A Framework for Mining Association Rules in Data Warehouses

Haorianto Cokrowijoyo Tjioe and David Taniar

School of Business Systems, Monash University
Clayton, Victoria 3800, Australia
{Haorianto.Tjioe,David.Taniar}@infotech.monash.edu.au

Abstract. The effort of data mining, especially in relation to association rules in real world business applications, is significantly important. Recently, association rules algorithms have been developed to cope with multidimensional data. In this paper we are concerned with mining association rules in data warehouses by focusing on its measurement of summarized data. We propose two algorithms: *HAvg* and *VAvg*, to provide the initialization data for mining association rules in data warehouses by concentrating on the measurement of aggregate data. These algorithms are capable of providing efficient initialized data extraction from data warehouses and are used for mining association rules in data warehouses.

1 Introduction

A traditional association rule for a transaction database was first introduced by Agrawal [1]. There are two steps to discover association rules: (i) generating large item sets that satisfy user minimum support; and, (ii) rule generation based on user minimum confidence [2]. Association rules for transactional data have been developed to handle hierarchical, quantitative and categorical attributes [9,10]. Moreover, they can also use more than one predicate or dimension [5] for transactional data without any classification or measurement of aggregate data. Unlike others, multidimensional guided association rules use minimum support quantity rather than minimum support count and are also capable of handling classification [4]. However, this approach is inadequate since minimum support quantity itself will misguide the user, preventing him from finding interesting patterns.

Apparently, both concepts in [1,2,9,10] are concerned only with applying association rules in transactional data and concepts in [4,5] miss the most important attribute which is the measurement of aggregate data in a Data Warehouse (DW). The data in the DW contains only summarized data such as quantity sold, amount sold, etc. No transaction data is stored. In this paper, we focus on providing a framework for mining association rules on data warehouses by concentrating on the measurement of aggregate data. We propose two algorithms, namely *HAvg* and *VAvg*, to create an initial table to be used as a framework for mining association rules relating to a DW. After this process, using various association rules techniques [2,7,8] we can mine interesting rules.

Z.R. Yang et al. (Eds.): IDEAL 2004, LNCS 3177, pp. 159–165, 2004.

2 Background: Data Modelling in Data Warehouses

A data warehouse is typically built using a star schema, where it has more than one dimension and each dimension corresponds to one or more facts [3] (see figure 1a). Dimensions store the description of business dimensions (eg. Product, customer, vendor and store), while fact tables consist of aggregate data of measurements such as quantity sold, amount sold [3]. DW contains aggregate data which are a summary of all the details of and operational database. These data will be kept based on the lowest granularity of data which will be stored in a data warehouse [3]. DW can also be viewed using a multidimensional model [3]. As illustrated in figure 1b, there is a multidimensional model which involves three dimensions: products, customers and time. Discovering Information on enquiry (such as products in number 10 to 100 and customers in numbers 20 to 100 in year 2000) can be done using a multidimensional model. Finally, as we can see, finding association rules for DW is different from transactional data, since in DW measurements of aggregate data are very important.

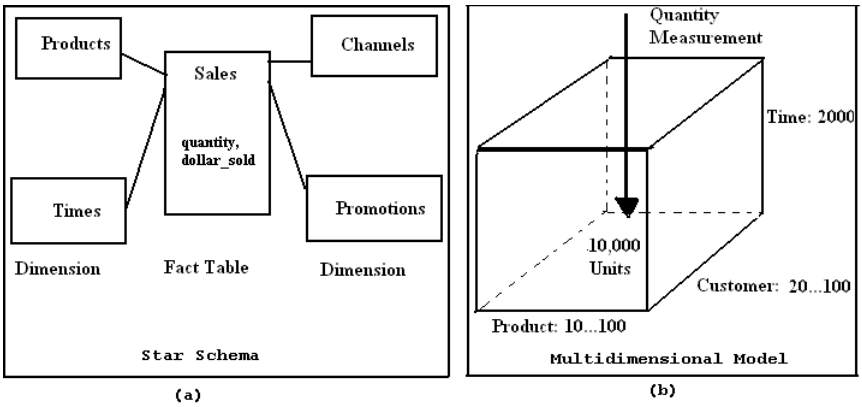

Fig. 1. Sample of a Star Schema and Multidimensional Model

3 Proposed Algorithms for Mining Data Warehouses

We propose *Vavg* and *Havg* algorithm to produce the extracted data from a DW to be used for mining association rules. Both algorithms focus on the quantitative attribute (such as quantity) of the fact table in order to prepare an initialized data for mining association rules in DW. We use the average of quantity in a fact table on m-dimensions. We prune all those rows in a fact table which have less than the average quantity.

```
SELECT <d_{m-1}.key>, <d_m.key>, SUM(quantity)AS Qty FROM Fact Table
WHERE (d_1 = duser_1) AND (d_2 = duser_2) AND … AND (d_{m-2} = duser_{m-2})
GROUP BY <d_{m-1}.key>, <d_m.key>;
```

Assume that in a fact table there is *m*-dimensions and the quantity attribute exists. The structure of the fact table is: *FactTab = ({d_1.key, d_2.key,…, d_m.key},quantity).* There are user input variables to decide which dimensions will be used. *UserVar = ($duser_1$, $duser_2$,…, $duser_m$)* where input variables for *($duser_1$,$duser_2$,…,$duser_{m-1}$)* can be treated as a single value or an interval. An interval on user input dimension variables can be treated as a classification when the interval has already been declared within the classification class. Note that notations on Table 1 are used in our proposed algorithms.

Table 1. Notations

Notation	Description
Db	Fact Table; contains(d_{m-1}.key, d_m.key, qty)
N	Total rows in fact table
VAvgTab	Vertical Avg quantity table; containts(d_m.key,d_m.qty,d_m.count,d_m.avg)
VInitTab	Vertical Initialize Table; contains(d_{m-1}.key,d_m.key,d_m.qty)
HAvgTab	Horizontal Avg quantity table; containts(d_{m-1}.key,d_{m-1}.qty,d_{m-1}.count, d_{m-1}.avg)
HInitTab	Horizontal Initialize Table ; contains (d_{m-1}.key,d_m.key,d_m.qty)

3.1 VAvg Algorithm

Using the *VAvg* algorithm, we find the average quantity of the defined dimensions vertically. As shown in figure 2, we use the *VAvg* algorithm to find the average product dimension quantity vertically from the first row of the fact table until the last row. This example uses *Time_Dim* as d_{m-1} dimension and *Product_Dim* as d_m dimension.

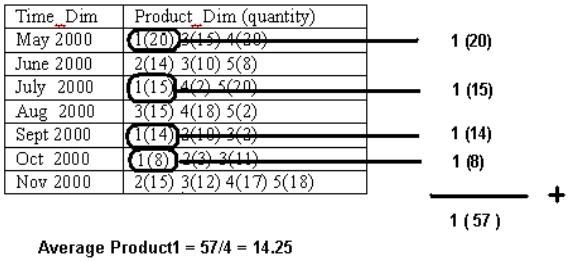

Fig. 2. An example of how the VAvg's algorithm works

We apply `Procedure VAvg` to create an initialized table (*VInitTab*) for mining association rules in a DW (see figure 3). As shown in figure 4, function `Check_Key` is used to check whatever the selected row (Db_i) on selected fact table (*Db*) already exist on *VAvgTab*. If the return value is `Found` then we update the contents of table *VAvgTab* (`Update_VAvgTab`). We update its quantity, count and average on the

selected *dm.key*. Otherwise, we insert a new record to table *VAvgTab* (`Insert_VAvgTab`). A function `Check_quantityVavg` is used to search all rows in the selected fact table (*Db*) which satisfied the minimum quantity based on table *VAvgTab*. If the row has satisfied the minimum quantity, we then insert a new row on table *VInit_Tab*.

After creating table *VInit_Tab*, the next step is mining association rules in DW using one association rule from various association rules algorithms (e.q Apriori) [2,7,8]. After identifying all the large item sets, we generate the interesting rules using a predefined confidence. Here we add an average quantity to show that these rules are satisfied only when they satisfy the average quantity (see figure 4).

```
Procedure VAvg
Begin
  For I = 1 to N loop
    Td = Check_Key(Db_i); IF Td Found then Update_VAvgTab(Db_i);
    Else Insert_VAvgTab(Db_i); End if; End loop;
  For J = 1 to N loop
    Sd = Check_quantityVavg(Db_i);
    IF Sd = True Then Insert_VInitTab(Db_i); End if; End Loop;
End;
```

Fig. 3. The VAvg Algorithm

$d_1, d_2, d_3, \ldots, d_{m-1}, d_m(\geq$ average quantity) $\rightarrow d_m(\geq$ average quantity)
d_m have more than one attribute.
$d_1, d_2, d_3, \ldots, d_{m-1}$ can be a single value or an interval

Fig. 4. Modified Rule Formula

Table 2. An example of how the proposed HAvg algorithm works

Prod	Country_Dim (quantity)	Total	Count	Avg
10	DE(8), IE(2), NL(32), PL(92), UK(131), US(120)	377	6	62.83
20	FR(125), IE(171), NL(69), UK(170), US(160)	570	5	114
30	NL(61), TR(93), US(2)	156	3	52
40	DE(109), ES(93), IE(39), NL(195), UK(25), US(108)	569	6	94.83

3.2 HAvg Algorithm

Using the proposed *HAvg* algorithm we find the average quantity of the defined dimensions horizontally. We use the *HAvg* algorithm to find the average product dimension quantity horizontally from the first row of fact table until the last row (see table 2). This example uses *Prod_Dim* as d_{m-1} dimension and *Country_Dim* as d_m dimension.

```
Procedure HAvg
Begin
  For I = 1 to N loop
    Td = Check_Key(Db_i); IF Td Found then  Update_HAvgTab(Db_i);
    Else Insert_HAvgTab(Db_i); End if; End loop;
  For J = 1 to N loop
    Sd = Check_quantityHAvg(Db_i);
    IF Sd = True Then Insert_HInitTab(Db_i); End if;
  End Loop;
End;
```

Fig. 5. Algorithm HAvg

We apply `Procedure HAvg` to create the initialized table (*HInitTab*) for mining association rules in DW (see figure 5). A function `Check_Key` is used to check whether the selected row (*Dbi*) in the selected fact table (*Db*) already exists in *HAvgTab*. If the return value is `Found` then we update the content of table *HAvgTab* (`Update_HAvgTab`). We update its quantity, count and average on selected *dm*-1.key. Otherwise, we insert new record on table *HAvgTab* (`Insert_HAvgTab`). A function `Check_quantityHAvg` is used to search all rows in the selected fact table (*Db*) which satisfied the minimum quantity based on table *HAvgTab*. If the row has satisfied the minimum quantity, then insert a new row in table *HInit_Tab*. After creating table *HInit_Tab*, we start mining association rules in DW. This is the same step as the one used after we discover *VInit_Tab*.

4 Performance Evaluation

In our performance experimentations, we used a sample sales DW [6] which contains five dimensions (e.q products, times, channels, promotions, customers) and one sales fact table with size one million rows. We performed our experiments using a Pentium IV 1,8 Gigahertz CPU with 512MB with Oracle9i Database as the DW repository.

As shown in figure 6a, we compare the number of rows produced by the *VAvg* and *HAvg* approaches when using a single attribute, with the 'no method' approach. As we see, from one to four dimensions, there is a significant reduction of rows when compared with the 'no method'. Both proposed methods have shown a similar trend across dimensions where both have reduced the rows up to 60%. However, on five dimensions, the results are similar. This is because the records involved in these dimensions are few. Moreover, in figure 6b, the results of classification or interval attributes across five dimensions have shown that both proposed methods have been significantly affected by the number of rows produced when compared with the 'no method' approach. Furthermore, as shown in figure 6c, we used a combination of five dimensions with interval and single attribute and compared its effectiveness with our approach. In general, the number of rows reduced on *HAvg* is similar to those for the 'no method' approach. Meanwhile, the *VAvg* approach has a significant gap in the

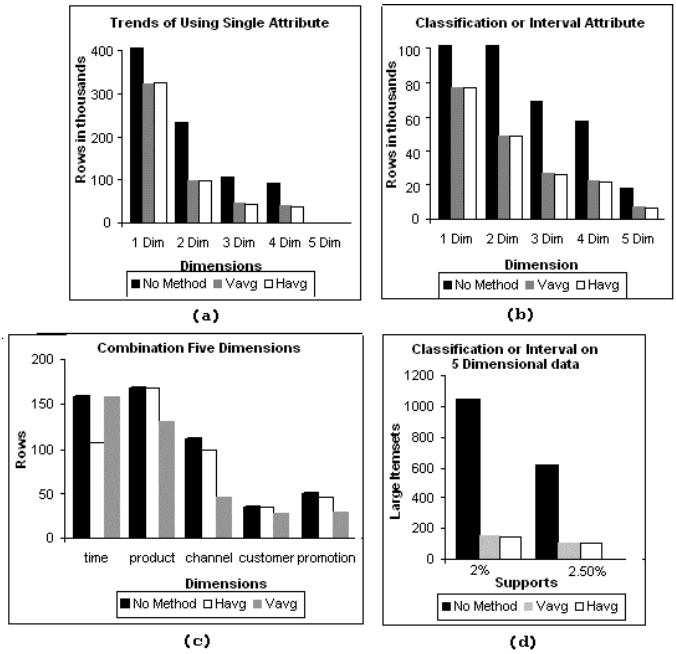

Fig. 6. Comparison HAvg, VAvg and No Method

number of rows when compared with 'no method'. However, an exception is given on the time dimension when the *HAvg* approach has reduced more rows rather than the *VAvg*. This happens each time when the dimension's attribute appears once in the vertical data. Finally, on figure 6d, we apply the Apriori algorithm implemented by Bodon [2] using two supports on interval or classification five dimensional data to discover large item sets. The results are clear with both our proposed algorithms having reductions of up to 86% for support 2% and 83% for support 2.5% compared with the 'no method' approach.

5 Conclusion and Future Work

Our proposed algorithms mainly work by filtering the data taken from data warehouses. The overall studies found that our algorithms significantly reduce the number of rows used as the data initialization for mining association rules in DW and also reduce the number of large item sets discoveries in DW. For future work, we consider developing various algorithms concerned only with mining DW.

References

1. Agrawal, R., Imielinski, T., Swami, A., Mining Association Rules between Sets of Items in Large Databases, *SIGMOD'93*, 207-216, 1993.

2. Bodon, F., A Fast Apriori Implementation, *FIMI'03*, November 2003.
3. Chaudhuri, S., Dayal, U., An Overview of Data Warehousing and OLAP Technology, *ACM SIGMOD Record*, 26, 65-74, 1997.
4. Guenzel, H., Albrecht, J., Lehner, W., Data Mining in a Multidimensional Environment, *ADBIS'99*, 191-204, 1999.
5. Kamber, M., Han, J., Chiang, J.Y., Metarule-Guided Mining of Multi-Dimensional Association Rules Using Data Cubes, *KDD'97*,207-210, Aug 1997.
6. Oracle, Oracle 9i Data Warehouse Guide, http://www.oracle.com, 2001.
7. Park, J.S., Chen M.S., Yu, P.S., An Effective Hash based Algorithm for Mining Association Rules, *SIGMOD'95*, 175-186, May 1995.
8. Savasere, A., Omiecinski, E., Navathe S., An Efficient Algorithm for Mining Association Rules in Large Databases, *VLDB'95*,432-444, 1995.
9. Srikant, R., Agrawal, R., Mining Generalized Association Rules, *VLDB'95*,407-419, 1995.
10. Srikant, R.; Agrawal, R., Mining Quantitative Association Rules in Large Relational Tables, *SIGMOD'96*,1-12, 1996.