Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 155 (2019) 575–580

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Applying Data Mining Techniques to E-Learning and Pedagogical Approaches (ADMEPA)

August 19-21, 2019, Halifax, Canada

Integration of Data Mining Techniques to PostgreSQL Database Manager System

Amelec Viloria[a]*, Genesis Camargo Acuña [b], Daniel Jesús Alcázar Franco [c], Hugo Hernández-Palma [d], Jorge Pacheco Fuentes [e], Etelberto Pallares Rambal[f]

[a,b] *Universidad de la Costa, Street 58 # 55 - 66. Barranquilla, Colombia.*
[c] *Corporación Universitaria Reformada, Street 38 #. 74 -179, Barranquilla, Colombia.*
[d,e,f] *Universidad del Atlántico, Street 30 # 8- 49 , Puerto Colombia – Colombia*

Abstract

Data mining is a technique that allows to obtain patterns or models from the gathered data. This technique is applied in all kind of environments such as in the biological field, educational and financial applications, industry, police, and political processes. Within data mining there are several techniques, among which are the induction of rules and decision trees which, according to various studies carried out, are among the most used. This research analyzes decision tree data mining techniques and induction rules to integrate several of its algorithms into PostgreSQL database management system (DBMS). Through an experiment, it was found that when the algorithms are integrated to the manager, the response times and the results obtained are higher.

Keywords: Data mining; database management system; PostgreSQL; decision-making trees; induction rules.

* Amelec Viloria. Tel.: +57-304-6238313
E-mail address: aviloria7@cuc.edu.co

10.1016/j.procs.2019.08.080

1. Introduction

There are several independent database management system tools for applying techniques to large volumes of data, however, most of these tools are proprietary and are not available to organizations because they are very costly [1]. Other tools like WEKA or YALE RapidMiner are GPL licensed, but when a lot of data is required for the analysis, the process becomes complex and slow. In addition, data security must be ensured as information travels over the network [2], [3].

To solve these problems, some companies such as Microsoft and Oracle have developed modules within their database management systems that include data mining techniques, allowing them to speed up response times as it would not be necessary to transform "raw data" into "actionable information" (data preparation) or import or link with the tool responsible for performing the analysis. In this way, it is not necessary to train the staff in the use of other data analytics tools providing the data analysts with direct but controlled access, thus accelerating productivity without compromising data security. However, despite these advantages, these kinds of software present the disadvantage of being proprietary [4], [5].

Many companies are currently migrating to open source software looking to ensure their economy [6]. One of the tasks to achieve this goal is the migration to PostgreSQL database technology as it is the most advanced open source DBMS in the world as it supports the vast majority of SQL transactions, concurrent control, offering modern features such as complex queries, triggers, views, transactional integrity, and allowing to add data type extensions, functions, operators, and procedural languages [7], [8]. However, this system has not integrated these data mining techniques [9]. For this reason, it is necessary to achieve the independence of PostgreSQL database management system to analyze the data using data mining techniques, induction rules, and decision trees. Hence, this research establishes the objective of integrating algorithms of data mining techniques, induction rules, and decision trees into the PostgreSQL database management system.

2. Method

Three algorithms were implemented in this study: 1R, PRISM and ID3. Generally, tables that perform data mining analyses have a large volume of information, which can delay the outcome of the study. One of the options offered by PostgreSQL to improve performance in these cases is the table partitioning, which allows a better performance when querying those tables. Table partitioning is a technique that involves breaking down a huge table (parent) into a set of child tables. This technique reduces the number of physical reads on the database when queries are executed. In PostgreSQL, the existing partitioning types are by range and by list [10].

- Partitioning by range: Partitions are created using defined ranges based on any column that does not overlap between the ranges of values assigned to different child tables.
- Partitioning by list: Partitions are created by values.

This research implements a function that allows the table to be partitioned according to the values of the class (partitioned by list). This makes it possible to speed up the search when sorting. The function has the table to be partitioned and the name of the class as the input parameter, creating as many partitions as values of the class. The parent tables created by the function will be named by the master concatenation and the previous name of the table, while the child tables will have the name of the concatenation of the previous table and the value of the class from which the partition was created [11] , [12], [13].

2.1 Creating the Data Mining Extension

To create the extension, two files must be created. The first one defines the extension characteristics; in the second one, the SQL objects that will be added. They should be located in the installation directory "C:'Program Files \ PostgreSQL\9.1\share\extension". In the file "mineria_data.control" created to add the extension in which the functions of the implemented algorithms, the following parameters were defined:

- comment: a brief description of the content of the created extension.
- Encoding: The type of used encoding.

- default_version: The extension version.
- relocatable: if available.
- schema: the schema in which the objects created by the extension will be stored.

Once the file "mineria_data.control" is defined, the file containing the code of the developed functions "mineria_datos-- 1.0.sql" is specified.

3. Results and Analysis

To validate the algorithms, an experiment defined by [14] was designed as "a research study in which one or more independent variables (alleged causes) are deliberately manipulated to analyze the consequences of the manipulation on one or more dependent variables (alleged effects), within a control situation for the researcher" [15]. In this case, the number of records and the tool used to apply data mining were defined as separate variables. The response time and the result of the algorithms were identified as dependent variables. For a better understanding of the experiment design, the operational definition is summarized in Tables 1 and 2.

Table 1. Operationalization of independent variables

Variable	*Variable type*	*Operationalization*	*Categories*
number of records	Independent	The number of rows in the to be analyzed.	-100002 -500010 -1000020
Tool	Independent	Tool used to apply data mining	Weka, PostgreSQL (Algorithms integrated into SGBD)
Partitioning	Independent	Whether or not the table to which the data mining algorithms are to be applied is partitioned	Partitioned Not partitioned

Table 2. Operationalization of dependent variables

Variable	*Unit of measurement*
Response time	Time interval (seconds)
Algorithm results	Degree of agreement (yes or no)

3.1 Ratio of the variable number of records to the result time

In the first case, a study was carried out on how response time behaves in the Weka tools and the PostgreSQL manager when manipulating the variable Number of Records for each of the studied algorithms. Table 3 shows that as the number of records increased, the analysis times for the 1R algorithm increased and, in the case of the Weka tool, the response times were higher than the analysis made using algorithms built into PostgreSQL DBMS. For the category or level of 1000020 records, the analyses could not be performed with the Weka tool as it returned error due to the large amount of data.

Table 4 shows that it was the same case with the PRISM algorithm, in which as the number of rows increased, the analysis times increased. For the case of category or level 500010 records, the Weka tool took 3600 seconds to run without displaying the result and, in the case of 1000020, an error occurred.

AS in the above tables, Table 5 makes clear the directly proportional relationship between the number of records and the analysis time of the ID3 algorithm, highlighting the resolution times proposed in the research which are shorter.

Table 3. Result of manipulating the variable Number of Records for the 1R algorithm

Number of records	*Weka (1R)*	*PostgreSQL(1R)*
100002	11.13	1.5
500010	17.24	8.18
1000020	---	16.43

Table 4. Result of manipulating the variable Number of Records for the PRISM algorithm

Amount of registration	*Weka(PRISM)*	*PostgreSQL(PRISM)*
100002	11.33	10.1
500010	3600	94.7
1000020	---	219.45

Table 5. Result of manipulating the variable Number of ecords for the ID3 algorithm

Amount of registration	*Weka (ID3)*	*PostgreSQL(ID3)*
100002	7.42	2.58
500010	23.78	14.36
1000020	---	41.19

3.2 Ratio of the variable Number of Records to the responses of the algorithms

In case number 2, the behavior of the variable resulting from the algorithms is analyzed by manipulating the number of records to be analyzed.

By analyzing the results of Tables 6, 7, and 8, it can be concluded that as the number of records increased, the analysis of the data was made difficult by the Weka tool.

Table 6. Results of the relationship between the number of records and the results for the 1R algorithm

Amount of registration	*Weka (1R)*	*PostgreSQL(1R)*
100002	Yes	Yes
500010	Yes	Yes
1 000020	No	Yes

Table 7. Results of the relationship between the number of records and the results for the PRISM algorithm

Amount of registration	*Weka(PRISM)*	*PostgreSQL(PRISM)*
100002	Yes	Yes
500010	No	Yes
1000020	No	Yes

Table 8. Results of the relationship between the number of records and the results for the ID3 algorithm

Amount of registration	*Weka (ID3)*	*PostgreSQL(ID3)*
100002	Yes	Yes
500010	Yes	Yes
1000020	No	Yes

3.3 Validation of the partitioning mechanism

To validate that the proposed table partitioning improves the performance of the algorithms, the test_d table, which has 8000160 records, will be partitioned. As a result of partitioning, 3 mastertest_d tables are obtained as the parent table, test_dsi that contains all the records with a value "if" and test_dno records with "no" values. When applying the built-in 1R algorithm to PostgreSQL DBMS without partitioning, the response time is 174,184 seconds, but after partitioning the table, it is 149.98 seconds.

4. Conclusions

A function was developed to take advantage of one of the optimization mechanisms of the manager to improve the response results of algorithms 1 R, PRISM, and ID3. The algorithms implemented were validated by means of an experiment design that allowed to observe that the analysis times of the algorithms integrated into the DBMS are less than the results of the Weka tool).

References

[1] Viloria A., Lis-Gutiérrez JP., Gaitán-Angulo M., Godoy A.R.M., Moreno G.C., Kamatkar S.J. (2018) Methodology for the Design of a Student Pattern Recognition Tool to Facilitate the Teaching - Learning Process Through Knowledge Data Discovery (Big Data). In: Tan Y., Shi Y., Tang Q. (eds) Data Mining and Big Data. DMBD 2018. Lecture Notes in Computer Science, vol 10943. Springer, Cham.

[2] Viloria, A., Viviana Robayo, P.: Virtual network level of application composed IP networks connected with systems - (NETS Peer-to- Peer). Indian J. Sci. Technol. (2016). ISSN 0974-5645.

[3] Balaguera MI., Vargas MC., Lis-Gutierrez JP., Viloria A., Malagón L.E. (2018) Architecture of an Object-Oriented Modeling Framework for Human Occupation. In: Tan Y., Shi Y., Tang Q. (eds) Advances in Swarm Intelligence. ICSI 2018. Lecture Notes in Computer Science, vol 10942. Springer, Cham.

[4] Fairley, R.E., "Recent advances in software estimation techniques", Proceedings of the 14th international conference on Software engineering, Melbourne, Australia, 1992, pp.382 – 391.

[5] Walkerden, F. y Jeffery, D., "Software cost estimation: A review of models, process, and practice", Advances in Computers, Vol. 44, 1997, pp. 59-125.

[6] Boehm, B., Abts, C. y Chulani, S., "Software development cost estimation approaches-a survey", Annals of Software Engineering 10, 2000, pp. 177-205

[7] Wieczorek, I. y Briand, L., Resource estimation in software engineering, Technical Report, International Software Engineering Research Network, 2001.

[8] Piotrowski, A.P., 2017. Review of Differential Evolution population size. Swarm Evol. Comput. 32, 1–24. https://doi.org/10.1016/j.swevo.2016.05.003

[9] Kaya, I., 2009. A genetic algorithm approach to determine the sample size for attribute control charts. Inf. Sci. (Ny). 179, 1552–1566. https://doi.org/10.1016/j.ins.2008.09.024

[10] Gaitán-Angulo M, Jairo Enrique Santander Abril, Amelec Viloria, Julio Mojica Herazo, Pedro Hernández Malpica, Jairo Luis Martínez Ventura, Lissette Hernández-Fernández. (2018) Company Family, Innovation and Colombian Graphic Industry: A Bayesian Estimation of a Logistical Model. In: Tan Y., Shi Y., Tang Q. (eds) Data Mining and Big Data. DMBD 2018. Lecture Notes in Computer Science, vol 10943. Springer, Cham.

[11] Amelec, Viloria. "Increased efficiency in a company of development of technological solutions in the areas commercial and of consultancy." Advanced Science Letters 21.5 (2015): 1406-1408.

[12] MAcQueen, J., 1967. Some methods for classification and analysis of multivariate observations. Proc. Fifth Berkeley Symp. Math. Stat. Probab 1, 281–297.

[13] Abdul Masud, M., Zhexue Huang, J., Wei, C., Wang, J., Khan, I., Zhong, M., 2018. Inice: A New Approach for Identifying

the Number of Clusters and Initial Cluster Centres. Inf. Sci. (Ny). https://doi.org/10.1016/j.ins.2018.07.034
[14] Ramadas, M., Abraham, A., Kumar, S., 2016. FSDE-Forced Strategy Differential Evolution used for data clustering. J. King Saud Univ. - Comput. Inf. Sci. https://doi.org/10.1016/j.jksuci.2016.12.005
[15] Yaqian, Z., Chai, Q.H., Boon, G.W., 2017. Curvature-based method for determining the number of clusters. Inf. Sci. (Ny). https://doi.org/10.1016/j.ins.2017.05.024.