International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011
ISSN: 1793-8201

Research and Application of Rough Lattice Construction Based on Compressed Matrix

Wei Kong, Shouning Qu and Yuhui Tian

Abstract—**This paper does some researches on rule mining according to the production of fluid industry, as its production process has the characteristics of complexity, strong correlation, nonlinear and uncertainty. Rough sets and concept lattice are two effective methods for data analysis and knowledge discovery in data mining. In this paper, according to CARCL(Construction Algorithm of Rough Concept Lattice), we combine the two methods' advantages and propose a new method for the construction of Rough Lattice based on compressed matrix. The new method can solve the redundancy of construction process and obtain corresponding rules .**

Index Terms—**Compressed matrix, Fluid industry, Rough lattice, Rule mining.**

I. Introduction

In the process of constructing Rough Lattice, the structure will be more complex as the number of nodes increases, meanwhile, produce massive redundant information. In order to better distinguish redundant information and useful information, this paper adopts matrix to construct Rough Lattice.

Rough Lattice theory integrates the concept lattice with the rough set theory [1]. Rough Lattice (RL) is the shortening of Rough Concept Lattice (RCL) and an effective tool for data mining. RL is a structure, which consists of nodes and line segments between two nodes. In RL, a node is called a concept, and the line is known as relationship. The concept lattice is composed of formal concepts and some orderly hierarchical relations among concepts. In concept lattice, formal concept is a node and made up of Intent and Extent. Intent is constructed by a set of attributes, while Extent is a set of objects, which have the same attributes. The relationship among these intent properties is "and", in other words, an extent object will be excluded if it does not meet even one intent property [2]. This reflects accuracy and determinacy of concept lattice. However, some possible and correct objects may be ignored in the process, this results in the concept lattice's limits on processing knowledge. The rough set theory is based on objects' equivalent relations decided by attributes. It makes equivalent devision of the object set and can further simplify attributes. The rough set is an effective method for dealing with fuzzy and uncertain knowledge [3]. The Rough Lattice theory that is put forward by this paper to overcome the shortcomings of concept lattice's excessive accuracy and rough set's incomplete and makes a combination of both. So it makes up for their respective deficiencies in data mining and extends the rough set's application fields.

Some scholars at home and abroad have studied how to combine the rough set and concept lattice. H. F. Yang and J. F. Zhang proposed the construction algorithm of Rough Lattice and mined classification rules on the constructed Rough Lattice [4]. S. P. Dai, T. He and X. M. Xie put forward the method of making concept approximation on concept lattice; it re-described the rough set's Upper and Lower Approximation [5]. W. P. Ding, Z. J. Guan and Z. G. Shi improved the concept lattice's approximation method, used the extended rough set model to achieve approximate combination of nodes in concept lattice [6]. L. Fu, G. J. Wang, Q. Wu, W. Zhou, et al. still studied the relationship and quality between rough set and concept lattice, but they just used Rough Lattice to deal with the concept lattice and even did not change the structure of concept lattice [7]-[11]. Yao. YY. A from Canada performed approximate calculations on the Extent and Intent of concept lattice, and carried out researches in three areas: the approximation of Extent, the approximation of Intent and the approximation of Extent and Intent [12]. These methods do not embody the structure and the construction process of Rough Lattice, information in the decision table that is constructed by the decision context has not described by the lattice structures. Even if the Rough Lattice has been built well, it still contained numerous nodes and mass information; this would increase traversal time of nodes and reduce the efficiency of data mining.

We put forward the CM_CARCL (Compressed Matrix Based Construction Algorithm of Rough Concept Lattice) algorithm based on the rough set and concept lattice theory. This algorithm absorbs construction method of data structure for CARCL and constructs a Compressed Matrix that can be used in CARCL algorithm. This Compressed Matrix can eliminate redundant paths of the Rough Lattice's structure effectively and improve the algorithm's accuracy. The cement flow production applies this algorithm to mine rules and has achieved good results. The experiment shows that information decision plays a more important role when the set has multiple attributes and a large collection of data, so

Manuscript received July 25, 2010. This work was supported by the National Natural Science Foundation of China (NSFC) (No. 69902005), the research plan of Shandong Province (No. 2008GG10001001) and the postgraduate innovative plan of Shandong Province (No. SDYY008).

Wei Kong is with College of Information Science and Engineering, University of Jinan, Jinan Shandong 250022, China (phone: 15169046905; e-mail: sfkw20082006@126.com).

Shouning Qu is with Information Network Center, University of Jinan, Jinan Shandong 250022, China (phone:13605315933; e-mail:qsn@ujn.edu.cn).

Yuhui Tian is with Industry Group of Shandong University, Jinan Shandong 250101, China.

the proposed algorithm is of vital practical significance in the application of Rough Lattice theory.

II. Elementary Definition in Rough Lattice

In view of the Rough Lattice's construction characteristics, when we construct extent, intent and relationship between nodes based on decision context, we need to use the following definitions:

Definition 1: $S = (U, A)$ is a sample decision table, U is a nonempty and finite set, which is consisted of individuals and denoted by $\{x_1, x_2, x_3\ldots x_n\}$. A is also a nonempty and finite set, which is consisted of attributes and denoted by $\{C\cup D\}$. C is a set of condition attributes, D is a set of decision attributes and $C\cap D=\phi$. For $\forall x \in U, \forall a \in A$, $a(x)$ represents the value of object x when its attribute is a.

Define the relation R: $R \subseteq U\times C^*D$, $\forall(x, a)\in U\times C^*D$, then $(x, a) \in R$, namely $a(x)$ is not null.

Base on the above, $Ds = (U, C^*D, R)$ is defined as decision context educed by decision table S.

Definition 2: Suppose Ds is decision context, for $\forall(X, Y)$, $X\subseteq U$, $Y\subseteq C^*D$, then we define two functions as follows:

$$f(X) = Y, t(Y) = X.$$

$f(X)$: common attributes are possessed by all the objects in X.

$t(Y)$: all objects set that contains attribute Y.

So, (X, Y) is defined as an object-attribute concept of decision context Ds.

If $X_1 \subseteq X_2$, $Y_1 \subseteq Y_2$, then(X_1, Y_1) is a sub-concept of (X_2, Y_2), expressed as $(X_1, Y_1) \leqslant (X_2, Y_2)$.

Definition 3: Suppose Ds is decision context, for $\forall x \in U$, $\forall Y\subseteq C^*D$, we defined M and N as follows:

$M = \{x \mid f(x) \cap Y \neq \phi\}$

Where we call M as Upper Approximation Extent, it represents some possible objects that meet the attribute Y.

$N = \{x \mid f(x) \cap Y = Y\}$

Correspondingly, N is Lower Approximation Extent; it indicates some objects set must meet the attribute Y.

Here, Y is called Intent, which describes the concept. The triple (M, N, Y) constructs any node of Rough Lattice. All the nodes construct the Rough Lattice.

Definition 4: Suppose $H_1 = (M_1, N_1, Y_1)$, $H_2 = (M_2, N_2, Y_2)$ are two different nodes of Rough Lattice. If $H_1 < H_2$, then $Y_1 \subset Y_2$, and we call H_1 is H_2's successor and H_2 is H_1's predecessor. If there is no node $H_3 = (M_3, N_3, Y_3)$ that makes $H_1 < H_3 < H_2$, then H_1 is H_2's direct successor and H_2 is H_1's direct predecessor, namely H_1 is H_2's child and H_2 is H_1's parent.

III. Construction of Rough Lattice Based on CM

The structure of Rough Lattice needs lots of nodes. In addition to the construction of every node, the relationship among nodes needs to be treated properly, otherwise it will result in confusion and paralysis of structure.

In the process of constructing Rough Lattice, we should construct all the nodes firstly, every node contains five parts: *M*, *N*, *Y*, *Parent*, *Children* [13]. The process of constructing nodes is the process of seeking for their corresponding Parent, Children.

The construction process of Compressed Matrix (CM, Compressed Matrix) is as follows:

Step 1. Construct the root node R and end node E. R is initialized as (U, ф, C∪D, ф, {E}); E is (ф, ф, ф, {R}, ф). The two nodes are inserted into the array nodes[]. At first, there are only R and E in Rough Lattice.

Step 2. Read each node from the database, a node is one record. Calculate M, N, Y of each node according to the function getM(), getN(), getY(). Then insert the constructed nodes into array nodes[].

Step 3. Construct the initial matrix bSrc[*i*][*j*] according to *Y*'s inclusion relation of each node. Only when the former node contains the latter one, the corresponding position in matrix is set to true, otherwise false.

Step 4. Initialize the outcome matrix bDst[i][j]. Set the value of each element in matrix as true.

Step 5. Implement logic subtraction, then write the acquired results in the destination matrix bDst[*i*][*k*]. bDst[*i*][*k*] = (bSrc[*i*][*k*] && (! bSrc[*j*][*k*])) && bDst[*i*][*k*]. The calculation is aimed at removing redundant edges in the lattice structure.

Step 6. At this time, every element in the outcome matrix bDst[i][j] whose value is true becomes the child node of current row node.

Based on the above idea, when we make use of the compressed matrix (CM, Compressed Matrix) to construct the Rough Lattice, we suppose the number of node is n, then the algorithm complexity is $O(n * n * n)$. If there are many records in database, the algorithm efficiency will reduce and the time complexity increase. So we can limit the nodes when constructing Rough Lattice:

Definition 6: $Ds = (U, C^*D, R)$ is defined as decision context, suppose (M, N, Y) is any node of Rough Lattice, if we call $p = |N| / |M|$ similarity, then $r = 1\text{-}p$ is called rough degree.

When $r = 0$, the Rough Lattice is changed into concept lattice. If $r = 1$, $|N| = 0$, the structure we constructed has no meaning, so $0 < r < 1$.

In the process of constructing Rough Lattice, we can set a value r and insert it into the Rough Lattice. Discard the node unless the value of new node is less than or equal to the set value r. After such disposal, the number of nodes will be greatly reduced and the robustness of Rough Lattice can be enhanced obviously. When constructing Rough Lattice, it is necessary to make an appropriate choice of nodes, especially in complex fluid industry environment.

The algorithm procedure of constructing Rough Lattice based on compressed matrix (CM, Compressed Matrix) shows in Fig. 1.

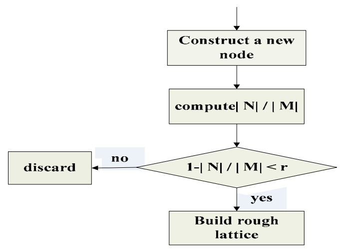

Fig. 1 Constructing process of Rough Lattice based on CM.

IV. Implementation of Algorithm

Take the cement production in fluid industry as an example, the construction procedure of Rough Lattice is as follows:

```
public BuildRL build ( BuildRL nodes[] )
{
int rs;
int i, j, k;
int iTotal = nodes.length;
// define the initial matrix bSrc[][]
boolean bSrc[][] = new boolean[iTotal][iTotal];
// define the outcome matrix bDst[][]
boolean bDst[][] = new boolean[iTotal][iTotal];
// Define an array nodes[] to store the nodes which have been constructed, and insert the head node R and the end node E into nodes[].
BuildRL nodes[] = new BuildRL[iTotal+2];
nodes[0] = R;
nodes[1] = E;
// initialize bSrc[][]
//Determine inclusion relationship between nodes according to rs. If these nodes exist inclusion relation, the appropriate location in bSrc[][] is set to true, otherwise is false.
rs = nodes[i].compare(nodes[j]);
// All the diagonal lines is set to false.
bSrc[i][i] = false;
// initialize bDst[][]
bDst[i][i] = true;
// Clear redundant edges in bSrc[][] and implement logic subtraction: bSrc[i][k]-bSrc[j][k], then insert the results into outcome matrix. Calculation formula is as follows:
bDst[i][k]= (bSrc[i][k]&&(!bSrc[j][k]))&&bDst[i][k];
// Dispose outcome matrix bDst[][]: link the node in the row whose value is true to the row that represents a node as its child node. nodes[i].Children.add(nodes[j]);
}
build( nodes );
public void searchild(BuildRL temp)
{
for (int j = 0; j < temp.Children.size(); j++)
{
p=temp; //The p is the location node that is used to output rules.
e = temp.Children.get(j);  // The e has child node of temp.
if (e != end)
{
if (e.M == temp.M)
{
//These rules can not be outputted until temp'M is equal to e'M.
temp.searchild (e);
}
}
}
System.out.println("rules are:" + p.Y);
}
```

V. Application of The Algorithm

We apply the construction of Rough Lattice based on compressed matrix to the process of cement production for rule mining. The cement production flow is affected by kinds of properties, such as the properties in Table I: C_1 to C_{10}. These attributes play a different role in different procedure. According to property changes, we can determine the appropriate rules in order to adjust the cement production. Owing to the continuity and complexity of production process, there exists a large number of incomplete, noisy and inconsistent data. If we use these data directly to mine data, it will give rise to low efficiency of data mining, and even lead to a wrong conclusion [14].

High quality decision must rely on high quality data. So inspecting abnormal data, adjusting data and reducing analyzed data as soon as possible will achieve high returns in the process of data mining. Therefore, data pretreatment is an important step in knowledge discovery. Data preprocessing is divided into two parts: data integration and data discrimination [15].

The main work of data integration is to integrate the scattered data and ensure its consistency. It includes data acquisition, sequence matching, edge blurring and noise processing. Since most of the existing machine learning methods have the input variables which possess discrete attributes, therefore, how to discretize the quantitative and continuous data in learning repository to the Boolean property which is easier for PC becomes an important problem for machine learning. This paper studies the influence of parameters' change in fluid industry on other important parameters. We discrete data using mean value technique. The mean value represents a common data, so the data can be discretized into two values, 1 or 0, which is corresponding to the two states of parameters: rise or fall. The concrete operations are as following: the data higher than mean value is discretized into 1, and lower than the mean value is discretized into 0. After data processing, the results are illustrated in Table I .

TABLE I: Decision ContexT of Decomposing Furnace Data in Data Mining

Object	C	D
	C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8	C_9 C_{10}
1	1 0 0 0 0 1 0 0	0 1
2	1 0 0 0 1 0 0 0	1 0
3	0 1 0 0 1 0 0 1	1 0
4	1 0 0 0 1 0 0 0	0 1
5	0 1 0 0 0 0 0 1	1 0
…	… … … … …	…
500	0 1 0 0 1 0 1 0	1 0

Adding: C_1: content of unbaked material C_2: content of CO from C_1 C_3: content of spouting coal in decomposing furnace C_4: temperature of tri-wind C_5: out temperature of decomposing furnace C_6: kiln speed C_7: kiln main motor current C_8: high-temperature fan speed C_9: coal production from wicket C_{10}: a comb speed

In this paper, C_1 to C_8 are the condition property, and C_9 and C_{10} which are re-marked as *da* and *db* are the decision attribute.

The rules have been mined from the constructed Rough Lattice are:

Rules: C_1 C_4 C_5 C_6 C_7 *da db*
Rules: C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 *da*
Rules: C_2 C_3 C_4 C_5 C_6 C_7 C_8 *da db*
Rules: C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 *db*
Rules: C_1 C_2 C_3 C_4 C_5 C_7 C_8 *da db*
Rules: C_1 C_2 C_3 C_5 C_6 C_7 C_8 da db
Rules: C_1 C_2 C_3 C_4 C_6 C_7 C_8 *da db*
For example: Rules: C_1 C_4 C_5 C_6 C_7 *da db*

It can be explained as: If C_1, C_4, C_5, C_6, C_7 increased, C_2, C_3, C_8 kept unchanged or fell, this led to C_9, C_{10} increased.

The rules have mined from the improved constructed Rough Lattice are:

Rules: C_1 C_4 C_5 C_6 C_7 *da db*
Rules: C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 *da*
Rules: C_2 C_3 C_4 C_5 C_6 C_7 C_8 *da db*
Rules: C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 *db*
Rules: C_1 C_2 C_3 C_4 C_5 C_7 C_8 *da db*
Rules: C_1 C_2 C_3 C_5 C_6 C_7 C_8 *da db*

Compared with the former rules which are not improved, the latter have reduced several rules: C_1 C_2 C_3 C_4 C_6 C_7 C_8 *da db*. It shows that the node is discarded because its M and N does not meet the given r value. At this point the program improves calculation speed, reduces rules and lessens interference of parameters. This algorithm used in controlling parameters of industrial processes will seize the important links, adjust the variation amplitude of all parameters in production process and improve the production efficiently.

VI. Conclusion

Because the Rough Lattice combines advantages of concept lattice and rough set, so it is widely used in Information Retrieval, Data Mining, Software Engineering and other fields [16]. When the rough degree is confirmed, the number of nodes can greatly reduced, the efficiency of constructing Rough Lattice is greatly improved and the algorithm in knowledge extraction from Rough Lattice advances veracity and efficiency. It is necessary to simplify information, especially in complex fluid industry environment. Setting the r is quite necessary. But in the process of constructing CM_CARCL, although the accuracy is assured and the space complexity is reduced by Compressed Matrix, time complexity is not significantly reduced. Then there is no limitation in setting r. So we will summarize experience in the late study, find a reasonable way to improve the algorithm's execution speed and effectiveness continuously.

Acknowledgment

My grateful acknowledgment goes first and foremost to my supervisor Mr. Shouning Qu who gave me considerable help by means of suggestion, comments and criticism. His encouragement and unwavering support has sustained me through frustration and depression. Without his consistent instruction, this thesis could not have reached its present form.

Second, I would like to express my heartfelt gratitude to my beloved family for their loving considerations and great confidence in me all through these years.

Last I owe my sincere gratitude to my classmates who gave me help in listening to me and helping me work out my problems during the difficult course of the thesis.

References

[1] L. Wei, J. J. Qi and W. X. Zhang, "Relation Research of Concept Lattice and Rough Sets", Computer Science, Vol. 3, No. 33, 2006, pp. 18-21.
[2] J. H. Wang and J. Y. Liang, "Concept Lattice and Rough Sets", Journal of Shanxi University, Vol. 4, No. 26, 2003, pp. 307-310.
[3] Z. C. Chen, K. Y. Qin. Structure of Remaining lattice in Rough Set Algebra. Fuzzy Systems and Mathematics, Vol. 4, No. 22, 2008, pp. 149-153.
[4] H. F. Yang and J. F. Zhang, "Rough Lattice and Constructing Algorithm", Computer Engineer and Application, Vol. 24, No. 43, 2007, pp. 172-175.
[5] S. P. Dai, T. He and X. M. Xie, "Data Analysis and Method Research of Concept Lattice and Rough Sets", Computer Engineer and Design, Vol. 6, No. 29, 2008, pp. 1423-1425.
[6] W. P. Ding, Z. J. Guan and Z. G. Shi, "Rule Mining Research of Similar Concept Lattice Based on Extent Rough Sets Model", Journal of Nanjing Youdian University, Vol. 2, No. 29, 2009, pp. 10-15.
[7] Z. Z. Chen and K. Y. Qin, "Structure of Remaining lattice in Rough Set Algebra", Fuzzy Systems and Mathematics, Vol. 4, No. 22, 2008, pp. 149-153.
[8] L. Fu and G. J. Wang, "Research on Rough Nature of Concept Lattice", Computer and Digital Engineering, Vol. 10, No. 36, 2008, pp. 20-24.
[9] Q. Wu, W. Zhou, H. Q. Chen and Z. T. Liu, "Attribute Reduction and Algorithm of Concept Lattice Based on Rough Set Theory", Computer Science, Vol. 6, No. 33, 2006, pp. 179-181.
[10] Q. Gu, "Rough Set Approximation Based on Concept lattice", Computer Engineer and Design, Vol. 2, No. 29, 2008, pp. 437-440.
[11] H. Wang and W. X. Zhang, "Comparative Study of Formal Concept Analysis and Rough Set", Computer Engineer, Vol. 8, No. 32, 2006, pp. 42-44.
[12] Yao. YY. A, "Comparative Study of Formal Concept Analysis and Rough Set Theory in Data Analysis", Manuscript, 2004.
[13] H. F. Yang, "Rough Concept Lattice and the Method of Classification Rules Mining Algorithm", Master Thesis in China: Taiyuan University of Technology.
[14] T. R. Ken, "Rough Concept Analysis". Ziarko, W. P. (1994), "Rough Sets, and Fuzzy Sets and Knowledge Discovery(RSKD'94)", London: Springer Verlag, 1994.
[15] G. Q. Cui, "Data Analysis and Optimization of Fluid Industrial Parameters Based on Data Engineering", Master Thesis in China: University of Jinan.
[16] S. N. Qu, and G. Q. Cui, "Research on Flow Object Knowledge Extration Method", International Conference on Intelligent Networks and Intelligent Systems(ICINIS 2008). China: Wuhan, 2008.

Wei Kong is with College of Information Science and Engineering, University of Jinan. She is postgraduate in University of Jinan and is researching rule mining now.

Shouning Qu, male, is with Information Network Center, University of Jinan. He is mainly engaged in researching the field of information integration and mobile computing.

Yuhui Tian, female, is with industry Group of Shandong University. She is currently engaged in researching the field of manufacturing industry information service.