Journal of Computer Science and Cybernetics, V.38, N.3 (2022), 277–292
DOI no 10.15625/1813-9663/38/3/17450

AN EFFECTIVE ALGORITHM FOR COMPUTING REDUCTS IN DECISION TABLES

DO SI TRUONG*, LAM THANH HIEN, NGUYEN THANH TUNG

Lac Hong University, 10 Huynh Van Nghe Street, Buu Long Ward, Bien Hoa City,
Dong Nai Province, Viet Nam

Abstract. Attribute reduction is one important part researched in rough set theory. A reduct from a decision table is a minimal subset of the conditional attributes which provide the same information for classification purposes as the entire set of available attributes. The classification task for the high dimensional decision table could be solved faster if a reduct, instead of the original whole set of attributes, is used. In this paper, we propose a reduct computing algorithm using attribute clustering. The proposed algorithm works in three main stages. In the first stage, irrelevant attributes are eliminated. In the second stage relevant attributes are divided into appropriately selected number of clusters by Partitioning Around Medoids (PAM) clustering method integrated with a special metric in attribute space which is the normalized variation of information. In the third stage, the representative attribute from each cluster is selected that is the most class-related. The selected attributes form the approximate reduct. The proposed algorithm is implemented and experimented. The experimental results show that the proposed algorithm is capable of computing approximate reduct with small size and high classification accuracy, when the number of clusters used to group the attributes is appropriately selected.

Keywords. Feature selection, attribute reduction, attribute clustering, partitioning around medoids clustering, normalized variation of information, rough set.

1. INTRODUCTION

Due to the rapid development in today's technology, the dimensionality of dataset becomes larger and larger. In most of applications such as gene data, text categorization, image retrieval and information retrieval, we often confront with the datasets involving huge numbers of features (or attributes). This may lead to the fact that the traditional mining or learning algorithms become slow and cannot process information effectively. One of the most feasible technique to cope with this problem is feature selection. Generally, feature selection can be viewed as the process of selecting a subset from the original set of features, removing as many irrelevant and redundant features as possible to improve the quality of data and reduce time and space complexity for analysis [1,2]. This is because firstly irrelevant features do not contribute to predictive accuracy. Secondly redundant features do not redound to getting a better predictor because they provide the most information which is already present

*Corresponding author.
E-mail addresses: truongds@lhu.edu.vn (D.S.Truong); lthien@lhu.edu.vn (L.T.Hien); nttung@lhu.edu.vn (N.T.Tung).

in other feature [1–3]. Feature selection is considered as NP-hard problem, since the number of all subsets 2^N grows exponentially with the number of features N. Several approximation algorithms have been proposed to find the near best feature subset in a reasonable time. Comprehensive surveys of feature selection algorithms are presented in [1,3].

When dealing with high dimensional data (datasets with hundreds or thousands of features), many feature selection algorithms can successfully remove irrelevant features but fail to pull redundant ones out [4,5]. To overcome this problem, in the last decades some feature selection algorithms using feature clustering were proposed in both supervised and unsupervised context [5–10]. Note that feature clustering is different from object clustering, here we are doing clustering for features rather than for objects. Feature clustering groups features into clusters so that the features within the same cluster are expected to possess high similarity, but within different clusters possess low similarity.

Clustering based feature selection algorithm follows the straightforward idea. It divides the initial feature space into a set of groups called clusters. Generally, correlation measures are used as clustering algorithm metrics which make features of the same group considered as redundant. This leads to the selection of one feature to represent each cluster. The resulting feature subset is considered to be relevant and non redundant [10]. However, for this type of approach, there are two core issues that need to be carefully considered, namely the choice of a similarity measurement function and a clustering method to use. The similarity function measures the similarity between two features in the feature space; clustering method collects features into groups using the selected similarity function.

Recent studies have demonstrated that the algorithms of selecting features through clustering have very important advantages. They can outperform the traditional feature selection algorithms by reducing the redundancy, reaching a high accuracy and, in some cases, reducing the calculation time. Besides, they also help users better understand the structure of the dataset to be analyzed and the relative importance between features [5–7,9,10].

Rough set theory proposed by [11], is a powerful mathematical tool for dealing with vague, imprecise, incomplete, and uncertain data. This theory has been successfully applied in different research fields such as machine learning, expert system, pattern recognition, and knowledge discovery in databases [11,12]. Feature selection is one important part researched in rough set theory. In rough set theory, the process of feature subset selection in a decision table is viewed as reduct computation process. A reduct is a minimal subset of the conditional attribute set which provide the same information for classification purposes as the entire set of available attributes. The classification task for the high dimensional dataset could be solved faster if a minimal reduct, instead of the original whole set of attributes, is used. However, computing a minimal reducts is an NP-hard problem [13]. Therefore, several studies on computing approximate reducts have been carried out. An approximate reduct is a minimal reduct with acceptable errors, but can be found in much shorter time relative to an exact minimal reduct. Many approaches for computing approximate reducts were proposed [14–17].

As mentioned above, clustering based feature selection algorithms have many important advantages. In recent years, some approximate reduct computing algorithms using attribute clustering have also been proposed by researchers such as Hong et al. [18–22] , Janusz and Slezak [23,24], Pacheco et al. [25]. In [18,21], the authors built a similarity measure for a pair of attributes based on the relative dependency. Using this similarity measure, an algorithm called Most Neighbors First (MNF) was also proposed to cluster the attributes into a fixed number of groups. The process starts with randomly selecting k representative attributes as cluster centers, then the dissimilarity measure is computed between the non-representative attributes. The non-representative attributes are allocated to their nearest centers and in the end of the process, the center is updated with the attribute with more neighbours in its surroundings. One of the big MNF deficiencies is that the convergence of the algorithm is not assured, in consequence it has to be executed several times to find tendencies or patterns in the results. Inspired by the MNF algorithm, in [25], the authors presented an approximate reduct computing technique for fault diagnosis in spur gears.

Attribute clustering is also considered NP-hard procedure, as the majority of feature selection algorithms, due to the similarity degree must be computed for all the pairs of attributes in order to arrange the clusters. Furthermore, genetic algorithms have become increasingly important for researchers in solving difficult problems since they could provide feasible solutions in a limited amount of time [26]. In [19, 20], the authors thus proposed a GA-based clustering method for attribute clustering and approximate reduct computing. Hong et al. [22] continued to improve the performance of the GA-based attribute clustering process based on the grouping genetic algorithm (GGA). In [23], the authors investigated methods for attribute clustering and their possible applications to a task of computation of optimal reducts from decision tables with a large number of attributes. They also proposed a discernibility-based attribute similarity measure, which is useful for identifying groups of attributes. In [24], the authors continued the research described in [23], and extended this work by an in depth investigation of the selected gene-clustering results.

Although clustering-based attribute reduction algorithms have received much attention in recent times, the number of publications is still relatively limited. In this paper, we propose a clustering based attribute reduction algorithm for high dimensional decision table. The proposed algorithm works in three main stages. In the first stage, irrelevant attributes are eliminated. In the second stage, relevant attributes are divided into a desired number of clusters by using Partitioning Around Medoids (PAM) clustering method integrated with a special metric in attribute space which is the normalized variation of information. In the third stage, the most representative attribute that is the most class-related is selected from each cluster to form a reduct. The proposed algorithm is implemented and experimented. The experimental results show that the proposed algorithm is capable of computing approximate reduct with small size and high classification accuracy when the number of clusters used to group the attributes is appropriately selected.

The rest of the paper is organized as follows. Section 2 reviews the theory used by our proposal. Section 3 presents the proposed attribute reduction algorithm. Section 4 describes and discusses the experimental results. Finally, Section 5 holds the conclusions and directions for further research.

2. PRELIMINARIES

In this section, we briefly review the theoretical guidelines that support our proposal. The concept of reducts in information system is first introduced, followed by the concept of Normalized Variation of Information (NVI), which is a special distance measure on feature space. Next, the famous clustering algorithm, k-medoids, is described. The contents are based on [11, 27], and [28].

2.1. Reducts in decision table

In many information processing systems, a set of objects are typically represented by their values on a finite set of attributes (features). Such information may be conveniently described in a tabular form. Each column corresponds to an attribute and each row corresponds to an object. In rough set theory, such a table is often called an information system.

Formally, an information system is a pair $IS = (U, A)$, where U is a non-empty finite set of objects, A is a nonempty finite set of attributes, and for every $a \in A$ there is a mapping $a : U \to V_a$, where V_a denotes the domain of a.

In the rest of this article, unless otherwise stated, we assume that all features in a given information system are categorical, i.e., that they have a finite and unordered domain.

In an information system $IS = (U, A)$, if some of the attributes are interpreted as outcomes of classification, then this information system can also be defined as a decision table by $DT = (U, C \cup \{d\})$, where $C \cup \{d\} = A$, $d \notin C$, C is called the condition attribute set, while d is called the decision attribute [11].

Given an information system $IS = (U, A)$, with any subset of attributes $B \subseteq A$, there is a binary indiscernibility relation $IND\,(B)$ as follows

$$IND\,(B) = \{(x, y) \in U \times U | \forall a \in B,\ a\,(x) = a\,(y)\}\,. \tag{2.1}$$

Obviously, $IND\,(B)$ is an equivalence relation, it partitions U into disjoint blocks (or equivalence classes), where two objects belong to the same block if they share the same value for B. Let $U/IND\,(B)$ or just U/B denote the family of all equivalence classes of $IND\,(B)$. For every object $x \in U$, let $[x]_B$ denote the equivalence class of relation $IND\,(B)$ that contains element x, called the equivalence class of x under relation $IND\,(B)$.

In a given information system $IS = (U, A)$, let $X \subseteq U$, $B \subseteq A$. One can characterize X by a pair of lower and upper approximation sets which are defined as follows.

$$\underline{B}(X) = \{x \in U | [x]_B \ \subseteq X = \bigcup_{X_i \in U/B \wedge X_i \ \subseteq X} X_i\}, \tag{2.2}$$

$$\overline{B}(X) = \{x \in U | [x]_B \cap X \neq \emptyset\} = \bigcup_{X_i \in U/B \wedge X_i \cap X \neq \emptyset} X_i. \tag{2.3}$$

The lower approximation set $\underline{B}(X)$ contains those objects in U that certainly belong to X, whereas the upper approximation set $\overline{B}(X)$ contains those objects in U that possibly belong to X. Obviously, there is $\underline{B}(X) \subseteq X \subseteq \overline{B}(X)$. A set X is said to be definable if $\underline{B}(X) = \overline{B}(X)$, otherwise, X is said to be rough. The difference between $\overline{B}(X)$ and $\underline{B}(X)$ is called the B-boundary region of X, which is denoted as $BN_B\,(X) = \overline{B}(X) - \underline{B}(X)$.

For a decision table, the most important task is attribute reduction, which means selecting or reserving those condition attributes that provide the same information for classification purposes as the entire set of available attributes. Such subsets are called reducts.

Let $DT = (U, C \cup \{d\})$ be a decision table, $B \subseteq A$. The positive region of the d with respect to B, denoted by $POS_B\,(d)$, is defined as follows

$$POS_B(d) = \bigcup_{X \in U/\{d\}} \underline{B}(X). \tag{2.4}$$

The positive region $POS_B\,(d)$ contains those objects that can be certainly classified to some decision classes by checking all attributes in B. If $POS_C(d) = U$, then the decision table DT is consistent, otherwise it is inconsistent.

Let $DT = (U, C \cup \{d\})$ be a decision table. A subset $R \subseteq C$ is called a (relative) reduct of DT if B is a minimal subset of condition attributes such that $POS_R(d) = POS_C(d)$.

In general, there are plural reducts in a decision table. Over the years, many methods for computing reducts have been proposed and researched in the rough set community [13,29,30]. Unfortunately, it has been proved that computing all reducts or computing an optimal reduct (a reduct with the least number of attributes) is an NP-hard problem [13]. In practice, most of the time only one reduct is required as typically only one subset of features is used to reduce a decision table. For this reason, many approaches in the literature [6, 15–17] adopt a forward greedy algorithm to find a approximate reduct on the basis of various significance measures of attributes. Two most widely applied attribute significance measures are defined based on the degree of Pawlak's dependency defined below and on Shannon conditional entropy defined in Subsection 2.3.

Given a decision table $DT = (U, C \cup \{d\})$, for any $B \subset C$, Pawlak defines the dependency degree of D on B in DT as follows

$$\gamma_B(d) = \frac{POS_B(d)}{|U|}. \tag{2.5}$$

Obviously, there is $0 \leq \gamma_B(d) \leq 1$. If $\gamma_B(d) = 1$, then we say that D depends totally on B, and if $0 < \gamma_B(d) < 1$, then we say that d depends on B in a degree $\gamma_B(d)$. If $\gamma_B(d) = 0$, then we say that d does not depend on B.

Given a decision table $DT = (U, C \cup \{d\})$, let $B \subseteq C$. For any $a \in B$, the significance of attribute a with respect to B and d in DT is defined as follows

$$SIG^{\gamma}(a, B, \{d\}) = \gamma_B(d) - \gamma_{B-\{a\}}(d). \tag{2.6}$$

$SIG^{\gamma}(a, B, \{d\})$ is the change of the coeficicient $\gamma_B(d)$ when removing the attribute a from B.

The QuickReduct algorithm [31] listed below is a typical algorithm that uses a greedy search strategy and the above attribute significance measure to find a approximate reduct. QuickReduct works as follows.

Algorithm 1. QuickReduct algorithm for computing the relative reduct.

```
inputs: Decision table $DT = (U, C \cup \{d\}, V, f)$.
Output: One relative reduct of $DT$.
Begin
    $red \leftarrow \{\}$;
    do
        $T \leftarrow red$;
        foreach $a \in C - red$
            if $\gamma_{red \cup \{a\}}(d) > \gamma_T(d)$ ;
              $T \leftarrow red \cup \{a\}$ ;
              $red \leftarrow T$;
        until $\gamma_{red}(d) = \gamma_C(d)$;
        return $red$;
End;
```

Wang et al. [32] developed the conditional entropy-based algorithm CEBARKNC for attributes reduction. The structure of the CEBARKNC algorithm is similar to the QuickReduct algorithm except that the conditional entropy based attribute significance measure is used, (see Subsection 2.3 for conditional entropy).

2.2. Normalized variation of information

The central idea of our work is to introduce an algorithm for attribute reduction that uses attribute clustering. So we need a special metric to measure the distance between attributes. Such a metric would be the normalized variation of information presented below.

Let $IS = (U, A)$ be an information system, attribute $X \in A$. The information system IS can be viewed as a statistical population and X is a discrete random variable. Suppose $V_X = \{x_1, x_2, \ldots, x_m\}$, $U/IND\,(X) = \{X_1, X_2, \ldots, X_m\}$. Then the probability distribution of X can be determined by

$$P(X = x_i) = P(x_i) = |X_i|/|U|,\ i = 1, ..., m. \tag{2.7}$$

where $|\ .\ |$ denotes the cardinality of a set.

Other related probability distributions can be similarly defined. In particular, $P\,(X, Y)$ is the joint probability distribution of X and Y, and $P(X|Y)$ is the conditional probability distribution of X given Y. Let $U/IND\,(X) = \{X_1, X_2, \ldots, X_m\}$, and $U/IND\,(Y) = \{Y_1, Y_2, \ldots, Y_n\}$, then

$$P\,(X = x_i, Y = y_j) = P\,(x_i, y_j) = |X_i \cap Y_j|\,/\,|U|\,,$$

$$P\,(X = x_i|Y = y_j) = P\,(x_i|y_j) = |X_i \cap Y_j|\,/\,|Y_j|\,,$$

with $i = 1, \ldots, m,\ j = 1, \ldots, n$.

For a given attribute X, (Shannon) entropy of X is an expression [27]:

$$H(X) = -\sum_{i=1}^{m} P(X = x_i) \log_2 P(X = x_i), \tag{2.8}$$

and by the convention $0\log_2 0 = 0$.

For an attribute X, its entropy $H\,(X)$ is related to the deviation of the probability distribution of X from the uniform distribution. A lower entropy suggests that the distribution is uneven and consequently one may have a better prediction using the value of X. The attribute entropy $H\,(X)$ serves as a measure of uncertainty or un-structuredness. An attribute with a larger domain normally divides the database into more smaller classes than an attribute with a smaller domain, and hence may have a higher entropy value. In fact, the maximum value of attribute entropy is $\log |V_X|$, which depends on the size of V_X. On the other hand, an attribute with smaller domain, i.e., a lower entropy value, usually divides the database into a few larger classes.

The notion of entropy may be generalized over two and more attributes, for instance [27]

$$H\,(X, Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} P\,(X = x_i, Y = y_j) \log_2 P\,(X = x_i, Y = y_j)\,. \tag{2.9}$$

The conditional entropy $H\left(X|Y\right)$ of X given Y is defined as [27]

$$H\left(X|Y\right)=-\sum_{j=1}^{n}P\left(Y=y_j\right)\sum_{i=1}^{m}P\left(X=x_i|Y=y_j\right)\log_2 P\left(X=x_i|Y=y_j\right). \tag{2.10}$$

Conditional entropy $H(X|Y)$ quantifies the remaining entropy (i.e., uncertainty) of an attribute X given that the value of another attribute Y is known. Applying formulas (2.8), (2.9), and (2.10) we have

$$H\left(X|Y\right)=H(X,Y)-H(Y). \tag{2.11}$$

The mutual information between the two attributes X and Y is defined as [27]

$$I\left(X;Y\right)=H\left(X\right)-H\left(X|Y\right)=H\left(Y\right)-H\left(Y|X\right). \tag{2.12}$$

Mutual information $I\left(X;Y\right)$ is non-negative and symmetric, i.e., $I\left(X;Y\right)\geq 0$ and $I\left(X;Y\right)=I\left(Y;X\right)$. It measures the information that X and Y share, and it tells us how much the knowledge on one of the two attributes reduces uncertainty about the other one.

Symmetric uncertainty of attributes X and Y is defined as [27]

$$SU(X,Y)=2.\frac{I(X;Y)}{H(X)+H(Y)}. \tag{2.13}$$

Symmetric uncertainty is a measure that allows to quantify the mutual dependence of two attributes. The numerator is mutual information. This uncertainty has been normalized by the total uncertainty on the attributes, given by the sum of the entropies $H\left(X\right)$ and $H\left(Y\right)$. Therefore, its values are in the range [0,1]. A value of 1 indicates that knowledge of the value of either one completely predicts the value of the other and the value 0 reveals that X and $Y$are independent.

The normalized variation of information between X and Y is defined by [27]

$$NVI(X,Y)=1-\frac{I(X;Y)}{H(X,Y)}=\frac{H(X|Y)+H(Y|X)}{H(X,Y)}. \tag{2.14}$$

$NVI\left(X,Y\right)$ is a metric on the space of attributes, that is, for any attributes X,Y, and Z it satisfies

(i) $NVI\left(X,Y\right)\geq 0$ and the equality holds iff $X=Y$,

(ii) $NVI\left(X,Y\right)=NVI(Y,X)$,

(iii) $NVI\left(X,Y\right)+NVI\left(Y,Z\right)\geq NVI(X,Z)$.

Values of $NVI\left(X,Y\right)$ are in the range [0,1]. $NVI\left(X,Y\right)$ is also a universal metric in that if any other distance measure places X and Y close-by, then the NVI will also judge them close.

Although the entropy-based measure handles categorical or discrete attributes, they can deal with continuous features as well if the values are discretized properly in advance [27].

2.3. k-medoids clustering algorithm

The k-medoids algorithm [28] is a clustering approach related to k-means clustering for partitioning a dataset into k groups or clusters. In k-medoids clustering, each cluster is represented by one of the data point in the cluster. These points are named cluster medoids. The term medoid refers to an object within a cluster for which average distance between it and all other members of the cluster is minimal. It corresponds to the most centrally located point in the cluster. These objects (one per cluster) can be considered as a representative example of the members of that cluster which may be useful in some situations. Recall that, in k-means clustering, the center of a given cluster is calculated as the mean value of all data points in the cluster. *The k-medoids algorithm can work with any distance matrix and is less affected by outliers than k-means* because it uses medoids as cluster centers instead of means [28].

The most common k-medoids clustering methods is the PAM algorithm (Partitioning Around Medoids) [28]. In summary, PAM algorithm proceeds in two phases as follows.

Build phase

1. Randomly select k objects to become the medoids;

2. Assign every object to its closest medoid, then calculate the total cost E for the resulting cluster configuration by using formula

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - m_i|. \tag{2.15}$$

where x is an object in cluster C_i, m_i is the current medoid of C_i, the absolute value $|x - m_i|$ means the distance between x and m_i;

Swap phase

```
3. For each medoid $m$
       For each non-medoid data point $x$
           swap $m$ and $x$; compute the total cost $E'$ of the resulting cluster configuration;
4. if $E' < E$, $m$ is replaced by $x$;
5. Repeat Steps 3-4 until there is no change in the medoids.
```

The complexity of PAM for each iteration (step 3-4) is $O\left(k\left(n-k\right)^2\right)$ where n is the number of objects in dataset, k is number of clusters. Moreover, the PAM algorithm complexity to recalculate the entire cost function is $O\left(n^2k^2\right)$ [28]. Therefore, the complexity of the k-medoids approach is in general higher than the k-means approach, but the former can guarantee that all centers of obtained clusters are objects themselves. This feature is important to us, since the attributes are not only clustered but also the representative attribute of each cluster has to be found. Note that, the goal of this paper is to select attributes using clustering. An attribute clustering method based on k-medoids, thus, can help us achieve this purpose.

PAM clustering algorithm is implemented in **R** programming language. To compute PAM, we can use the pam() function in the "cluster" package [33]. For PAM algorithm, a user has to specify k, the number of clusters to find. There is also an enhanced version of pam(), function pamk() in R package "fpc". pamk() does not require a user to choose k. Instead, it performs a partitioning around medoids clustering with the number of clusters estimated by optimum average silhouette width method, described in [34].

Briefly, the average silhouette approach measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering. Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximize the average silhouette over a range of possible.

3. PROPOSED METHOD

This section introduces our proposed algorithm for computing an approximate reduct in a decision table. It is called ACBRC (attribute clustering based reduct computing).

In a decision table, irrelevant attributes that do not contribute to predicting accuracy and redundant attributes do not redound to better prediction because most of the information they provide is already in the other attribute. Irrelevant attributes, along with redundant attributes, severely affect the accuracy of the learning machines [35]. Therefore, attribute reduction algorithm should be able to identify and remove as much of the irrelevant and redundant information as possible. Furthermore, good attribute subsets must contain attributes that are close to the decision attribute, but not close to each other. Keeping these in mind, we propose ACBRC, an attribute clustering based reduct computing algorithm which can efficiently and effectively deal with both irrelevant and redundant attributes in a decision table, and give a good approximate reduct.

In order to more precisely introduce the algorithm, we firstly present our definitions.

Definition 3.1. Let $DT = (U, C \cup \{d\})$ be a decision table. Attribute clustering in DT can be defined as the partitioning of the set C of conditional attributes into a collection $C_X = \{C_1, C_2, \ldots, C_k\}$ of mutually disjoint subsets C_i of C, such that $C_1 \cup C_2 \cup \ldots \cup C_k = C$, $C_i \neq \emptyset$, and $C_i \cap C_j = \emptyset$, for $i \neq j$.

Definition 3.2. Let $DT = (U, C \cup \{d\})$ be a decision table, $C \cup \{d\}$ is the full set of attributes. The distance between any pair of attributes X_i and X_j $(X_i, X_j \in C \cup \{d\}, i \neq j)$ is measured by $NVI\left(X_i, X_j\right)$, defined as in (2.14).

Note that for any $X_i \in C$ we have $0 \leq NVI\left(X_i, d\right) \leq 1$.

Definition 3.3. The irrelevance between the condition attribute $X_i \in C$ and the decision attribute d is measured by the distance value $NVI\left(X_i, d\right)$. The greater the value $NVI\left(X_i, d\right)$ is, the lower the relevance between them. If $NVI\left(X_i, d\right)$ is greater than a threshold $\delta = 0.98$, we say that X_i is an irrelevant attribute; otherwise X_i is relevant one.

Definition 3.4. Let G be a cluster of attributes. A feature $X^R \in G$ is a representative attribute of the cluster if and only if

$$X^R = \arg\min_{X \in G} NVI\left(X, d\right).$$

This means X^R is the strongest relevant attribute and can act as a relevant attribute for all attributes in the cluster G.

Using the above definitions, ACBRC algorithm is the process consisting of the two connected parts: irrelevant attribute elimination and redundant attribute removal. The former obtains relevant attributes by eliminating irrelevant ones, the latter removes redundant attributes from relevant ones via choosing representatives from different attribute clusters, and thus produces the final subset of attributes. Framework of ACBRC is shown in Figure 1.

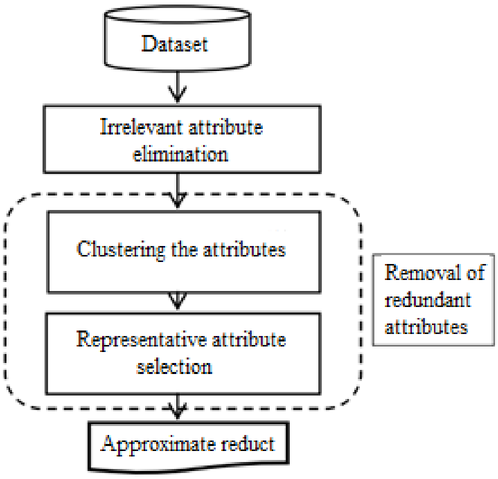
Dataset
Irrelevant attribute elimination
Clustering the attributes
Representative attribute selection
Removal of redundant attributes
Approximate reduct

Figure 1. Framework of the proposed Reduct Computing algorithm ACBRC

ACBRC algorithm consists of three stages.

(1) First, irrelevant attributes are eliminated. For this purpose, the distance $NVI\,(X, d)$ is measured between each attribute X and the decision attribute d. We assume that the greater an attribute has irrelevance value is, the lower its ability to distinguish between classes. Here, the attribute with irrelevance greater than 0.98 will be removed from the initial attribute set.

(2) Clustering the relevant attributes using function pamk() in R package "fpc", integrated with metric NVI

pamk() is an enhanced version of pam(), which can work with any distance matrix and does not require a user to choose the number of clusters k. Instead, it performs a PAM clustering with the number of clusters estimated by optimum average silhouette width method, described in [34].

(3) Finally, selecting from each cluster the attribute which has the strongest decision-relevance. This attribute can act as a representative attribute for all attributes in the cluster. Once a attribute is selected, attributes belonging to the same cluster are removed. The selected attributes form the approximate reduce.

The main steps of the ACBRC algorithm are as follows.

Algorithm 2 The ACBRC algorithm

```
inputs: The given decision table $DT = (U, C \cup \{d\})$, $\delta = 0.98$ - the irrelevance threshold.
output: Red – approximate attribute reduct.
step 1. Irrelevant attributes elimination.
    For each $X \in C$ compute irrelevance $= NVI\,(X, d)$. If irrelevance $> \delta$ then
    $C^R = C \backslash \{X\}$.
step 2. calculate the distance matrix $NVI$ for all attribute pairs.
    For each attribute pair $X_i$ and $X_j$ in $C^R$ compute
            $NVI\,[i, j] = NVI\,(X_i, X_j)$ (equation (2.14).
step 3. Using pamk() fuction in R package "fpc" to cluster the attributes in $C^R$.
step 4. for each cluster G do $X^R = \arg\min_{X \in G} NVI\,(X, d)\,.\ Red = Red \cup X^R$.
```

4. EXPERIMENTAL RESULTS

The proposed ACBRC attribute reduction algorithm was implemented in R programming language and on a personal computer with Pentium dual core 2.70 GHz CPU and 2.00 GB RAM.

Experimental computations were carried out on 5 benchmark datasets obtained from UCI repository [36]. The characteristics of these datasets are shown in Table 1. The first two columns show the names and abbreviations of datasets, the next two columns show the number of samples and attributes, and the last column shows the number of class labels. All attributes of the selected datasets are categorical. Thus, discretization is not necessary.

Table 1. Descriptions of datasets in the experiment

Datasets	Abbreviations	Nr. of instances	Nr. of condition attributes	Nr. of classes
Chess	Chess	3196	36	2
Mushroom	Mushroom	8124	22	7
Soybean (small)	Soybean	47	35	4
Lung-cancer	**Lung**	32	56	3
Votes	Votes	435	16	2

To evaluate the performance of our proposed ACBRC algorithm, we compare it with QuickReduct and CEBARKNC algorithms, in terms of the number of selected attributes, and the classification performance.

For comparing the classification performance of ACBRC, QuickReduct and CEBARKNC, we used C5.0 and Native Bayes, which are two popular classification algorithms and widely applied in various research fields.

In order to make the best use of the data and obtain stable results, a 3-trials 10-fold cross-validation strategy is used. That is, for each dataset, each attribute reduction algorithm and each classification algorithm, the 10-fold cross-validation is repeated 3 times, with each time, the order of the instances of the dataset is randomized. Randomizing the order of the instances can help diminish the order effects. For each classification algorithm, we obtain 3-trials 10-fold classification accuracy for each attribute reduction algorithm and each dataset. Averaging these accuracies, we obtain mean accuracy of each classification algorithm under each attribute reduction algorithm and each dataset.

A. Comparison of number of selected attributes for three attribute reduction algorithms

Table 2 shows the attributes selected by three attribute reduction algorithms - ACBRC, QuickReduct and CEBARKNC when they are applied for each dataset in Table 1.

From Table 2, we can see that all three algorithms achieve significant reduction of dimensionality by selecting only a small portion of the original attributes. ACBRC generally obtains the best proportion of selected attributes.

From Table 3, we see that the execution time of the three algorithms depends on the characteristics of each dataset. In general, the execution time of the ACBRC algorithm is slightly larger than that of the QuickReduct and CEBARKNC algorithms, but the execution time of ACBRC is acceptable..

Table 2. Selected attributes using three algorithms

Datasets	ACBRC	QuickReduct	CEBARKNC
Chess	7 29 32 8 10 18 33 14 16 21	21 10 29 14 28 1 15 16 6 33 7 35 18 34 11 5 17 23 36 26 20 30 4 24 12 27 25 31 3 9 13	21 10 33 32 6 35 15 1 34 7 16 23 17 4 2 30 5 27 3 9 20 25 31 12 13 24 18 28 26 36
Mushroom	13 20 5 9	15 6 22 11 12 5 1	5 20 22 21
Soybean	22 21	22 1	22 4
Lung	9 48	1 42 7 4	9 43 3 4
Votes	4 12	1 9 14 4 11 16 13 3 2 6	4 11 3 13 16 2 9 15 1

Table 3. Execution time of the proposed algorithms (in sec.)

Datasets	ACBRC	QuickReduct	CEBARKNC
Chess	18.52	28.41	12.82
Mushroom	1.14	2.29	1.11
Soybean	0.64	0.12	0.36
Lung	0.84	0.31	0.44
Votes	0.64	0.73	0.53

B. Evaluation of the classification performance of the ACBRC attribute reduction algorithm

Table 4 shows 95% confidence intervals of 3-trials 10-fold classification accuracy of two classifiers on 5 datasets without attribute reduction.

Table 4: Classification accuracy without attribute reduction

Datasets	C5.0	Bayes
Chess	0.9928 ± 0.0024	0.87868 ± 0.0126
Mushroom	1	0.94088 ± 0.0057
Soybean	0.975 ± 0.049	1
Lung	0.7667 ± 0.1701	0.56667 ± 0.2396
Votes	0.9674 ± 0.0139	0.90465 ± 0.0349

Table 5 shows 95% confidence intervals of 3-trials 10-fold classification accuracy of two classifiers on 5 datasets after ACBRC attribute reduction algorithm is used.

Generally, for all five datasets, the classification accuracy of attributes selected by ACBRC is greater than the classification accuracy of the original attributes.

C. Comparison of classification accuracy for three attribute reduction Algorithms

Table 6 and Table 7 show 95% confidence intervals of classification accuracy by 3-trials 10-fold cross-validation when C5.0 and Naïve Bayes classifiers are used for the datasets with attributes selected by ACBRC, QuickReduct, and CEBARKNC.

From Table 6, we see that for datasets “Soybean”, “Lung”, and “Votes”, C5.0 classification accuracy with attributes selected by ACBRC is greater than the classification accuracy with attributes selected by QuickReduct and CEBARKNC. And for two other datasets, the classification result after attribute reduction by ACBRC is comparable with the results after attribute reduction by QuickReduct and CEBARKNC.

From Table 7, we see that for all five datasets, the Bayes classification accuracy with

Table 5: Classification accuracy using attributes selected by ACBRC

Datasets	C5.0	Bayes
Chess	0.9928 ±0.0022	0.8906 ± 0.0129
Mushroom	1	0.9473 ± 0.0031
Soybean	1	1
Lung	0.8 ± 0.1996	0.6 ± 0.2134
Votes	0.9674 ±0.0217	0.9581 ± 0.0164

Table 6: C5.0 classification accuracy using different attribute reduction algorithms

Datasets	ACBRC	QuickReduct	CEBARKNC
Chess	0.9928 ±0.0022	0.9931 ± 0.0024	0.9937 ± 0.0035
Mushroom	1	1	1
Soybean	1	0.975 ± 0.049	0.975 ± 0.049
Lung	0.8 ± 0.1996	0.8 ±0.1445	0.7667 ± 0.1701
Votes	0.9674 ±0.0217	0.9651 ± 0.014	0.9558 ± 0.0126

selected attributes by ACBRC is greater than the classification accuracy with attributes selected by QuickReduct and CEBARKNC.

5. CONCLUSION

In this paper, we have proposed a clustering based attribute reduction algorithm for high dimensional decision table. The proposed algorithm, called ACBRC, consists of three stages: (1) removing irrelevant attributes, (2) clustering the relevant attributes into appropriately selected number of clusters using Partitioning Around Medoids (PAM) clustering method integrated with Normalized Variation of Information as distance measure, and (3) selecting from each cluster the representative attribute which has the strongest relevance. Once an attribute is selected from a cluster, attributes belonging to the same cluster are removed and thus the dimensionality of decision table is drastically reduced. Only the selected attributes form the approximate reduct.

Experimental computations were carried out on five benchmark datasets obtained from UCI repository. To evaluate the performance of the proposed attribute reduction algorithm, we compare it with QuickReduct and CEBARKNC algorithms, in terms of the number of selected attributes, and the classification performance. Generally, ACBRC obtained the best proportion of selected attributes, the best classification accuracy for C5.0, and Naive Bayes. The classification accuracy after attribute reduction by ACBRC even outperforms the classification accuracy using whole dataset in some cases. These experimental results

Table 7: Bayes classification accuracry using different feature selection methods

Datasets	ACBRC	QuickReduct	CEBARKNC
Chess	0.8906 ± 0.0072	0.8881 ±0.015	0.8947 ± 0.0061
Mushroom	0.9473 ±0.0036	0.982 ±0.0027	0.9807 ± 0.0031
Soybean	1	0.875 ±0.1096	1
Lung	0.5334 ±0.2425	0.4667 ± 0.2789	0.4667 ± 0.2425
Votes	0.9581 ±0.0164	0.9279 ± 0.023	0.9302 ± 0.0192

show that ACBRC is a promising algorithm for attribute reduction.

For the future work, we will attempt to apply the proposed attribute reduction algorithm to some real application domains with high dimensional datasets like DNA analysis and text categorization.

REFERENCES

[1] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2013.11.024

[2] E. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[3] H. Liu and H. Motoda, *Computational Methods of Feature Selection.* Chapman and Hall/CRC Press, 2007.

[4] K. Kira and L. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proceedings of Ninth National Conference on Artificial Intelligence*, 1992, pp. 129–134. [Online]. Available: https://www.aaai.org/Papers/AAAI/1992/AAAI92-020.pdf

[5] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 1–14, 2013.

[6] M. Alimoussa, A. Porebski, N. Vandenbroucke, R. Thami, and S. E. Fkihi, "Clustering-based sequential feature selection approach for high dimensional data classification," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, 2021, pp. 122–132. [Online]. Available: https://www.scitepress.org/Papers/2021/102595/102595.pdf

[7] S. Chormunge and S. Jena, "Correlation-based feature selection with clustering for high dimensional data," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 542–549, 2018. [Online]. Available: https://doi.org/10.1016/j.jesit.2017.06.004

[8] D. Harris and A. Niekerk, "Feature clustering and ranking for selecting stable features from high dimensional remotely sensed data," *International Journal of Remote Sensing*, vol. 39, no. 23, pp. 8934–8949, 2018. [Online]. Available: https://doi.org/10.1080/01431161.2018.1500730

[9] K. Zhu and J. Yang, "A cluster-based sequential feature selection algorithm," in *2013 Ninth International Conference on Natural Computation (ICNC)*, 2013, pp. 848–852.

[10] X. Zhu, Y. Wang, Y. Li, Y. Tan, G. Wang, and Q. Song, "A new unsupervised feature selection algorithm using similarity-based feature clustering," *Computational Intelligence*, vol. 35, no. 1, pp. 2–22, 2019. [Online]. Available: https://doi.org/10.1111/coin.12192

[11] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, 1991.

[12] Q. Zhang, Q. Xie, and G. Wang, "A survey on rough set theory and its applications," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 4, pp. 323–333, 2016. [Online]. Available: https://doi.org/10.1016/j.trit.2016.11.001

[13] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," in *Intelligent Decision Support. Theory and Decision Library*, R. Słowiński, Ed. Springer, Dordrecht, 1992, vol. 11. [Online]. Available: https://doi.org/10.1007/978-94-015-7975-9_21

[14] Q. Al-Radaideh, M. Sulaiman, M. Selamat, and H. Ibrahim, "Approximate reduct computation by rough sets based attribute weighting," in *2005 IEEE International Conference on Granular Computing*, vol. 2, 2005, pp. 383–386.

[15] K. Gao, M. Liu, K. Chen, N. Zhou, and J. Chen, "Sampling-based tasks scheduling in dynamic grid environment," in *Proceedings of the 5th WSEAS International Conference on Simulation, Modeling and Optimization*, Corfu, Greece, 2005, pp. 25–30. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.582.5929&rep=rep1&type=pdf

[16] H. Q. Sun and Z. Xiong, "Finding minimal reducts from incomplete information systems," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, vol. 1, 2003, pp. 350–354.

[17] J. Wroblewski, "Computing minimal reducts using genetic algorithms," in *The Second Annual Joint Conference on Information Sciences*, 1995, pp. 186–189. [Online]. Available: http://www.cs.sjsu.edu/~khuri/Aalto_2017/ge_short.pdf

[18] T. Hong and Y. Liou, "Attribute clustering in high dimensional feature spaces," in *2007 International Conference on Machine Learning and Cybernetics*, 2007, pp. 2286–2289.

[19] T. Hong, P. Wang, and Y. Lee, "An effective attribute clustering approach for feature selection and replacement," *Cybernetics and Systems: An International Journal*, vol. 40, no. 8, pp. 657–669, 2009.

[20] T. Hong, P. Wang, and C. Ting, "An evolutionary attribute clustering and selection method based on feature similarity," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–5.

[21] T. Hong, Y. Liou, S. Wang, and B. Vo, "Feature selection and replacement by clustering attributes," *Vietnam Journal of Computer Science*, vol. 1, pp. 47–55, 2014. [Online]. Available: https://doi.org/10.1007/s40595-013-0004-3

[22] T. Hong, C. Chen, and F. Lin, "Using group genetic algorithm to improve performance of attribute clustering," *Applied Soft Computing*, vol. 29, pp. 371–378, 2015. [Online]. Available: https://doi.org/10.1016/j.asoc.2015.01.001

[23] A. Janusz and D. Slezak, “Utilization of attribute clustering methods for scalable computation of reducts from high-dimensional data,” in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2012, pp. 295–302.

[24] ——, “Rough set methods for attribute clustering and selection,” *Applied Artificial Intelligence*, vol. 28, no. 3, pp. 220–242, 2014.

[25] F. Pacheco, M. Cerrada, R. Sánchez, D. Cabrera, C. Li, and J. V. de Oliveira, “Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery,” *Expert Systems With Applications*, vol. 71, pp. 69–86, 2017. [Online]. Available: https://doi.org/10.1016/j.eswa.2016.11.024

[26] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Boston, MA, USA: Addison Wesley, 1989.

[27] A. Jakulin, “Machine learning based on attribute interactions,” Ph.D. dissertation, University of Ljubljana, Faculty of Computer and Information Science, 2005. [Online]. Available: http://eprints.fri.uni-lj.si/205/1/jakulin05phd.pdf

[28] L. Kaufman and P. Rousseeuw, *Computing Groups in Data: An Introduction to Cluster Analysis.* Toronto: John Wiley & Sons, 1990.

[29] R. Bello and R. Falcon, “Rough sets in machine learning: A review,” in *Studies in Computational Intelligence*, 2017.

[30] L. C. Molina, L. Belanche, and A. Nebot, “Feature selection algorithms: A survey and experimental evaluation,” in *2002 IEEE International Conference on Data Mining, Proceedings*, 2002, pp. 306–313.

[31] R. Jensen and Q. Shen, “A rough set-aided system for sorting www bookmarks,” in *Web Intelligence: Research and Development. WI 2001. Lecture Notes in Computer Science.* Springer, Berlin, Heidelberg, 2001, vol. 2198. [Online]. Available: https://doi.org/10.1007/3-540-45490-X_10

[32] G. Wang, H. Yu, and D. Yang, “Decision table reduction based on conditional information entropy,” *Chinese Journal of Computers*, vol. 25, no. 7, pp. 759–766, 2002.

[33] Y. Zhao, *R and Data Mining: Examples and Case Studies.* Elsevier, 2012. [Online]. Available: https://www.webpages.uidaho.edu/~stevel/517/RDataMining-book.pdf

[34] P. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: https://doi.org/10.1016/0377-0427(87)90125-7

[35] R. Kohavi and G. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997. [Online]. Available: https://doi.org/10.1016/S0004-3702(97)00043-X

[36] D. Dua and C. Graff, “Uci machine learning repositories,” 2019. [Online]. Available: http://archive.ics.uci.edu/ml/

Received on August 22, 2022
Accepted on September 23, 2022