International Journal of Hybrid Information Technology
Vol. 9, No.8 (2016), pp. 333-346
http://dx.doi.org/10.14257/ijhit.2016.9.8.29

Feature Selection based on Rough Sets and Minimal Attribute Reduction Algorithm

Khaled Alwesabi[1*], Weihua Gui[1], Chunhua Yang[2] and Hamdi Rajeh[3]

[1*,1,2]*School of Information Science and Engineering, Central South University Changsha, Hunan (HN), 410000/ Time (UTC+8), China*
[3]*Hunan University, Changsha, Hunan, China*
[1*]*k_alwesabi@csu.edu.cn,* [1]*gwh@mail.csu.edu.cn,* [2]*ychh@mail.csu.edu.cn*
[3]*hamdi@hnu.edu.cn*

Abstract

Numerous studies have focused on feature selection using many algorithms, but most of these algorithms encounter problems when the amount of data is large. In this paper, we propose an algorithm that handles a large amount of data by partitioning the data to process a reduction, and then selecting the intersection of all reducts as a stable reduct. This algorithm is successful but may suffer from loss of information if the samples are unsuitable. The proposed algorithm is based on discernibility matrix and function. Furthermore, the method can address the case in which the data consist of a significant amount of information. Our results show that the proposed algorithm is powerful and flexible enough to successfully target a range of different domains and can effectively reduce computational complexity as well as increase reduction efficiency. The efficiency of Proposed Algorithm is illustrated by experiments with UCI datasets further.

Keywords*: Rough set algorithm, Minimal attributes reduction, Partition algorithm, Reduct and Core*

1. Introduction

A significant amount of data in the real world contain noise and other peculiarities that cause difficulty in extracting particular features out of them. Thus, we address the problem of feature selection from a sample of decision table data. In other words, in most organizations, data are rarely specifically collected or stored in a database for the purpose of mining knowledge. Thus, a database always contains numerous attributes that are redundant and not necessary for rule discovery. Also, many attributes exist that are not necessary for rule discovery. Moreover, an attribute that should be deleted is difficult for both non-experts and experts to determine. Clearly, developing methods to select the attribute subset is necessary. The problem of feature subset selection is that of finding an optimal subset of attributes of a database according to a certain criterion so that a classifier with the highest possible accuracy can be generated by an inductive learning algorithm that is run on data containing only the subset of features.

To solve this problem, many methods for selecting a subset of features have been proposed. Among these methods are the filter approach, which involves sieving the irrelevant and/or redundant features without considering the induction algorithm [13], and the wrapper approach, which uses the induction algorithm itself as a black box in attribute selection to select a good feature subset that improves the accuracy of the induction algorithm [8]. Although the wrapper approach has significantly improved the accuracy of well-known algorithms, such as C4.5 and Naive Bayes, its generalization is limited for the following reasons:

* Corresponding Author

ISSN: 1738-9968 IJHIT

I) high computational cost, which results from calling the induction algorithms for each feature subset considered,
II) impracticability of dealing with large datasets; for the filter approach, the main limitations are the following:
   a. it ignores the effects of selected feature subsets on the performance of the induction algorithm [8], and
   b. various heuristics tend to overestimate the multi-valued attributes [10].

Many rough set algorithms are used for feature selection, but most of these algorithms encounter problems in dealing with large amounts of data.

This paper proposes a simple method based on partitioning the data before processing them and after coding them in a perfect way. This method enables data processing in a simple way. In this study, a code based on discernibility matrix and function is used. This code selects the minimum reduct, which represents the important features of the data directly and effectively. These features represent the data in a way that causes them to lose validity and importance when these features are taken out. Compared with the results of other algorithms used to solve the same problem, excellent results are obtained by the proposed method in terms of accuracy and processing time. The method exhibits the following merits:

I) It is suitable for situations with a tremendously large amount of data.
II) It is suitable for situations with incremental data.
III) It splits data randomly into parts and then "divides and conquers" them.
IV) It is suitable for parallel computing.

The feature selection step of the construction procedure of our new classification technique is based on the calculation of dynamic reduct. This process involves the reduction of an uncertain and noisy decision table by using a dynamic approach that extracts relevant information. A reduct is a minimal subset of attribute reduction and is one of the most fundamental and important ideas in rough set theory.

The rest of this paper is structured as follows. Section 2 presents an overview of related work. Section 3 introduces the fundamentals of rough set theory and several properties of discernibility matrix and function. Section 4 provides background information on minimal attribute reduction based on the discernibility function and an overview of the complete algorithm for minimal attribute reduction based on discernibility function (CAMARDF). Section 5 briefly describes the approach of feature selection based on rough set theory. Section 6 presents concepts related to the minimum attribute reduction and provides an overview of our partition algorithm. Section 7 shows experimental results and compares our approach with a related method. Section 8 concludes our paper.

2. Related Work

Owing to the complexity of the real world, knowledge discovery from real-world databases is a multi-phase process that involves discretization of continuous attributes, feature selection, inductive learning, and other steps. Rough set theory provides a useful mathematical tool that can be used not only for selection minimal attribute reduction but also for other steps in the discovery process. We are developing a rough-set-based knowledge discovery process.

Rough set theory was introduced by Pawlak [17] in the 1980s and applied in knowledge discovery systems to identify and remove redundant variables [20], as well as to classify imprecise and incomplete information [11]. A reduct of a decision table is a subset of condition attributes that suffice to define the decision attributes. More than one reduct that is a constructing associative classifier from decision tables may exist. The intersection of all possible reducts is called the core [7], which represents the most important information of the original dataset [2]. Feature

selection is a basic problem in pattern recognition and has been a fertile field of research and development since the 1970s.

Feature selection has been effective in removing irrelevant and redundant features, increasing efficiency in learning tasks, and enhancing learning performance. Feature selection methods are classified into two broad categories: filter model and wrapper model [8]. The wrapper model requires one predetermined learning algorithm in feature selection and uses its performance to evaluate and determine which features are selected. The filter model relies on general characteristics of the training data to select certain features without involving any specific learning algorithm. Evidence proves that the wrapper model often performs better on small-scale problems [8]; however, on large-scale problems, such as text classification, this model is impractical because of its high computational cost. Therefore, in text classification, filter methods that use feature scoring metrics are popularly used. In this section, we review recent studies on feature selection for both topic-based and sentiment classification. In the past decade, feature selection studies mainly focused on topic-based classification where the classification categories are related to the subject content, such as sport or education. Yang and Pedersen [2] investigate five feature selection metrics and report that good feature selection methods improve the categorization accuracy with an aggressive feature removal using DF, IG, and CHI. In 2003, Forman empirically compares 12 feature selection methods on 229 text classification problem instances and proposes a new method called bi-normal separation (BNS) [5].

Experimental results show that BNS can perform effectively in evaluation metrics of recall rate and F-measure. However, in terms of precision, BNS often loses to IG. Be-sides these two comparison studies, many other researchers have contributed to this topic [22] and an increasing number of new feature selection methods have been generated, such as Gini index [21], distance to transition point [17, strong class information words [12], and parameter tuning-based feature selected for Rocchio classifier [14]. Sentiment classification has also become popular because of its widespread application [23].

Recently, sentiment classification has become popular because of its wide range of applications [19]. Its classification criterion is the attitude expressed in the text (e.g., recommended or not recommended, positive or negative) rather than facts (e.g., sport or education). To the best of our knowledge, no related work has focused on comparing feature selection methods on this special task. There are only some scattered reports in their experimental studies. Riloff *et al*. [21] report that the traditional FS method, which only uses the IG method, performs worse than the baseline in some cases. However, Cui *et al*. [4] present experiments on the sentiment classification for large-scale online product reviews to show that using the FS method of CHI does not degrade the performance but can significantly reduce the dimension of the feature vector. Moreover, Ng *et al*. [17] examine the feature selection of the weighted log-likelihood ratio (WLLR) on the movie review dataset and achieves an accuracy of 87.1%, which is higher than the result re-ported by Pang and Lee [17] with the same dataset. From the preceding analysis, we believe that the performance of the sentiment classification system is also dramatically affected by feature selection.

In this paper, we use two algorithms to select the minimum reducts, which represent the important features from the data directly and effectively. The first algorithm, which is called simplification matrix, is proposed by Yao and Zhao [21], while the second algorithm called CAMARDF is used to find a minimal reduct and is proposed by Zhou, Miao, and Feng [23].

3. Rough Set Base Approach

The Rough set theory proposed by Pawlak [18] provides a mathematical tool that can be used to determine all possible feature subsets [10]. Unfortunately, the number of possible subsets is always very large when N is large because 2N subsets exist for N features. Thus, examining exhaustively all subsets of features to select the optimal one is NP-hard. Most practical algorithms attempt to fit the data by solving the NP-hard optimization problem [3].

In rough set theory, knowledge is represented in information systems. An information sys-tem is a dataset represented in a table. Each row in the table represents an object, such as a case or an event. Each column in the table represents an attribute, such as a variable, an observation, or a property. Some attribute values are assigned to each object (row).

Some basic terms and notations on rough set theory must be explained first. In rough set theory, a decision table is denoted by

$$S = (U, A, C, D) \tag{1}$$

where U is the non-empty finite set of objects called universe of discourse; A is the non-empty finite set primitive features; and C, D $\subset$ A are two subsets that are called condition and decision features, respectively [23].

3.1. Information System and Indiscernibility Relation

Given a subset of attributes P$\subseteq$ A, each subset defines an equivalence relation IND (P) called an indiscernibility relation. This indiscernibility relation is defined as

. $$IND(P) = \{(x, y) \in U^2 : for\, a \in P, a(x) = a(y)\} \tag{2}$$

Let U/IND (P) denote the family of all equivalence classes of the relation IND (P). For simplicity of notation, U/P is written instead of U/IND (P). Equivalence classes U/INDI and U/IND (D) are called condition and decision classes, respectively.

3.2. Class Approximation

- **Lower approximation**: The R-lower approximation set of X is the set of all elements of U, which can be classified with certainty as elements of X, based on the assumption of knowledge R. This approximation can be presented formally as

$$\underline{R}X = \bigcup \{Y \in U / R, Y \subseteq X\} \tag{3}$$

- **Upper approximation**: The R-upper approximation set of X is the set of all elements of U, which can be classified as elements of X, based on the assumption of knowledge R. This approximation can be presented formally as

$$\overline{R}X = \bigcup \{Y \in U / R, Y \bigcap X \neq \phi\} \tag{4}$$

- **Positive region**: The C-positive region of D is the set of all objects from universe U, which can be classified with certainty to classes of U/D by employing attributes from C as follows:

$$POS_c(D) = \bigcup_{X \in U / D} \underline{C}X \tag{4}$$

3.3. Dispensable and Indispensable Features

Every dataset contains conditional and decision features. Some of these features are indispensable and are very important in the analysis [23]. The problem of feature selection is searching for indispensable features and eliminating the dispensable ones. Let c $\in$C. A feature c is dispensable in S if $POS_{C-\{c\}}(D) = POS_C(D)$; otherwise, feature c is considered as indispensable in S. If c is an indispensable feature, then deleting it from S makes S inconsistent. S is said to be independent if all of its features are indispensable.

3.4. Reduct and CORE

In the following, we present the definitions of reduct and CORE:

- **Reduct**: A system $S=(U,A,C,D)$ is independent if all c in C are indispensable. A set of feature R in C is called a reduct if $S=(U,A,C,D)$ is independent and $POS_R(D)=POS_C(D)$.

A reduct is a minimal set of features that preserves the indiscernibility relation produced by a partition C. Several subsets of attributes such as R may exist.

- **CORE**: The set of all features indiscernible in C is denoted by CORE (C). The CORE is the set of all single element entries of the discernibility matrix, that is,

$$CORE(C)=\{a\in C: m_{ij}=\{a\} \ \ for\ \ some\ i,\ j\} \tag{6}$$

We have

$$CORE(C)=\bigcap RED(D).$$

Where *RED(C)* is the set of all reducts of *C*. Thus, the CORE is the intersection of all reducts of an information system. The CORE does not consider the dispensable features and can be expanded using reducts. The feature subset obtained is good enough to enable information induction.

3.4. The Discernibility Matrix and the Discernibility Function

A prime implicant of a Boolean function is an implicant that cannot be covered by a more general implicant. Skowron [22] has proved that all reducts are in one to one correspondence with the prime implicants of the discernibility function in a given decision table.

The problem of finding minimal reducts is polynomially equivalent to the problem of searching for prime implicants of the discernibility function with the shortest length. A prime implicant with the shortest length means that the number of its variables is minimal [23].

Some detailed descriptions of rough set theory can be found in the works by [19] [22].

Definition 1 [22]. Decision table $DT=(U,C\bigcup D,V,\rho),\ \ U=\{x_1,x_2,...,x_n\}$ The discernibility matrix can be defined as $n \times n$ matrix $DM(DT)=(c_{ij})_{n\times n}$, where the element c_{ij} satisfies the following:

$$c_{ij}=\begin{cases}\{a\mid a\in C\wedge\rho(x_i,a)\neq\rho(x_j,a)\} & \Omega\\ \phi & Otherwise\end{cases} \tag{7}$$

For i, j = 1, 2, 3… n,

In Eq. (7), Ω means $1\le j<i\le n,\rho(x_{i,},D)\neq\rho(x_j,D)$ and at least one object between x_i and x_j is consistent.

Definition 2 [22]. Given a decision table $DT=(U,C\bigcup D,V,\rho),$ the discernibility function of *DT* is a Boolean function where each Boolean variable *a* is identified with attribute $a\in$ C and is defined as follows:

$$DF(DT)=\wedge\{\vee c_{ij}:1\le j<i\le n,c_{ij}\neq\phi\} \tag{8}$$

Where $c_{ij}\in DM(DT)$, and $\vee c_{ij}=\vee a\left(a\in c_{ij}\right)$ is the disjunction of all variables such that $a\in c_{ij}$. Absorption law is often adopted to reduce the discernibility function, and the reduced discernibility function is also a conjunctive normal from obviously.

In the sequel, we use reduced discernibility function in the discussion. Suppose a reduced discernibility function $DF=f_1\wedge f_2\wedge...\wedge f_s$, we consider $DF=\{f_1,f_2,...,f_s\}$

instead and if $f_i = a_1 \vee a_2 \vee ... \vee a_{ki}$, we consider instead when no confusion arises. The set of all variables of DF is denoted as ψDF.

4. Minimal Attribute Reduction Based on Discernibility Function

To find a minimal reduct of a decision table, an iterative algorithm can be constructed by applying theorems 2 and 3 repeatedly. Based on theorems 2 to 5 (see [23]), some search strategies can be added to minimal attribute reduction based on depth-first search method.

I) We choose the decomposition variable according to its significance from maximal to minimal because choosing the attribute with higher significance reduces the search space faster.

II) If the order of variables is constructed for the first time according to their significance and this order is unchanged in the sequel decomposition procedures, then the order is called static variable. By contrast, if the attribute significance is changed dynamically based on different Boolean functions in the sequel decomposition and the relevant order of attributes is also changed simultaneously, then the order is called dynamic variable. The latter type of order is applied in the algorithm implementation.

III) If the length of the current variable sequence in a depth search path is equal to the length of the candidate minimal reduct, then the current depth search is terminated and the path returns to the upper layer for width search continually.

We suppose that $\psi DF = \{a_1, a_2, ..., a_t\}$ and the variable order is $a_1 > a_2 >, ... > a_t$ based on significance. According to I, a_k is preferential to a_{k+1}. After the search path beginning from a_k is terminated, the shortest implicate that includes a_k is found. For the search path beginning from a_{k+1}, we can only deal with Boolean function $\wedge\{f_i \mid f_i \in DF \wedge a_k \notin f_i\} \wedge \{(f_i - \{a_k\}) \mid f_i \in DF \wedge a_k \in f_i\}$ using theorems 2 and 3 iteratively. If one clause is empty after certain variables are removed during the decomposition procedure, then the algorithm returns to the upper layer.

The complete algorithm for minimal attribute reduction based on discernibility function (CAMARDF) is described as follows:

Algorithm 1: (CAMARDF)

Input: decision table $S = (U, C \bigcup D, V, \rho)$;

Output: a minimal reduct of *S*.

Initialization: *Reduct.length* = 0, *MinReduct.length* = |C| and reduced discernibility function *DF* has been constructed.

```
CAMARDF (DF)
{
1    computeSIG( a , a ∈ ψDF );
2    SortSIG( sig(a) , a ∈ ψDF );
3    i = 0;
4    do {
5    Reduct.length ++;
6   If(Reduct.length=minReduct.legth){
7      Reduct.length --;
8      return;
9   }//end if
```



```
10  if (  i>0  ){
11        DF = Df\{Attribute[i  -  1]};
12        if  ( ∃f_i ∈ DF,  f_i = 0  ) {
13              Reduct.length --;
14                    return;
15        }//end if
16  }//end if
17  Reduct=reduct∪Attribute[i];
18  { DF' = DF − { f_i | f_i ∈ DF
    { ∧ Attribute[i] ∈ f_i }
19  if (DF' = φ ){
20   if(MinReduct.length>reduct.length)
21       MinReduct = Reduct
22   }/end if
23   else
24        CAMARDF  (DF');
25     Reduct=Reduct - Attribute[i]
26     Reduct.length --;
27     I ++;
28   }while( sig (Attribut [i]) >1∧i <|C| );
29  }//end CAMARDF
```

Where, DF\ {Attribute [i – 1]} denotes $\forall f_i \in DF$, if Attribute [i - 1] $\in$ f_i, then $f_i = f_i - \{Attribute[i-1]\}$.

Reduct and MinReduct are global variables in the algorithm. For more details about the work of the algorithm can be found in [23].

5. Feature Selection Using Rough Set

Feature selection is considered as an important research topic in machine learning [5], and is an effective means to identify relevant features for dimensionality reduction [6]. In many applications, especially during the age of information explosion, many features that are potentially useful are collected. Rough set theory is a mathematical tool that has been used successfully to discover data dependencies and reduce the number of attributes contained in a dataset by purely structural methods [8]. Reducts that are obtained by using rough sets are highly informative and all the other attributes can be removed with minimal information loss because of the use of the degree of dependency measure suggested by [20] and others used by many others authors [23] [8].

For example:

Consider the knowledge representation system presented in Table 1 with $U = \{x_1, x_2, x_3, ... x_7\}$, $C = \{a, b, c, d\}$, $D = \{d\}$.

Table 1. A Sample DataBase

U	(a)	(b)	(c)	(d)	(E)
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

Table 2. The Discernibility Matrix

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	ϕ					
x_3	b, c, d	b, c				
x_4	b	b, d	C, d			
x_5	a,b,c,d	a,b,c	ϕ	a,b,c,d		
x_6	a,b,c,d	a,b,c	ϕ	a,b,c,d	ϕ	
x_7	ϕ	ϕ	a,b,c,d	a, b	c, d	c, d

The discernibility function corresponding to the discernibility matrix is as follows:

= (b ∨ c ∨ d) b (a ∨ b∨ c ∨ d) (b ∨ c) (b ∨ d) (a ∨ b ∨ c) (c ∨ d)
= b (c∨ d)
= bc ∨ bd.

There are two reduct sets, namely, {b, c} and {b, d}. Thus, {b} is the CORE of C = {a, b, c, d} (see Table 2).

The feature is the CORE = {b}. We can see that b is the unique feature for discerning x1 and x4. Furthermore, the two reducts are {b, c} and {b, d}. Since feature α is not contained in any reduct, it should be deleted. In other words, the CORE can be defined as the set of all singleton entries in the discernibility matrix. The reduct is the minimal element in the discernibility matrix, which intersects all the elements of this matrix. The reducts can be obtained by using the complete algorithm for minimal attribute reduction based on the discernibility function.

The features in CORE must be included in an optimal result and in an approximate result. Clearly, if the accuracy of a decision table is unchanged, then all indispensable features in CORE cannot be deleted from C.

6. Selection Minimal Attributes Reduction

A reduct is a minimal subset of attribute reduction, which is one of the most fundamental and important ideas in rough set theory. A reduct is a minimal attribute that preserves the same information considered as provided by the entire set of attributes [23] that are directly derived from reducts that will be distinguished. Thus, we intend to find the minimal reducts, that is, the shortest reducts, so that attributes can be removed as much as possible.

6.1. Proposed Algorithm

Yao and Zhao in their work [23] did not provide an optimized implementation of their algorithm (simplification matrix). Furthermore, according to our tests, the algorithm did not exhibit good performance when dealing with big data sets, but it delivers good results when applied to small data sets. By contrast, CAMARDF provides good results when applied to larger data sets (less than 3000). For these reasons, and to take advantage of both algorithms, we propose an algorithm that controls the flow of the CAMARDF algorithm according to the size of the input data set as presented in Figure 1.

The proposed algorithm uses a simple method based on partitioning the data into several parts before processing Parts, for Example ($part_1, part_2, \ldots\ldots part_n$), and after coding it in a perfect way. This method enables data processing in a simple way. I have used a code based on the discernibility matrix and function. This code selects the minimum reduct, which represents the important features from the data in a direct and strong way. These features represent the data in a way that if these features are taken out, the data lose its validity and importance. Excellent results were obtained in comparison with the results of other algorithms for the same problem, with consideration of the accuracy and the time of processing factors. An excellent and high-accuracy factor was obtained. The method is explained at the end of this paper.

To find the minimum reduct, we first test the data volume using the following algorithm in which we combined two algorithms: the first is the simplification matrix algorithm (see [21]), and the second is the complete algorithm for minimal attribute reduction based on discernibility function CAMARDF (see [23]). After an organization identifies opportunities for performance improvement through data analysis.

6.2. How Does The Algorithm Work?

Since our data is in Excel format, we first designed a code that transfers the data to a. mat file. We have written another code that works immediately after the code translates the data from Excel, and its purpose is to partition the data randomly. The number of parts is to be determined by the user only if the amount of data is extremely large.

Thereafter, we use a complete algorithm for minimal attribute reduction if the data is extremely large; otherwise, we use a row-wise simplification matrix reduct construction algorithm (Figure 1).

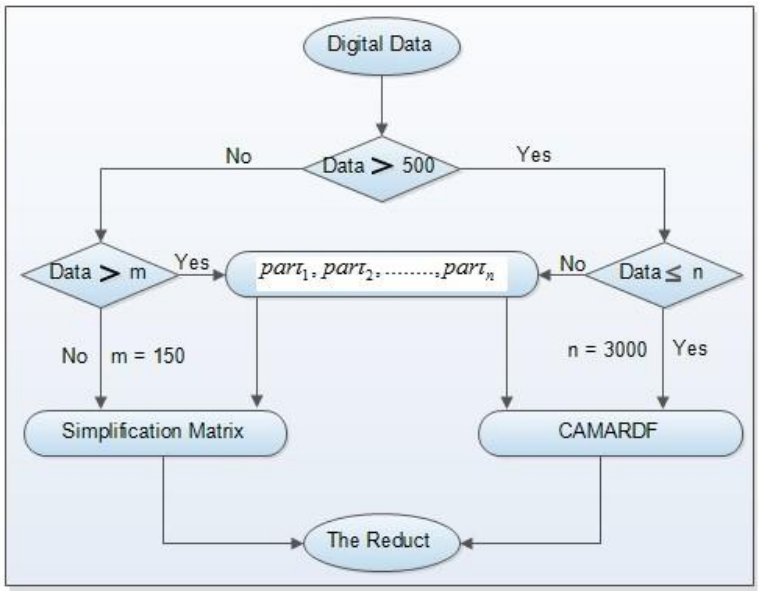

Figure 1. Partition Algorithm

We run both that require the a. mat data file. These algorithms take a long time to function if the data algorithms are extremely large. The results of these algorithms are saved in other files to be mainly used in a row-wise simplification matrix reduct construction algorithm. When we run this algorithm, it asks for the a. mat data files, and then it constructs and displays the classification rules and tree. It also calculates the accuracy of the classification tree by comparing the result from the tree and the original class with the data (the last field of the data file, which is the decision field).

6.3. Illustrative Example

Given that we are working on a large amount of data, we have proposed to partition the data randomly and to process each part. Then, we obtain the required reducts and the processing time for each part. The final reduct is obtained from the intersection of the reducts of each part. The results of our algorithm are excellent when compared with those obtained from other algorithms. Table 3 shows the comparison.

The method is illustrated as follows:

Part1: (Mush_1_room) R1= {3,4,5,10,12,13,14,19,20,22}
Part2: (Mush_2_room) R2= {1,2,3,4,5,9,11,12,13,14,15,19,20,21,22}
Part3: (Mush_3_room) R3= {2,3,4,5,8,9,10,11,12,13,5,19,20,21,22}
Part4: (Mush_4_room) R4= {1,3,4,5,9,10,12,13,14,19,20,21,22}
R1∩ R2∩ R3∩ R4= {3, 4, 5, 12, 13, 19, 20, 22}.

To partition the data, we start the partitioning program randomly. When this program starts, the window in Figure 2 appears.

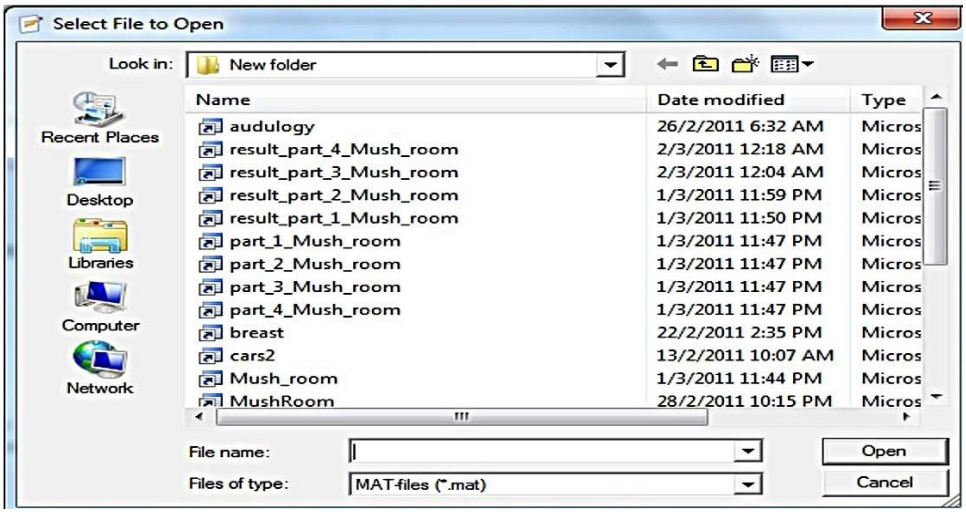

Figure 2. Selecting Data for Partitions

Figure 2: shows how data is randomly segmented into four parts and how every part of data is separately processed in which results of all parts are saved in special files.

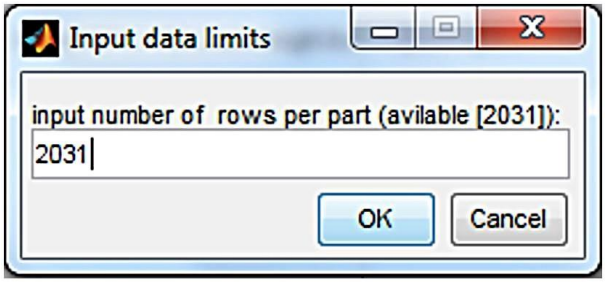

Figure 3. Inputting the Number of Partitions

Using the window in Figure 3, we determine the data that we intend to partition. According to the previous example, the data is partitioned into four parts, with each part containing 2,031 objects.

Thereafter, we run the program according to the data size that should be processed, and using another window similar to the first window, we determine the first part from which we obtain the minimum reduct. The result is saved in a file (see a window in Figure 2) called Result_part1_mush_room.

Following the previous procedures, the data are processed one part after another, and each result is saved in a separate file. Then, we apply the intersection procedure on the result, and the result of the intersection is the minimum reduct for the total data. From this minimum reduct, we determine the accuracy using a regression tree viewer that determines the tree shape that represents the learning algorithm and the accuracy degree. In our example, the tree shape is shown in Figure 4.

7. Algorithm Testing and Comparison (Implementation)

Our algorithm has already been implemented in MATLAB. We compare it with the CAMARDF algorithm (see [23]) by searching for the reductions of six datasets selected from UCI machine learning library [2] to validate our algorithm. The result presented in Table 3 shows that our algorithm is better than CAMARDF especially for large datasets. In addition, we use the minimum discernibility matrix algorithm (see [21]) to search for all reductions to test whether our algorithm has found the minimal reduction.

Table 3. Comparison of our Algorithm with CAMARDF

Data sets	No of object	No of attributes	CAMARDF	Proposed Algorithm	Accuracy of Proposed Algorithm
Zoo	101	17	7	6	96.04%
Soy	47	36	4	5	91.5%
Tik-tac-toc	958	10	9	4	72.96%
Mushroom	8124	23	13	8	99.6
Breast	699	10	8	8	90.99
Cars	1728	7	-	6	92.47%

Figure 4. An illustrative example how to determine the accuracy using a regression tree viewer that determines the tree shape which in turn represents the learning algorithm and the accuracy degree.

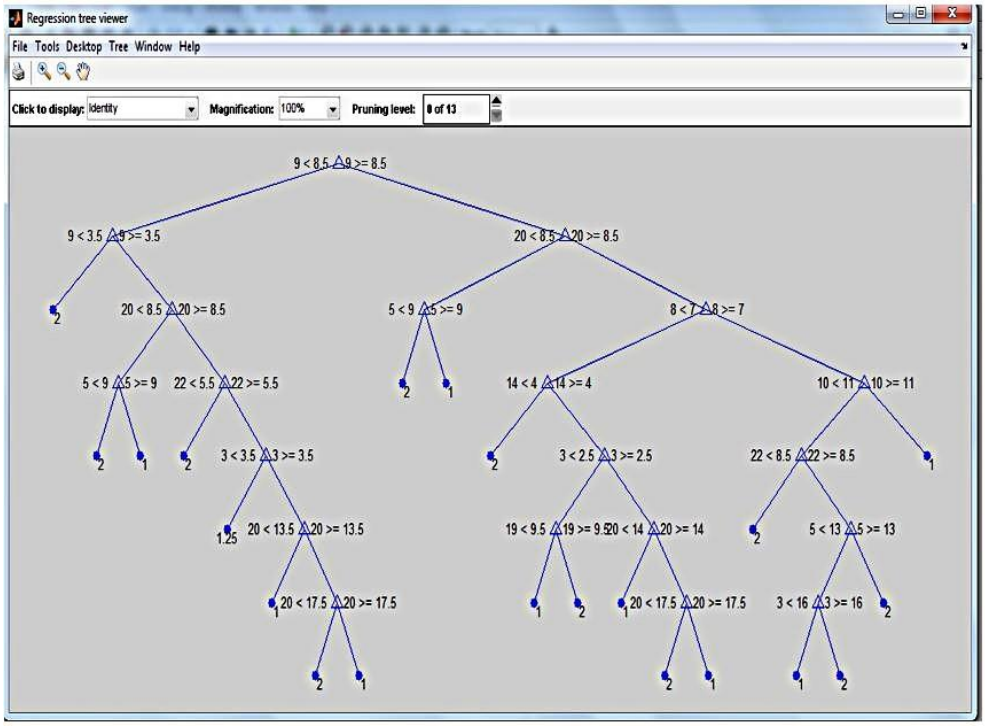

Figure 4. Decision Tree Accuracy (Mush_room)

8. Conclusion

In this paper, we presented a simple approach for feature selection based on data partitioning before processing them and after coding them in a perfect way. This method enables simple data processing. We used a code based on the discernibility matrix and function as well as on rough set theory and greedy heuristics. The main advantages of our approach are that it can select an improved subset of features quickly and effectively from a large database with numerous features, and the selected features do not damage the induction performance because this performance is considered in the evaluation criterion for feature selection.

Our study showed that data partitioning before its processing not only better in accuracy, but also in Minimum Redundancy Feature Selection. in the future, it will be interesting, to study the analysis and explanation of this phenomenon.

References

[1] J. G. Bazan, “A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables”, Rough sets in knowledge discovery, vol. 1, **(1998)**., pp. 321-365.
[2] C. Blake and C. Merz, “UCI repository of machine learning databases", Department of Information and Computer Science, University of California, Irvine, CA, 1998, URL:< http://www. archive. ics. uci. edu/ml, **(2008)**.
[3] M. Boussouf, A hybrid approach to Feature Selectio", Principles of Data Mining and Knowledge Discovery: Springer, **(1998)**, pp. pp. 230-238.
[4] H. Cui, V. Mittal and M. Datar, "Comparative experiments on sentiment classifica-tion for online product reviews", AAAI, **(2006)**, pp. 1265-1270.
[5] J. G. Dy and C. E. Brodley, “Feature selection for unsupervised learning”, The Journal of Machine Learning Research, vol. 5, **(2004)**, pp. 845-889.
[6] E. Gabrilovich and S. Markovitch, "Text categorization with many redundant fea-tures: Using aggressive feature selection to make SVMs competitive with C4. 5", Proceedings of the twenty-first international conference on Machine learn-ing: ACM, **(2004)**, pp. 41
[7] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection”, The Journal of Machine Learning Research, vol. 3, **(2003)**, pp. 1157-1182.
[8] M. Adamczyk, in Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on.,IEEE, **(2014)**, pp. 43-50.
[9] G. H. John, R. Kohavi and K. Pfleger, "Irrelevant features and the subset selection problem", Machine Learning: Proceedings of the Eleventh International Con-ference, **(1994)**, pp. 121-129
[10] R. Kohavi and D. Sommerfield, "Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology", KDD, **(1995)**, pp. 192-197
[11] I. Kononenko, "On biases in estimating multi-valued attributes", IJCAI: Citeseer, **(1995)**, pp. 1034-1040.
[12] M. Kryszkiewicz, “Rough set approach to incomplete information systems”, Infor-mation sciences, vol. 112, no. 1, **(1998)**, pp. 39-49.
[13] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classifica tion and clustering”, Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 4, **(2005)**, pp. 491-502.
[14] A. Moschitti, "A study on optimal parameter tuning for Rocchio text classifier", Springer, **(2003)**.
[15] S. H. Nguyen and H. S. Nguyen, “Pattern extraction from data”, Fundamenta In-formaticae, vol. 34, no. 1-2, **(1998)**, pp. 129-144.
[16] Z. Pawlak, "Rough sets-theoretical aspect of reasoning about data", Kluwer Aca-demic Publishers Dordrecht, **(1991)**.
[17] Z. Pawlak, and A. Skowron, “Rudiments of rough sets”, Information sciences, vol. 177, no. 1, **(2007)**,pp. 3-27.
[18] P. Piñero, L. Arco, M. M. García, Y. Caballero, R. Yzquierdo and A. Morales, "Two new metrics for feature selection in pattern recognition", Progress in Pat-tern Recognition, Speech and Image Analysis: Springer, **(2003)**, pp. 488-497.
[19] E. Riloff, S. Patwardhan and J. Wiebe, "Feature subsumption for opinion analysis", Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing: Association for Computational Linguistics, **(2006)**, pp. 440-448.
[20] Y. Zheng and C.K. Kwoh, "A feature subset selection method based on high-dimensional mutual information", Entropy, vol. 13, no. 4, **(2011)**, pp. 860-901.
[21] Y. Yao and Y. Zhao, “Discernibility matrix simplification for constructing attribute reducts”, Information sciences, vol. 179, no. 7, **(2009)**, pp. 867-882.
[22] N. Zhong, J. Dong and S. Ohsuga, “Using rough sets with heuristics for feature selection”, Journal of intelligent information systems, vol. 16, no. 3, **(2001)**, pp. 199-214.
[23] J. Zhou, D. Miao, Q. Feng and L. Sun, "Research on complete algorithms for minimal attribute reduction", Rough Sets and Knowledge Technology: Springer, **(2009)**. p.^pp. 152-159

