Łukasz NOZDRZYKOWSKI, Magdalena WRÓBEL

THE USE OF SQL AS A TOOL SUPPORTING THE IMPLEMENTATION OF A METHOD OF ATTRIBUTE SIGNIFICANCE ANALYSIS BASED ON SOFT REDUCTION OF ATTRIBUTES IN THE ROUGH SET THEORY

Abstract

The article presents a way to quickly implement a method of analyzing the significance of attributes by using soft reduction of conditional attributes in the rough set theory. The analysis is a universal instrument for testing the significance of attributes and may be successfully used in many fields, including transport. It uses the rules that can be considered useful and allows reducing those attributes that do not cause a significant decrease in the number of rules generating entirely certain rules. At the same time it is a rapid mechanism of analyzing large data sets such as encoded attributes of rules. For implementation purposes we propose to use the mechanisms of modern relational databases and the capabilities presently offered by the SQL language, including its expansion with conditional CASE queries.

INTRODUCTION

Application of the rough set theory allows, inter alia, the induction of decision rules and the reduction of data sets and an analysis of attribute significance. The presented method allows analyzing large data sets by making possible their reduction to a minimum. It also permits to remove the inconsistency of data and generate minimum rules that simulate expert's decisions. Advantages of the rough set theory are: no assumptions to data (e.g. probability), fast data analysis algorithms, facilitated data analysis and mathematical simplicity [1]. This article presents the possibility of using databases and SQL in the implementation of a modified analysis of significance of conditional attributes in the rough set theory, based on the soft reduction of attributes in those sets.

The universal character of the attribute significance analysis based on soft reduction of attributes in the rough set theory makes the analysis applicable in any scientific discipline or field, for instance in maritime transport, where the significance analysis of attributes can refer to models built to describe transport processes.

1. SIGNIFICANCE ANALYSIS IN THE ROUGH SET THEORY

The use of rough set theory for the reduction of conditional attributes is composed of four main stages. In the first step of the analysis of the significance of the attributes obtained data should be presented in the form of an information table containing a set T = (U, Q, D, V, f), where U is a set of examples, Q is a set of attributes, D is a set of decision attributes, V is a set of all possible values of attributes, and f is an information function [2]. The next step is to divide data into sets: an expert or a group of experts, on the basis of crisp membership functions, distribute the data into sets. The number of these sets is dependent on the experts. Another method of dividing data into sets is formation of equinumerous sets which allows automating this stage. In the third stage, the table with input data is converted into an encoded form by generating a secondary, discrete database. For a decoded table a reduced one is created, where the decision attribute is not taken into account. All identical occurrences of attributes are saved as decision rules E_i. In the fourth step all decision rules are classified as a series X_i, in other words decision attributes classified as decision rules are divided into sets. Then a reduced information table is generated for a decision attribute. For such a reduced set, the number of certain rules is defined and on this basis the significance of a reduced attribute is determined in terms of preserving the quality of rule generation.

Soft reduction of conditional attributes based on the relative probability of useful rules in the rough set theory is supposed to allow the rejection of those attributes, the removal of which has no meaningful effect on the decrease in the number of useful rules generating entirely certain rules.This reduction is carried out with a slight decrease in the number of examples generating entirely certain rules. The quality of rules is assessed here on the basis of relative probability of atomic rules. This probability is expressed by this formula:

$$P_w = \frac{P}{L} \tag{1}$$

where P - sum of the probabilities of useful atomic rules.
L - number of elementary conditional sets.

An atomic rule of an elementary set is considered to be useful if its probability is greater than a preset threshold for which it is recognized as useful. The significance analysis is done in the traditional way, where their significance in generating certain rules is determined by coding decision and conditional attributes and by the reduction of subsequent conditional attributes [3].

2. THE USE OF THE SQL DATABASE IMPLEMENTATION METHOD

Implementation of attribute significance analysis method with soft reduction of conditional attributes in the rough set theory requires the programmer to store large chunks of data, which increases the complexity of the memory of a program so written. In addition, the source code written in this way must have all the searching and rule counting mechanisms used in the method itself, such as reduction of decision rules equal to the count of their repetitions in order to generate a table of reduced rules as well as determine the reduced sets of rules during the reduction of conditional attributes. These operations are quite difficult to implement. For this reason, the authors propose using relational databases and SQL capabilities to facilitate the implementation of the whole method.

Such program can be implemented in programming languages capable of working with relational databases, and thus should pro-

vide a mechanism of connecting to databases, sending queries and process acquired data. Additionally, the authors assume that the database used will allow making conditional expressions (CASE queries) [4]. Possibilities of performing such queries are provided by database management systems PostgreSQL [5] and MySQL [6]. The authors, during the verification of the correctness of formulated queries and program tests of the herein described method used the PostgreSQL database, version 9.5.4.

3. DETERMINATION OF RELATIVE PROBABILITIES OF ALL USEFUL ATOMIC RULES

The first stage of the analysis is to present the problem in the full space of conditional attributes. To do this, we should prepare a base table in a database that stores encoded conditional and decision attributes. The data should be prepared in a manner assumed for encoding all attributes. Any coding method can be chosen for this purpose, e.g. through equal distribution of samples in the compartments. In the presented method the first attribute is the decision attribute and it will be denoted as k0, then successively conditional attributes are stored. For simplification, the following attributes are denoted with symbols k1 to kn. We propose to store the data in the form of integers for the subsequent levels of encoding. The table, storing attributes in integer columns, is created by a query presented in Listing 1.

Listing 1 - Creation of a base table

```
CREATE TABLE Base
        {k0 integer DEFAULT 0, k1 integer DEFAULT 0,…, kn integer DEFAULT 0}
```

Such table principally does not call for defining an identifying key as no relationships will be established. The authors assume that data for such table will be prepared and delivered, e.g. in the form of INSERT queries. The method of preparing encoded data is arbitrary, as the data set can be easily prepared even in Matlab based on a CSV file.

On the basis of this table, you can easily prepare a simplified table where repeated rules will be eliminated and the table itself will contain information on the number of examples for that rule. To this end, we propose to create another table (ElementaryTable) for storing unique rules and their numbers, in which the following data will be stored:

- DistinctAttr - a unique decision attribute for individual rules,
- CountAttr - the count of single unique rules,
- k1-kn - subsequent encoded conditional attributes.

The table has the structure shown in Listing 2.

Listing 2 - Creation of an ElementaryTable

```
CREATE TABLE ElementaryTable
        { distinctAttr integer DEFAULT 0,
countAttr integer DEFAULT 0,
k0 integer DEFAULT 0, k1 integer DEFAULT 0,…, kn integer DEFAULT 0}
```

Data for this table come from a SQL query using the grouping according to the decision attribute. The use of a query is presented in Listing 3.

Listing 3 - A query generating data into an ElementaryTable

```
SELECT DISTINCT(k0), COUNT (k0), k1, k2…, kn FROM Base GROUP BY k0, k1, … , kn;
```

At this stage, the implemented program receives information on all of the unique rules together with their numbers. These data allow for quick creation of INSERT queries to an ElementaryTable by rewriting all data in the software loop FOR to a new query. Such loop iterates over the number of rows returned as a result of a SELECT query.

The next step in the analysis is to determine its basic parameters:

- number of elementary sets,
- number of useful rules,
- total absolute probability of useful atomic rules,
- relative probability of all useful atomic rules.

Because the analysis presupposes the determination of these parameters on the basis of all possible values of encoded conditional attributes, the database should be successively queried about all possible combinations of these attributes. For this purpose, while implementing a program for the analysis we should utilize the mechanism of mutually nested program loops. An example is given in Listing 4, for attributes encoded with values 0 to 5.

Listing 4 - Example of program loops generating queries with all combinations of attributes.

```
for (indexK1=1; indexK1<5; indexK1++)
        for (indexK2=1; indexK2<5; indexK2++)
                …
        for (indexKn=0; indexKn<5; indexKn++)
{
                //loop body

}
```

In the body of the furthest nested loop all the mentioned parameters will be counted. The basic query inside the body of the furthest nested loop is one shown in Listing 5.

Listing 5 - Query to a database counting all of the parameters

```
SELECT distinctAttr, countAttr FROM ElementaryTable WHERE k1=indexK1 AND k2=indexK2 AND... AND kn=indexKn
```

The number of elementary sets is obtained from information about all unique rules (including useless rules) without taking into account a decision attribute. If the query above returns the result (regardless of the number of returned rows), then the variable counting elementary sets is increased by 1, because such rule exists in the database.

Another parameter is the number of useful rules. A rule is considered useful when the probability of its occurrence is greater than an assumed threshold. To this end, find certain rules (well defined part without contradictions) and take into account rules from the ill-defined parts that can be considered useful. Because the ill-defined part contains rules contradictory in terms of the decision attribute (there are rules with the same conditional attributes but a different decision attribute), rules with a probability of occurrence below a

predetermined threshold are rejected. The number of obtained examples K is calculated for a given certain rule defined by the conditional attributes and decision attribute, then this number is divided by the number L of examples for the rule defined by conditional attributes (the number of elementary conditional sets). The obtained value is compared to the established threshold, and if it is greater than the threshold T the rule is considered to be useful.

Listing 6 shows the acquisition of the parameter divisor L.

Listing 6 - A query calculating the parameter L

```
SELECT SUM(countAttr) FROM ElementaryTable WHERE k1=indexK1 AND k2=indexK2 AND... AND kn=indexKn
```

The parameter dividenda K is obtained on the basis of a conditional query in the database having the form shown in Listing 7.

Listing 7 - Query calculating parameter K

```
SELECT
        COUNT(
                CASE WHEN
((countAttr::float)/L)>T
THEN
((countAttr::float)/L)
 END
        ) FROM ElementaryTable WHERE k1=indexK1 AND k2=indexK2 AND... AND kn=indexKn
```

The parameter of total absolute probability of useful atomic rules isobtained by performing queries from Listing 8.

Listing 8 - A query calculating the total absolute probability of useful atomic rules

```
SELECT
        SUM(
                CASE WHEN
 ((countAttr::float)/L)>T
THEN
((countAttr::float)/L)
END
        ) FROM ElementaryTable WHERE k1=indexK1 AND k2=indexK2 AND... AND kn=indexKn
```

Through this query information is obtained about the partial value of the total probability for each value of encoded conditional attributes. The full value of this probability is the sum of the partial values.

At the same time, in both cases of using conditional queries the parameter L can be reduced by inserting into that space a query defining this parameter. This operation, however, is unprofitable due to the readability of the code and the need to perform repeatedly an additional nested query. The only reduction is that of a number of queries in the same code, while the number of queries performed grows. An example correct query for the last presented query would have the form presented in Listing 9.

Listing 9 - A query generating full absolute probability of atomic rules

```
SELECT
        SUM(
                CASE WHEN
((countAttr::float)/
SELECT SUM(countAttr) FROM ElementaryTable WHERE k1=indexK1 AND k2=indexK2 AND... AND kn=indexKn
) > T
THEN
((countAttr::float)/
SELECT SUM(countAttr) FROM ElementaryTable WHERE k1=indexK1 AND k2=indexK2 AND... AND kn=indexKn
)
END
        ) FROM ElementaryTable WHERE k1=indexK1 AND k2=indexK2 AND... AND kn=indexKn
```

The last parameter in the introductory part of the analysis before the reduction of attributes is the calculated relative probability of useful rules. This value is equal to the total absolute probability of useful atomic rules divided by the number of elementary sets.

4. SOFT REDUCTION OF CONDITIONAL ATTRIBUTES

The next step in the analysis is to attempt the rejection of subsequent conditional attributes and the final verification by an expert checking whether it does not cause a significant reduction in the relative probability of useful atomic rules. If the drop is not too high, then that attribute is considered as insignificant and can be finally reduced.

To facilitate the implementation, it is worth preparing another table in the database to store information about the rules after a test reduction of a selected conditional attribute. This table is similar to the table for the full outline. Like the previously encoded decision attribute distinctAttr, this table stores the number of examples confirming the rule for a given decision attribute countAttr and kn-1 of conditional attributes without the attribute that is tentatively being removed. The table is generated by a query shown in Listing 10.

Listing 10 - A query forming a table after an attempt to reduce a selected attribute

```
CREATE TABLE ReductionTable
        { distinctAttr  integer DEFAULT 0,
countAttr integer DEFAULT 0,
k0 integer DEFAULT 0, k1 integer DEFAULT 0,…, kn-1 integer DEFAULT 0}
```

The data for the reduction table come from a table that stores information about all the atomic rules and are obtained by a query shown in Listing 11.

Listing 11 - A query obtaining data for the reduction table

```
SELECT DISTINCT(k0), COUNT (k0), k1, k2…, kn FROM Base  GROUP BY k0, k1, … , kn;
```

This query does not comprise information about the attribute subject to a trial reduction.

The significance analysis method at this stage involves another determination of analysis parameters, i.e. number of elementary sets, number of useful atomic rules, total probability of useful rules and relative probability of useful atomic rules. The last step is to determine the significance of the attribute. This value makes up a basis for assessing whether the tested conditional attribute will be removed or not.

A new table simplifies the implementation process to the duplication of operations included in the initial analysis phase, where the probabilities were tested with a full set of conditional attributes.

The function or method that determines the significance, like in earlier operations, is based on n-1 number of jointly nested loops, where n is the number of conditional attributes. The method of testing the probabilities is based on the same queries in the full analysis, except for the table from which the data are taken. The main query is presented in Listing 12.

Listing 12 - A quety generating the number of elementary sets and the number of useful atomic rules

```
SELECT distinctAttr, countAttr FROM ReductionTable WHERE
k1=indexK1 AND k2=indexK2 AND... AND kn-1=indexKn-1
```

In the other queries the threshold parameter T is still used for determining the rules that are considered useful.

The significance of the tested attribute to be reduced is determined on the basis of obtained parameters. To determine this value, we use calculated parameters of the relative probability of useful atomic rules pfull for all attributes and the relative probability of useful atomic rules with a reduced i-th conditional attribute qi. If this probability is denoted as pred (qi), a general formula for the determination of the level of significance of attribute qi has this form (2):

$$\text{significance}(q_i) = \frac{p_{full} - p_{red_{qi}}}{p_{full}} \quad (2)$$

Such attribute may be reduced if its significance is low. Whether reduction is possible is decided by an expert performing the analysis. However, because during the analysis you may find that there is another attribute of low significance, they should not be reduced at the same time, until you try to perform the analysis of the simultaneous reduction of these two attributes. This happens because simultaneous reduction of several attributes can significantly degrade the total value of the relative probability of rules, which will make such reduction unacceptable.

The final step after the significance analysis of a selected attribute is to clear the data from the table ReductionTable to prepare that table for data for the next significance analysis.

5. EFFICIENCY OF THE METHOD

The authors carried out performance tests of the proposed solutions. The tests made use of a laptop computer MSI GE72-2QD with 8-thread processor Intel Core i7-5700HQ clocked at 2.7 to 3.5 GHz, 16GB DDR3 RAM and SSD Kingston SM2280S3120G. The computer runs the operating system Windows 10. The programming environment is Microsoft Visual Studio 2015 Enterprise, and the database was created in the system PostgreSQL 9.5.4.

A test analysis was made using a table storing seven conditional attributes and one decision attribute, each encoded into five compartments. Encoding was performed by the method of equinumerous compartments. The table stored about 22,000 rules. Tables 1 and 2 show the results of the proposed implementation in a test environment.

Tab. 1. *RAM usage of the proposed solution [MB]*

Number of rules / Number of attributes	14000	18000	22000
5	3.2	3.3	3.6
6	5.2	5.5	5.8
7	7.1	7.8	8.8

Tab. 2. *Execution times of the proposed solution [s]*

Number of rules / Number of attributes	14000	18000	22000
5	3	4	6
6	17	18	23
7	95	105	124

With a maximum number of rules in the database the program performed calculations in 2 minutes 4 seconds using up to 9 MB of RAM, of which the main analysis process without reduction took 52 seconds. This is because in the course of a full analysis an extra nested loop is performed compared to the reduction. One can see from the table that the number of attributes uses the most RAM and has longest execution times of such implementation.

SUMMARY

The article presents a convenient way of rapid implementation of the algorithm for determining the significance of conditional attributes based on their soft reduction in the rough set theory. Utilizing SQL and relational databases we were able to quickly implement this method. Its main advantage is improvement of all implementation processes through the use of aggregating, searching and conditional functions available in SQL to prepare an appropriate program code without the need to implement the critical moments of the significance analysis method itself.

BIBLIOGRAPHY

1. Pawlak Z., Zbiory przybliżone nowa matematyczna metoda analizy danych, w: Miesięcznik Politechniki Warszawskiej nr 5/2004. Dostępny w Internecie: http://bcpw.bg.pw.edu.pl/Content/1949/zb_przyb.pdf.
2. Łatuszyńska M, Wawrzyniak A, Wąsikowsk B, Galindo E, Sandoval J. D., Teoria zbiorów przybliżonych w wykrywaniu reguł zachowań zakupowych kobiet i mężczyzn podczas kupowania telefonów komórkowych, Zeszyty Naukowe Uniwersytetu Szczecińskiego, Studia Inormatica Nr 35, 2014
3. Moudani W, Shahin A, Chakik F, Mora-Camimo F, Dynamic Rough Sets Features Reduction, International Journal of Computer Science and Information Security, Vol. 9, No. 4, April 2011
4. Żmuda K., SQL, Jak osiągnąć mistrzostwo w konstruowaniu zapytań, Helion 2015
5. PostgreSQL Reference Manual: https://www.postgresql.org/docs/7.4/static/functions-conditional.html
6. MySQL Reference Manual: https://dev.mysql.com/doc/refman/5.7/en/case.html

UŻYCIE JĘZYKA SQL JAKO NARZĘDZIE WSPOMAGAJĄCE IMPLEMENTACJĘ METODY ANALIZY ISTOTNOŚCI ATRYBUTÓW W OPARCIU O MIĘKKĄ REDUKCJĘ ATRYBUTÓW W TEORII ZBIORÓW PRZYBLIŻONYCH

Streszczenie

W artykule przedstawiono sposób na szybką implementację metody analizy istotności atrybutów poprzez wykorzystanie miękkiej redukcji atrybutów warunkowy w teorii zbiorów przybliżonych. Analiza ta wykorzystuje reguły, które można uznać za użyteczne i pozwala na redukcję atrybutów, które nie powodują znacznego spadku liczby reguł generujących całkowicie pewne reguły. Jest przy tym szybkim mechanizmem analizy dużych zbiorów danych jakim są zakodowane atrybuty reguł. Do celów implementacyjnych zaproponowano wykorzystanie mechanizmów współczesnych relacyjnych baz danych oraz możliwości jakie obecnie daje język SQL, w tym rozbudowanie go o zapytania warunkowe typu CASE.

Autorzy:
dr inż. **Łukasz Nozdrzykowski** – Akademia Morska w Szczecinie, Wydział Nawigacyjny, ul. Wały Chrobrego 1-2, 70-500 Szczecin
mgr inż. **Magdalena Wróbel** – Akademia Morska w Szczecinie, Wydział Nawigacyjny, ul. Wały Chrobrego 1-2, 70-500 Szczecin