

Quantization of Rough Set Based Attribute Reduction

Bing Li*, Peng Tang, Tommy W. S. Chow

Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong.
Email: *lib675@163.com.

Received 2012.

ABSTRACT

We demonstrate rough set based attribute reduction is a sub-problem of propositional satisfiability problem. Since satisfiability problem is classical and sophisticated, it is a smart idea to find solutions of attribute reduction by methods of satisfiability. By extension rule, a method of satisfiability, the distribution of solutions with different numbers of attributes is obtained without finding all attribute reduction. The relation between attribute reduction and missing is also analyzed from computational cost and amount of solutions.

Keywords: Attribute Reduction; Extension rule; Distribution of Solutions; Quantization.

1. Introduction

The size of dataset has been increasing dramatically, so it has been an important issue to reduce huge objects and large dimensionality. Attribute reduction finds a subset of attributes to reduce dimensionality. Reducing attributes can save cost of computational time and memory. It is also useful to improve classification accuracy as a result of removing redundant and irrelevant features [1,2].

A hypothesis done in classical attribute reduction theory is that all available objects are completely described by available attributes. This hypothesis contrasts with the empirical situation where the information concerning attributes is only partial.

Several methods have been used for handling missing values. One is complete case analysis ignoring the samples with missing values [3]. This method can only be used when the proportion of missing values is not large; moreover, much effective information is directly discarded. The second approach, called imputation method, imputes the values of missing data by statistical methods or machine learning methods [4,5]. This kind of approach leads to additional bias in multivariate analysis [6]. Third method assumes a model for the covariates with missing values [7]. A disadvantage of this kind method assumes implicitly that data are missed at random.

For attribute reduction, the above concepts are not suitable. They all make assumption, so complete analysis of missing value is reduced. In addition, it is hard or even

impossible to check the correctness of assumption in practice. In contrast, rough set can hold complete analysis, since it considers missing value as “everything is possible” [8, 9]. Rough set is proposed by Pawlak [10] as a mathematical theory of set approximation, which is now widely used in information system. In order to find the solution of incomplete system by rough set, tolerance relation is defined through relaxing the equivalence relation [8,11]. It is a NP-hard problem to find an optimal solution. Heuristic approaches have been proposed, keeping the positive region of target decision unchanged [12] or employing conditional entropy to obtain a solution [13]. Each method aims at some basic requirement according to their mechanisms of reduction, so no one can give a fair evaluation among these methods.

In this paper, we find that rough set based attribute reduction with missing value is a sub-problem of propositional satisfiability problem (SAT). SAT is one of the most classical problems because of its significance in both theoretical research and practical application, so it is smart to find solutions of attribute reduction with missing value by method of SAT. Exclusion rule [14,15] checks the satisfiability by inverse of resolution. It can obtain all satisfiable solutions without trying every combination of attributes. So it provides a possible way to find all possible reductions. According to the result of exclusion rule, the distribution of solution with different amount of attributes shows that the total number of solutions is decided by the size of minimal reduction. The total number decreases with addition of missing before threshold. In general, threshold is smaller than 50%. The relationship between computational cost and missing is

*Corresponding author.

also illustrated.

The rest of paper is organized as follows. The basic knowledge about attribution reduction using rough set is given in Section 2. Its relationship with SAT is introduced in Section 3. Experimental results are shown in section 4. Finally, the conclusion is drawn in Section 5.

2. Background

In this section, the basic notions [11] related to incomplete information systems and rough set are presented.

Definition 2.1: Let $IS = (U, A)$ be an information system, where

U - a nonempty, finite set called universe and

A - a nonempty, finite set of attributes i.e.

$a : U \rightarrow V_a$ for $a \in A$

where V_a is called the value set of a .

Definition 2.2: In an incomplete information system $IS = (U, A)$, an $n \times n$ matrix (c_{ij}) called *discernible matrix* of IS is defined as

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j) \text{ and } a(x_i) \neq \text{null}, \text{ and } a(x_j) \neq \text{null}, x_i, x_j \in U\} \text{ for } i, j = 1, \dots, n. \quad (1)$$

The discernible matrix is denoted as $M(IS)$. It is straightforward to find $M(IS)$ is symmetric and $c_{ii} = \emptyset$.

Definition 2.3: A *discernible function* f_{IS} for an information system $IS = (U, A)$ is a Boolean function of m variables a_1, \dots, a_m , which is defined as

$$f_{IS}(a_1, \dots, a_m) = \bigwedge \{ \bigvee (c_{ij}) : 1 \leq j < i \leq n, \bigvee (c_{ij}) \neq \emptyset \}, \quad (2)$$

where a_i denotes an attributes in A and $\bigvee (c_{ij})$ is the disjunction of the variable a' such that $a' \in c_{ij}$.

Definition 2.4: For any subset of attributes $B \subseteq A$, a relation $IND(B)$ called *B-indiscernible relation* is defined as follows:

$$IND(B) = \{ \{x, y_1, \dots, y_l\} : a(x) = a(y_i) \text{ or } a(x) = \text{null} \text{ or } a(y_i) = \text{null}, \forall a \in B, x, y_i \in U, 1 \leq i \leq l \} \quad (3)$$

Definition 2.5: $POS_B(a^*)$ representing *B-positive region* of a^* is defined as

$$POS_B(a^*) = \bigcup BX \quad (4)$$

where, for each $BX \in IND(B)$, $\exists Z \in IND(\{a^*\}) BX \subseteq Z$.

3. Quantization of attribute reduction

In this work, we just show the concepts between attribute reduction and SAT. The details of the demonstrations can be found in our future work.

Definition 3.1: For an information system $IS = (U, A)$, a $n \times (n+1)$ matrix (C_{ij}^*) called *a*-discernible-checked matrix* is defined in two steps:

i.

$$C_{ij}^* = \begin{cases} \{a \in A - \{a^*\} : a(x_i) \neq a(x_j) \text{ and } a(x_i) \neq \text{null} \\ \text{and } a(x_j) \neq \text{null}\} \text{ if } a^*(x_i) \neq a^*(x_j) & \text{for } i, j = 1, \dots, n \quad (5) \\ A - \{a^*\} & \text{if } a^*(x_i) = a^*(x_j) \end{cases}$$

$$C_{i(n+1)}^* = \begin{cases} \emptyset & \text{if } \exists j \in \{1, \dots, n\}, c_{ij}^* = \emptyset \\ A - \{a^*\} & \text{otherwise} \end{cases} \text{ for } i = 1, \dots, n; \quad (6)$$

ii.

$$C_{ij}^* = A - \{a^*\} \text{ if } c_{i(n+1)}^* = c_{j(n+1)}^* = \emptyset, \text{ for } i, j = 1, \dots, n. \quad (7)$$

a^* -discernible-checked matrix is denoted as $M^*(IS)$, and $n+1$ column is called *state checking*. In step ii., the c_{ij}^* of two samples neither in $A - \{a^*\}$ -Positive Region is reduced, making sure that every attribute in minimal reduction is necessary. *a*-discernible-checking function* is $f_{IS}^*(a_1, \dots, a_{m-1}) = \bigwedge \{ \bigvee (c_{ij}^*) : 1 \leq i < j \leq n+1, c_{ij}^* \neq \emptyset \}$.

Definition 3.2 [16] An information system $IS = (U, A)$ is consistent if all objects, which have the same value concerning $A - \{a^*\}$, also have the same value of decisional attribute a^* ; otherwise, the information system is inconsistent.

Theorem 3.1: In a consistent information system, there is not \emptyset in its a^* -discernible-checked matrix, and all samples are in $POS_{A-\{a^*\}}(a^*)$.

Theorem 3.2: In an inconsistent information system, only the sample having \emptyset value of state checking is not in $POS_{A-\{a^*\}}(a^*)$.

Definition 3.3: For a subset B of $A - \{a^*\}$, it is an attribute reduction when $POS_B(a^*) = POS_{A-\{a^*\}}(a^*)$.

Theorem 3.3: B is an attribute reduction iff $f_{IS}^*(V_B(a_1), \dots, V_B(a_{m-1})) = 1$, where $V_B(\cdot) : a \rightarrow \{0, 1\}$ such that $a \in B$ iff $V_B(a) = 1$.

Definition 3.4 *a*-improved-discernible function* f_{IS}^{**} for an information system $IS = (U, A)$ is a Boolean function of $m-1$ variables a_1, \dots, a_{m-1} , which is defined as $f_{IS}^{**}(a_1, \dots, a_{m-1}) = \bigwedge \{ \bigvee (C_{ij}^{**}) : 1 \leq i < j \leq n, a^*(x_i) \neq a^*(x_j) \}$ (8) where a_i denotes an attributes in A .

Theorem 3.4 $f_{IS}^{**} = f_{IS}^*$.

Definition 3.5 [17] An instance of SAT is a Boolean formula in conjunctive normal formula. Each disjunction formula is called a clause, and the instance is called a clause set.

An example of conjunctive normal formula is $\{t \vee l\} \wedge \{\bar{t} \vee \bar{l}\} \wedge \{t \vee m\}$. t , l and m are Boolean variables; overbar is logical operation denoting "not"; \bar{t} , \bar{l} , and $t \vee m$ are clauses. Each Boolean variable can be assigned either true or false.

Theorem 3.5 Attribute reduction based on rough set is a problem of SAT.

According to Theorem 3.3, a set of attributes is a solution of reduction iff the related $f_{IS}^* = 1$. The set of attributes can be translated as an assignment of variables. The values of attributes are true, when they are included in the reduction; otherwise, the values are false.

Not all of instances in SAT are converted into attribute reduction. Give an example to explain this. $(t \vee l) \wedge (\bar{t} \vee \bar{l}) \wedge (\bar{t} \vee \bar{l})$ is an instance of SAT, but it can not be converted into attribute reduction. There are 3

clauses, so the number of objects in the universe is also 3. According to the definition of a^* -discernible-checking function, \bar{t} means the values of two objects are the same. In the instance, two clauses include \bar{t} . That is to say that there are two pairs having the same values concerning the attribute t . As a result, the 3 objects appear to have the same value of t because of transitivity. But it is inconsistent with the third clause. So the clause set can not be written as an instance of attribution reduction. Based on this analysis, we have concluded that attribute reduction based on rough set is the subset of SAT.

Definition 3.6 [14]: $C' = \{C \vee t\} \wedge \{C \vee \bar{t}\}$ is called the result of extension rule on a clause C , where t is an attribute not in C .

Definition 3.7 [14]: A clause is a maximum term iff it contains all attributes in either positive form or negative form.

Theorem 3.6 [14]: For a clause set with $m-1$ attributes, the clause set is unsatisfiable iff 2^{m-1} maximum term can be extended by the clauses.

Theorem 3.7 [14]: Given a clause set $\Sigma = C_1 \wedge C_2 \wedge \dots \wedge C_n$, let P_i be the set of all the maximum terms obtained from C_i by using the extension rule, and set NM the number of distinct maximum terms got from Σ .

$$NM = \sum_{i=1}^n |P_i| - \sum_{1 \leq i < j \leq n} |P_i \cap P_j| + \sum_{1 \leq i < j < k \leq n} |P_i \cap P_j \cap P_k| - \dots + (-1)^{n+1} |P_1 \cap P_2 \cap \dots \cap P_n|, \quad (9)$$

$$|P_i| = 2^{m-1+|C_i|},$$

$$|P_i \cap P_j| = \begin{cases} 0 & \text{complementary forms in } C_i \text{ and } C_j, \\ 2^{m-1-|C_i \cup C_j|} & \text{otherwise.} \end{cases}$$

By knowledge compilation using extension rule, each pair of clauses in the new clause set contains complementary form of the same attribute [15]. So $NM = \sum_{i=1}^n |P_i|$. Algorithm 3.1 shows process of knowledge compilation [15].

Algorithm 3.1 Knowledge compilation using the extension rule

Input: Let $\Sigma_1 = C_1 \wedge C_2 \wedge \dots \wedge C_n$ be a set of clauses, $\Sigma_2 = \Sigma_3 = \emptyset$

While $\Sigma_1 \neq \emptyset$

Loop

Select a clause from Σ_1 , say C_1 , and add it into Σ_2

While $i <$ the number of clauses in Σ_1

Loop

While $j <$ the number of clauses in Σ_2

Loop

If C_i and C_j contain complementary forms of the same attribute, skip.

Else if C_i subsumes C_j , eliminate C_j from Σ_2 .

Else if C_j subsumes C_i , eliminate C_i from Σ_1 .

Else extend C_j on a variable using extension rule.

$j = j + 1$

End loop

$i = i + 1$

End loop

$\Sigma_3 = \Sigma_3 \cup \Sigma_2$; $\Sigma_2 = \emptyset$

End loop

Output: Σ_3 is the result of the compilation process.

Theorem 3.8: Set $f_c(a_1, \dots, a_{m-1}) = \vee(C_{ij}^*)$ and $f_c'(a_1, \dots, a_{m-1}) = (\vee(C_{ij}^*) \vee t) \wedge (\vee(C_{ij}^*) \vee \bar{t})$, where t is an attribute not included $\vee(C_{ij}^*)$. $f_c(V_B(a_1), \dots, V_B(a_{m-1})) = 1$ iff $f_c'(V_B(a_1), \dots, V_B(a_{m-1})) = 1$, where $V_B(\cdot): a \rightarrow \{0, 1\}$ such that $a \in B$ iff $V_B(a) = 1$.

Theorem 3.9: If a maximum term of conditional attributes dose not subsume any clause in f_{IS}^{**} , the attributes with negative form in the maximum term is a solution of attributes reduction; if a maximum term of conditional attributes dose not subsume any clause in the extended results of f_{IS}^{**} , the attributes with negative form is a solution of attributes reduction.

Theorem 3.10: The number of attribute reduction is $2^{m-1} - NM$, where NM is the number of distinct maximum terms got from f_{IS}^{**} and $m-1$ is the number of conditional attributes.

According to the results of complication knowledge, the distribution of solution can be found directly. That is because every clause has complementary literature of other clauses. That is to say that the maximum terms extended by each clause are totally different from others. The supplementary of maximum terms extended by the clauses after complication knowledge is the set of all solutions. It is not generally necessary to obtain all solutions, because it will take memory cost of 2^{m-1} . In this work, the distribution of solution is found without extending maximum terms, which is shown in Algorithm 3.2.

Algorithm 3.2 Distribution of attribution reduction

Input: Clause set after knowledge compilation $f' = C_1' \wedge C_2' \wedge \dots \wedge C_{n'}'$; a count set include m elements, where $m-1$ is number of conditional attributes.

Set $i = 0$.

Initial count set:

While $i \leq m-1$

Loop

Value of the $(i+1)$ th element in count set is $\frac{C(m-1, i)}{(m-1)!(m-1-i)!}$.

End loop

Set $i = 1$.

While $i \leq n'$

Loop

Select C_i from f' .

Denote the number of literatures L , and number of negative

form NL .

The value of $(NL+1)$ th element in count set is decreased 1.

Set $j = NL + 1$.

While $j < m-1$

Loop

The $(j+1)$ th element in count set is decreased

$\frac{C(m-1-L, j-NL)}{(j-NL)!}$.

End loop

End loop

Output: The distribution of attribute reduction is in the count set.

4. Experiments

In this section, we show the process finding distribution of solutions to analyze the characteristics of attribute reduction. Table 1 shows the details of 3 UCI datasets.

Table 1 Dataset description

Datasets	Samples	Attribute	Classes	Proportion of missing
Zoo	101	18	7	0%
Soybean	47	36	4	0%
Voting	435	17	2	14.7%

Table 2 Number of clause

Missing \ Dataset	Zoo	Soybean	Voting
Equation 8	3873	810	44859
Original dataset	956	615	8460
1%	1144.9	664	-
3%	1458.9	711.3	-
5%	1685.2	744	-
10%	2125.9	771.7	-
20%	2451.1	784.5	4122
30%	2081.8	751	2378
40%	1326.4	694.9	1018
50%	230.1	450.9	465
60%	3.4	114.1	0
70%	0	0	0
80%	0	0	0
90%	0	0	0

4.1. Analysis of a^* -improved discernible

a^* -improved-discernible function depicts the discernible relationship among samples. It reduces unnecessary elements of a^* -discernible-checked matrix for correct classification purpose. First, element about 2 samples in the same class is abandoned. Second, element about 2 samples neither in the $A - \{a^*\}$ -positive region is omitted. It is not possible that these samples are correctly classified by rough set. The clauses in a^* -improved-discernible function is

$$\frac{n(n-1)}{2} - \frac{l_1(l_1-1)}{2} - \frac{l_2(l_2-1)}{2} - \dots - \frac{l_w(l_w-1)}{2} - \frac{\bar{n}(\bar{n}-1)}{2}, \quad (10)$$

$$= \frac{n^2 + \bar{n} - \bar{n}^2 - (l_1^2 + l_2^2 + \dots + l_w^2)}{2}$$

where n is the total number of samples, \bar{n} is the number of samples not in $A - \{a^*\}$ -positive region, and l_i is the number of samples in the same class.

By the characteristic of conjunction, a^* -improved-discernible function just keeps one copy of the same

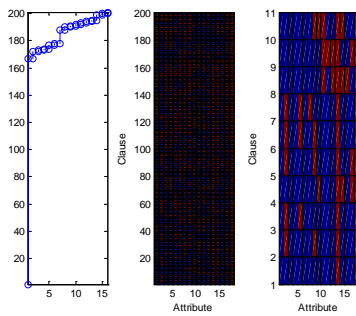
clauses to reduce memory burden. So its exact number of clauses is smaller than Equation 8 in Definition 3.4. Null value is randomly set in order to observe the variation rule of clauses with varying of missing proportion. Table 2 shows a summary of exact clauses in a^* -improved-discernible function. It can be observed that exact number of clauses increase with addition of missing. Missing damages the distribution of values in each attribute. As a result, the relationship of samples in different class is more complex with increasing of missing. However, the addition of clauses stops, when the missing proportion is large enough. That is because that the samples in the $A - \{a^*\}$ -positive region decreases. The maximum of clauses are found with missing from 10% to 20%.

4.2. The number of Clauses after Knowledge Compilation

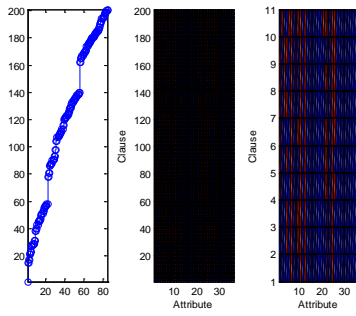
During the process of knowledge compilation using extension rule, one clause is deleted from the clause set, when the clause subsumes another clause [15]. So the subsuming plays important role in the computational cost of memory and time. Since the number of clauses in a^* -improved-discernible function is large, authors can not find anything in the figures showing all the clauses. In Fig.1, 200 clauses in the original datasets are randomly picked as example to explain subsuming.

Every figure in Fig.1 includes three subfigures. The middle one describes the 200 clauses of a^* -improved-discernible matrix. Every column expresses one feature; one row is a clause. Red denotes the feature in the related clause, and blue means the feature not in a clause. For a clause, its set of red features contains all red features of another clause. Then, this clause is directly deleted in knowledge compilation. The third subfigure shows this detail of the first 10 clauses. For example, in the third subfigure of Fig.1a, the red features of the first clause are included by other 9 clause, so the 9 clauses are deleted. The first subfigure gives a summary of subsuming in the second subfigure. The clauses in every two circles are in a subsuming set where only the first clause is kept. It can be observed that the clauses of Zoo and Voting are dynamically deleted, shown in Fig. 1(a) and 1(c).

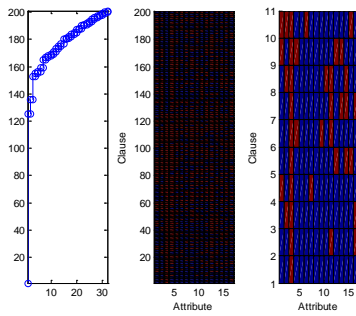
Fig. 2 illustrates the amount variation of clauses under increasing of missing during the process of knowledge compilation. It is very interesting that the variation of deleted clauses is similar with a^* -improved discernible function. We can conclude that the subsuming characteristic of dataset is not changed by missing proportion. In Figs. 2(a) and 2(b), clauses in the extended sets increase with addition of missing, and their increasing stops in enough large missing proportion. The rule is also found in Figs. 2(c) although the variation is not dramatic.



(a)

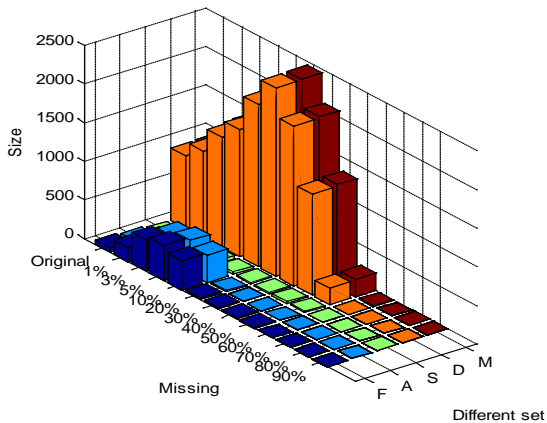


(b)

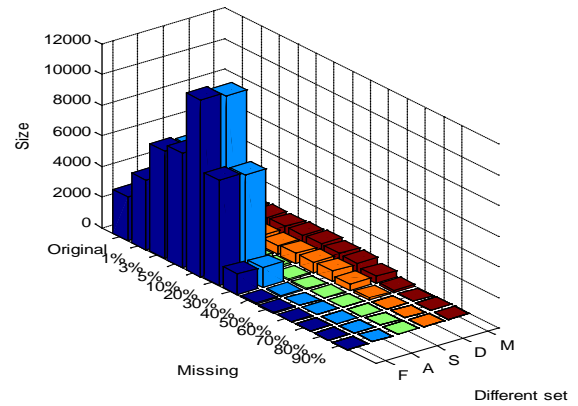


(c)

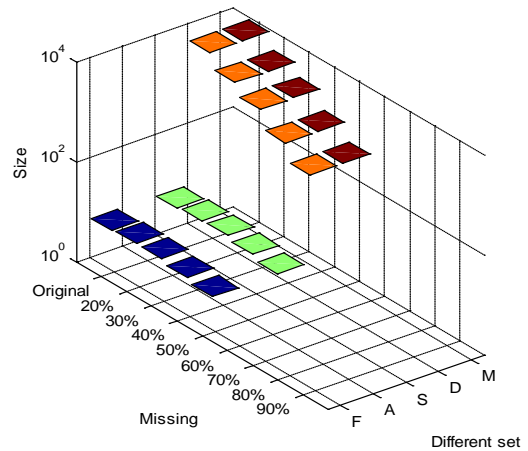
Figure 1. Subsuming of feature values. (a): Zoo; (b): Soybean; (c): Voting.



(a)

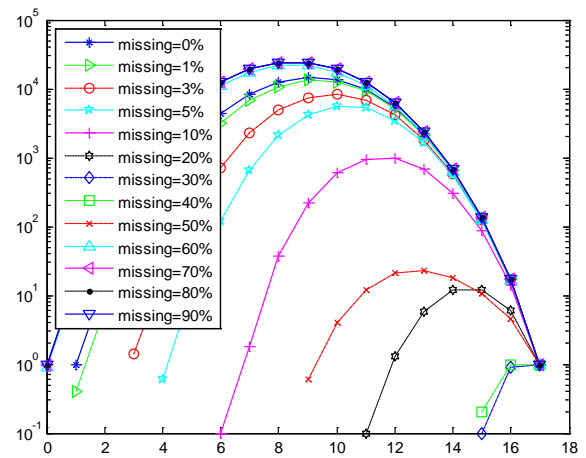


(b)



(c)

Figure 2. Amount variation of clause. The value “F” in Different set axis expresses the number of clauses in the extended set; A denotes the additional clauses in the processes of knowledge compilation; S is the number of subsuming set in α^* -improved discernible function; D expresses the deleted clauses; M is the clauses in α^* -improved discernible function. (a): Zoo; (b): Soybean; (c): Voting.



(a)

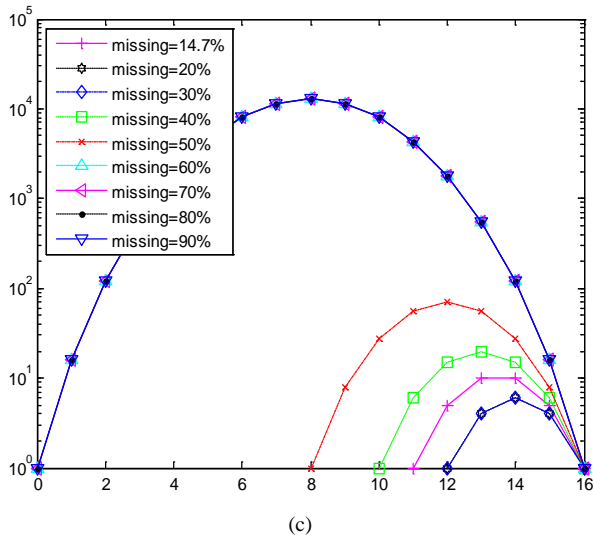
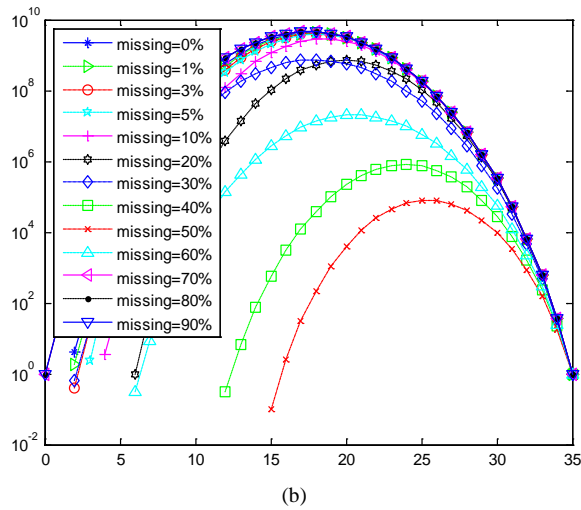


Figure 3. Distribution of solutions. (a): Zoo; (b): Soybean; (c): Voting.

4.3. Distribution of Solutions

According to Algorithm 3.2, the distribution of all possible solutions can be found, shown in Fig.3. It can be observed that the amount of all solutions is decided by the size of minimal reduction. The number of all solutions is approximated to the area delimited by the curve and x-axis. For a dataset, a reduction having the smallest size among all possible solution is minimal reduction. In every figure, if the minimal reduction has smaller size, its number of all solutions is larger. The reason is that any set containing minimal reduction is also a solution.

Table 3 is the summary about amount of solutions. The number of all solutions decreases with addition of missing before threshold. Then, the number increases. For each attribute, the original distribution of values in different class is disordered by missing. This brings

Table 3. Number of solutions.

Dataset	Zoo	Soybean	Voting
Original dataset	74448	3.16×10^{10}	32
1%	65622.6	3.022×10^{10}	-
3%	37285.3	2.837×10^{10}	-
5%	23799	2.659×10^{10}	-
10%	3869.9	1.916×10^{10}	-
20%	38.3	4.622×10^9	16
30%	2	5.485×10^9	16
40%	2.2	4870963.2	64
50%	94.8	445644.8	256
60%	118067.2	144703488	65536
70%	131072	3.44×10^{10}	65536
80%	131072	3.44×10^{10}	65536
90%	131072	3.44×10^{10}	65536

about decrease of discernible ability of every attribute, so more attributes are needed to provide the same samples in $A - \{a^*\}$ - positive region. It results in decrease of all solution with the addition of missing. However, the samples in $A - \{a^*\}$ - positive region are so fewer that small amount of attributes is enough to provide the same discernible ability, after missing is larger than threshold. So solution begins increasing. In Table3, the minimum of solutions is obtained before missing is smaller than 50%.

5. Conclusions

Rough set based attribute reduction with missing value is demonstrated as a sub-problem of satisfiability problem. Extension rule, a satisfiability method, is employed to obtain distribution of solutions without finding all attribute reductions. The relation between attribute reduction and missing is analyzed from computational cost and amount of solutions.

REFERENCES

- [1] Liu, Y., Dimensionality reduction and main component extraction of mass spectrometry cancer data. Knowledge-Based Systems, 2011.
- [2] Lu, J., T. Zhao, and Y. Zhang, Feature selection based-on genetic algorithm for image annotation. Knowledge-Based Systems, 2008. 21(8): p. 887-891.
- [3] Chmielewski, M.R., et al., The rule induction system LERS-a version for personal computers. Foundations of

- Computing and Decision Sciences, 1993. 18(3-4): p. 181-212.
- [4] Booth, D.E., Analysis of Incomplete Multivariate Data. *Technometrics*, 2000. 42(2): p. 213-214.
- [5] Hudak, A.T., et al., Nearest neighbor imputation of species-level, plot-scale forest structure attributes from Li-DAR data. *Remote Sensing of Environment*, 2008. 112(5): p. 2232-2245.
- [6] Pelckmans, K., et al., Handling missing values in support vector machine classifiers. *Neural Networks*, 2005. 18(5-6): p. 684-692.
- [7] Myers, J.W., K.B. Laskey, and T. Levitt. Learning Bayesian networks from incomplete data with stochastic search algorithms. 1999. Morgan Kaufmann Publishers Inc.
- [8] Kryszkiewicz, M., Rough set approach to incomplete information systems. *Information sciences*, 1998. 112(1): p. 39-49.
- [9] Slowinski, R. and J. Stefanowski, Rough classification in incomplete information systems. *Mathematical and Computer Modelling*, 1989. 12(10-11): p. 1347-1357.
- [10] Pawlak, Z., Rough sets. *International Journal of Parallel Programming*, 1982. 11(5): p. 341-356.
- [11] Leung, Y. and D. Li, Maximal consistent block technique for rule acquisition in incomplete information systems. *Information sciences*, 2003. 153: p. 85-106.
- [12] Meng, Z. and Z. Shi, A fast approach to attribute reduction in incomplete decision systems with tolerance relation-based rough sets. *Information sciences*, 2009. 179(16): p. 2774-2793.
- [13] Liang, J., Z. Shi, and D. Li, The information entropy, rough entropy and knowledge granulation in rough set theory. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 2004. 12(1): p. 37-46.
- [14] Hai, L., S. Jigui, and Z. Yimin, Theorem proving based on the extension rule. *Journal of Automated Reasoning*, 2003. 31(1): p. 11-21.
- [15] Hai, L. and S. Jigui, Knowledge compilation using the extension rule. *Journal of Automated Reasoning*, 2004. 32(2): p. 93-102.
- [16] Grzymala-Busse, J.W., Managing uncertainty in expert systems. Vol. 143. 1991: Springer.
- [17] Kirkpatrick, S. and B. Selman, Critical behavior in the satisfiability of random boolean expressions. *Science*, 1994. 264(5163): p. 1297-1301.