Data Quality Mining using Genetic Algorithm

Sufal Das sufal.das@gmail.com
Lecturer, Department of Information Technology
Sikkim Manipal Institute of Technology
Rangpo-737136, India

Banani Saha bsaha_29@yahoo.com
Reader. Department of Computer Science & Engineering
Calcutta University
Kolkata, Pin 700073, India

ABSTRACT

Data quality mining (DQM) is a new and promising data mining approach from the academic and the business point of view. Data quality is important to organizations. People use information attributes as a tool for assessing data quality. The goal of DQM is to employ data mining methods in order to detect, quantify, explain and correct data quality deficiencies in very large databases. Data quality is crucial for many applications of knowledge discovery in databases (KDD). In this work, we have considered four data qualities like accuracy, comprehensibility, interestingness and completeness. We have tried to develop Multi-objective Genetic Algorithm (GA) based approach utilizing linkage between feature selection and association rule. The main motivation for using GA in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction that the greedy rule induction algorithms often used in data mining.

Keywords: Data Quality, Genetic Algorithm, Multi-objective Optimization, Association Rule Mining.

1. INTRODUCTION

The main contribution of this paper is to give a first impression of how data mining techniques can be employed in order to improve data quality with regard to both improved KDD results and improved data quality as a result of its own. That is, we describe a first approach to employ association rules for the purpose of data quality mining [1]. Data mining is about extracting interesting patterns from raw data. There is some agreement in the literature on what qualifies as a "pattern", but only disjointed discussion of what "interesting" means. Problems that hamper effective statistical data analysis stem from many source of error introduction. Data mining algorithms like "Association Rule Mining" (ARM) [2,3] perform an exhaustive search to find all rules satisfying some constraints. Hence, the number of discovered rules from database can be very large. Typically the owner of the data is not fully aware of data quality deficiencies. The system might have been doing a good job for years and the owner probably has its initial status in mind. Doubts concerning data quality may raise astonishment or even disaffection. We often have been facing exactly this situation. By trying to make the best of it we employed our skills – data mining techniques – as a patched-up solution to measure, explain, and improve data quality. Based on the earlier works, it is clear that it is difficult to identify the most effective rule. Therefore, in many applications, the size of the dataset is so large that learning might not work well. The

generated rule may have a large number of attributes involved in the rule thereby making it difficult to understand. If the generated rules are not understandable to the user, the user will never use them. Again, since more importance is given to those rules, satisfying number of records, these algorithms may extract some rules from the data that can be easily predicted by the user. It would have been better for the user, if the algorithms can generate some of those rules that are actually hidden inside the data. Also, the algorithm should capture all attributes which are useful. By introducing data quality mining (DQM) we hope to stimulate research to reflect the importance and potentials of this new application field. In this paper the authors have considered Association Rule Mining and tried to improve this technique by applying Multi-objective Genetic Algorithms (MOGA) [9] on the rules generated by Association Rule Mining based on four data qualities (objectives): accuracy, comprehensibility, interestingness and completeness. A brief introduction about Association Rule Mining and GA is given in the following sub-sections, followed by methodology, which will describe the basic implementation details of Association Rule Mining and GAs. The authors will discuss the results followed by conclusion in the last section.

2. RELATED WORKS

2.1 Association Rule Mining

Let $I = \{I_1, I_2 \ldots\ldots, I_m\}$ be a set of m distinct attributes, also called literals. Ai = r is an item, where r is a domain value is attribute, Ai in a relation, R (A1, …, An). A is an itemset if it is a subset of I. D =. { ti , t_{i+1} ,……, t_n) is a set of transactions, called the transaction (tid, t-itemset). A transaction t contains an itemset A if and only if, for all items $i \in A$, i is in t-itemset. An itemset A in a transaction database D has a support, denoted as Supp (A) (we also use p(A) to stand for Supp (A)), that is the ratio of transactions in D contain A. Supp(A) = |A(t)| / |D|, Where A(t) = {t in D/t contains A}. An itemset A in a transaction database D is called a large (frequent) itemset if its support is equal to, or greater than, a threshold of minimal support (minsupp), which is given by users or experts. An association rule is an expression of the form IF A THEN C(or A -> C), $A \cap C = \phi$, where A and C are sets of items. The meaning of this expression is that transactions of the databases, which contain A, tend to contain C. Each association rule has two quality measurements: support and confidence, defined as:

1) The support of a rule A->C is the support of $A \cup C$, where $A \cup C$ means both A and C occur at the same time.
2) The confidence or predictive accuracy [2] of a rule A->C is conf (A->C) as the ratio: $|(A \cup C)(t)| / |A(t)$ or $Supp(A \cup C) / Supp(A)$.

That is, support = frequencies of occurring patterns; confidence = strength of implication. Support-confidence framework (Agrawal et al. 1993): Let I be the set of items in database D, A, $C \subseteq I$ be itemset, $A \cap C = \phi$, p(A) is not zero and p(C) is not zero. Minimal support minsupp) and minimal confidence (minconf) are given by users or exparts. Then A->C is a valid rule if

1. Supp ($A \cup C$) is greater or equal to minsupp,
2. Conf (A->C) is greater or equal to minconf.

Mining association rules can be broken down into the following two sub-problems:

1. Generating all itemsets that have support greater than, or equal to, the user specified minimal support. That is, generating all large itemsets.
2. Generating all the rules that have minimum confidence.

2.2 Genetic Algorithm

Genetic Algorithm (GA) [8] was developed by Holland in 1970. This incorporates Darwinian evolutionary theory with sexual reproduction. GA is stochastic search algorithm modeled on the process of natural selection, which underlines biological evolution. GA has been successfully applied in many search, optimization, and machine learning problems. GA process in an iteration manner by generating new populations of strings from old ones. Every string is the encoded binary, real etc., version of a candidate solution. An evaluation function associates a fitness measure to every string indicating its fitness for the problem. Standard GA apply genetic operators such selection, crossover and mutation on an initially random population in order to compute a whole generation of new strings.

- **Selection** deals with the probabilistic survival of the fittest, in that more fit chromosomes are chosen to survive. Where fitness is a comparable measure of how well a chromosome solves the problem at hand.
- **Crossove**r takes individual chromosomes from P combines them to form new ones.
- **Mutation** alters the new solutions so as to add stochasticity in the search for better solutions.

In general the main motivation for using GAs in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining. This section of the paper discusses several aspects of GAs for rule discovery.

3. METHODOLOGY

Representation of rules plays a major role in GAs, broadly there are two approaches based on how rules are encoded in the population of individuals ("Chromosomes") as discussed in Michigan and Pittsburgh Approach [12]; The pros and cons as discussed in [12] is as follows, Pittsburgh approach leads to syntactically-longer individuals, which tends to make fitness computation more computationally expensive. In addition, it may require some modifications to standard genetic operators to cope with relatively complex individuals. By contrast, in the Michigan approach the individuals are simpler and syntactically shorter. This tends to reduce the time taken to compute the fitness function and to simplify the design of genetic operators. However, this advantage comes with a cost. First of all, since the fitness function evaluates the quality of each rule separately, now it is not easy to compute the quality of the rule set as a whole - i.e. taking rule interactions into account. In this paper Michigan's approach is opted i.e. each individual encodes single rule. The encoding can be done in a number of ways like, binary encoding or expression encoding etc. For example let's consider a rule "If a customer buys milk and bread then he will also buy butter", which can be simply written as

If milk **and** bread **then** butter

Now, in the Michigan approach where each chromosome represents a separate rule. In the original Michigan approach we have to encode the antecedent and consequent parts separately; and thus this may be an efficient way from the point of space utilization since we have to store the empty conditions as we do not known a priori which attributes will appear in which part. So we will follow a new approach that is better than this approach from the point of storage requirement. With each attribute we associate two extra tag bits. If these two bits are 00 then the attribute next to these two bits appears in the antecedent part and if it is 11 then the attribute appears in the consequent part. And the other two combinations, 01 and 10 will indicate the absence of the attribute in either of these parts. So the rule AEF->BC will look like 00A 11B 11C 01D 00E 00F. In this way we can handle variable length rules with more storage efficiency, adding only an overhead of 2k bits, where k is the number of attributes in the database. The next step is to find a suitable scheme for encoding/decoding the rules to/from binary chromosomes. Since the positions of attributes are fixed, we need not store the name of the attributes. We have to encode the values of different attribute in the chromosome only.

3.1 Multi-objective Optimization

Although it is known that genetic algorithm is good at searching for undetermined solutions, it is still rare to see that genetic algorithm is used to mine association rules. We are going to further investigate the possibility of applying genetic algorithm to the association rules mining in the following sections. As genetic algorithm is used to mine association rule, among all measurements, one measurement is accuracy or confidence factor. In the present work we have used another three measures of the rules like comprehensibility [9], interestingness [10] and completeness, in addition to predictive accuracy. Using these four measures, some previously unknown, easily understandable and compact rules can be generated. It is very difficult to quantify understandability or comprehensibility. A careful study of an association rule will infer that if the number of conditions involved in the antecedent part is less, the rule is more comprehensible. To reflect this behavior, an expression was derived as comp = N – (Number of conditions in the antecedent part). This expression serves well for the classification rule generation where the number of attributes in the consequent part is always one. Since, in the association rules, the consequent part may contain more than one attribute; this expression is not suitable for the association rule mining. We require an expression where the number of attributes involved in both the parts of the rule has some effect. The following expression can be used to quantify the comprehensibility of an association rule,

$$\text{Comprehensibility} = \log (1 + |C| / (|D| - |A|)) * (1 / |A|)$$

Here, |C| and |A| are the number of attributes involved in the consequent part and the antecedent part, respectively and |D| is the total number of records in the database.

It is very important that whatever rule will be selected for useful one this rule should represent all useful attributes or components. For that we have to select compact association rule with all useful features. So, we have to find out the frequent itemset with maximum length. The antecedent part and consequent for an association rule should cover all useful features as well as the two parts should be frequent. The following expression can be used to quantify the completeness of an association rule,

$$\text{Completeness} = (\log (1 + |C| + |A|) / |D|)* \text{Supp}(A) * \text{Supp}(C)$$

Here, |C| and |A| are the number of attributes involved in the consequent part and the antecedent part, respectively and |D| is the total number of records in the database. Supp (A) and Supp(C) are the occurrences of Antecedent part and consequent part respectively.

Since association rule mining is a part of data mining process that extracts some hidden information, it should extract only those rules that have a comparatively less occurrence in the entire database. Such a surprising rule may be more interesting to the users; which again is difficult to quantify. For classification rules it can be defined by information gain theoretic measures. This way of measuring interestingness for the association rules will become computationally inefficient. For finding interestingness the data set is to be divided based on each attribute present in the consequent part. Since a number of attributes can appear in the consequent part and they are not pre-defined, this approach may not be feasible for association rule mining. The following expression can be used to define as interestingness of an association rule,

$$\text{Interestingness} = \text{Supp}(A)* [(1- \text{Supp}(C)) /(1-\text{Supp}(AUC))]* [\text{Supp}(AU\ C) / \text{Supp}(A)] *[\text{Supp}(AUC) /\text{Supp}(C)].$$

This expression contains two parts. The first part, Supp (A)* [(1– Supp(C)) /(1– Supp(AUC))], compares the probability that A appears without C if they were dependent with the actual frequency of the appearance of A. The remaining part measures the difference of A and C appearing together in the data set and what would be expected if A and C where statistically dependent.

3.2 Genetic Algorithm with Modifications

• **Individual Representation** can be performed using Michigan's approach, i.e. each individual encodes single rule, as discussed in previous section.

• **Selection** is performed as the chromosomes are selected (using standard selection scheme, e.g. roulette wheel selection) using the fitness value. Fitness value is calculated using their ranks,

which are calculated from the non-dominance property of the chromosomes. A solution, say a, is said to be dominated by another solution, say b, if and only if the solution b is better or equal with respect to all the corresponding objectives of the solution a, and b is strictly better in at least one objective. Here the solution b is called a non-dominated solution. The ranking step tries to find the non-dominated solutions, and those solutions are ranked as one. Among the rest of the chromosomes, if p_i individuals dominate a chromosome then its rank is assigned as $1 + p_i$. This process continues till all the chromosomes are ranked. Then fitness is assigned to the chromosomes such that the chromosomes having the smallest rank gets the highest fitness and the chromosomes having the same rank gets the same fitness. After assigning the fitness to the chromosomes, selection, replacement, crossover and mutation operators are applied to get a new set of chromosomes, as in standard GA.

3.3 Our Approach

Our approach works as follows:
1. Load a sample of records from the database that fits in the memory.
2. Generate N chromosomes randomly.
3. Decode them to get the values of the different attributes.
4. Scan the loaded sample to find the support of antecedent part, consequent part and the rule.
5. Find the confidence, comprehensibility, completeness and interestingness values.
6. Rank the chromosomes depending on the non-dominance property.
7. Assign fitness to the chromosomes using the ranks, as mentioned earlier.
8. Select the chromosomes, for next generation, by roulette wheel selection scheme using the fitness calculated in Step 7.
9. Replace all chromosomes of the old population by the chromosomes selected in Step 8.
10. Perform crossover and mutation on these new individuals.
11. If the desired number of generations is not completed, then go to Step 3.
12. Decode the chromosomes in the final stored population, and get the generated rules.
13.Select chromosomes based on accuracy, comprehensibility, completeness and interestingness.

4. INPLEMENTATION & RESULT

The proposed technique has been implemented on different data sets with satisfactory results. Here we present the results on one such data set having 47 attributes and 5338 records. Crossover and mutation probabilities were taken respectively as 0.87 and 0.0195; the population size was kept fixed as 50. Number of generations was fixed as 300. Best four rules which were selected based on accuracy, comprehensibility completeness and interestingness, are put in the following table.

Rule No	Antecedent part	Consequent Part	A value	C value	I value	Co value
1	{4->3}, {18->0}, {28->3}, {38->0}, {39->2}	{1->1}, {5->2}, {9->0}, {12->3}, {15->0}, {17->1}, {21->1}, {24->2}, {26>2}, {29->0}, {31->2}, {33->2}, {34->3}, {37->1}	0.2151	0.0601	0.0128	0.0168
2	{5->3}, {12->2}, {13->2}, {15->0}, {24->1}, {26->3}	{1->2}, {4->2}, {9->0}, {16->3}, {17->1}, {20->1}, {21->2}, {28->3}, {30->1}, {31->1}, {33->3}, {34->2}, {35->0}, {37->2}, {38->1}	0.1921	0.0519	0.0148	0.01279
3	{1->2}, {3->2}, {5->2}, {7->1}, {12->3}, {16->2}, {21->3}	{2->0}, {4->2}, {9->1}, {13->1}, {15->0}, {17->2}, {20->3}, {24->2}, {26->3}, {28->1}, {30->0}, {31->0}, {32->0}, {34->2}, {38->1}, {39->3}	0.2129	0.0418	0.0173	0.01085
4	{5->2}, {12->2}, {15->0}, {24->1}, {31->1}, {38->2}, {39->0}	{3->0}, {4->3}, {8->3}, {9->2}, {17->1}, {21->2}, {26->2}, {28->0}, {30->3}, {31->2}, {34->0}, {36->3}	0.1874	0.0329	0.0164	0.00983

N.B.: [A, C, Co and I stand for accuracy, comprehensibility, completeness and interestingness respectively and {1->0} stands for attribute no.1 having value 0.]

If we consider the first rule, then 19 attributes are involved. Now, we can say that out of 47 attributes, these 19 attributes are useful.
Here, we have got some association rules which optimal according their accuracy, comprehensibility, interestingness and completeness. If we study the result then, we can select 19 to 22 attributes among 47 attributes. Using completeness measurement we can say that these 19 to 22 attributes are only the useful, no more is left out. If completeness measurement is not considered then we have seen that 14 to 17 attributes were selected and all were useful according the other three measurements.

5. CONSLUSION & FUTURE WORK

The use of a multi-objective evolutionary framework for association rule mining offers a tremendous flexibility to exploit in further work. In this present work, we have used a Pareto based genetic algorithm to solve the multi-objective rule mining problem using four measures—completeness, comprehensibility, interestingness and the predictive accuracy. We adopted a variant of the Michigan approach to represent the rules as chromosomes, where each chromosome represents a separate rule. The approach can be worked with numerical valued attributes as well as categorical attributes.
This work is able to select all useful attributes for any sort of dataset. One big advantage of this approach is that user or expert do not need to have any knowledge the dataset. No threshold value is not used here.
This approach may not work properly is the given dataset is not homogeneous as this is applied on a sample of dataset. Sample of any dataset does not represent the whole dataset completely. In future we can work to remove this disadvantage.

6. REFERENCES

1. Jochen Hipp,Ulrich G¨untzer and Udo Grimmer, *"Data Quality Mining - Making a Virtue of Necessity",* In Proceedings of the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD) 2001.

2. R. Agrawal, R. Srikant, *"Fast algorithms for mining association rules",* in Proceeding of the 20th Int'l Conference on Very Large Databases, Chile, 1994.

3. Imielinski, T., R. Agrawal and A. Swami, "*Mining association rules between sets of items in large databases"*. Proc. ACM SIGMOD Conf. Management of Data, pp: 207–216.

4. K.M. Faraoun, A. Rabhi, "*Data dimensionality reduction based on genetic selection of feature subsets",* EEDIS UDL University- SBA, (2002).

5. Cheng-Hong Yang, Chung-Jui Tu, Jun-Yang Chang Hsiou-Hsiang Liu Po-Chang Ko, "*Dimensionality Reduction using GA-PSO"(2001).*

6. P_adraig, "*Dimension Reduction",* Cunningham University College Dublin Technical Report UCDCSI-2007-7 August 8th, 2007

7. Erick Cantu-Paz, *"Feature Subset Selection, Class Separability, and Genetic Algorithms",* Center for Applied Scientic Computing Lawrence Livermore National Laboratory Livermore, CA, (1994).

8. M. Pei, E. D. Goodman, F. Punch, "*Feature Extraction using genetic algorithm", Case Center for Computer-Aided Engineering and Manufacturing* W. *Department of Computer Science,(2000).*

9. Sufal Das, Bhabesh Nath, "*Dimensionality Reduction using Association Rule Mining"*, IEEE Region 10 Colloquium and Third International Conference on Industrial and Information Systems (ICIIS 2008) December 8-10, 2008, IIT Kharagpur, India

10. Hsu, W., B. Liu and S. Chen, *"Ggeneral impressions to analyze discovered classificationrules",.* Proc. Of 3rd Intl. Conf. On Knowledge Discovery & Data Mining (KDD-97), pp: 31–36.AAAI Press.(1997)

11. Freitas, A.A., E. Noda and H.S. Lopes,. *"Discovering interesting prediction rules with a genetic algorithm"'*. Proc. Conf. Evolutionary Computation, (CEC-99), pp: 1322–1329.(1999)

12. Cristiano Pitangui, Gerson Zaverucha, "*Genetic Based Machine Learning:Merging Pittsburgh and Michigan, an Implicit Feature Selection Mechanism and a New Crossover Operator"*, Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06).(2006).