Web Application for discovering Association Rules in Social Welfare Data Base

Carlos Enrique Gutierrez[1] and Mohammad Reza Alsharif[1]
[1] Department of Information Engineering, University of the Ryukyus
1 Senbaru, Nishihara, Okinawa 903-0213, Japan
E-mail: [1]k068519@eve.u-ryukyu.ac.jp, [1]asharif@ie.u-ryukyu.ac.jp

Abstract: The current algorithms for fiding association rules are mainly batch processes. Several efforts are carried out to improve the algorithm's performance and response's speed. Besides that the dynamic and quick search of knowledge is becoming a necessity in the traditional Data Mining techniques.

In this paper we explain step by step our web implementation of a fast asociation rules' retrieval system in order to provide useful information to take decisions. Our system is based on the creation of temporary tables and the use of Structured Query Language *"SQL"* that allow a good exploitation of the database engine's advantages. We present a simple web interface where the user chooses the attributes on which the mining algorithm will be executed.

1. Introduction

Association rules discovery is a data mining technique to investigate the possibility of multiple occurrence of items [2]. The rules provide knowledge. An example of an association rule in basket-market databases is the statement that 90% of transaction that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule [3].

Let define $I = [I_1, I_2, I_3, ..., I_n]$ as the set of items. An item is an attribute from a certain table denoted as T that will be scanned. By an association rule we mean an implication $X \Rightarrow I_j$, where X is a set of some attributes in I and I_j is a single element in I.

The *support* of a rule is defined as the fraction of records from T that *"satisfies"* the rule. For example, the support of the rule with 3 attributes $" I_1 = 2, I_2 = 5 \Rightarrow I_3 = 3 "$ is the amount of records that satisfies $" I_1 = 2$ *and* $I_2 = 5$ *and* $I_3 = 3 "$ calculated as: *"select count (*) from T where* $I_1 = 2$ *and* $I_2 = 5$ *and* $I_3 = 3$ *"*. We are usually interested only in rules with *support* above some threshold for business and practical reasons. The *minimum support* is defined as the threshold value that allows to generate rules above that constrain. The sets of items that have support above the minimum support are called *large itemsets*.

The rule $X \Rightarrow I_j$ is satisfied in the table T with *"confidence factor"*, denoted as c, if at least c% of the records in T that satisfied X also satisfied I_j, where $0 \le c \le 1$.

$$\frac{Support(X \cup I_j)}{Support(X)} \ge c$$

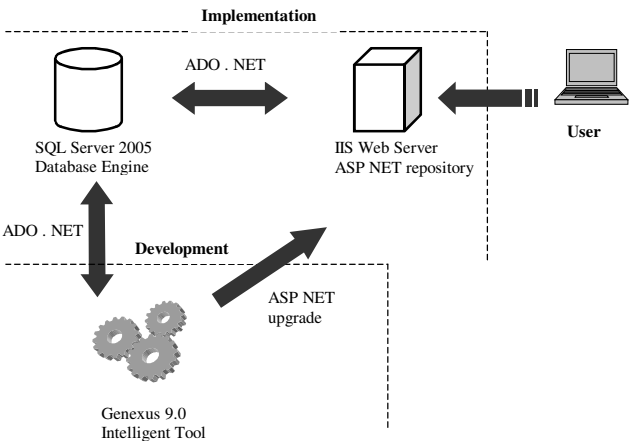
2. The Data Set

From a social welfare database, a table T called *Families* has been prepared following the guidelines of the CRISP-DM version 1.0 (Cross Industry Standar Process for Data Mining). Each record refers to a family of a certain location in Argentina and it has 18 attributes, 9 refer to the social condition and 9 to the housing situation ($I = [I_1, I_2, I_3, ..., I_{18}]$).

The attributes store integer values (indexes) that refer to a certain state. For example, the attribute "floor type" can take the values 1, 2 or 3 which refer to mosaic, cement or ground, respectively. The quantity of states or values, that each attribute takes, are depending on the database design definition.

3. The Web Application Specifications

Microsoft SQL (Structured Query Language) Server 2005 is used as database engine. The development was carried out using the intelligent tool Genexus 9.0. Web pages with technology ASP.NET (Active Server Pages) were generated. IIS (Internet Information Services) was used as web server. The system was implemented using a PC Intel Centrino with Windows XP service pack 2 with 1Gb of RAM and processor's speed of 1.73 Ghz. ADO.NET (ActiveX Data Objects) was used to establish connection among the web pages and the database engine. Fig. 1 shows the web system's architecture.



"Fig. 1. Web system's architecture."

4. Association Rules Extraction Method

The knowledge extraction in a simple way is the aim of our web application. In each step we build and scan temporary tables in order to calculate the support and to delete records.

The system's operations and processes are summarized in the following steps:

4. 1 Selection of attributes and minimum support

We may be interested only in rules that have an specific I_j in the consequent or in the antecedent. We may request all rules from a certain large itemset. We may need rules with a certain set X as antecedent. There are many sintactic contrains to consider depending on the problem and the necessities. In our system, the rules are generated from the selected attributes. Also, a value as minimum support is defined (Fig. 2).

At the end of the process all the rules with support above the minimum support will be shown ordered by their confidence factor c.

"Fig. 2. Selection of Attributes."

4. 2 Large 1-itemsets

A temporary table *temp01* is created. We append records based on the attributes selected. Its records are the attributes selected in their different values or states (see section 2).

After built *temp01*, it is scanned in order to calculate the support for each record. Those records below the minimum support are eliminated *"delete from temp01 where support < min_support"*. This step's output is a temporary table where each record is a large itemset of a single element (Fig. 3).

2. Large 1-itemsets

Id	Attribute	Description	Atribute value	Support
1	Water	Current to Home	1	715
2	Water	River, Stream or Can	7	303
3	Water	Slope	8	486
4	Roof Type	Foil of zinc	3	1496
5	Numerous Family?	Yes	1	283
6	Numerous Family?	No	2	1382
7	It receives social h	Yes	1	610
8	It receives social h	No	2	1055

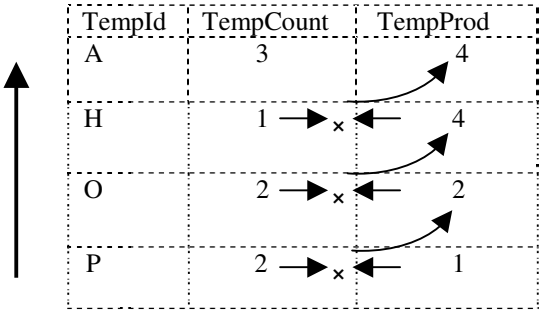
<| < > |>

"Fig. 3. Large itemsets of a single element (*temp01*)."

4. 3 All combination sets

Based on *temp01* the temporary table *temp02* is created, where *TempId* is the selected attribute, *TempCount* is the quantity of states above the minimum support that the attribute takes (see section 2) and *TempProd* is calculated by scanning *temp02* from bottom to top following the steps (Table 1):

1) Initialize the last record's *TempProd* attribute with 1.
2) Multiply the attributes *(TempCount * TempProd)* and store the result in the next record *TempProd* attribute.
3) Skip to the next record, if the current record is not the first one repeat from step 2).



"Table 1. Temporary table *temp02*."

The temporary table *temp03* is created based on *temp02*. This table store all the combinations of the selected atributtes's values above the minimum support (Table 2).

The combinations are generated using the attributes *Temprod* and *TempCount* from the table *temp02*. In *temp03* the columns of the table are the selected atributtes.

We append records in *temp03* based on the frequency of each attribute and the quantity of values (states). The frecuency is indicated by *TempCount* and the quantity of values by *TempCount*.

We can observe, based on Table 1 and Table 2, that the attribute *TempId* in *temp02* appears *TempProd* times in *temp03* for each of its *TempCount* values. The first record *(TempCount * TempProd)* operation in *temp02* indicates the total quantity of records in *temp03*. After complete the first attribute (column) in *temp03* we proceed with the others attributes repeating cycles until completing the total quantity of records indicated. Fig. 4 shows the combinations sets formed for the selected attributes.

Id	A	H	O	P
1	1	3	1	1
2	1	3	1	2
3	1	3	2	1
4	1	3	2	2
5	7	3	1	1
6	7	3	1	2
7	7	3	2	1
8	7	3	2	2
9	8	3	1	1
10	8	3	1	2
11	8	3	2	1
12	8	3	2	2

"Table 2. Temporary table *temp03*."

3. All combination sets

Id	A	H	O	P
1	Current to Home	Foil of zinc	Yes	Yes
2	Current to Home	Foil of zinc	Yes	No
3	Current to Home	Foil of zinc	No	Yes
4	Current to Home	Foil of zinc	No	No
5	River, Stream or Canal	Foil of zinc	Yes	Yes
6	River, Stream or Canal	Foil of zinc	Yes	No
7	River, Stream or Canal	Foil of zinc	No	Yes
8	River, Stream or Canal	Foil of zinc	No	No
9	Slope	Foil of zinc	Yes	Yes
10	Slope	Foil of zinc	Yes	No
11	Slope	Foil of zinc	No	Yes
12	Slope	Foil of zinc	No	No

<| < > |>

"Fig. 4. All combination sets (*temp03*)."

4. 4 Large itemSets

The support for each record of *temp03* is calculated. For example, the support of the record *Id=1* (Table 2) is calculated as: *"select count (*) from Families where A=1 and H=3 and O=1 and P=1"*.

The records below the minimum support are eliminated *"delete from temp03 where support < min_support"*. The result is a temporary table *temp03* where each record is a large itemset with number of items (elements) defined by the user. Fig. 5 shows the large itemsets generated for the selected attributes.

4. Large itemsets

Id	A	H	O	P	Support
4	Current to Home	Foil of zinc	No	No	327
12	Slope	Foil of zinc	No	No	310

<| < > |>

"Fig. 5. Large itemsets (*temp03*)."

4. 5 Generating rules

Our system generates rules with a single attribute as consequent. In this step, the temporary table *temp04* is created where each of its records represents a rule.

The rules are built by scanning *temp03*. For each record of *temp03* will be generated in *temp04* a quantity of rules equal to the quantity of attributes (columns) present in *temp03*. We append each record from *temp03* into *temp04* replacing a single attribute's value with zero, in each step it is replaced with zero a different attribute's value (Table 3).

The zero indicates the consequent of the rule and the others attributes's values the antecedent of the rule. For example, for the record *Id=4* and *Num=1* the rule is: *"If H=3 and O=2 and P=2 Then A=1"*. In Table 3 we can observe the rules generated for the records *Id=4* and *Id=12* of *temp03*.

Id	Num	A	H	O	P
4	1	**0**	3	2	2
4	2	1	**0**	2	2
4	3	1	3	**0**	2
4	4	1	3	2	**0**
12	1	**0**	3	2	2
12	2	8	**0**	2	2
12	3	8	3	**0**	2
12	4	8	3	2	**0**

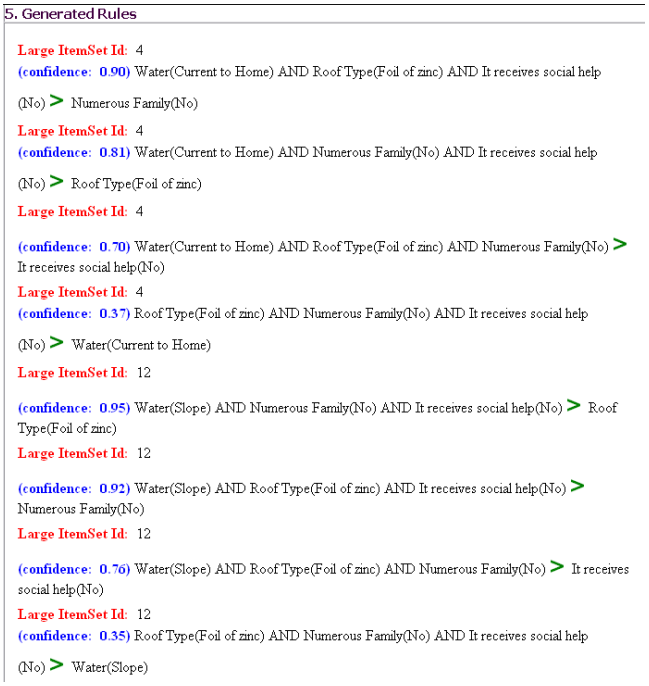
"Table 3. Temporary table *temp04*."

The support is calculated for each record of *temp04* without considering the attribute equal to zero, for example, for the record *Id=4* and *Num=1* the support is calculated as: *"select count (*) from Families where H=3 and O=2 and P=2"*.

The confidence factor *"c"* of the rule is calculated by dividing the support among *"temp03"* and *"temp04"*. For example, *"c"* of the rule *Id=4* and *Num=1* is calculated as : *support(record Id=4 in temp03) / support(record Id=4 and Num=1 in temp04)*.

In order to obtain the final output of the system we translate the attributes' values of *temp04* by their literal descriptions.

The rules ordered by their *confidence factor* are the final output of the system. Fig. 6 shows the generated rules for the selected attributes.



"Fig. 6. Generated rules."

5. Results

Some generated rules are well-know and can be verified with the experience and the opinion of the specialists, but others are new, that is the hidden knowledge discovered.

Through the execution of the system we obtained some obvious rules. Their validity was analyzed and confirmed by specialists in social development of Argentine's government, for example:

** (confidence: 0.99) Critical Family(No) AND Numerous Family(No) > It receives institutional attention(No)*

** (confidence: 0.94) Water Treatment(Chlorine) AND Numerous Family(No) > Critical Family(No)*

Also new rules were discovered and confirmed by the specialists, these rules constitute the information to take decisions, useful to design assistance programs, for example:

** (confidence: 0.56) Quantity of Rooms(1) > Bathroom Type(Latrine)*

** (confidence: 0.72) Water(Slope) > It receives social help(No)*

We have executed the system several times with many variations of selected attributes. The response's speed of the system was acceptable.

The operation of the system is easy and understandable, it consists in a sequence of steps.

The system shows all the generated rules with their confidence factor leaving to the user's criteria the evaluation of the rules.

6. Conclusion and Future Work

We have designed a web application system of association rules' extraction, where the result is obtained quickly through a simple web interface.

Our procedure doesn't extend the itemsets generating candidates [1], the amount of itemsets (attributes) is defined by the user to generate the rules.

The user attributes's definition, the usage of temporary tables and the Structured Query Language *"SQL"* are the system's features that permit a fast response time.

We propose a simple algorithm to generate all combination sets from a set of attributes.

It is neccesary to design a method (or metric) to pre-evaluate the rules validity automatically before the user evaluation.

The web systems needs to incorporate an interface for the user's feedback in order to uses this evaluation in future implementations.

The web systems is executed manually by the user, but it could be modified in order to be executed automatically and periodically on a dynamic database or data stream.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules", VLDB, 20th International Conference on Very Large Data Bases, pp. 487-499, 1994.

[2] Chai, Duck Jin; Jin, Long; Hwang, Buhyun; Ryu, Keun Ho, "Frequent Pattern Mining using Bipartite Graph", DEXA, 18th International Conference on Database and Expert Systems Applications, pp. 182-186, 2007.

[3] Rakesh Agrawal, Tomasz Imielinski, Arun Swami, "Mining Association Rules between Sets of Items in Large Databases", ACM SIGMOD, International Conference on Management of Data, pp. 207-216, 1993.

[4] Sergey Brin and Lawrence Page, "Dynamic Data Mining: Exploring Large Rule Spaces by Sampling", VLDB, 1998.