A ROBUST APPROACH FOR DATA CLEANING USED BY DECISION TREE INDUCTION METHOD IN THE ENTERPRISE DATA WAREHOUSE

Raju Dara¹ and Dr.Ch.Satyanarayana²

¹Research Scholar, Department of Computer Science & Engineering, Jawaharlal Nehru Technological University, Kakinada, India
²Professor, Department of Computer Science & Engineering, Jawaharlal Nehru Technological University, Kakinada, India

ABSTRACT

Now a day's every second trillion of bytes of data is being generated by enterprises especially in internet. To achieve the best decision for business profits, access to that data in a well-situated and interactive way is always a dream of business executives and managers. Data warehouse is the only viable solution that can bring the dream into veracity. The enhancement of future endeavours to make decisions depends on the availability of correct information that is based on quality of data underlying. The quality data can only be produced by cleaning data prior to loading into data warehouse since the data collected from different sources will be dirty. Once the data have been pre-processed and cleansed then it produces accurate results on applying the data mining query. Therefore the accuracy of data is vital for well-formed and reliable decision making. In this paper, we propose a framework which implements robust data quality to ensure consistent and correct loading of data into data warehouses which ensures accurate and reliable data analysis, data mining and knowledge discovery.

KEYWORDS

Data Cleaning, Data Preprocessing, Data Tree Induction, Data Quality, Data Integration

1.INTRODUCTION

Data received at the data warehouse from external sources usually contains various kinds of errors, e.g. Spelling mistakes, inconsistent conventions across data sources, and/or Missing fields, Contradicting data, Cryptic data, Noisy values, Data Integration problems, Reused primary keys, Non unique identifiers, inappropriate use of address lines, Violation of business rules etc. Data warehouse of an enterprise consolidates the data from several sources of the organization/enterprise in hoping to provide a unified view of the data that can be used for decision making, report generation, analyzing, planning, etc. The processes which can be performed on data warehouse for above mentioned activities are highly sensitive to achieve the quality of data, and depends upon the accuracy and consistency of data. Data cleansing deals with the dirty data so as to load the high data quality in a warehouse. The principle of data cleaning is to investigate errors and inconsistencies in data and it is a sub task of data preprocessing. The DOI:10.5121/ijcsa.2015.5406
The main objective of data cleaning is to necessarily ensure the quality, productive strategies and helping in decision support system. Eventually immediately after preprocessing, data is loaded in the Data warehouse. However data cleaning is not generalized i.e. the attributes will be differed from one domain to other. Hence the way of applying the data cleaning rules on a variety of attributes may often be different. In this framework the data is collected from Notepad, Microsoft Word, Microsoft Excel, Microsoft Access and Oracle 10g. In this paper, it is assumed a database named “Indiasoft” with Nine Attributes including int, char, String, Date data types, which contains dirty data. After applying algorithms the data from different sources is cleaned and inserted into a Data Warehouse. The Decision Tree Induction Algorithm is used to fill the Missing Values in different data sources and also provided solutions to clean Dummy Values, Cryptic Values, and Contradicting data.

2.RELATED WORK

Data cleaning is exclusively required when integrating heterogeneous data sources into data warehouses, and is a major part of the so-called ETL process [1]. Data quality perfection is an essential aspect of enterprise data management, data characteristics can change with customers, domain and geography making data quality improvement a challenging task [2], the problem of detecting and eliminating duplicated data is one of the major problems in the broad area of data cleaning and data quality [3]. Authors presented a declarative framework for collective deduplication of entity references in the presence of constraints and the constraints occur naturally in many data cleaning domains and can improve the quality of deduplication and discussed Dedupalog, the first language for collective deduplication, that is declarative, domain independent, is expressive enough to encode many constraints considered in prior art and scales to large data sets [4]. Discussed the aggregate constraints that often arise when integrating multiple sources of data, can be leveraged to enhance the quality of deduplication [5]. The interaction of the heterogeneous constraints by encoding them in a conflict hypergraph and have shown that this approach to holistic repair improves the quality of the cleaned database w.r.t. the same database treated with a combination of existing techniques [6]. Cleansing data from impurities is an integral part of data processing and maintenance [7]. Data preprocessing is a data mining technique that involves transforming raw data into an understandable format, real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors and data preprocessing is a proven method of resolving such issues [8]. Data collection has become a ubiquitous function of large organizations and not only for record keeping, but to support a variety of data analysis tasks that are critical to the organizational mission, data analysis typically drives decision-making processes and efficiency optimizations, data quality remains a pervasive and thorny problem in almost every large organization [9]. To ensure high data quality, data warehouses must validate and cleanse incoming data tuples from external sources, [10] proposes a new similarity function which overcomes limitations of commonly used similarity functions, and develop an efficient fuzzy match algorithm.

Considering the data cleansing as a vital part of renowned ETL in enterprise level data warehouse development to achieve data quality and this article proposes a pragmatic, interactive and easy to implementable data cleansing framework.

This paper is organized in the following order: Section 2, reviews literature in this area. Section 3 describes the data cleaning stages and process of data cleaning. The problem statement is explained in Section 4. Section 5, states the algorithm used in this paper. Section 6 states the implementation. Section 7, depicts experimental results as test cases, Section 8, gives the conclusion, limitation of the research and future work.
3. DATA CLEANING STAGES

Data quality improvement is achieved through data cleansing which has four stages namely, investigate, standardize, deduplication, and survivorship.

3.1. Investigation Stage

This is auditing phase in which the client data is analyzed for identifying errors and patterns in the data. This stage requires all or a sample of data to discover the data quality. Investigation results contain frequency reports on various tokens, labels and record patterns and the reports provide the basis for tuning standardization rule sets for a given customer data.

3.2. Standardization Stage

In this stage, the data is transformed into a standard uniform format which is accepted by the customer. Most commonly this stage involves in segmenting the data, canonicalization, correcting spelling errors, enrichment and other cleansing tasks using rule sets. This stage usually requires iterative tuning of rule sets and the tuning can be applied on the whole data or a sample data-set.

3.3. De-duplication/Matching Stage

This is an optional phase where standardized data from previous stages is used to identify similar or duplicate records within datasets. This stage is configured by providing parameters for the blocking and the matching steps. Blocking is used to reduce the search scope and matching is used to find the similarity between the records using edit distance or other known matching methods. It generally takes little iteration to decide on the thresholds in practice.

3.4. De-duplication/Matching

In this stage, the customer decides what data has to be retained after the match stage. If data is being merged from different sources then how the overall merging should take place are defined using rules. This is done by incorporating the inputs from the customer who decides the credibility of data sources.

3.5. Rules configuration data base

Rules configuration data base is a central repository that comprises of three varieties of rules, which are data extraction rules, transformation rules and business rules. These rules are the driving force throughout the data cleaning process and enter the rules configuration database based on data profiling results, user experience and data warehouse model requirements to execute data cleansing process.

The data extraction rules are needed to extract required data from larger data set that requires to be cleaned up and these rules are generally based on data profiling input after the data profiling of source systems. Basically the transformation rules define that what parameters, functions and approaches are required to clean the dataset, the data cleansing process uses transformation rules to clean the data according to inputs. The transformation rules can be data formatting type, removal of duplication records, default values of missing values, other related inconsistencies, etc.
3.6. Data cleansing process

Data goes through a series of steps during preprocessing [11]:

3.6.1. Data Cleaning

Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

3.6.2. Data Integration

Data with different representations are put together and conflicts within the data are resolved. Data Transformation: Data is normalized, aggregated and generalized.

3.6.3. Data Reduction

This step aims to present a reduced representation of the data in a data warehouse.

3.6.4. Data Discretization

Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

The data cleansing process takes two inputs such as one is data required to be cleaned, next is rules of cleansing from rules configuration database and the overall process is shown the following fig 1. This is the area where actual data cleansing processing takes place based on the rules from rules configuration repository and output of this process is error-free, consistent data that is ready to be loaded into data warehouse. This output data is standardized, uniform, accurate and complete with accordance to the business requirements. The cleaned data not only provides data quality, also expedite the processing speed and performance of overall ETL process.

Figure 1. Schematic diagram of Data Cleaning Stages
4. PROBLEM STATEMENT

Develop a framework with a list of data cleaning techniques and heterogeneous data sources in an easy way to understand and select the options. The user selects the data source and supplies the data cleaning technique on the data source. The cleaned data should be produced on single data source and multiple data sources at a time and integrated into a Relational database management system such as 10g.

5. ALGORITHM

5.1. Decision Tree Induction

Decision Tree Induction algorithm has been used to fill the missing values. For each attribute one decision tree will be constructed. Decision tree consists of a set of nodes. Each node is a test on an attribute. Two branches will come out from each node except for the leaf nodes. One branch is the “Yes” and the other is “No”. The Missed value is filled with the Leaf Nodes.

Algorithm: Generate _decision _tree.

Method:

a. Create a node N
b. If SAMPLES are all of the same class C, then
c. Return N as a leaf node labeled with the class C
d. If attribute_list is empty, then
e. Return N as a leaf node labeled with the most common class in SAMPLES; //majority visiting
f. Select test-attributes, the attribute among attribute_list with the highest information gain.
g. Label node N with test attribute
h. For each known value ai of test-attribute
i. Grow a branch from node N for the condition_test_attribute= ai
j. Let si be the set of samples in samples for which test_attribute= ai
k. If si is empty, then
l. Attach a leaf labeled with the most common class in samples
m. Else attach the node returned by Generate_decision_tree (si,attribute_list-test_attribute)

6. IMPLEMENTATION

6.1 Missing Values

Missing Values are replaced by Values in the leaf nodes of Decision Tree. Initially Data in the form of rows is collected from different Data Sources. Each row is a combination of several attributes. Each attribute is separated by comma “,”. The Nine Attributes which are considered in this work for cleaning are as follows

Empid, Empname, DOB, Age, Gender, Highqual, Income, Experience, CompanyName.

The following are the examples of missing values in the rows:

t1568,,50,,b.tech,5,7,wipro,

ramesh,,60,eee,,30,ibm,
Decision Tree for Age:

DOB Available?

YES

Age=2011-DOB.getYear

NO

Experience Available?

YES

Age = Experience+25

NO

Age=35

Decision Tree for Empid:

Figure 2: Diagram of Decision Tree for Age
Similarly Decision Trees will be constructed for every attribute and Missing values are replaced with leaf Node values.

6.2. Dummy Values

Internet is a place where Dummy Values will be generated in a huge range, for example, Email account registrations, download registrations, etc., and there no algorithm does exist at present to identify the dummy Values. Each domain follows its own method to identify and solve dummy values. The following are the examples of Dummy values.

abc  xyz  pqr  efg  hfg  yur  tur  wws  hbn  aaaa  bbbb  cccc  dddd  eeee  ffff  gggg  hhhh  iii  jjjj  abcdefg  higkls  bvbv  ddkd  lslls  slss  qqq  ppppp

After identifying this kind of values, we replace them with a global value “Indiasoft”, later when a Data Mining query is applied, whenever mining progression finds a value “Indiasoft”; it immediately knows that it is a dummy value. The mining will not be done based on this kind of values. Therefore accurate results will be obtained by cleaning dummy values in preprocessing.
6.3. Cryptic Values

A simple example for Cryptic Values is CGPA. If a user enters CGPA of a student as 9.0, the cleaning process should convert it to the percentage before inserting into the database because most of the universities prefer Percentages than CGPA.

7. TEST CASES

An Example for Contradicting data is Date of Birth (DOB) and Age. If Age and Year of DOB mismatches the cleaning process identifies this kind of Contradicting data. Then the Year of DOB is changed to Year + Age.

7.1. Test Case: 1

Input Specification: User Clicks the Start Cleaning Button without Selecting Algorithm
Output Specification: Displays a dialog “Please Select an Algorithm”
Pass/Fail: Pass

![Diagram](image)

Figure 4. Diagram shows the result of test case 1

7.2. Test Case: 2

Input Specification: User Clicks the Show Clean Table button without selecting Checkbox
Output Specification: Displays a dialog “Please Select Clean Table CheckBox”
Pass/Fail: Pass
7.3. Test Case: 3

Input Specification:

a. User Selects Microsoft Word Data Source & Missing Values Algorithm
b. Clicks Start Cleaning button
c. Clicks show Intermediate data button

Output Specification: Displays cleaned Missing Values in a Table format.
Pass/Fail: Pass
7.4. Test Case: 4

Input Specification: User Clicks the Show Oracle Table button without selecting Checkbox
Output Specification: Displays a dialog “Please Select Oracle Table CheckBox”
Pass/Fail: Pass

Figure 7. Diagram shows the result of test case 4

7.5. Test Case: 5

Input Specification: User Clicks the Show Access Table button without selecting Checkbox
Output Specification: Displays a dialog “Please Select Access Table CheckBox”
Pass/Fail: Pass

Figure 8. Diagram shows the result of test case 5
7.6. Test Case: 6

Input Specification:

a. User Selects the Microsoft Access Data Source & Missing Values Algorithm.
b. Clicks Start Cleaning button
c. Clicks show Intermediate data button.

Output Specification: Displays filled Missing Values in a Table format.
Pass/Fail: Pass

Figure 9. Diagram shows the result of test case 6

7.7. Test Case: 7

Input Specification:

a. User Selects the Microsoft Word, Microsoft Excel Data Sources & Missing Values, Dummy Values Algorithm.
b. Clicks Start Cleaning button
c. Clicks show Intermediate data button.

Output Specification: Displays filled Missing Values, Replaced dummy values in a Table format
Pass/Fail: Pass
8. CONCLUSIONS AND FUTURE WORK

In this Paper the sample database of Employees is designed with attributes such as Empid, Empname, Income, etc and named this database as “Indiasoft”. However some sample instances are considered for each attribute where the data cleaning is more effective by taking more instances for an attribute. Then the most probable values can be filled in missing values as well as contradicting values. We assumed some Dummy Values before comparing with the database values so that we replaced this with “indiasoft”. Whenever data mining queries discover a term indiasoft that immediately knows it as a dummy value, so mining approach will not be led through that path. In the future a data-cleansing product or application designed and developed using this framework, which can be building in-house or for commercial use. Warnings are issued and stored for any record that does not meet cleansing standards, so that it may be recycled through a modified cleansing process in the future.

REFERENCES