ISSN 2277-3061

Reducing the Impurity of Object-Oriented DatabaseThrough Gini Index

Sudhir Kumar Singh, Dr. Vipin Saxena
Deptt. of Computer Science and Engineering
Bagwant University (Rajasthan) 305001, INDIA
sudhir07singh@yahoo.com
Department of Computer Science
B.B. Ambedkar University (A Central University)
Lucknow (U.P.) 226025, INDIA
vsax1@rediffmail.com

ABSTRACT

In the current scenario, the size of database is increasing due to audio and video files. In the database, irregularities occur due to duplication of data at many places, therefore, it needs reconstruction of database size. The present work deals with reducing of impurity through a well-known Gini index technique. Since many of software's are using the object-oriented databases, therefore, an object-oriented database is considered, A real object-oriented database for Electricity Bill Deposit System is considered. A sample size of 15 records is considered, however the present technique can be applied for large size or even for the complex database. A decision tree is constructed and sample queries are performed for verifying the result and Gini index is computed for minimizing the impurity in the presented object-oriented database.

Indexing terms/Keywords

UML; Decision Tree; Classification rules; GINI Index; Gain ratio; Object-Oriented database

.

Council for Innovative Research
Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 13, No 11.

www.ijctonline.com , editorijctonline@gmail.com

INTRODUCTION

Classification rules are defined as a predefined data in the forms of groups and classes. A real case study of Customer Deposit Electrical Bill System (CDEBS) using Gini Index and gain ratio is considered. Classification rule is useful in data mining which arranges the data in a group wise fashion. UML shows the graphical representation of any database problem. Classification rules have been successfully used to propose the new system which pertains to the electric bills of the customer. The results establish the relationship between the income factor of the customer and the payments for bill received against the same. It is observed that performance factor of object-oriented database is higher in comparison to the relational database. If customer's income is low and billed amount is very high then probability of the customer is not depositing and the bill will be higher side and vice versa. In the present work another method of Decision Tree construction technique is also used and it forms a tree-like structure where each branch represents the nodes involved in a decision process.

Watanabe [1] has described object-oriented query language which is to be complex in comparison with relational query languages. The author showed that object-oriented database deals with complex objects and object-specific methods and addressed a formal model of object-oriented databases to attach it to a query language on the basis of the formal model. Karlapalem and Vieweg [2] described the object-oriented database systems which are becoming popular and are being used in a large number of application domain. Khoshgoftaar [3] has described decision trees to be attractive for a software quality classification problem which predict the quality of program module in terms of risk-based classes. Yin et al [4] have postulated that Multi relational classification is the procedure of building a classifier based on information stored in multiple relations and making predictions with it. Existing approaches of Inductive Logic Programming (ILP) has proven effectiveness his with high accuracy in multi relational classification. Alsaadi [5] described the class diagram to be the most important diagrammatic representation of object-oriented software systems and includes both the static and behavioral aspects. This can serve as a pattern for a persistent collection of objects, or as a scheme for a database system, and as a set of communication diagrams at the same time.

Ali et al. [6] have described that the Unified Modelling Language (UML) is which the most widely known and used notation for object-oriented analysis and design. UML consists of various graphical notations, which capture the static system structures, system component behaviour and system component interact-ions. UML notations can be produced with the help of CASE (Computer-aided software engineering) tools such as Rational Rose. Kwak and Moon [7] has described published Query Graph (QG) as an easy-to-use visual query language which facilitates formulating a query. Unlike relational databases, object-oriented databases (OODBs), the basic entity of QG, i.e. a class, may consist of several entities to which the operations of a query actually apply, which causes the increase of query complexity and lack of expressiveness. We propose a visual query language Object Query Diagram (OQD) for OODBs, where a class is specialized as a number of object sets which are the primitive entities of designation. Park et al. [8] have proposed a new complete GINI-Index text feature selection algorithm for text classification. This new algorithm obtains an unbiased feature values and from the feature subsets. This algorithm eliminates many irrelevant and redundant features and also retains many representative features. They also compared the new algorithm with the original versions of algorithm and demonstrated the classification performance. Zhongyang et al. [9] have proposed that decision tree based classification method is better than the other traditional statistical classification methods as it can deal with noise and lost information without depending on normal school data but does not need the requirement of normal distribution. It has been proved that the decision tree-based classification method has obvious advantages, such as exact classification, efficient, definite classification criterion, intuitive classification structure, controllable classification precision automated classification etc. Tsang et al. [10] described traditional decision tree classifiers which work with data values which are known and precise. They extend such classifiers to handle data with uncertain information. Value uncertainty arises in many applications during the data collection process. Example sources of uncertainty include measurement/quantization errors, data staleness, and multiple repeated measurements. With uncertainty, the value of a data item is often represented not by one single value, but by multiple values forming a probability distribution. Rather than abstracting uncertain data by statistical derivatives. They discover that the accuracy of a decision tree classifier and can be much improved if the "complete information" of a data item. One Versus All (OVA) decision trees learn k individual binary classifiers, each one to distinguish the instances of a single class from the instances of all Other classes. Thus OVA are different from existing data stream classification schemes whose majority uses multiclass classifiers, each one to discriminate among all the classes. Basheer et al. [11] have inferred that data mining has a goal to discover knowledge from huge volume of data. Rule mining is a beneficial and one of the most usable mining methods in order to obtain valuable knowledge from stored data on database.

Classification of Object-Oriented Database

The main objective of the object-oriented database system is to provide encapsulation, abstraction, and polymorphism data hiding concepts to implement the real world environment in data storage structure. The classes are formed and they are accessed using the created objects of the concerned classes. The classes can be reorganized without affecting its usage in any application. There are certain classification rules that describe the predetermined set of data and classes. In the present work the concerned data is stored in table format in the form of field Id, Name, Unit, Amount, and Decision. Classification rules are applied to table elecbill.t1 according to the prevalent or standard conditions. Classification process seeks to divide the compiled data into two parts: training data and test data. Utility of training data lies in the analysis of classification algorithm. There exists class labels that consist of various attributes such as Name, Unit, Amount, Decision and these represent the form of classification rules. On the other hand, the test data is used to predict the accuracy of classification rules. When accuracy rules are acceptable then these rules can also be applied to the classification of new data which have been added in the database. A bill depicting high usage may indicate a risky decision in bill submission and vice versa. A table is created in object-oriented database SQL server 2008. SQL server 2008 supports object-orieneted database property feature. The object-oriented database query performed by following step. First user creates the database name then creates the table name object-oriented database syntax.

```
    create database database_name;

      create table database_name.table_name
(
        column_name [ constraints ] [ default default ] ,
        [ column_name [ constraints ] [ default default ] , ]
        [ additional_columns ]
        [ unique ( column_name ) , ]
        [ counter ( column_name ) , ]
        [ timestamp ( column_name ) , ]
);

create database elecbill;

 create table elecbill.t1(Id INT NOT NULL, Name TEXT   NOT NULL, Unit INT, Amount  INT, Decision TEXT);

insert into elecbill.t1 values( Id, Name, Unit, Amount, Decision
)
values
(  101, "Ajay", 1000, 5000, “yes”);

insert into elecbill.t1 values(
        Id, Name, Unit, Amount, Decision
)
values
(
        102, "Vikash", 800, 4000, “no”
);

                select * from electbill.t1;
```

The output is recorded in the following table 1.

Table I: A Sample of Database (tablename:elecbill.t1)

Id	Name	Unit	Amount	Decision
101	Ajay	1000	5000	Yes
102	Vikash	800	4000	No
103	Aman	500	2785	Yes
104	Sohan	900	4656	Yes
105	Vinay	750	4000	No
106	Rama	802	4690	Yes
107	Somesh	347	2191	Yes
108	Akash	425	2734	Yes
109	Sanjeev	710	3810	No
110	Pranay	1210	6210	Yes
111	Ram	1321	7240	No
112	Sanjay	1121	6410	Yes
113	Manoj	1312	6812	Yes
114	Mahesh	850	4380	No
115	Suresh	465	3100	Yes

UML Class Diagram

UML class diagram shows graphical representation of any database problem. It is very useful technology in the software field and a standard visual modeling language and is divided into three parts. First part shows the class name, second part shows the attributes name and third parts shows the operations. Fig 1 shows the one to one relationship of UML class diagram which consists is linked to the Electical_office. Fig 2 shows the two many associations for the customer who deposits the electricity bill in electrical office. After defining these classes let us compute the Gini Index for reducing the impurity in the object-oriented database.

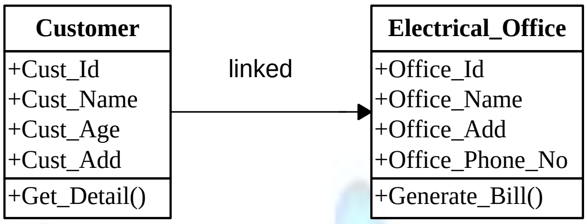


Fig 1: One to One Association Between UML Class

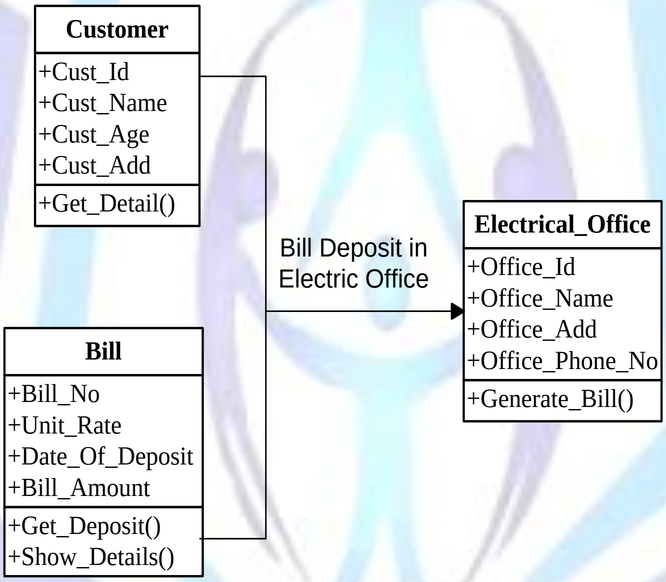


Fig 2: Two Way Association Between UML Classes

Impurity of Object-Oriented Database

Large amount of data can be stored in the object-oriented form which may contain a lot of useless data. Impurity removes the useless data from the stored object-oriented database. Impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. Table 1 stores the data field as ID, Income, Age, Deposit_bill, Deposit _decision and Deposit _ways. Some fields are not required in customer deposit in the electrical bill of customer and these fields are Age and Income. Necessary data is stored as object-oriented database and impurity removes the useless data from the database. When we apply the impurity then we face the problem of data redundancy. Redundancy removes the duplicate data from the object-oriented database. Electrical office provides the Cust_Id which is a primary key in object-oriented database table. The data partition is defined by following formula

$$\textbf{Info (D)} = -\sum p_i \log_2 (p_i) \qquad (1)$$

Where p_i is the nonzero probability of data tuple in D. Table 2 represents a training sets D, of class-labeled tuples D which are randomly selected from the Customer Electrical Bill Deposit System. The class label attribute Deposit_ways has two distinct values i.e. Yes and No. Therefore, there are two distinct classes defined as m=2. In the present work, we have taken as class C_1 which corresponds to Yes and class C_2 corresponds to No. There are 10 tuples of class Yes and 5 tuples of class No.

Table 2: Class Labeled Training Tuples From the Customer Electrical Bill Deposit System Database

ID	Age	Income	Deposit _bill	Deposit _decision	Deposit _ways
101	Youth	High	Yes	Fair	Yes
102	Youth	High	Yes	Fair	No
103	Middle	Low	No	Poor	Yes
104	Senior	Middle	No	Excellent	Yes
105	Senior	Middle	Yes	Excellent	No
106	Middle	Low	Yes	Fair	Yes
107	Youth	Low	No	Fair	Yes
108	Youth	High	Yes	Poor	Yes
109	Middle	High	Yes	Excellent	No
100	Youth	High	No	Excellent	Yes
111	Senior	Low	Yes	Fair	No
112	Middle	Middle	Yes	Poor	Yes
113	Youth	High	No	Excellent	Yes
114	Senior	Low	Yes	Fair	No
115	Youth	High	Yes	Fair	Yes
Entropy	**0.2896**	**0.0083**	**0.08508**	**0.03406**	**0.9184**
Gini Index	**0.45**				

compute the entropy by using the formula given in (1) and recorded in the above table. From the above table, there are 10 tuples with Yes and 5 for No.

Info (D)= -10/15 log2 10/15 – 5/15 log $_2$ 5/15

= 0.9184164

By applying similar steps, we compute the entropies for each column available in the above table 2. Next step we need to compute the expected information in each attribute. We find out value of Age and count the Yes and No tuples for each category of Age and Age category Youth are six Yes and one No tuples. For the category of Middle, there are three Yes and one No tuples, for the category of Senoir one Yes and three No tuple. Now applying these values in equation (1), we get

Info age(D)=7/15(-6/8 log$_2$ 6/8 – 1/15 log$_2$ 1/15) +4/15(-3/4 log$_2$ 3/4 -1/4 log$_2$ 1/4)+ 3/15(-2/3 log$_2$ 2/3 – 1/3 log $_2$ 1/3)
=**0.628811208**
Gain ($_{age}$)=Info(D)-Info $_{age}$(D)
= 0.9184164-0.628811208
=0.289605192

The entropy for the attribute Income is computed below:
Info $_{Income}$ (D) =7/15(-5/7 log$_2$ 5/7- 2/7 log$_2$ 2/7)+ 5/15(-3/5 log$_2$ 3/5- 2/5 log $_2$ 2/5)+3/15(-2/3 log$_2$ 2/3- 1/3 log$_2$ 1/3)
=**0.9100978**
Gain ($_{Income}$) =Info (D)-Info $_{Income}$(D)
=0.9184164-0.9100978
=**0.0083186**

The entropy for Deposit_bill is computed below:
Info $_{Deposit\_bill}$ (D) = 10/15(- 5/10 log2 5/10 – 5/10 log2 5/10)+ 5/15(- 5/10 log 2 5/10 – 0/10 log 2 0/10)
=0.833333
Gain$_{(Depositt\_bill)}$(D)=Info(D)- Info$_{(Depositt\_ bill)}$(D)

=0.9184164-0.8333333

=0.0850831

The entropy for Deposit_ decision is also computed below:

Info $_{\text{Deposit\_decision}}$ (D) = 7/15(- 4/7 log $_2$ 4/7 – 3/7 log $_2$ 3/7) + 3/15(-2/3 log$_2$ 2/3 – 1/3 log$_2$ 1/3)+ 5/15(-4 /5 log $_2$ 4/5 – 1/5 log$_2$ 1/5)

=**0.88407491**

The GINI Index measures the impurity of D, a data partition or set of training tuples and is given by:

$$\text{Gini (D)} = 1 - \sum_{i=1}^{m} p_i^2 \qquad (2)$$

In order to compute the Gini Index of Electrical Bill Deposit System, we take D be the training data from the table 2 where, there are 10 tuples belonging to the Deposit_ways Yes and the remaining 05 tuples belong to the Deposit_ways No and it is given below:

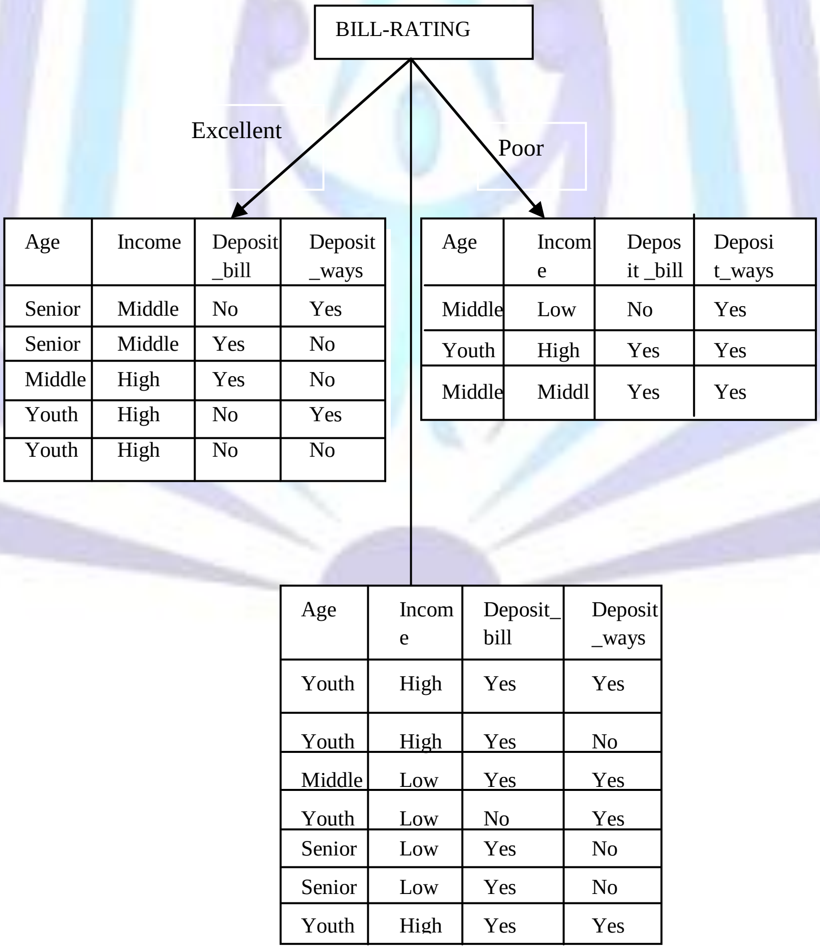
$$\text{Gini (D)} = (1-(10/15)^2 - (5/15)^2)$$
$$=0.45$$

The value of Gini Index from the table1=0.45 which shows the maximum reduction in impurity of tuples.

Decision Tree

A decision tree is like a flow chart but shows a tree-like structure. Decision tree represents a graphical symbol. Decision tree comprises of root nodes, internal node and leaf nodes. Internal node is also called non leaf node and each leaf node is called terminal node. Top position node in tree is root node. Internal node is denoted by rectangle and leaf node is denoted as oval. Principal advantage of decision tree is that it can handle multidimensional data. A decision tree for the said data is designed in figure 3. This diagram indicates customer is a root node whereas unit and income are internal nodes. Low and high are non-leaf nodes. This diagram shows customer submits the electrical bill depending on the unit and his income. Low income coupled with high unit is a typical condition wherein the customer does not deposit the bill.

.



Age	Income	Deposit _bill	Deposit _ways
Senior	Middle	No	Yes
Senior	Middle	Yes	No
Middle	High	Yes	No
Youth	High	No	Yes
Youth	High	No	No

Age	Incom e	Depos it _bill	Deposi t_ways
Middle	Low	No	Yes
Youth	High	Yes	Yes
Middle	Middl	Yes	Yes

Age	Incom e	Deposit_ bill	Deposit _ways
Youth	High	Yes	Yes
Youth	High	Yes	No
Middle	Low	Yes	Yes
Youth	Low	No	Yes
Senior	Low	Yes	No
Senior	Low	Yes	No
Youth	High	Yes	Yes

Fig 3: Design of Decision Tree for Electrical Bill Deposit System

Conclusions

From the above work, it is concluded that we can use UML for the representation of object-oriented database through UML class diagram. A powerful method for computation of entropy and Gini index is used for minimizing the impurity in the object-oriented database. When the size of database grows, then obviously impurity will grow and by using above technique, one can reduce the impurity in the database. The graphical representation of the database in the form of object-oriented is also presented through decision tree. The above work can also be implemented by the use of any higher level object-oriented programming language as UML is not dependent on the programming language.

ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

REFERENCES

[1] Watanabe, T., “A formalization of object-oriented database and its functional query language”, This paper appears in TENCON '94. IEEE Region 10's Ninth Annual International Conference. Theme: Frontiers of Computer Technology. Proceedings of 1994, Page 511-515, Date of Conference: 22-26 Aug 1994.

[2] Karlapalem, K. , Vieweg, S., “Method-induced partitioning schemes for object-oriented databases”, This paper appears in Distributed Computing Systems, 1996., Proceedings of the 16th International Conference, Page 377-384, 27-30May1996.

[3] Khoshgoftaar, T.M., “ IEEE Genetic programming-based decision trees for software quality classification”, This paper appears in IEEE Page 374–383, 3-5 Nov 2003.

[4] Yin, X., Han, X., Yang, Philip, “Efficient Classification across Multiple Database Relations: A Cross Mine Approach”, This paper appears in IEEE, Page No 770-783, June 2006.

[5] Alsaadi, A., “Checking Data Integrity via the UML Class Diagram”, This paper appears in Software Engineering Advances, International Conference, Page 37, Oct. 2006.

[6] Ali, N.H., Shukur, Z. , Idris, S. ,“A Design of an Assessment System for UML Class Diagram”, This paper appears in: Computational Science and its Applications, ICCSA, International Conference, Page 539-546 ,26-29 Aug,2007.

[7] Kwak, C.J., Moon, S., “Object Query Diagram: An Extended on Query graph for object-oriented databases”, Computational Science and its Applications, ICCSA, International Conference, Page 539-546, 26-29 Aug. .

[8] Park, H., Kwon, S., Kwon, C.H., "Complete Gini-Index Text (GIT) feature-selection algorithm for text classification," Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on , vol., no., pp.366-371, 23-25 June .

[9] Zhongyang, L., Huailiang, C., Chunhui, Z., “IEEE study on the land use and cover classification of zhengzhou based on decision tree”, Knowledge Acquisition and Modeling (KAM), 2011 Fourth International Symposium on, Page 405-408, Oct 2011.

[10] Tsang, S., Kao, B., Kevin Y. Y., Ho, W., S Lee, S.D., “Decision trees for uncertain data ”,This paper appears in IEEE, vol. 23 Page . 64-78, 2011.

[11] Basheer M.AL-Maqalesh, Shahbazkia, H., “A Genetic Algorithm for discovering classification rules in data mining “, International Journal of Computer Application Vol.41,Page no18.

Author’ biography with Photo

Dr. Vipin Saxena is a Professor & Head, Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India. He got his M.Phil. Degree in Computer Application in 1991 & Ph.D. Degree work on Scientific Computing from University of Roorkee (renamed as Indian Institute of Technology, Roorkee, India) in 1997. He has more than 16 years teaching experience and 19 years research experience in the field of Scientific Computing & Software Engineering. Currently he is proposing various software designs by the use of Unified Modeling Language for the research problems related to the Software Domains & Advanced Computer Architecture. He has published more than 100 International and National research papers in various refereed Journals and authored four books covering Software Engineering, E-Learning and Operating System. Dr. Saxena is a life time member of Indian Congress.

Mr. Sudhir Kumar Singh has completed master degree in computer science and applications. Mr. Singh has supervision some project. He has more than eight master students supervised in various project. He has good grip in various software language ie. Java, C,C++ etc. Currently Research Area data mining,Object-Oriented Database and UML .He has also Published research paper in his respective field.