Attribute Reduction Based on Granular Computing*

Jun Hu[1,2], GuoYin Wang[1,2], QingHua Zhang[2,3], and XianQuan Liu[2,3]

[1] School of Electronic Engineering,
XiDian University,
Xi'an, Shaanxi, 710071, P.R. China
[2] Institute of Computer Science and Technology,
Chongqing University of Posts and Telecommunications,
Chongqing, 400065, P.R. China
[3] School of Information Science and Technology,
Southwest Jiaotong University,
Chengdu, Sichuan, 610031, P.R. China
{hujun, wanggy, zhangqh, liuxq}@cqupt.edu.cn

Abstract. Attribute reduction is a very important issue in data mining and machine learning. Granular computing is a new kind of soft computing theory. A novel method for encoding granules using bitmap technique is proposed in this paper. A new attribute reduction method based on granular computing is also developed with this encoding method. It is proved to be efficient.

Keywords: Granular computing, rough set, attribute reduction, bitmap technique.

1 Introduction

Attribute reduction is a very important issue in data mining and machine learning. It can reduce redundant attributes, simplify the structure of an information system, speed up the following process of rule induction, reduce the cost of instance classification, and even improve the performance of the generated rule systems. There are many methods for attribute reduction in rough set theory. Rough set theory was developed by Pawlak in 1982[1]. For its ability to process inconsistent and imperfect information, it has been applied successfully in many fields in the past years. A lot of attribute reduction algorithms based on rough set theory have been developed in the last decades, which can be classified into two categories: (1)attribute reduction from the view of algebra[2][3]; (2)attribute reduction from the view of information[4][5]. The challenging issues of

* This paper is supported by National Natural Science Foundation of P. R. China (No.60373111, No.60573068), Program for New Century Excellent Talents in University (NCET), Science & Technology Research Program of Chongqing Education Commission(No.040505), and Natural Science Foundation of Chongqing University of Posts and Telecommunications(A2006-56).

S. Greco et al. (Eds.): RSCTC 2006, LNAI 4259, pp. 458–466, 2006.

these methods are multiple computation for equivalent classes and huge number of objects. The major objective of this paper is to design an encoding method for granules based on bitmap, and develop an efficient method for attribute reduction.

The rest of this paper is organized as follows: basic concepts of granular computing are introduced in section 2. In section 3, an encoding method for granules using bitmap technique is developed. A new method for attribute reduction based on granular computing is proposed in section 4. Then section 5 try to prove the efficiency of this method by a simple simulation. At last this paper is concluded in section 6.

2 Basic Concepts of Granular Computing

In many cases, it is impossible or unnecessary to distinguish individual objects or elements in a universe, which force us to think of a subset of the universe as one unit, instead of many individuals. In other words, one has to consider groups, classes, or clusters of elements. They are referred to as granules. The concept of information granule was first introduced by Zadeh in 1979[6]. Information granules arise in the process of abstraction of data and derivation of knowledge from information[7].

Granular Computing (GrC) is an emerging conceptual and computing paradigm of information processing. As the name stipulates, GrC concerns processing with information granules. Ever since the introduction of the term of GrC by T.Y Lin in 1997[8], a rapid development of this topic has been observed. Many models and methods for granular computing have been proposed and studied[9][10][11].

Although there does not exist a general agreement about what is GrC, nor is there a unified model, the basic notions and principles are the same. Granular computing focuses on problem solving based on the commonsense concepts of granule, granularity, granulated view, and hierarchy. The objective of granular computing is to translate problems into a hierarchy structure, and search the solution by order relations. This method is also consistent with human problem solving experiences. It is a method simulating human problem solving.

3 Encoding Granules with Bitmap Technique

The construction of granular computing and computation with granules are the two basic issues of GrC. The former deals with the formation, representation, and interpretation of granules, while the later deals with the utilization of granules in problem solving[12]. The bitmap technique was proposed in the 1960's[13] and has been used by a variety of products since then. Recently, many attempts have been paid to applying bitmap techniques in knowledge discovery algorithms[14][15], for bitmaps improve the performance and reduce the storage requirement. In this section, we will introduce a method for encoding granules using bitmap.

3.1 Partition Matrix

It is convenient to describe a finite set of objects called the universe by a finite set of attributes in an information table[16]. Formally, an information table can be expressed as: $S(U, At, L, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$, where U is a finite nonempty set of objects, At is a finite nonempty set of attributes, L is a language defined using attributes in At, V_a is a nonempty set of values for $a \in At$, $I_a : U \rightarrow V_a$ is an information function. Each information function I_a is a total function that maps an object of U to exactly one value in V_a[17]. An information table could be encoded using bitmap, called encoding information table.

Table 1 is an example of information table.

Table 1. An information table

Object	Height	Hair	Eyes	Class
O_1	short	blond	blue	+
O_2	short	blond	brown	-
O_3	tall	red	blue	+
O_4	tall	dark	blue	-
O_5	tall	dark	blue	-
O_6	tall	blond	blue	+
O_7	tall	dark	brown	-
O_8	short	blond	brown	-

The encoding rule is as follows:

1. For every attribute $a \in At$, the code length of a is equal to the cardinality of V_a;

2. Every bit of the code denotes a value in V_a;

3. For every attribute value, its code can be represented by a $\mid V_a \mid$-length bitmap, in which the corresponding bit is set to be 1, other bits 0.

For example, the cardinality of *Height* is 2, so the length of its code is two. Let the first bit denote *short* and the second bit *tall.* The attribute value *tall* will be encoded as 01, and *short* as 10. According to this rule, the information table shown in Table 1 could be encoded like Table 2.

Each subset A of At determines an equivalent relation on U, in which two objects are equivalent iff they have exact the same values under A[18]. An equivalence relation divides a universal set into a family of pair-wise disjoint subsets, called the partition of the universe. Here we use a matrix to represent a partition induced by an attribute. For an attribute $a \in At$ in an information system S, the partition matrix can be defined as $P(a) = \{P_a(i,j)\}_{n \times n}, 1 \leq i, j \leq n = \mid U \mid$, where

$$P_a(i,j) = \begin{cases} 1, \ I_a(i) = I_a(j) \\ 0, \ else \end{cases} \tag{1}$$

To generate the partition matrix on an attribute, the traditional way, according to above definition, is to compare the attribute values and $P_a(i,j)$ is set to

Table 2. Encoded information table

Object	Height	Hair	Eyes	Class
O_1	10	100	10	10
O_2	10	100	01	01
O_3	01	010	10	10
O_4	01	001	10	01
O_5	01	001	10	01
O_6	01	100	10	10
O_7	01	001	01	01
O_8	10	100	01	01

be 1 if the object i and j have the same value on attribute a, otherwise $P_a(i,j)$ is set to be 0. Here we could have another way using bitmap. In terms of the definition of the encoded information table, if two objects have the same value on an attribute, then they have the same code value on this attribute. To judge whether two objects have the same code value, the logic operation AND can be applied, i.e. $P_a(i,j)$ is set to be 1 for the result of non-zero, otherwise is set to be 0. Because the partition is symmetrical, it can be simplified to reduce the storage. For example, Table 3 is the partition matrix on *Height*, and Table 4 is the partition matrix on *Eyes*.

Table 3. Partition matrix on *Height*

P_{Height}	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
O_1	1							
O_2	1	1						
O_3	0	0	1					
O_4	0	0	1	1				
O_5	0	0	1	1	1			
O_6	0	0	1	1	1	1		
O_7	0	0	1	1	1	1	1	
O_8	1	1	0	0	0	0	0	1

Table 4. Partition matrix on *Eyes*

P_{Eyes}	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
O_1	1							
O_2	0	1						
O_3	1	0	1					
O_4	1	0	1	1				
O_5	1	0	1	1	1			
O_6	1	0	1	1	1	1		
O_7	0	1	0	0	0	0	1	
O_8	0	1	0	0	0	0	1	1

Having the partition matrix on each attribute, the partition matrix on a subset of At can be further computed. For a subset A of At, the partition matrix $P(A) = \{P_A(i,j)\}_{n\times n}, 1 \leq i,j \leq n =\mid U \mid$, can be computed using the following formula:

$$P_A(i,j) = P_{a_1}(i,j)\, AND P_{a_2}(i,j)\, AND ... AND\, P_{a_m}(i,j),\ a_1, a_2 ... a_m \in A \quad (2)$$

For instance, we can get the partition on $\{$*Height, Eyes*$\}$ based on the above two partition matrixes. It is shown in Table 5.

For convenience, we complement the partition matrix by symmetry. A partition matrix represents the equivalence relation holding between all the objects. In each line or column of a partition matrix, the subset consists of all objects which

Table 5. Partition matrix on {*Height, Eyes*}

$P_{\{Height,Eyes\}}$	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
O_1	1	0	0	0	0	0	0	0
O_2	0	1	0	0	0	0	0	1
O_3	0	0	1	1	1	1	0	0
O_4	0	0	1	1	1	1	0	0
O_5	0	0	1	1	1	1	0	0
O_6	0	0	1	1	1	1	0	0
O_7	0	0	0	0	0	0	1	0
O_8	0	1	0	0	0	0	0	1

are equivalent to the object denoted by the line. In other words, every line or column represents an equivalence class. Using partition matrix, we can easily get all equivalent classes, for example, $U/\{Height, Eyes\} = \{\{O_1\}, \{O_2, O_8\}, \{O_3, O_4, O_5, O_6\}, \{O_7\}\}$. Moreover, the computing process is incremental, that is, we can get the partition on {*Height, Eyes*} step by step.

3.2 Computation with Encoded Granules

In the partition model, a granule is a subset of the universe. The objects in a granule are gathered together by indiscernibility. On the other hand, a formula put the objects satisfying the formula in a granule. Therefore, we have a formal description of a granule. A definable granule in an information table is a pair $(\phi, m(\phi))$, where $\phi \in L$, and $m(\phi)$ is the set of all objects having the property expressed by the formula ϕ. In other words, ϕ can be viewed as the description of the set of objects $m(\phi)$. For $\phi, \psi \in L$, the following properties hold[16]:

(1) $m(\neg\phi) = \neg m(\phi)$
(2) $m(\phi \wedge \psi) = m(\phi) \cap m(\psi)$
(3) $m(\phi \vee \psi) = m(\phi) \cup m(\psi)$

Suppose the cardinalities of $m(\phi), m(\psi)$ and U are p, q and n respectively. The time complexities for calculating $m(\phi), m(\phi \wedge \psi)$ and $m(\phi \vee \psi)$ are $O(pn), O(pq)$ and $O(pq)$ respectively using traditional set operation. In the following paragraph, an encoding method for granules will be introduced. The time complexities of the above three computations will be reduced obviously with it.

Suppose the number of objects in the universe is n, then the length of code is n, and every bit denotes an object in the universe. Given a granule, if an object of the universe belongs to the granule, then the corresponding bit is set to be 1, otherwise, 0. A granule encoded by this rule is called as an encoded granule. According to the definition, the empty set is encoded by $\underbrace{00...0}_{n}$, and the universe $\underbrace{11...1}_{n}$, labeled as G_ϕ and G_U respectively.

Using encoded granules, the set operation of granules can be translated into logic operation (AND, OR, NOT, XOR) on codes of encoded granules.

Let $(\phi, m(\phi))$ and $(\psi, m(\psi))$ be two granules, and their codes is $a_1a_2...a_n$, $b_1b_2...b_n$, where n is the cardinality of the universe. The complement, intersection and union are defined as:

(1) $m(\neg\phi) = NOT\ a_1a_2...a_n$
(2) $m(\phi \wedge \psi) = a_1a_2...a_n\ AND\ b_1b_2...b_n$
(3) $m(\phi \vee \psi) = a_1a_2...a_n\ OR\ b_1b_2...b_n$

Based on the above analysis, if we use set operation, the time complexity of complement, intersection and union are $O(np), O(pq)$ and $O(pq)$, while the time complexity is $O(p)$,$O(p+q)$ and $O(p+q)$ respectively using logic operation.

We often need to determine whether a granule is included in another granule. If we check every object one by one, the complexity is $O(pq)$. Here we can get a simplified method with encoded granules, and its complexity is $O(p+q)$. If $((\phi, m(\phi))\ AND\ (\psi, m(\psi)))\ XOR\ (\phi, m(\phi)) = 0$, $(\phi, m(\phi))$ is included in $(\psi, m(\psi))$. For example, $\{O_2, O_4\}$ and $\{O_2, O_4, O_5\}$ are encoded as 01010000 and 01011000 respectively. Because (01010000 *AND* 01011000) *XOR* 01010000 = 0, we can conclude that $\{O_2, O_4\}$ is included in $\{O_2, O_4, O_5\}$, while needn't to check every object one by one. So, this method is more efficient than the traditional way.

Each equivalent class is a subset of the universe. It is a granule also. Therefore, every equivalent class can be encoded using the method developed in the last section. For example, {10000000, 01000001, 00111100, 00000010} is the code of $U/\{Height, Eyes\}$. Comparing these codes with Table 5, we can find that each line or column of a partition matrix is a code of an equivalent class.

In conclusion, the partition matrix not only describes the relationship between objects, but also gives the codes of equivalent classes. In this way, the challenging issues of attribute reduction can be solved by partition matrix and encoded granules.

4 Attribute Reduction Based on Granular Computing

Attribute reduction aims to find minimal subsets of attributes, each of which has the same discrimination power as the entire attributes. A minimal subset of attributes is called a reduction if it cannot be further reduced without affecting the essential information. Owning to the limitation of space, the basic concepts about attribute reduction is omitted here, but one can consult them in [19].

Attribute reductions based on algebra and information views were discussed in [18]. Although their definitions are different, both of them need to compute equivalent classes, so we can use the methods developed in the last section. Moreover, the set operation can be replaced by logic operation, which could improve the performance.

Let S be an information system. Using partition matrix we can get the codes of equivalent classes on condition attributes and decision attributes. Suppose they are $\{C_1, C_2, ..., C_i\}$ and $\{D_1, D_2, ..., D_j\}$, we can develop the following algorithm to compute the positive region of C with respect to D.

Algorithm 1 (Computing Positive region of C with respect to D)
Input: $IND(C) = \{C_1, C_2, ..., C_i\}$ and $IND(D) = \{D_1, D_2, ..., D_j\}$
Output: The positive region of C with respect to D, $POS_C(D)$
Step 1: Let $POS_C(D) = G_\phi$
Step 2: If $IND(C) \neq \phi$, then select an element C_m from $IND(C)$, let $IND(C) = IND(C) - \{C_m\}$, $T = IND(D)$. Otherwise go to Step 5
Step 3: If $T \neq \phi$, then select an element D_n from $IND(D)$, let $T = T - \{D_n\}$, $t = C_m\ AND\ D_n$
Step 4: If $t = 0$, then go to Step 3. Otherwise if $t\ XOR\ C_m = 0$, then let $POS_C(D) = POS_C(D)\ OR\ C_m$. Go to Step 2
Step 5: End

According to the definition of attribute reduction in the algebra view, here we can develop a new attribute reduction algorithm.

Algorithm 2 (Attribute Reduction Based on Granular Computing, *ARBGrC*)
Input: An information system S
Output: A reduction of condition attribute C, *RED(C)*
Step 1: Let $RED(C) = \phi$, $A = C$, compute the significance of each attribute $a \in A$, and sort the set of attributes based on significance
Step 2: Compute $POS_C(D)$ with Algorithm 1
Step 3: Compute $POS_{RED(C)}(D)$ with Algorithm 1
Step 4: If $(POS_{RED(C)}(D)\ XOR\ POS_C(D))$=0, then let $A = RED(C)$, go to Step 6
Step 5: Select an attribute a from A with the highest significant value, $RED(C) = RED(C) \cup \{a\}$, $A = A - \{a\}$, go to Step 3
Step 6: If $A = \phi$, then go to Step 8. Otherwise, select an attribute a from A, $A = A - \{a\}$
Step 7: Compute $POS_{RED(C)-\{a\}}(D)$ with Algorithm 1, if $POS_C(D)\ XOR\ POS_{RED(C)-\{a\}}(D) = 0$, then let $RED(C) = RED(C) - \{a\}$. go to Step 6
Step 8: End

The time complexity of $ARBGrC$ is $O(mn^2)$, where n is the number of objects and m is the number of attributes.

5 Simulation Result

In order to test the efficiency of $ARBGrC$ algorithm, we compare it with other two algorithms. One is the algorithm introduced in [19] whose time complexity is $O(m^2n^2)$, the other is the algorithm developed in [20] whose time complexity is $O(m^2nlogn)$. These two algorithms are labeled as Algorithm 3 and Algorithm 4 respectively in Table 6. We implement all algorithms using Visual C++6.0. Some classical data sets from UCI used by many other researchers are used in our experiment. To make the test result more validity, every algorithm on each data set was tested 100 times, and the time consuming is their average. For the reason of simplification, we might suppose NI is the number of instances, NOA is the number of original attributes, NAR is the number of attributes after reduction and TC is the time consuming. The experiment results are shown in Table 6.

Table 6. Test result for different algorithms

Data set	NI	NOA	Algorithm 3		Algorithm 4		ARBGrC	
			NAR	TC(s)	NAR	TC(s)	NAR	TC(s)
Zoo	101	16	5	0.0745323	5	0.0236871	5	0.0199314
Glass	214	9	8	0.1363940	8	0.0436963	8	0.0439632
Wine	178	13	2	0.1882960	2	0.1238230	2	0.0251350
Bupa	345	6	3	0.3069120	3	0.1986720	3	0.0435812
Letter-recognition	5000	16	9	202.7750	10	138.4930	9	58.9536

From the experiment, we can find that the results are almost the same in NAR, while different in TC. More concretely, these three algorithms have the same NAR, except that Algorithm 4 got 10 attributes on Letter-recognition, while the other two algorithms got 9 attributes. Moreover, Algorithm 4 has higher speed than Algorithm 3, and *ARBGrC* consumes less time than Algorithm 4 except on Glass. From the view of time complexity, Algorithm 4 and *ARBGrC* are better than Algorithm 3, but it is difficult to say which one is better than the other of these two algorithms. Theoretically, Algorithm 4 is better than *ARBGrC* when $n/\log_2 n$ is greater than m, while *ARBGrC* is better than Algorithm 4 when m is greater than $n/\log_2 n$. However, from the view of computing method, bitmap technique is applied in *ARBGrC*, which is machine oriented, and makes the computing positive region process incremental, avoiding computing repeatedly. For this reason, *ARBGrC* is more effective than the other two algorithms in time consuming.

6 Conclusion

As a new general computation theory, granular computing has appeared in many related fields. Granule is everywhere, such as classes, clusters, subsets, groups and intervals. Computing with granules can improve the performance, and reduce the time complexity, etc. An encoding method for granules using bitmap is proposed in this paper. Based on it, a new method of attribute reduction is developed. Compared with former methods, this method translates the set operation into logic operation, which improves the performance. This method can be also used in other fields, such as distributed data processing, and this will be our future work.

References

1. Pawlak,Z.: Rough set. *International Journal of Computer and Information Sciences.* 11(1982)341-356
2. Hu,X.H.: Mining Knowledge Rules from Database: A Rough Set Approach. Proceedings of the Twelfth International Conference on Data Engineering. (1996)96-105

3. Chang,L.Y., Wang,G.Y., and Wu,Y.: An Approach for Attribute Reduction and Rule Generation Based on Rough Set Theory. *Chinese Journal of Software*. 10(1999)1206-1211
4. Wang,G.Y., Yu,H., and Yang,D.C.: Decision Table Reduction Based on Conditional Information Entropy. *Chinese Journal of Computers*. 25(2002)759-766
5. Wang,G.Y., Yu,H., Yang,D.C., and Wu,Z.F.: Knowledge Reduction Based on Rough Set and Information Entropy. The 5th World Multi-Conference on Systemics and Informatics. (2001)555-560
6. Zadeh,L.A.: Fuzzy sets and information granularity. In Advances in Fuzzy Set Theory and Applications, Gupta, N., Ragade, R., Yager, R. (Eds.). North-Holland, Amsterdam. (1979)3-18
7. Pedrycz,W.: *Granular computing: An Introduction*. Kluwer Academic Publishers(2001)
8. Lin,T.Y.: Granular computing, announcement of the BISC Special Interest Group on Granular Computing(1997)
9. Zadeh,L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*. 90(1997)111-127
10. Yao,Y.Y.: A partition model of granular computing. LNCS Transactions on Rough Sets. (2004)232-253
11. Ma,J.M.: A covering model of granular computing. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou. (2005)1625-1630
12. Yao,Y.Y.: Granular computing: basic issues and possible solutions. Proceedings of the 5th Joint Conference on Information Sciences. (2000)186-189
13. Bertino,E., etc.: *Indexing techniques for advanced database system*. Kluwer Academic Publisher(1997)
14. Lin,T.Y.: Data mining and machine oriented modeling: a granular computing approach. *Journal of Applied Intelligence*. 10(2000) 113-124
15. Louie,E., Lin,T.Y.: Finding association rules using fast bit computation: machine-oriented modeling. Proceedings of the 12th International Symposium on Foundations of Intelligent Systems. (2000)486-494
16. Pawlak,Z.: *Rough Sets, Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht(1991)
17. Yao,Y.Y.: Modeling data mining with granular computing. Proceedings of the 25th Annual International Computer Software and Applications Conference. (2001)638-643
18. Wang,G.Y., Zhao,J., An,J.J., Wu,Y.: Theoretical Study on Attribute Reduction of Rough Set Theory: in Algebra View and Information View. Third International Conference on Cognitive Informatics. (2004) 148-155
19. Wang,G.Y.: *Rough Set Theory and Knowledge Acquisition*. Xi'an: Xi'an Jiaotong University Press(2001)
20. Liu,S.H., Sheng,Q.J., Wu,B., Shi,Z.Z., Hu,F.: Research on Efficient Algorithms for Rough Set Methods. *Chinese Journal of Computers*. 26(2003)524-529