarrow w_1*conf1+w_{2*}conf_{2+\ldots..+}w_{m*}conf_{m;}$
6. rank all rules in S by their supports;
7. output the high rank rules in S whose support and confidence are at least minsupp and minconf, respectively.
8. end all

The algorithm above generates high frequency valid association rules from different sized data sources. Step 2 does the pruning of low frequency rules. Low frequency rules in S according to its frequency. Step 3 assigns a weight to each rule in S according to its frequency. Step 4 assigns a weight to each data source by finding the average of size weight and rule occurrence weight. The rule occurrence weight is calculated from the number of high frequency rules that rule set supports. The size weight is calculated from the ratio of the size of data source to the sum of sizes of all available data sources. Step 5 discovers the support and confidence of each rule in S by the weights of different data sources. According to the weighted supports, the rules of S are ranked in Step 6. The output in Step 7 is the high rank rules selected by the user requirements.

C. Data Source selection procedure

The number of data sources considered for discovery of valid rules may itself become huge and there is a need to limit the data sources sent for centralized processing.

This can be achieved by the Data Source Selection procedure. A data source selection procedure is followed for filtering out the data sources below the threshold specified hence making the model more efficient.

Procedure: Data Source selection (D)

Input: D-set of n Data Sources, size of n Data sources ,β-minimum threshold;

Output: D-reduced set of data sources

For i=1 to n
Let Size(D_i)←the size of data source

$$\text{Let sum} \leftarrow \sum_{1}^{D} Size(D_i)$$

If((Size(D_i)/sum)< β)
D←D-{ D_i };
End for;
Output D;
End procedure;

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/10650/pdf

The data source selection procedure above generates a reduced data source set, D, from the original Data sources.

If a data source does not satisfy the user specified minimum threshold then it will be pruned off along with its rules.

Algorithm2: Discovering valid rules from selected data sources

Input: $S_1,S_2,\ldots S_m$: rulesets, minsupp, minconf: threshold values, n-number of data sources, γ-minimum voting degree

Output: $\mathbf{X}\rightarrow Y$: discovered valid association rules; D-set of n Data Sources, size of n Data sources,β-minimum threshold;

1. call DataSourceSelection(D);
2. call Algorithm1(Discovering valid rules from different sized data sources)
3. end all

Thus step1 filters out the data sources having a minimum size and those which are not satisfying the user specified minimum threshold. Step2 applies both the size weight and rule occurrence weight to find out the overall weight of that data source. Thus high frequency valid association rules can be discovered with/without data source selection.

IV. Support Equalization

When data sources are of different sizes, weighting model can be applied to calculate the weights of the data sources. When the weighting model is applied along with the user specified threshold for the weight, it generates high frequency rules by eliminating the low weighted data sources. However, low frequency or low weight rules are not eliminated from the individual data sources in the first instance.

The support equalization method helps in equalizing the supports of the data sources and prunes the low frequency rules at the local data sources. This helps in easier and an efficient method of computation of the weights.

A. Procedure

Support equalization helps in synthesizing high frequency valid rules in the following steps:

The user is prompted to enter a minimum support value.

Weight of the individual data source is found.

The data source with the highest weight is found and the minimum support is assigned to it.

For each of the data source, the support is calculated as

$$Support_{Di}=(Minsupp*maxweight)/W_{Di} \quad (7)$$

where $W_{Di}=size(D_i)/\sum^{m}_{i=1} size(D_i)$, minsupp is the user specified value, maxweight is the weight of the data source that has the maximum value.

In each data source, only the rules satisfying this support value is taken and used for calculating the weight.

The rules obtained in this method are applied to the rule selection procedure and this helps in synthesizing high frequency rules.

The output will be high frequency valid rules.

Example

Let us consider three data sources, D1, D2 and D3 of sizes 10k, 20k and 30k respectively. Let the user specified minimum support be 0.2.

As per equation (5), we get the weights of the data sources as below

$W_{Di}=size(D_i)/\sum^{m}_{i=1} size(D_i)$

Weight[D_1] = 10/(10+20+30) = 0.167

Weight[D_2] = 20/(10+20+30) = 0.33

Weight[D_3] = 30/(10+20+30) = 0.5

Here the greatest weight is 0.5. So the maxweight value becomes 0.5. As per equation

$Support_{Di}=(Minsupp*maxweight)/W_{Di}$

Support[D_1] = (0.2*0.5)/0.167= 0.625

Support[D_2] = (0.2*0.5)/0.33= 0.303

Support[D_3] = (0.2*0.5)/0.5= 0.2

D_1 Resultset = null

D_2 Resultset

Rule	Support	Confidence
$B\rightarrow C$	0.45	0.87
$A\rightarrow D$	0.36	0.7
$B\rightarrow E$	0.4	0.6

D_3 Resultset

Rule	Support	Confidence
A, $B\rightarrow C$	0.5	0.82
$A\rightarrow D$	0.25	0.62

When this is applied to the rule selection procedure, we get the following:

Rule	Occurrence	Weight
R1	A, $B\rightarrow C$	1	1/5 = 0.2
R2	$A\rightarrow D$	2	2/5 = 0.4
R3	$B\rightarrow C$	1	1/5 = 0.2
R4	$B\rightarrow E$	1	1/5 = 0.2

$wD_2 = 2*0.4 + 1*0.2 + 1*0.2 = 1.2$

Net weight = 1.2/(1.2+1.0) = 0.54

$wD_3 = 1*0.2 + 2*0.4 = 1.0$

Net weight = 1.0/(1.2+1.0) = 0.4545

$Supp(A\cup D)=Weight[D_2]*supp_2(A\cup D)+Weight[D_3]*supp_3(A\cup D)$

$= 0.5454*0.36 + 0.4545*0.25 = 0.309969$

$Conf(A\rightarrow D)=Weight[D_2]*conf_2(A\rightarrow D)+Weight[D_3]*conf_3(A\rightarrow D)$

$= 0.5454*0.7 + 0.4545*0.62 = 0.66357$

$Supp(A\cup B\cup C)= Weight[D_3] *supp_3(A\cup B\cup C)$

$= 0.4545*0.5 = 0.22725$

$Conf(A, B\rightarrow C)= Weight[D_3] *supp_3(A, B\rightarrow C)$

$= 0.4545*0.82 = 0.37269$

$Supp(B\cup E)= Weight[D_2] *supp_2(B\cup E)$

$= 0.5454*0.4= 0.21816$

$Conf(B\rightarrow E)= Weight[D_2] *conf_2(B\rightarrow E)= 0.5454*0.6 = 0.32724$

$Supp(B\cup C)=Weight[D_2]*supp_2(B\cup C)= 0.5454*0.45 = 0.24543$

$Conf(B\rightarrow C)= Weight[D_2] *conf_2(B\rightarrow C)$

$= 0.5454*0.87= 0.474498$

The resulting high frequency rules will be as follows:

Rule	Support	Confidence
$A\rightarrow D$	0.309969	0.66357
A, $B\rightarrow C$	0.22725	0.37269

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/10650/pdf

B→E 0.21816 0.32724
B→C 0.24543 0.474498

The advantage of going in for this support equalization method is that low frequency rules can be eliminated at the individual sites. This reduces the number of rules for centralized processing and makes the system efficient for finding high frequency valid rules from data sources that are of different sizes.

V. Performance Comparison

A. Generating valid rules from different sized data sources by weighting

When data sources of different sizes need to be synthesized, we used the weighting model with and without data source selection as described. The efficiency of the two methods, by data source selection (SWDS) and without data source selection (SWNDS) can be compared by using benchmarking methods.

A graph can be drawn based of the number of data sources used, comparing their thresholds and corresponding supports and confidences. The time complexity varies for each of these methods.

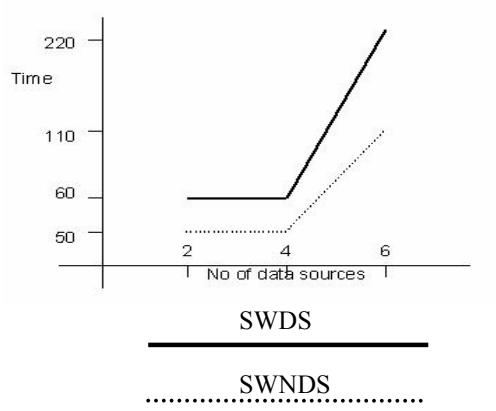


Fig. 2 Comparison between SWDS and SWNDS

Fig 2 shows the comparison between the above mentioned two methods and it has been found experimentally that discovering valid rules with data source selection is more efficient when time complexity is considered and shows better performance results.

B. Support equalization

The support equalization method helps in equalizing the supports of the data sources and helps in eliminating the low frequency or low weighted rules at the individual sites in the first instance. This greatly reduces computation complexity thus minimizing the number of rules to be synthesized. The time taken by the system for synthesizing rules by the support equalization technique can be benchmarked graphically as shown in fig 3.

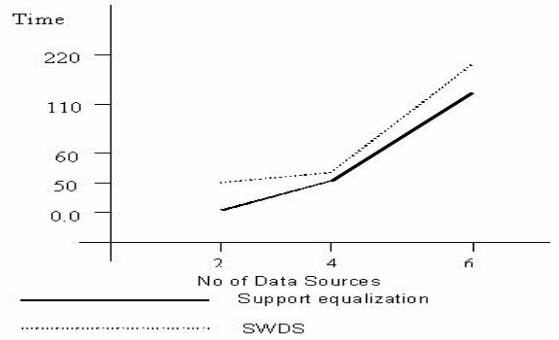


Fig. 3 Comparison between support equalization and SWDS

VI. Experiments

The comparison is also done with the synthesizing model for similar sized data sources. In our approach we are specifying the size of data sources as equal say (100 k) and the results obtained are compared with the results obtained using synthesizing model by weighting for similar sized data sources.

Using this approach, we have performed some experiments. We used JDK1.4 and implemented our model using the rules mined by familiar mining algorithms.

The results obtained are shown below:

Synthesizing model for similar sized data sources [12] is applied to a fruit stall database for experimental purpose with minsupp = 0.2 and minconf = 0.3 and the results obtained from it:

Rule	**Support**	**Confidence**
Jackfruit→Persimmon	0.27958	0.4886
Lemon, Orange→Pineapple	0.22249	0.6297
Lemon→Apple	0.21491	0.4814
Lichie→Grapes	0.40112	0.75416
Persimmon→Quinces	0.32125	0.52722

Results obtained using our approach specifying the same size as input for all data sources and minsupp = 0.2 and minconf = 0.3.

Rule	**Support**	**Confidence**
Jackfruit→Persimmon	0.28187	0.49248
Lemon, Orange→Pineapple	0.22271	0.62976
Lemon→Apple	0.2167	0.4851
Lichie→Grapes	0.40134	0.75442
Persimmon→Quinces	0.31652	0.51947

It can be noted that our approach can also be applied to similar sized data sources by specifying the sizes as equal hence making this approach usable for multiple purposes for similar as well as different sized data sources.

Results obtained from support equalization when compared with that of the weighting model with different sized data sources are no lesser in accuracy. But consumes much less time and is fast in processing when multiple rules are sent

from multiple data sources of different sizes

VII. Conclusion

Rules from data sources of different sizes can be discovered using a weighting model designed for the purpose. A data source selection procedure has been constructed in addition to the rule selection procedure in order to improve efficiency for obtaining rules from different sized data sources. Support equalization is another method used to synthesize rules by equating supports of the data sources and eliminating low frequency rules in the first instance at the data source before centralizing.

The work proposed here is advantageous in the sense that it eliminates the time and computation complexity involved in synthesizing rules as only high frequency rules are involved. This greatly reduces the number of rules and reduces processing time.

References

[1] Agarwal, R. and Srikant, R,'Fast Algorithms for Mining Association Rules, Proc. Very Large Database Conf. 1994.
[2] R.Agarwal. T.Imielinski and A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proc. ACM International Conferences on Management of Data, 1993, pp.207-216.
[3] Cheung, D. Lee, S. and Kao, B., Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique, Proc. 12th Int'l Conf. Data Eng., 1996, pp. 106-114.
[4] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, 1996.
[5] Han, J. Pei, J. and Yin, Y. , Mining Frequent Patterns Without Candidate Generation, Proc. ACM SIGMOD Int'l Conf. Management of Data, 2000, pp. 1-12.
[6] Jia-Wei Han and Micheline Kamber (2001), Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers.
[7] R.Nedunchezhian and K.Anbumani, Single Scan Frequent set Generation in Association Rule Mining, Proc. 1st International Computer Engineering Conference New Technologies for the Information Society, Cairo University, Egypt, 2004, 300-305.
[8] Park, J.S. Chen, M.S. and Yu, P.S., An Effective Hash Based Algorithm for Mining Association Rules, Proc. ACM SIGMOD Conf. Management of Data, 1995.
[9] Rastogi, R. and Shim, K., Mining Optimized Support Rules for Numeric Attributes, Proc. ACM SIGMOD Conf. Management of Data, 1999.
[10] Simovici, Dan A. Cristofor, Laurentiu and Cristofor, Dana, Galois Connections and Data mining, J.UCS: Journal of Universal Computer Science, 2000.
[11] Webb, G.I., Efficient Search for Association Rules, Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2000, pp. 99-107.
[12] Wu, Xindong and Zhang, Shichao, Synthesizing High- Frequency Rules from Different Data Sources, IEEE Trans. Knowledge and Data Eng., vol. 15, no.2., Mar/Apr 2003.

Raju Nedunchezhian is currently pursuing PhD in Bharathiar University. He received his BE degree in Computer Science and Engineering from the Bharathidasan University and ME degree in Computer Science and Engineering from the Bharathiar University. His research interests are knowledge discovery and data mining, distributed computing, and information security.

Kalirajan Anbumani obtained his BE with Honours from the College of Engineering, Guindy, Madras (Presently Anna University) (1962), ME with Distinction and University First rank from the College of Engineering, Pune (1967), and Ph D with GPA of 4/4 from the Indian Institute of Sciences, Bangalore (1982), all in India. Dr.K.Anbumani has a total of 40 years of teaching engineering, mostly in government engineering colleges of Tamilnadu state, India, in addition to two years of industrial experience. Currently, he is the Director, School of Computer Science and Technology, Karunya Deemed University, Coimbatore-641114,India. Current research interests of Dr.Anbumani include information security, image processing, business intelligence, and real-time systems.

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/10650/pdf