Vol. 4, No. 10 October 2013 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

Enhancement of Missing Values Prediction and Estimation Using Data Mining Algorithms

[1] Alaa. H. Al-Hamami, [2] Sarah Al-Samadi
[1] College of Computer Sciences and Informatics Amman Arab University
[2] Oil Products Distribution Company Western Division-Mosul Branch-Iraq
[1]Alaa_hamami@yahoo.com, [2] sarah_alsamady@yahoo.com

ABSTRACT

In real life, users of most of the database systems face problems related to missing values, whether these values are unknown to the users or inapplicable. The Database is an image of the real life and life is always incomplete. For this reason many researchers tried to complete the database information through estimating the missing values by applying several algorithms to gain high estimated accuracy rates. This estimation is through predicting and replacing the missing values with approximate values obtained from the development of an algorithm that combines two data mining algorithms (Decision Tree and K-nearest neighbor) for estimations and predictions. The treatment also checks and validates the stored and estimated values through comparing them with internally set business rules. Using Oracle DBMS, a framework is designed and implemented for the estimation of Null Value utilizing the standard database "ADULT", consisting of (32561) tuples, obtained from UCI Data Repository. The implementation of the present treatment shows treatment validity with a success rate of up to 80%.

Keywords: *Null value, Missing Value, Decision Tree Algorithm, Data Mining, and K-Nearest Neighbor Algorithm.*

1. INTRODUCTION

Any system designed for a database aims to provide data integrity and completeness. Some of the problems that are often faced when retrieving data or predicting values from the database appear when the databases contain missing values called "Null Values". The Null Values represent the values of attributes that may be unknown at the moment or may not apply to a tuple. These special values are called null and are also known as undefined values [1]. The reasons behind the Null Values can be classified according to the type of the Null Value which can be one of two types according to E.F.Codd way of representing the missing data in the relational database model which are [2]:

a. A-MARKS: Data applicable but not known; meaning that the data in fact exists but then missing such as the date of birth which is inevitable for everyone, but is unknown to the user entering the data in the database at the time or that the available database contains missing values.
b. I-MARKS: Data is Inapplicable; meaning that the data does not exist in reality such as the bonus of an unskilled typewriter clerk who doesn't have a bonus. Null Value here cannot be considered zero or a value.

When a DBMS deals with a database containing missing values, problems may appear of which few are shown hereunder:

a. When joining more than one table (LEFT /RIGHT JOIN, FULL JOIN) a problem may appear with the Null Value, even if the tables are originally complete, as it is possible that the problem of the Null Value appears when joining these tables.
b. When aggregate functions are used such as AVG, COUNT, SUM with a database containing Null Values, false results may appear.
c. When performing any mathematical operation and one of the entries is a Null Value, the result will be a Null Value whatever the other values are.
d. When performing any logical process or implementing program instructions which are essentially logical operations. The implementation of these instructions is a logical operation, and as there are three logical values as mentioned by E.F.Codd which are (TRUE, FALSE, UNKNOWN) [3], the UNKNOWN value represents the logical value of the Null Value.

There are different methods that process and solve the problem of missing values; some of which are shown in the following [4]:

- **Removal:** Ignoring or deleting the records that contain missing values and dealing only with complete data records. This option becomes non effective when there is a large amount of missing values within a limited amount of data and the deletion of the records that contain missing values leads to the loss of real and important data.

- **Imputation:** Filling in the missing values with default values (max, min or mean) for an attribute which contains Null Values. This option is non-effective because it is possible that, for example, one value, the highest value, is placed instead of all the missing values and this solution is not accurate.

- **Special Coding:** Filling in the missing values with a certain code indicating the missing value. This option is also not effective as placing indications of the absence of values does not solve the problem.

- **Estimation:** Filling in the missing values with estimated values using relationships that exist among the complete values in the dataset. This entails the estimation of the missing data and, filling in the missing values with the estimated values using techniques and algorithms such as decision trees, fuzzy neural network, or other methods which may represent more effective solutions [4].

In this research, the treatment of the missing values is done through estimating their values using two of data mining algorithms which are: Decision Tree for prediction and classification and K-nearest neighbor for estimation.

2. DATA MINING ALGORITHMS

Two data Mining Algorithms are used for the classification, prediction and estimation.

2.1 Decision Tree Algorithm

The use of the decision tree algorithm is either for classification or prediction and this flexibility makes it very attractive option. The use of the decision tree is not only widespread in the fields of probability and statistical pattern recognition but also in various fields such as medicine (diagnosis), computer science (data structures), botany (classification) and psychological (decision theory). It is possible to display the classification trees as images of schemes to facilitate their translation rather than just being digital translations [5].

2.2 K- Nearest Neighbor Algorithm (K-NN)

The k- nearest neighbor technique is a classification technique, i.e. it classifies similar cases and calculates the number of cases for each class and allocates the new case to the same class to which most of its neighbors belong. As well this technique can be used for estimation [5].

3. RELATED WORK

Most of the popular techniques for estimating missing values (Null Values) are using rules generation; for example replacing all the missing attribute values by the most frequently occurring attribute values and thereby completing the information table. Decision rules are generated using the redact of the decision table and these rules are also validated using ROSE2 (Rough Sets Data Explorer) software [6].

Other estimation techniques do not rely on the generation of rules or on the estimations based on these rules and base their estimates on the data available in the database; for example they estimate Null Values in relational database systems by applying K-means method along with genetic algorithm. The aim is to generate the right number of clusters and also to attain a high accuracy [7].

The present technique for estimating the missing values integrates the use of the rules, selected by the user to provide flexibility to change these rules when needed, together with the use of the data stored in the database. The rules are used to check the data stored in the database and then estimate the missing values using K-nearest neighbor algorithm and the Decision tree algorithm and finally checking the results against the available rules to verify the accuracy of estimations.

4. THE PROPOSED SOLUTION

The databases are images of the real world and thus could contain missing values. Upon entering the data the followings can be faced:

- The value is wrong, as a result of a human error.
- The value being unknown to the user entering the data in the database.
- There is no value that can possibly be entered in this field.
- An estimated value is stored in the database.
- The value is unknown to the user entering the data in the database and No Nearest Neighbor records exist.

In the present work the presence of missing values in the database is treated through a framework to process these missing values by estimating these values through the use of data mining algorithms (Decision Tree and K-nearest neighbor). The components of the framework are:

a. Decision Tree Algorithm

The use of the decision tree algorithm in the framework helps in adding big flexibility to the treatment and estimation of the missing values by classifying the stored values in the database as true, estimated or false, as well as predicting that the values being inapplicable or applicable. The decision tree is used in many processes in the framework, whether for the purpose of classification or prediction, adding flexibility to the treatment.

b. K mean – Nearest Neighbor Algorithm

The use of K mean - nearest neighbor algorithm forms the basis for the process of estimation in the present treatment. The main objective behind the use of this algorithm is to estimate the missing value by comparing it with the average of similar or near records in case these records are equal in frequency.

c. K frequency – Nearest Neighbor Algorithm

The use of K frequency - nearest neighbor algorithm forms the basis for the process of estimation in the present treatment. The main objective behind the use of this algorithm is to estimate the missing values by comparing them with similar or near records that are more frequent in case these records have different frequency.

d. Checking the Database

The process of checking and validating the database within the framework is done in two parts of the algorithm. The first part is checking each value stored in the database with the business rules and in case it is a true value, it remains stored as it is, and in case it is false, it is replaced by a Null Value and gets re-estimated, while in case the value is estimated it is also replaced by a Null Value and also gets re-estimated for flexibility in estimation, as when we add new records we may get better accuracy of the estimate compared to the previous estimate.

The second part of checking and validation comes at the end of the estimation whereby the true estimated value is checked whether it is within the allowed rules in the business table and in case it is, the estimated value gets stored while if it is false, re-estimation is made from the business table.

Thus, the checking and validation is a process to check whether the value in the database is true or false and to correct the error either by estimation or through an approximate value and in this case it helps in maintaining the integrity and correctness of the data.

5. PROPOSED ALGORITHM TO TREAT MISSING VALUES

The main aim of the present work is to treat the missing values which also help in the checking and validation of the database to ensure the correctness of the data contained in the database and as can be seen in the following detailing of the steps of the framework:

// - Input: two tables
- The input algorithms are represented by main table and business rules table.
- The main table containing the missing values is divided into two tables: The first table contains complete data with no Null Values, and the second table contains incomplete data.
- The output algorithms are represented by a table containing estimated values in place of an incomplete table.
- Output:- table //

Step 1: Start

Step 2: Check the database values with business rules.

Step 3: Use the Decision tree algorithm to classify and decide whether the value is true, false or estimated.

Step 4: In the event the value is classified as a true value, the stored value remains as is.

Step 5: In the event the value is classified as false or estimated, the false or estimated values are replaced by the Null Value.

Step 6: The Decision tree algorithm is used to classify and decide whether the Null Value is applicable or inapplicable.

Step 7: In the event the Null Value is classified as inapplicable, a value is placed as an indicator for the value of the inapplicable Null Value.

Step 8: In the event the Null Value is classified as containing applicable Null Value the K – nearest neighbor algorithm is used to calculate the distance between the values in tuple of the relevant fields and the values corresponding to the Null Values and the closest distance is found as the shortest distance being within the established limits or under the conditions placed and the tuples corresponding to the given condition are used for estimation.

Step 9: The Decision tree algorithm is used to classify and decide whether K-nearest neighbor records exist or do not exist.

Step 10: In the event no K-nearest neighbor records exist, a value is placed from the business rules table.

Step 11: In the event the K-nearest neighbor records exist, the tree algorithm is used to classify and decide whether one or more k-nearest neighbor records exist.

Step 12: In the event only one-nearest neighbor record exists, the Null Value is estimated using this record.

Step 13: In the event more than one K-nearest neighbor record exist, the tree algorithm is used to classify and decide whether different frequency records or equal frequency records exist.

Step 14: In the event the records are classified as different frequency records, the Null Value is estimated using K-nearest neighbor (the highest frequency).

Step 15: In the event the records are classified as equal frequency records, the Null Value is estimated using K-nearest neighbor (the average attribute of the nearest neighbor records).

Step 16: The estimated values are checked with business rules.

Step 17: The Decision tree algorithm is used to classify and decide whether the estimated values are true or false.

Step 18: In the event the estimated values are classified as true, the estimated values are stored.

Step 19: In the event the estimated values are classified as false, they are replaced with values from the business rules table.

Step 20: The database values are checked and if the database contains other Null Values, step (5) is returned and the process continues.

Step 21: Return (table containing estimated values in place of an incomplete table).

Step 22: Stop

The framework applies the previous algorithm to cure the problems of Null Values. The treatment of Null Values in the present work is done through checking the data stored in the database and estimating / re - estimation the Null Values which enables the generation of the reports, statistics and retrieval of data that reflect the real and true image of the database.

6. DISCUSSION

The present work formulates a technique for estimating the missing values in database systems based on the use of certain rules, selected by the user, to provide flexibility to change these rules when needed. These rules are used to check the data stored in the database and to estimate the missing values using K-nearest neighbor algorithm and the Decision tree algorithm, and the results are finally checked against the set rules to verify the accuracy of the estimations.

The success rate (as defined by number of correct predictions / total predictions) for the first stage implementation using the estimation algorithms without the adoption of the set rules is up to 76%, while the success rate for the second stage, using the estimation algorithms with the adoption of the set rules, rises up to 80%. The limitations of the proposed framework and the estimation method lie with small size databases as the accuracy of the estimation increases markedly with the increase in the size of the database due to the fact that the estimations are based on the nearest neighbor records.

REFERENCES

[1] Elmasri,R., Navathe ,S.B. ,"Database Systems models, languages, design, and application programming (6TH ed.)". Pearson, 2011.

[2] Codd, E.F. , "The Relational Model for Database Management": Version 2. Addison-Wesley , 2000.

[3] CODD, E.F. ,"Missing Information (Applicable and Inapplicable)in Relational Databases" SIGMOD RECORD, Vol. 15, No. 4, December 1986

[4] Du,H. , "Data mining techniques and applications an introduction " , nelson education ,ltd ,Canada , 2010.

[5] Al-Hamami, A.H., "Data Mining: Concepts, Techniques and Applications" , ITHRAA Publishing and Distribution, Jordan ,2008.

[6] ST.R. ,Garg M.L , "A Rough Set Approach for Generation and Validation of Rules for Missing Attribute Values of a Data Set ", International Journal of Computer Applications, volume 42-no 14 , 2012.

[7] Pandole,K. , Bhargava , N.," Comparison and Evaluation for Grouping of Null Data in Database Based on K-Means and Genetic Algorithm", International Journal of Computer Technology and Electronics Engineering (IJCTEE) , Volume 2, Issue 3,2012.

[8] Repository,UML: http://archive.ics.uci.edu/ml/datasets.html access at 1/3/2013

AUTHOR PROFILES

Dr. Alaa. H. Al-Hamami is presently professor of Database Security and dean of computer sciences and Informatics College, Amman Arab University, Jordan. He is a reviewer for several national and International Journals and a keynote speaker for many International Conferences. He is supervising a lot of PhD, MSc, and Diploma Theses. His research is focused on Distributed Databases, Data Warehouse, Data Mining, Cryptography, Steganography, and Network Security.

Sarah Al-Samadi received the bachelor degree in mathematical science from the university of Mosul in year 2000 and the bachelor degree in computer science from hadbaa university college in year 2004 and the master degree in computer science from Amman Arab university in year 2013, Currently, she is working as programmer for the Oil Products Distribution Company/Western Division-Mosul Branch-Iraq.