International Journal of Engineering Research And Advanced Technology(IJERAT)
ISSN: 2454-6135 [Volume. 02 Issue.07, July – 2016]
www.sretechjournal.org

COMBINING ROUGH SET WITH CERTAINTY RATIO BASED ALGORITHM

Sura N. Maryoush[1], Ahmad T. Sadiq[2]

[1]Computer science Department, College of Education

Al-Mustansiriya university

[2]Department of computer science, University of Technology

Baghdad, IRAQ

ABSTRACT

Rough set theory (RS) is one of the important methods that concerned with analysis of data tables for the classification purposes. In this paper, a certainty ratio based algorithm is proposed. The proposed algorithm uses certainty ratio as criterion for selecting the attribute values that have a good certainty ratio, and ignore the attribute values that has low certainty level. This algorithm is combined with rough set theory in order to enhance the generated rules. The experiments show that the proposed approach can effectively increase the accuracy of the generated rules.

KEYWORDS: Rough set theory, Certainty ratio, Certainty ratio based algorithm, Generated rules.

1.INTRODUCTION

Rough set theory was introduced by Zdzislaw Pawlak in 1982. It is one of the effective methods that concerned with analysis of data tables for the classification purposes, and dealing with uncertainty and vagueness that exist in the data tables [1]. Rough sets consider as an extension to the set theory, which uses approximations in order to make decisions[2]. In recent years, there has been a fast growing interest in this new emerging theory. The successful applications of the rough set model in a variety of problems have amply demonstrated its usefulness and versatility [3]. It is turning out to be rationally significant to artificial intelligence and cognitive science, especially in the representation of and reasoning with vague and/or imprecise knowledge, machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems and pattern recognition [4].

In this paper, a new algorithm is proposed to enhance the generated rules by rough set theory called a certainty ratio based algorithm. This algorithm aims to select only the attribute values that has a good level of certainty by using the certainty ratio as a criterion for selecting these values, and ignore the attributes values that have low level of certainty. The proposed algorithm are combined

with rough set theory in order to decrease the complexity of the generated rules. The proposed approach is tested using four datasets, which contains different data from different sources

The other sections in this paper are organized as follows: section 2 presents a literature related works. Section 3 explains the main concepts of rough set theory. Section 4 describes the proposed certainty ratio based algorithm and rough set after combination with the proposed algorithm. The experimental results and discussion are presented in section 5, followed by the conclusion in section 6.

2. RELATED WORKS

Many literatures related of enhancing the performance of rough set theory are proposed. In 2004,X. Hu, J. Han and T. Y. Lin proposed a scalable rough set model. In order to have the benefits of the efficient set- oriented operations of database, core attributes and reducts are redefined by depending on relational algebra. These new definitions are used to introduce a new algorithm for calculating core attributes and another algorithm to calculate reducts. The introduced algorithms can be applied to a wide range of database systems and can be accommodated in a wide range of real world applications, and that because of the relational algebra has been accomplished efficiently on these database systems [5].

Also in 2008, J. Liu, Q. Hu and D. Yu developed a weighted method based on rough set for dealing with the problem of class imbalance, and weights introduced into the rough sets for balancing the class distribution of data sets. At the beginning, a weighted algorithm is designed for reduction of attribute by extending and presenting Guiasu weighted entropy. After that, a weighted algorithm for rule extraction is designed by presenting a heuristic strategy which it is also weighted, and eventually weighted decision algorithm is designed by presenting some weighted factors in order to evaluate the extracted rules. The empirical evaluation proves The developed method is effective in learning imbalance classes [6].

In 2009, S. Trabelsi, Z. Elouedi and P. Lingras proposed a classifier based on rough sets. This classifier induced from not a totally uncertain decision table. It assumes that the uncertainty is in decision attribute and not exist in condition attributes. The belief function is selected to typify uncertainty, which empowers a flexible representation of total or partial ignorance. This belief function which is used in the proposed classifier is based on the transferable belief model [7].

In 2014, H. Feng, et al. proposed a new rough set theory as a classification algorithm to generate rules and presented a new concept of discernibility degree of condition attribute. The main feature of the proposed algorithm is it can calculate core attributes without calculating reducts before; another feature is that it does not calculate the core values for inconsistent elements and it select condition attribute value that has a maximum discernibility degree to generate rules for these elements [8].

3. ROUGH SETS CONCEPTS

As mentioned before, rough sets deals with analyzing the uncertainty and vagueness that may be found in data [9]. The vagueness concepts are approximated by rough set, where the rough set presents two precise concepts, named lower approximation and upper approximation, which represents a classification of the interest domain into disconnected categories. The elements of the domain that certainly belong to the interest subset are described by means of the lower approximation, while the elements that not certainly belong to the interest subset are described by means of the upper approximation [2].

The vagueness is expressed in rough sets by using boundary region of the set, in additional to the means of set elements membership. The boundary region consists of elements that exist in the upper approximation and not exist in the lower approximation (the difference between the upper and lower approximations). The set is counted as crisp set, if its boundary region is empty, otherwise it is considered as a rough set [10]. In real life, the data have different levels of complexity and sizes, which makes the data is difficult to be analyzed and also hard to be managed from computational view point. The main aims of Rough Set analysis are to handle inconsistency that exist in data and to reduce the size of data [9].

The basic concepts of rough set theory can be explained by the following definitions[1] :

Definition 3.1: let S be an information system, S=(U, A), where U represents a finite set of instances called a universe, and A represents a finite set of attributes. S is called a decision table if A is distinguished and partitioned into C and D, where C is a set of condition attributes and D is decision attribute, such that S=(U,C∪D). For every a∈C, a:V→Va, where Va represents a set of values of the attribute a, which called the domain of attribute a.

Definition 3.2: let S=(U,C∪D) for any B, where B is a subset of C, there is an equivalence relation, which represents a binary relation called indiscernibility relation and defined as

$$IND(B) = \{(x, y) \in U * U: a(x) = a(y) for\ all\ a\ in\ B\} \quad (1)$$

Definition 3.3: let S=(U,C∪D) , B⊆C and X⊆U. The lower approximation set $\underline{B}$(X) is a set of all instances in U that can be surely classified as elements that belongs to X when using B

$$\underline{B}(X) = \{x \in U | [x]_B \subseteq X\} \quad (2)$$

And the upper approximation set $\overline{B}$(X) is a set of elements that not surely classified as elements that belongs to X when using B, and can be defined as

$$\overline{B}(X) = \{x \in U | [x]_B \cap X \neq \emptyset\} \quad (3)$$

The boundary region of X expresses the uncertainty of knowledge, and defined as

$$BN_B(X) = \overline{B}(X) - \underline{B}(X) \quad (4)$$

The accuracy of approximations is defined as

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|} \quad (5)$$

$0 \leq \alpha_B(X) \leq 1$, if $\alpha_B(X) = 1$ then X is crisp otherwise X is rough.

Definition 3.4: let S=(U,C∪D), x∈U, the membership of x to X given B is defined as

$$\mu_x^B(x) = \frac{|X \cap B(x)|}{|B(x)|} \quad (6)$$

Where $\mu_x^B: U \rightarrow < 0,1 >$, and $|X|$ symbolizes the cardinality of X.

Definition 3.5: let S=(U,C∪D), the dependency between C and D, symbolized C⇒D, defined as

$$k = \gamma(C,D) = \frac{POS_C(D)}{|U|} \quad (7)$$

Where $0 \leq k \leq 1$ and $POS_C(D) = \cup_{X \in U/D} \underline{C}(X)$. If k=1 this means that D depends totally on C, otherwise D depends partially on C.

Definition 3.6: let S=(U,C∪D), and a∈ C , a is dispensable in C if
$\gamma(C,D) = \gamma(C - \{a\}, D)$ otherwise a is indispensable.
If all the attributes in C are indispensable, then C will be called independent.
Let $\acute{C}$ be a subset, where $\acute{C} \subseteq C$, $\acute{C}$ is reduct of C if

$$\gamma(C,D) = \gamma\left(\acute{C}, D\right) \quad (8)$$

Definition 3.7: let S=(U,C∪D), and a is an attribute, where a∈ C. the significance of attribute a is defined as

$$\sigma_{(C,D)}(a) = \frac{(\gamma(C,D) - \gamma(C - \{a\}, D))}{\gamma(C,D)} = 1 - \frac{\gamma(C - \{a\}, D)}{\gamma(C,D)} \quad (9)$$

Where $0 \leq \sigma(a) \leq 1$. Let B a subset of attributes, B⊆C, the significance of B is defined as

$$\sigma_{(C,D)}(B) = \frac{(\gamma(C,D) - \gamma(C - B, D))}{\gamma(C,D)} = 1 - \frac{\gamma(C - B, D)}{\gamma(C,D)} \quad (10)$$

If B is reduct of C, then $\sigma_{(C,D)}(B)=0$, Any subset of C is called the approximate reduct, and any set has a number, called error of reduct approximation and can be defined as

$$\varepsilon_{(C,D)}(B) = \frac{\gamma(C,D) - \gamma(B,D)}{\gamma(C,D)} = 1 - \frac{\gamma(B,D)}{\gamma(C,D)} \quad (11)$$

Symbolized as $\varepsilon(B)$, and demonstrates how the subset B approximates the set C (condition attributes). Clearly $\varepsilon(B) = 1 - \varepsilon(C - B)$ and also $\varepsilon(B) = 1 - \sigma(B)$.

4. THE PROPOSED APPROACH

The section consists of two parts, the first part presents the proposed certainty ratio based algorithm, and the second part will present the combination between rough set and the certainty ratio based algorithm, as will see below:

4.1. The proposed certainty ratio based algorithm

The proposed algorithm offers a new method for selecting the best values in each attribute depending on the certainty ratio of each value.

This proposal performs a type of selection on the attributes to select only the attribute values that has a certainty ratio equal or greater than a specified threshold. The certainty ratio represents the probability of the attribute value elements that belong to a decision class. The certainty ratio of each attribute value must be equal or exceed the threshold value in order to be selected. Also, this certainty ratio will be applied on the equivalence classes that generated by two or more attributes. It's worth to mention that this algorithm will be used in the beginning of rough set theory, but the resulting data from this algorithm will be used in writing the certain rules and uncertain rules only, which will contribute in decrease the complexity of some generated rules. The main steps of the proposed algorithm will be explained as follows:

A. Finding the certainty ratio for each attribute value

The first step in the algorithm is checking the certainty ratio of each attribute value and checking if it is equal or exceed the threshold. The threshold is specified by the user, and can be changed according to the user requirements. This step aims to select the best values from each attribute. Algorithm (1) shows how to find the certainty ratio of an attribute value or an equivalence class and checking if it is equal or exceed the threshold. This algorithm returns 1 or 0. If the certainty ratio of the value or equivalence class with any decision class equals or exceed the threshold, then the algorithm returns 1, otherwise, it returns 0.

It is worth to mention that all algorithms are written as a pseudo code.

Algorithm (1): Finding the certainty ratio and check with a threshold

Input: V[v1, v2, …., vn], Threshold // V instances of the attribute value, n is partitions number of the value elements according to its decision classes //
Output: CertaintyInfo // result of checking certainty ratio of the value with threshold//
Begin:
CertaintyInfo =0
T=0
un=number of elements in V
n= number of possible decision classes
for i=1 to n **do**
m=length of v_i
CertaintyR(v_i)=m/un //CertaintyR(v_i) is the certainty ratio of the value with each possible decision class //
if CertaintyR ≥ Threshold **then**
T=T+1
End if
End for
If T>0 **then**
CertaintyInfo=1
End if
Return CertaintyInfo
End

B. Check the attribute values

The next step in the algorithm is checking all values of an attribute. If the attribute has one or more values that have certainty ratio equal or exceed the threshold, then the attribute will pass to the next step, otherwise it will not be used in the next step.

Algorithm (2) shows how to check all values of attribute or all equivalence classes of a set of attributes. This algorithm also returns 1 or 0. If the attribute has values more than zero have output equal to one from algorithm (1), the algorithm returns 1, otherwise it returns 0.

Algorithm (2): Checking all the values of attributes.
Input: Attributes A, Threshold // A, or B which is a subset consists of one or more attributes //
Output: AttrCert // value of checking all attribute values //
Begin:
AttrCert = 0
M=0
SubN= number of subsets generated by attribute A
For i = 0 to SubN-1 **do**
Cert= CertaintyRatio($A_i, Threshold$) // calling algorithm (1) to check the certainty ratio of each attribute value or equivalence class //
If Cert=1 **then**
M=M+1
End if
End for
If M >0 **then**
AttrCert=1
End if
Return AttrCert
End

C. Generating candidates from the passed attributes

After checking every attribute, generating candidates of more than one attribute step is coming. This step aims to discover if the passed attribute can be passed again if it's combined with other attributes. The length of these candidates ranged between two and the number of passing attributes. In this step, the algorithm first combined two attributes, then check the certainty ratio of each equivalence class obtained from these attributes, if these attributes have one or more of equivalence class that has the certainty ratio equal or exceed the threshold, then these attributes will combined with a third attribute, and so on.

The algorithm continues in generating candidates until obtaining the candidate that contain the biggest number of passed attributes. This candidate will represent the final result of the algorithm with the attribute values that have the certainty ratio equal or exceed the threshold except the elements that was in the equivalence classes of the final candidate, which haven't the certainty ratio equal or exceed the threshold.

The main steps of the proposed certainty ratio based algorithm are shown in the algorithm (3).

Algorithm (3): Certainty ratio based algorithm

Input: attributes B , attribute D, Threshold // B a set of condition attributes , D is decision attribute //
Output: SelAttr //a set of selected condition attributes and their values after checking the certainty ratio//
Begin:
M=0
Initialze PassA[M]
N= number of attributes in B
For i=0 to N-1 **do**
CertI=0
CertI=CertaintyChecking(B[i], Threshold) // call algorithm (2) for checking the certainty ratio of all attribute classes //
If CertI=1 **then**
PassA[M]=B[i]
M=M+1
End if
End for
// start to generate candidates //
Initialize Attr[K] // set the candidate of attribute subset of size K//
Initialize AttrA[K] // set the excepted accepted subset of attributes of size k //
AttrA[1] // each attribute in PassA represent an excepted candidate of 1//
While AttrA[K] <>∅ **do**
Attr[k+1]= candidate generated from AttrA[k] // candidate generated from AttrA[k] join AttrA[k] and make intersection between the values of the two AttrA[k] //
CertI=0
CertI=CertaintyChecking(Attr[k+1])
If CertI=1 **then**
AttrA[K+1]= Attr[K+1] // the candidate is accepted //
End if
End while
SelAttr= maximum accepted AttrA[k]with the attributes accepted values according to its certainty ratio, except the elements of not accepted equivalence classes of AttrA[k]
Return SelAttr
End

4.2. The combination between rough set and certainty ratio based algorithm

The proposed certainty ratio based algorithm contributes in reducing the condition attributes, decreasing the number of the generated rules, and increasing the accuracy of the generated rules, and that happened due to using the certainty ratio as criterion for selecting the attributes values that has a good level of certainty or has a certainty exceed the specified threshold. The result of this algorithm will be used in writing rules step in rough set theory, because if the result is used when performing the indiscernibility relation and approximations, that maybe will lead to disappear some important rules

due to the operations that performed by indiscernibility relation, which ignore the objects that don't have a value in each available condition attribute. If the object has an attribute value less than the threshold, then the indiscernibility relation will ignore this object.

5. THE EXPERIMENTAL RESULTS

In this section, the experimental results of the proposed approach are presented. The comparison between the original size of the data and the size of data after combining the certainty ratio based algorithm with rough set theory is introduced, the number of rules and accuracy are also compared. Four different data sets are used to evaluate the performance of the proposed approach, includes dataset of the terrorist attacks in Iraq [11], weather in Iraq, and two medical datasets, one of diabetes disease and the other of heart disease [12].

In this research the threshold is specified by the value 65, which can give more suitable results than other values. If we use a threshold less than 65, the results of rough set after combining with the certainty ratio based algorithm will not be changed, and if we use a threshold more than 65, probably many of the generated rules will be disappearing.

The first comparison in this section is between the attribute size before and after combining rough set with certainty ratio based algorithm. The result of this comparison is given in table 1.

Table(1): the attributes size before and after combining rough set with certainty ratio based algorithm.

Name of dataset	Original attribute size	Certainty ratio based alg.
Terrorist attacks	4	4
Weather	11	10
Diabetes	7	6
Heart disease	12	10

From the table above, on can see that the proposed doesn't select all the available attributes in most of used datasets. The certainty ratio based algorithm selects the attributes that had at least one value, which its certainty ratio equal or exceed the specified threshold, therefore, even the attribute that is less effective, maybe will be selected if it is has one value where its certainty ratio equal or exceed the threshold.

The other comparison shows the number of the generated rules by rough set theory before and after combining with certainty ratio based algorithm, which given in table 2.

Table (2): the generated rules by rough set theory before and after combining with certainty ratio based algorithm.

Name of dataset	Rough set theory	RS+ Certainty ratio based alg.
Terrorist attacks	1460	1393
Weather	151	113
Diabetes	41	35
Heart disease	212	199

In additional to removing some redundant attributes, certainty ratio based algorithm removes the attribute values that have a low level of certainty. The certainty ratio based algorithm removed some attribute values that have a certainty ratio less than the specified threshold, and that lead to generate rules with different complexity, where these rules consists of the values that has the required level of certainty. So, when the complexity of the rules reduced, that will lead to generate a less number of rules, because many objects will be classified or predicted by the same rule by using only the attribute values that these objects are have.

The final comparison is between the accuracy of classification by using rough set before and after combining with the certainty ratio based algorithm, which given in table 3.

Table (3): the accuracy of classification.

Name of dataset	Rough set theory	RS+ Certainty ratio based alg.
Terrorist attacks	71.71%	71.71%
Weather	14.58%	35.41%
Diabetes	70%	90%
Heart disease	76.38%	80.55%

From above, one can see that when the certainty ratio based algorithm is used, the accuracy is increased and that because of removing some redundant attributes, which was decreasing the accuracy of the generated rules. Also, some attributes values are removed when certainty ratio based algorithm is used, which contributes in increasing the accuracy of the generated rules.

6. CONCLUSION

In order to decrease the complexity of the generated rules by rough set theory, this paper proposed a new algorithm called a certainty ratio based algorithm. This algorithm selects the effective values from each attribute or select only the attribute values that have high certainty level that exceed a specified threshold, and ignore the attributes values that has a low level of certainty or its certainty ratio can't exceed the specified threshold. If the attribute doesn't have at least one value can exceed the threshold, then the attribute will be ignored. The proposed algorithm is combined with rough set

theory, where certainty ratio based algorithm is applied at the beginning of the proposed approach and its results are used in the step of writing rules only of rough set (certain rules and uncertain rules). Four different data sets are used in the experiments. The results, analysis, and discussion show that the proposed certainty ratio based algorithm can effectively and ignore the attribute values that its certainty ratio cannot exceed the specified threshold and ignore the attributes that doesn't have at least one value can exceed the threshold. Also, the accuracy of the generated rules is increased when it combined with rough set theory and decreasing the number of the generated rules.

REFERENCES

[1] J. Komorowski, L. Polkowski and A. Skowron, "**Rough sets: A tutorial**", Springer, 1999.

[2] R. Jensen, "**Combining rough and fuzzy sets for feature selection**". PhD thesis, School of Informatics, University of Edinburgh, Scotland, 2005.

[3] Z. Pawlak, **''Rough Sets: Theoretical Aspects of Reasoning About Data''**, Kluwer Academic Publishers, Boston, MA,1991

[4] Z. Bonikowski, and U. Wybraniec-Skardowska, **''Vagueness and Roughness. Transactions on Rough SetsIX''**, Springer-Verlag, Heidelberg - Berlin, Germany, pp. 1-13, 2008.

[5] X. Hu, T. Y. Lin and J. Han, "**A New Rough Sets Model Based on Database Systems**", Fundamental Informaticae, IOS Press,Vol.1, issue 18, 2004.

[6] J. Liu, Q. Hu, and D. Yu, " **A weighted rough set based method developed for class imbalance learning**", Elsevier, Vol.178, issue 1, pp.1235-1256, 2007.

[7] S., Z. Elouedi, and P. Lingras, "**Belief Rough Set Classifier**", Springer Berlin Heidelberg, Vol. 5549, Issue 1, pp.257-261, 2009.

[8] H. Feng, et al., "**A Discernibility Degree and Rough Set Based Classification Rule Generation Algorithm (RGD)'',** International Multi Conference of Engineers and Computer Scientists (IMECS), Hong Kong, China , Vol.1, 12- 14 March 2014.

[9] P. Mahajan, R. Kandwal and R. Vijay," **Rough Set Approach in Machine Learning: A Review**", International Journal of Computer Applications, Vol. 56– No.10, October 2012.

[10] Z. Pawlak and A. Skowron, "**Rudiments of Rough Sets**", Elsevier, Vol. 177- No. 1, pp. 3 – 27, January 2007.

[11] www.iraqbodycount.org

[12] http://mldata.org/repository/data/viewslug/datasets-uci-heart-c/