International Journal of Approximate Reasoning 53 (2012) 912–926

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

# An efficient rough feature selection algorithm with a multi-granulation view

Jiye Liang [a,*], Feng Wang [a,b], Chuangyin Dang [b], Yuhua Qian [a]

[a] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China
[b] Department of System Engineering and Engineering Management, City University of Hong Kong, Hong Kong

ARTICLE INFO

*Article history:*
Received 15 April 2011
Received in revised form 27 February 2012
Accepted 29 February 2012
Available online 13 March 2012

*Keywords:*
Feature selection
Multi-granulation view
Rough set theory
Large-scale data sets

ABSTRACT

Feature selection is a challenging problem in many areas such as pattern recognition, machine learning and data mining. Rough set theory, as a valid soft computing tool to analyze various types of data, has been widely applied to select helpful features (also called attribute reduction). In rough set theory, many feature selection algorithms have been developed in the literatures, however, they are very time-consuming when data sets are in a large scale. To overcome this limitation, we propose in this paper an efficient rough feature selection algorithm for large-scale data sets, which is stimulated from multi-granulation. A sub-table of a data set can be considered as a small granularity. Given a large-scale data set, the algorithm first selects different small granularities and then estimate on each small granularity the reduct of the original data set. Fusing all of the estimates on small granularities together, the algorithm can get an approximate reduct. Because of that the total time spent on computing reducts for sub-tables is much less than that for the original large-scale one, the algorithm yields in a much less amount of time a feature subset (the approximate reduct). According to several decision performance measures, experimental results show that the proposed algorithm is feasible and efficient for large-scale data sets.



## 1. Introduction

As a common technique for data preprocessing in pattern recognition, machine learning and data mining, feature selection has attracted much attention in recent years [5–7,20,21,23,26,30,40]. In practices, databases increase quickly not only in the rows (objects) but also in the column (features) nowadays. Tens, hundreds even thousands of features are stored in databases in some real-world applications, which has resulted in data with high dimension. However, only a limited amount of features is useful in practice, that is, an excessive amount of features may cause a significant slowdown in the learning process and irrelevant or redundant features may deteriorate the performance of learning algorithms [12,13,38]. To ease this situation, it is desirable to reduce redundant features and select informative features for decreasing the cost of measuring, storing and transmitting, shortening the process time and gaining more compact classification models with a better generalization.

Rough set theory, proposed by Pawlak [31–33], is a relatively new soft computing tool for the analysis of a vague description of an object, and has become a popular mathematical framework for pattern recognition, image processing, feature selection, rule extraction, neuro-computing, conflict analysis, decision supporting, granular computing, data mining and knowledge discovery from large data sets [3,4,8,28,36,50,51]. In rough set theory, an important concept is attribute reduction (or approximate reduct), which can be considered a kind of specific feature selection. In other words, based on rough set theory,

* Corresponding author. Tel./fax: +86 0351 7018176.
*E-mail addresses:* ljy@sxu.edu.cn (J. Liang), sxuwangfeng@126.com (F. Wang), mecdang@cityu.edu.hk (C. Dang), jinchengqyh@126.com (Y. Qian).

0888-613X/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2012.02.004

one can select useful features from a given data table. Attribute reduction does not attempt to maximize the class separability but rather to retain the discernible ability of original features for the objects from the universe [15,16,41,44,52].

As one of the most important research topics along with the fast development of rough set theory, attribute reduction has aroused wide concern and study, and many attribute reduction techniques have been developed in last twenty years. Applying discernibility matrix, Skowron [42] proposed an attribute reduction algorithm by computing disjunctive normal form, which is able to obtain all attribute reducts of a given table whereas finding the minimal reduct of a decision table is an NP-hard problem. Kryszkiewicz and Lasek [22] proposed an approach to computing the minimal set of attributes that functionally determine a decision attribute. These two attribute reduction algorithms are usually computationally very expensive, especially for dealing with large-scale data sets of high dimensions. Therefore, to overcome this difficulty, many heuristic attribute reduction algorithms have been developed in rough set theory [11,13,24,25,39,35,43,45,46,48]. A heuristic attribute reduction algorithm can extract a single reduct from a given table in a relatively short time. In order to further reduce computational time, based on four kinds of common heuristic reduction algorithms, Qian et al. [37] developed a common accelerator to improve the time efficiency of a heuristic search process. According to the accelerator, certain objects are deleted from the universe every time when a new attribute is selected and added into the core. However, if the core is a reduct for some table, the accelerator will have no effect on the time reduction. And for some very large-scale data sets, the accelerated computational time is still very long. Besides, in view of the computational space utilization, it is space-consuming as well to find a reduct to a large-scale data set. Therefore, it is desired to have an efficient and space-saving feature selection algorithm to large-scale data tables.

In biological research, social survey, product testing, etc., since it usually is very difficult or impossible to collect all the samples, one often has to use some of the samples to estimate the totality. This leads to the main idea of this paper, namely, estimating on sub-tables the reduct of a large-scale data set. The idea of reduct computation on sub-tables was also mentioned in the process of dynamic reducts [1]. It should be pointed out that, this idea in the dynamic reducts aim to find stable reducts of a given decision table, however, not to get the estimates. In addition, we remark that dynamic reducts need to select lots of sub-tables and the size of each sub-table is very close to the size of the original table [2]. Therefore, the computation of dynamic reducts is also very time-consuming for a large-scale data table. In addition, based on reduct subspaces, Miao et al. [27] constructed a classifier for partially labeled data, and the subspace was not introduced to get the estimates, either.

Drove by the above analysis, an efficient rough feature selection algorithm is devised in this paper. The algorithm targets estimating reduct of a large-scale data set from a multi-granulation view. A sub-table of a large-scale data set can be considered as single small granularity; and one can estimate on this small granularity the reduct of the original table. By collecting different sub-tables together, one can compute a reduct on each small granularity to which the original large-scale table is mapped. Fusing together these reducts on all small granularities, we obtain a feature subset of the original large-scale table. It should be noted that the available feature subset usually is not an exact reduct (Pawlak's reduct) on the original large table but an approximate reduct. The total time spent on computing reducts for sub-tables is much less than that for the original large-scale one and the space utilization is also much smaller. In practices, to save computational time, it is good enough to find an approximate reduct. In order to further illustrate the feasibility of the proposed feature selection algorithm, several experimental tests are given in this paper, which have been carried out according to four common decision measures on reducts.

The rest of this paper is organized as follows: some preliminaries in rough set theory are briefly reviewed in Section 2. In Section 3, by introducing the approach for selecting small granularities and finding a reduct on each small granularity, we give an efficient rough feature selection algorithm for large-scale data sets. In Section 4, ten UCI large-scale data sets are employed to illustrate the feasibility and efficiency of the proposed algorithm. Section 5 concludes the paper with some discussions.

## 2. Preliminary knowledge on rough sets

In this section, we will review several basic concepts in rough set theory. Throughout this paper, we assume that the universe $U$ is a finite nonempty set.

An information table, as a basic concept in rough set theory, provides a convenient framework for the representation of objects in terms of their attribute values. An information system $S$ is a pair $(U, A)$ , where $U$ is a finite nonempty set of objects and is called the universe and $A$ is a non-empty, finite set of attributes. For each $a \in A$, a mapping $a : U \to V_a$ is determined by a given decision table, where $V_a$ is the domain of $a$.

Each non empty subset $B \subseteq A$ determines an indiscernibility relation in the following way,

$$R_B = \{(x, y) \in U \times U \mid a(x) = a(y), \forall a \in B\}.$$

The relation $R_B$ partitions $U$ into some equivalence classes given by

$$U/R_B = \{[x]_B \mid x \in U\},$$

where $[x]_B$ denotes the equivalence class determined by $x$ with respect to $B$, i.e.,

$$[x]_B = \{y \in U \mid (x, y) \in R_B\}.$$

When relation $R$ is know by default or unimportant for consideration, $U/R_B$ can be replaced by $U/B$.

Given an equivalence relation $R$ on the universe $U$ and a subset $X \subseteq U$, one can define a lower approximation of $X$ and an upper approximation of $X$ by

$$\underline{R}X = \bigcup\{x \in U \mid [x]_R \subseteq X\}$$

and

$$\overline{R}X = \bigcup\{x \in U \mid [x]_R \cap X \neq \emptyset\},$$

respectively [6]. The order pair $(\underline{R}X, \overline{R}X)$ is called a rough set of $X$ with respect to $R$. The positive region of $X$ is denoted $POS_R(X) = \underline{R}X$.

We define a partial relation $\preceq$ on the family $\{U/B \mid B \subseteq A\}$ as follows: $U/P \preceq U/Q$ (or $U/Q \succeq U/P$) if and only if, for every $P_i \in U/P$, there exists $Q_j \in U/Q$ such that $P_i \subseteq Q_j$, where $U/P = \{P_1, P_2, \ldots, P_m\}$ and $U/Q = \{Q_1, Q_2, \ldots, Q_n\}$ are partitions induced by $P$, $Q \subseteq A$, respectively. In this case, we say that $Q$ is coarser than $P$, or $P$ is finer than $Q$. If $U/P \preceq U/Q$ and $U/P \neq U/Q$, we say $Q$ is strictly coarser than $P$ (or $P$ is strictly finer than $Q$), denoted by $U/P \prec U/Q$ (or $U/Q \succ U/P$).

It is clear that $U/P \prec U/Q$ if and only if, for every $X \in U/P$, there exists $Y \in U/Q$ such that $X \subseteq Y$, and there exist $X_0 \in U/P$ and $Y_0 \in U/Q$ such that $X_0 \subset Y_0$.

A decision table is an information system $S = (U, C \cup D)$ with $C \cap D = \emptyset$, where $C$ is called a condition attribute set and its element is called a condition attribute, $D$ is called a decision attribute set and its element is called a decision attribute. Given $P \subseteq C$ and $U/D = \{D_1, D_2, \ldots, D_r\}$, the positive region of $D$ with respect to the condition attribute set $P$ is defined by $POS_P(D) = \bigcup_{k=1}^{r} \underline{P}D_k$. Then, one can extract decision rules from a decision table.

## 3. Rough feature selection algorithm with a multi-granulation view

In rough set theory, feature selection is also called attribute reduction, which is a studying focus in many fields. With the development of attribute reduction in application, one of the bottlenecks is the computational time of reduction computation, especially for the large-scale data sets. Therefore, according to the idea of using samples to estimate the totality, we devise in this section a highly efficient rough feature selection algorithm from a multi-granulation view. In the design of the algorithm, we remark that there are three key problems should be considered. The first problem is selecting sub-tables from the large-scale one, the second one is finding reduct on sub-tables, and the last one is the fusing the all the reducts on sub-tables together.

A sub-table of a large-scale data table can be considered as single small granularity; and one can estimate on this small granularity the reduct of the original table. By collecting different sub-tables together, one can compute a reduct on each small granularity to which the original large-scale table is mapped. Fusing together these reducts on all selected small granularities, we obtain a feature subset of the original large-scale table. It should be noted that we only discuss the reduct on the decision table in this paper.

### 3.1. Selecting small granularity

In the process of selecting small granularity, one of the most important issues is how to determine the size of a small granularity. Hence, with the use of some concepts and formulas in statistics, we first introduce a familiar approach to determine sample size [17]. This approach is very common in statistics, which has been widely used to estimate the sample size in many instances such as estimating the annual salary of college graduates, average consumption of customers and average deposit of residents.

Let $S$ be a data table (the original large-scale data table) and let the size of $S$ be denoted by $N$. Then, the sample size $M'$ is defined as [17]:

$$M' = \frac{Z^2 \times \sigma^2}{E^2}, \tag{1}$$

where $\sigma$ means the standard deviation on $S$, $Z$ means the $Z-$statistic under confidence intervals (e.g., the $Z-$statistic corresponding to confidence interval 95% is 1.96, and confidence interval 99% is 2.58), and $E$ means the acceptable tolerance error which can be adjusted as requested.

It can be seen from formula (1) that there is no direct relation between sample size $M'$ and table size $N$. In fact, if sample size $M'$ is larger than 5% of the overall size, the sample size $M'$ needs to be adjusted. In [18,29], by introducing the adjustment coefficient FPC, the above formula is adjusted to reduce the sample size, which is defined as follows:

$$M = \frac{M'N}{M' + N}. \tag{2}$$

In view of that the decision tables in rough set theory are categorical data, we introduce the coefficient of unalikeability $u$ to replace the standard deviation $\sigma$. For the data table $S$, let its universe be denoted by $U$. Then, the coefficient of unalikeability $u$ on $U$ is defined as [19,34]:

$$u = \frac{\sum_{i=1}^{|U|}\sum_{j=1}^{|U|} c(x_i, x_j)}{|U|^2}, \tag{3}$$

where $x_i, x_j \in U$, and

$$c(x_i, x_j) = \begin{cases} 1, & x_i \neq x_j\ , \\ 0, & x_i = x_j\ . \end{cases}$$

Note that $x_i \in U$ is a one-dimensional vector in [19,34], whereas the data in a decision table is usually multi-dimensional. Thus, we expand the definition of $c(x_i, x_j)$ into multi-dimensional data, which is denoted by $c_m(x_i, x_j)$. Let $S = (U, C \cup D)$ be a decision table, $a \in C$, $x_i \in U$ and $x_i = (a_1(x_i), a_2(x_i), \ldots, a_{|C|}(x_i))$. Then $c_m(x_i, x_j)$ is defined as:

$$c_m(x_i, x_j) = \sum_{k=1}^{|C|} \delta(a_k(x_i), a_k(x_j)), \tag{4}$$

with the function $\delta$ being given by

$$\delta(a_k(x_i), a_k(x_j)) = \begin{cases} 1, & a_k(x_i) \neq a_k(x_j) \\ 0, & a_k(x_i) = a_k(x_j). \end{cases}$$

Hence, for a decision table $S$, the coefficient of unalikeability can be redefined as

$$u_1 = \frac{\sum_{i=1}^{|U|}\sum_{j=1}^{|U|} c_m(x_i, x_j)}{|U|^2}, \tag{5}$$

and sample size is redefined as

$$M_1' = \frac{Z^2 \times u_1^2}{E^2}. \tag{6}$$

Based on the above introduction, an algorithm is given to determine the size of sub-table (small granularity) on a large-scale decision table as follows:

**Algorithm 1**. An algorithm to determine the sample size on a large-scale decision table

**Input**: Decision table $S = (U, C \cup D)$.
**Output**: Sample size $M_1$.
*Step 1*: Compute the coefficient of unalikeability $u_1$ (according to equations (4) and (5));
*Step 2*: Compute sample size $M_1'$(according to equation (6));
*Step 3*: If $M_1' > 0.05|U|$, then compute adjusted sample size $M_1 = \frac{M_1' \times |U|}{M_1' + |U| + 1}$;
else $M_1 \leftarrow M_1'$.
*Step 4*: Return $M_1$ and end.

Here, we employ an example to illustrate above concepts and computations involved in the determination of $M_1$. UCI data set *Breast-cancer-wisconsin* with 699 objects, 9 attributes and 2 decision classes is used in the example. For convenience, we remove the objects with missing values from the data set, and the number of remaining objects used in the example is 683.

**Example 1.** For the data set *Breast-cancer-wisconsin*, we have $|U| = 683$ and $|C| = 9$.
Then, $u_1 = \frac{\sum_{i=1}^{|U|}\sum_{j=1}^{|U|} c_m(x_i,x_j)}{|U|^2} = 5.88$.
We take $Z = 2.58$ (confidence interval is 99%) and $E = 1.01$, then $M_1' = \frac{Z^2 \times u_1^2}{E^2} \approx 243$.
Because $243 > 683 \times 0.05 \approx 34$, then $M_1 = \frac{M_1' \times |U|}{M_1' + |U| + 1} = 179$.

In view of that, according to $M_1'$ in equation (6), $Z$ is a constant value and $E$ is a desired level of precision, we can give an estimation for $M_1'$. If we select the confidence interval equal to 99% ($Z = 2.58$) and $E = 1.01$, then we get that $M_1' \approx 6.5u_1^2$. Hence, Algorithm 1 can be further described as follows:

**Algorithm 1'**. An algorithm to determine the sample size on a large-scale decision table

**Input**: Decision table $S = (U, C \cup D)$.
**Output**: Sample size $M_1$.
*Step 1*: Compute sample size $M_1' = 6.5u_1$ (according to equations (4 - 6));
*Step 2*: If $M_1' > 0.05|U|$, then compute adjusted sample size $M_1 = \frac{M_1' \times |U|}{M_1' + |U| + 1}$;
else $M_1 \leftarrow M_1'$.
*Step 3*: Return $M_1$ and end.

Note that, for a large-scale data set, the sample size $M_1$ found by Algorithm 1(or 1') can be relatively adjusted, but not quite different from the original value. For example, instead of $M_1 = 179$ obtained in Example 1, one can use 180 as the sample size is also fine.

For a decision table, we know that the reduct is directly related to its decision distribution. Thus, the decision distribution on a small granular space may also affect the estimated result. To ensure the decision distribution on a small granular space is close to the large-scale one, we set in the algorithm the ratio of decision attribute values of a small granular space equal to the ratio of the original large-scale one. Besides, there should be some similarities among small granularity, which make the reducts on small granularity are close to each other relatively and are more convenient for the fusion of feature subset. Hence, in the selection process of small granularity, we make each small granularity contains some objects that are identical to those in another one.

According to the above discussion, we propose the algorithm for selecting sub-table (small granularity) on a large-scale decision table as follows:

**Algorithm 2**. An algorithm for selecting small granularity on a large-scale decision table

**Input**: Decision table $S = (U, C \cup D)$.
**Output**: $n$ small granularity $S_j = (U_j, C \cup D)$ $(j = 1, 2, \ldots, n)$.

*Step 1*: Compute the size of small granularity $M_1$ (according to Algorithm 1);
*Step 2*: Compute $U/D = \{D_1, D_2, \ldots, D_r\}$, and the decision attribute value proportions $p_i = |D_i|/|U| (i = 1, 2, \ldots, r)$;
*Step 3*: Compute the numbers of each decision attribute value in the small granularity $m_i = [M_1 \times p_i] (i = 1, 2, \ldots, r)$ (function $[\cdot]$ is the rounding function);
*Step 4*: Select the first granularity $S_1$ on $U$, $U_1 \leftarrow \emptyset$:
for $(i = 1; i \le r; i++)$
{
Select $m_i$ objects from $D_i$ randomly, which is denoted by $X$;
$U_1 \leftarrow U_1 \cup X$;
}
*Step 5*: Select granularity $S_j$ repeatedly, $j \leftarrow 2$:
while$(|U - \bigcup_{k=1}^{j-1} U_k| < M_1)$
{
Given threshold $\alpha$ $(0 < \alpha < 1)$;
*Step 5.1*: Select $\alpha M$ objects from table $S_{j-1}$:
{
Compute $U_{j-1}/D = \{D_1', D_2', \ldots, D_r'\}$;
Select $\alpha m_i$ objects from $D_i'$ $(i = 1, 2, \ldots, r)$ randomly, which is denoted by $X'$;
$U_j \leftarrow U_j \cup X'$;
}
*Step 5.2*: $U'' = U - \bigcup_{k=1}^{j-1} U_k$, and select $(1 - \alpha)M$ objects from $U''$:
{
Compute $U''/D = \{D_1'', D_2'', \ldots, D_r''\}$;
Select $(1 - \alpha)m_i$ objects from $D_i$ $(i = 1, 2, \ldots, r)$ randomly, which is denoted by $X''$;
$U_j \leftarrow U_j \cup X''$;
}
$j \leftarrow j + 1$;
}
*Step 6*: $n \leftarrow j - 1$ and end.

Here are some explanations about Algorithm 2. In *Steps 2-3*, the algorithm aims to ensure the decision distribution on sub-tables is close to the large-scale one. Besides, because of that $m_i$ are integers, one can get that $\sum_{i=1}^{r} m_i \approx M$. In the process of selecting sub-tables in *Step 5*, some objects are selected from the existing sub-tables, which ensure there are certain similarities among selected sub-tables. In addition, threshold $\alpha$ should not be too small to weaken the similarity, we propose an empirical value of $\alpha = 0.5$.

**Example 2** (*Continued from Example 1*)**.** Select sub-tables from *Breast-cancer-wisconsin* by using Algorithm 2.

From Example 1, we get that there are two decision classes in data set *Breast-cancer-wisconsin* and $M_1 = 109$. According to *Step 2 and 3*, by computing $U/D = \{D_1, D_2\}$, we get the decision attribute value ratio is $\theta_1 = 0.65$ and $\theta_2 = 0.35$, respectively. Then, $m_1 = [179 \times 0.65] = 116$ and $m_2 = [179 \times 0.35] = 63$.

According to *Step 4*, we select 116 objects from $D_1$ and 63 objects from $D_2$ randomly, and form the first sub-table $U_1$.

According to *Step 5*, because $\alpha = 0.5$, we have $\alpha \cdot 179 \approx 90$. Then, we select 90 objects from $U_1$, 89 objets from $U - U_1$ and form the second sub-table $U_2$. By doing so, we select in turn sub-tables $U_3, U_4, \ldots$. For $|U - \bigcup_{k=1}^{5} U_k| = 59 < M_1$, we have $n = 5$, namely, we obtain 5 sub-tables from *Breast-cancer-wisconsin*.

### 3.2. Reduction algorithm to small granularity

In practices, a given decision table usually has multiple reducts, and finding its minimal reduct is an NP-hard problem. Therefore, some heuristic algorithms that can find one reduct in a shorter time were proposed in Refs. [11,13,24,25,39,43,45,46,48], most of which are greedy and forward search algorithms. Starting with a nonempty set, these search algorithms keep adding one or several attributes of high significance into a pool at each iteration until the dependence no longer increases. A common accelerator was proposed in [37] to save the computational time of existing heuristic algorithms. We employ in this section the accelerated reduction algorithm to find reduct of sub-tables. Four representative heuristic reduction algorithms were employed to devise the accelerated algorithm in [37], which are reviewed as follows:

The idea of attribute reduction using positive-region was first originated by Grzymala-Busse in [9,10]. Hu and Cercone proposed a heuristic attribute reduction algorithm, known as positive-region reduction (PR), which keeps the positive region of target decision unchanged [11]. The definition of positive region of a decision table can be found in Section 2, and the attribute dependence degree based on positive region is as follows [31]:

**Definition 1.** Let $S = (U, C \cup D)$ be a decision table and $B \subseteq C$. The attribute dependence degree of $B$ relative to $D$ is defined as

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}. \tag{7}$$

In a classical rough set model, Shannon's information entropy was introduced to find reduct in [43], and its conditional entropy was used to find the relative reduct of a decision table in [45]. The reduction algorithm in [45] keeps the conditional entropy of target decision unchanged, and is denoted by SCE, where a conditional entropy is defined as follows [45]:

**Definition 2.** Let $S = (U, C \cup D)$ be a decision table and $B \subseteq C$. Then, one can obtain the condition partition $U/B = \{X_1, X_2, \ldots, X_m\}$ and decision partition $U/D = \{Y_1, Y_2, \ldots, Y_n\}$. Based on these partitions, a conditional entropy of $B$ relative to $D$ is defined as

$$H(D|B) = -\sum_{i=1}^{m} \frac{|X_i|}{|U|} \sum_{j=1}^{n} \frac{|X_i \cap Y_j|}{|X_i|} \log\left(\frac{|X_i \cap Y_j|}{|X_i|}\right). \tag{8}$$

In [24], the complementary entropy was used to measure the uncertainty of an information system. And its conditional entropy can be used to measure the uncertainty of a decision table. In [24,25], based on the complementary entropy, a heuristic reduction algorithm was introduce to reduce the redundant features. The conditional entropy in this algorithm will be used in this study and is as follows [24,25]:

**Definition 3.** Let $S = (U, C \cup D)$ be a decision table and $B \subseteq C$. Then, one can obtain the condition partition $U/B = \{X_1, X_2, \ldots, X_m\}$ and decision partition $U/D = \{Y_1, Y_2, \ldots, Y_n\}$. Based on these partitions, a conditional entropy of $B$ relative to $D$ is defined as

$$E(D|B) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{|Y_i \cap X_j|}{|U|} \frac{|Y_i^c \cap X_j^c|}{|U|}, \tag{9}$$

where $Y_i^c$ and $X_j^c$ are complement sets of $Y_i$ and $X_j$ respectively.

Qian and Liang in [39] presented a combination entropy for measuring the uncertainty of information systems and used its conditional entropy to obtain a feature subset. This reduction algorithm can find an attribute subset that possesses the same number of pairs of indistinguishable elements as that of the original decision table, and is denoted here by CCE. The definition of the conditional entropy is as follows [39]:

**Definition 4.** Let $S = (U, C \cup D)$ be a decision table and $B \subseteq C$. Then one can obtain the condition partition $U/B = \{X_1, X_2, \ldots, X_m\}$ and decision partition $U/D = \{Y_1, Y_2, \ldots, Y_n\}$. Based on these partitions, a conditional entropy of $B$ relative to $D$ is defined as

$$CE(D|B) = \sum_{i=1}^{m} \left( \frac{|X_i|}{|U|} \frac{C^2_{|X_i|}}{C^2_{|U|}} - \sum_{j=1}^{n} \frac{|X_i \cap Y_j|}{|U|} \frac{C^2_{|X_i \cap Y_j|}}{C^2_{|U|}} \right). \tag{10}$$

where $C^2_{|X_i|}$ denotes the number of pairs of objects which are not distinguishable from each other in the equivalence class $X_i$.

Based on the above four measures, the common attribute significance in a heuristic reduction algorithm is defined as follows [37]:

**Definition 5.** Let $S = (U, C \cup D)$ be a decision table and $B \subseteq C$. $\forall a \in B$, the significance measure (inner significance) of $a$ in $B$ is respectively defined as

$$Sig_1^{inner}(a, B, D, U) = \gamma_B(D) - \gamma_{B-\{a\}}(D),$$

$$Sig_2^{inner}(a, B, D, U) = H(D|B - \{a\}) - H(D|B),$$

$$Sig_3^{inner}(a, B, D, U) = E(D|B - \{a\}) - E(D|B),$$

$$Sig_4^{inner}(a, B, D, U) = CE(D|B - \{a\}) - CE(D|B).$$

**Definition 6.** Let $S = (U, C \cup D)$ be a decision table and $B \subseteq C$. $\forall a \in C - B$, the significance measure (outer significance) of $a$ in $B$ is respectively defined as

$$Sig_1^{outer}(a, B, D, U) = \gamma_{B\cup\{a\}}(D) - \gamma_B(D),$$

$$Sig_2^{outer}(a, B, D, U) = H(D|B) - H(D|B \cup \{a\}),$$

$$Sig_3^{outer}(a, B, D, U) = E(D|B) - E(D|B \cup \{a\}),$$

$$Sig_4^{outer}(a, B, D, U) = CE(D|B) - CE(D|B \cup \{a\}).$$

Given a decision table $S = (U, C \cup D)$ and $a \in C$. From the literature [24,31,37,39], we know that if $Sig_\Delta^{inner}(a, C, D) > 0$ ($\Delta = 1, 2, 3, 4$), then the attribute $a$ is indispensable, i.e., $a$ is a core attribute of $S$. Based on the core attributes, a heuristic attribute reduction algorithm can find an attribute reduct by gradually adding selected attributes to the core.

For convenience, we introduce a uniform notation $ME(D|B)$ to denote those four measures. For example, if one adopts Shannon's entropy to define an attribute significance, then $ME(D|B) = H(D|B)$. Based on above measures for attribute significance, a feature selection accelerated algorithm was proposed in [37], which is as follows:

**Algorithm 3**. An accelerated attribute reduct algorithm to a decision table (FSPA)

**Input**: Decision table $S = (U, C \cup D)$

**Output**: One reduct *red*

*Step 1*: $red \leftarrow \emptyset$;

*Step 2*: *for* $(j = 1; j \le |C|; j++)$

{ if $Sig^{inner}(a_j, C, D, U) > 0$, then $red \leftarrow red \cup \{a_j\}$;

}

*Step 3*: Let $i \leftarrow 1, P \leftarrow red, U_i \leftarrow U$;

*Step 4*: *while* $(ME^{U_i}(D|P) \neq ME^{U_i}(D|C))$ *do*

{ $i \leftarrow i + 1$;

$U_i = U_{i-1} - POS_P^{U_{i-1}}(D)$;

Compute and select sequentially $Sig^{outer}(a_0, red, D, U_i) = max\{Sig^{outer}(a_i, red, D, U_i)\}$, $a_j \in C - red$;

$red \leftarrow red \cup \{a_0\}$;

$P \leftarrow red$;

}

*Step 5*: return *red* and end.

In *Step 4*, $POS_P^{U_{i-1}}(D)$ denotes the positive region of $D$ with respect to the condition attribute subset $P$ on universe $U_{i-1}$. And the time complexity of the above algorithm is $O(|U||C| + \sum_{i=1}^{|C|} |U_i|(|C| - i + 1))$ according to the literature [37].

However, in [37], the time complexity does not include the computational time of entropy and positive region. For a decision table, computing entropy and positive region is a key step in the above reduction algorithm, which is not computationally costless. Thus, to analyze the exact time complexity of the algorithm, we need to give the time complexity of computing entropy and positive region as well.

For a decision table, according to Definitions 1–4, we first need to compute the conditional classes and decision classes, respectively, and then compute the value of entropy or positive region. Xu et al. in [49] gave a fast algorithm for partition with time complexity being $O(|U||C|)$. So, the time complexity of computing entropy or positive region is

$$O(|U||C| + |U| + \sum_{i=1}^{m} |X_i| \cdot \sum_{j=1}^{n} |Y_j|) = O(|U||C| + |U| + |U||U|) = O(|U|^2),$$

where the specific introduction of $m$, $n$, $X_i$ and $Y_j$ is shown in Definitions 1-4. Thus, the time complexity of Algorithm 2 should be modified as

$$O(|U|^2|C| + \sum_{i=1}^{|C|} |U_i|^2(|C| - i + 1)).$$

**Example 3** (*Continued from Example 2*)**.** Find reduct of sub-tables of *Breast-cancer-wisconsin* by using FSPA.

For convenience, based on positive region, we only employ in this example one sub-table of *Breast-cancer-wisconsin* to illustrate the reduct computation.

In step 2, according to Definition 5, we have $Sig_1^{inner}(a_j, C, D, U) = \gamma_C(D) - \gamma_{C-\{a\}}(D)$. Hence, we get the attribute significance in order are $Sig_1^{inner}(a_1, C, D, U) = Sig_1^{inner}(a_2, C, D, U) = Sig_1^{inner}(a_3, C, D, U) = Sig_1^{inner}(a_4, C, D, U) = Sig_1^{inner}(a_5, C, D, U) = Sig_1^{inner}(a_6, C, D, U) = Sig_1^{inner}(a_7, C, D, U) = Sig_1^{inner}(a_8, C, D, U) = 0$. Then, we have $red = \emptyset$ now.

In step 4, for the first loop, because $red = \emptyset$, we have $POS_P^{U_1}(D) = \emptyset$. Then according to $Sig_1^{outer}(a, red, D, U) = \gamma_{red\cup\{a\}}(D) - \gamma_{red}(D)$, we get the attribute significance in order are $Sig_1^{outer}(a_1, C, D, U_1) = 0.1611$, $Sig_1^{outer}(a_2, C, D, U_1) = 0.7500$, $Sig_1^{outer}(a_3, C, D, U_1) = 0.7556$, $Sig_1^{outer}(a_4, C, D, U_1) = 0.1222$, $Sig_1^{outer}(a_5, C, D, U_1) = 0.0833$, $Sig_1^{outer}(a_6, C, D, U_1) = 0.0500$, $Sig_1^{outer}(a_7, C, D, U_1) = 0.1778$ and $Sig_1^{outer}(a_8, C, D, U_1) = 0.1111$. Then, we have $a_0 = a_3$ and $red = \emptyset \cup a_3 = a_3$ now.

Because $\gamma_{red}(D) \neq \gamma_C(D)$, we continue to add attribute to *red*. In view of that the calculation of following loops are similar to the first one, we didn't give the specific precesses here. By through three loops, the final reduct is $\{a_3, a_5, a_6\}$, which is simplified as $\{3, 5, 6\}$.

### 3.3. An efficient feature selection algorithm for large-scale decision tables

For a large-scale decision table, from Algorithm 1 and Algorithm 2, we obtain a group of estimates to the reduct. By fusing together these estimates, we get a valid feature subset for the large-scale decision table.

Given a decision table, under the introduction of discernibility matrix and core in [11], we first develop the following theorem, which will be used in our further development.

**Theorem 1.** *Let $S = (U, C \cup D)$ be decision table, $a \in C$ and $U_0 \subseteq U$. If $a_0$ is a core attribute on $U_0$, then $a_0$ is a core attribute on $U$.*

**Proof.** As mentioned in [11], if an element in the discernibility matrix is a single attribute, then this attribute belongs to the core attribute set.

Let $M' = \{m'_{ij}\}$ be the discernibility matrix on $U_0$ and $M = \{m_{ij}\}$ be the discernibility matrix on $U$. Then, we have

$$M' = \begin{bmatrix} m'_{11} & & & \\ m'_{12} & m'_{22} & & \\ \vdots & \vdots & & \\ m'_{|U_0|1} & m'_{|U_0|2} \cdots & & m'_{|U_0||U_0|} \end{bmatrix} \quad and \quad M = \begin{bmatrix} m_{11} & & & \\ m_{12} & m_{22} & & \\ \vdots & \vdots & & \\ m_{|U|1} & m_{|U|2} \cdots & & m_{|U||U|} \end{bmatrix}, \tag{11}$$

respectively. Because of $U_0 \subseteq U$, matrix $M$ can be rewritten as

$$M = \begin{bmatrix} m'_{11} & & & & \\ m'_{12} & m'_{22} & & & \\ \vdots & \vdots & & & \\ m'_{|U_0|1} & m'_{|U_0|2} \cdots & m'_{|U_0||U_0|} & & \\ m'_{|U_0|+11} & m'_{|U_0|+12} \cdots & m'_{|U_0|+1|U_0|} & m'_{|U_0|+1|U_0|+1} & \\ \vdots & \vdots & \vdots & & \\ m_{|U|1} & m_{|U|2} \cdots & m_{|U||U_0|} & m_{|U||U_0|+1} \cdots & m_{|U||U|} \end{bmatrix}. \tag{12}$$

It is easy to see that $M'$ is a sub-matrix of $M$. Obviously, if element $m'_{ij} \in M'$ contains just one attribute, then $m'_{ij} \in M$ also contains one attribute only. Namely, if $a_0$ is a core attribute on $U_0$, then $a_0$ is also a core attribute on $U$. This completes the proof. □

As mentioned above, core attribute is the indispensable attribute in a reduct, which means the intersection of all reducts of a data table. Hence, an effective feature subset should contains as many core attributes as possible. From Theorem 1, we know that if an attribute is a core attribute on a sub-table, then this attribute is a core attribute on the original table. To ensure the final feature subset include as many core attributes as possible, we form a set by all the estimates on sub-tables and use it as the final feature subset. And an rough feature selection algorithm is proposed as follows:

**Algorithm 4**. An efficient rough feature selection algorithm(E-FSA)

**Input**: A large-scale decision table $S = (U, C \cup D)$
**Output**: Feature subset *Red*

*Step 1*: Select $n$ small granularity according to Algorithm 2 from $S$: $S_1 = (U_1, C \cup D), S_2 = (U_2, C \cup D), \cdots, S_n = (U_n, C \cup D)$;
*Step 2*: *Red* $\leftarrow \emptyset$;
*for* $(j = 1; j \le n; j++)$
{
Compute the attribute reduct $red_j$ of table $S_j = (U_j, C \cup D)$ using Algorithm 3 ;
*Red* = *Red* $\cup$ $red_j$ ;
}
*Step 3*: return *Red* and end.

Time complexity of Algorithm 4 : the time complexity of *Step 1* is $n|U_j||C|$ according to Algorithm 2; in *Step 2*, regarding to Algorithm 3, the time complexity of finding reducts on $n$ sub-tables is $O(n|U_j|^3|C| + n\sum_{i=1}^{|C|}|U_j^i|^3(|C| - i + 1))$; and time complexity of *Step 3* is $n|C|$. Thus, the time complexity of E-FSA is $O(n|U_j|^2|C| + n\sum_{i=1}^{|C|}|U_j^i|^2(|C| - i + 1))$.

From the discussion in the previous subsections, we know that, for a large-scale decision table $S = (U, C \cup D)$, the time complexity of the accelerated algorithm in [37] is $O(|U|^2|C| + \sum_{i=1}^{|C|}|U_i|^2(|C| - i + 1))$. Usually, $|U|^2$ and $|U_i|^2$ are much larger than $n|U_j|^2$ and $n|U_j^i|^2$, respectively. Therefore, the computational time of algorithm E-FSA is much smaller than that of the accelerated algorithm.

Note that, for a sub-table, most of the reduction algorithms can be employed to find reducts. We mainly focus on in this paper how to select sub-tables from a large-scale data table, and fuse the final reducts on all selected sub-tables. In the process of finding reduct on a sub-table, we employ the accelerated framework based on four kinds of representative heuristic reduction algorithms in this paper. This is also the reason why we select those four kinds of algorithms to test our proposed algorithm in the experiment part (Section 4.2). In addition, based on the framework that is dividing and fusing on a large-scale data set proposed in this paper, by employing other reduction algorithms to find reduct on a sub-table, one can also construct appropriate efficient algorithms.

**Example 4** (*Continued from Example 3*)**.** Find feature subset of *Breast-cancer-wisconsin* by using E-FSA.

In this example, we find feature subset based on positive-region reduction algorithm. According to Example 2, we obtain five sub-tables of *Breast-cancer-wisconsin*.

By using Algorithm 3, we get reducts on above five sub-tables, which are {3, 5, 6}, {1, 2, 3, 5}, {1, 2, 5}, {1, 2, 5} and {1, 2}. Then, the final feature subset is {1, 2, 3, 5, 6}.

For data set *Breast-cancer-wisconsin*, the reduct found by FSPA(Algorithm 3) based on positive-region reduction is {1, 3, 5, 6}. Comparing with algorithm FSPA, there is one redundant attribute in the feature subset found by E-FSA.

Example 4 illustrated the process of finding feature subset by using E-FSA based on positive-region reduction. In the same way, other three reduction algorithms (algorithms based on entropy) mentioned in Section 3.2 can be also used to select features. In addition, for the convenience of calculation, data set *Breast-cancer-wisconsin* employed in Examples 1-4 is relatively in a small scale. In the following section of experiments, we employ several larger-scale data sets to test the algorithm E-FSA.

## 4. Experimental analysis

The objective of the following experiments is to show the computational efficiency of the proposed algorithm E-FSA. The data sets used in the experiments are outlined in Tables 1, 5 and 14, which were all downloaded from UCI repository of machine learning databases. All the experiments were carried out on a personal computer with Windows XP and Inter(R) Core(TM)2 Quad CPU Q9400, 2.66 GHz and 3.37 GB memory. The software being used is Microsoft Visual Studio 2005 and programming language is C#.

In order to illustrate the feasibility and efficiency of the algorithm E-FSA, we employ in this section ten UCI data sets to test the algorithm. The experiments are divided into three parts, which are illustration of the feasibility, efficiency and high-efficiency for large scale data tables, respectively. In the first two parts, the feasibility and efficiency of E-FSA are illustrated mainly through comparing it with the usual and representative attribute reduction algorithms. To further illustrate the efficiency, two larger-scale data sets are employed in the last part to conduct the experiment. The specific design of experiment of each part is in the following.

### 4.1. Feasibility analysis

As mentioned above, a given data table usually has multiple reducts. Based on the introduction of discernibility matrix in a decision table, an attribute reduction algorithm was proposed in [11], which is able to obtain all attribute reducts of the decision table. Given a feature subset, if it is very close to one reduct of a decision table, then it is commonly considered as an effective approximated reduct; and if it is quite different from all reducts, it is ineffective apparently. In this section, the experiment aims to illustrate that if algorithm E-FSA can find an effective approximated reduct, that is, E-FSA is feasible.

In this section, two UCI data sets used in the experiments are outlined in Table 1. For each data set, we first find all reducts by applying the above algorithm in [11], and then compute the feature subset *Red* using E-FSA. The feasibility of the algorithm E-FSA is demonstrated by comparing *Red* with all the reducts. In these two data sets, *Mushroom* is a data set with missing values, and for a uniform treatment of all data sets, we remove the objects with missing values. Moreover, *Winequality-white* is preprocessed using the data tool Rosetta.

By carrying out the algorithm in [11] on these two data sets, we get that there are 156 reducts of *Mushroom* and 2 reducts of *Winequality-white*, which are shown in Table 2 and Table 3, respectively. In view of that there are many reducts of *Mushroom*, we only list a small part of the result here. The feature subsets found by algorithm E-FSA are shown in Table 4. In these three tables, each element denotes an attributes subset of the data set, for example, the first element {2, 3, 10, 11, 20} in Table 2 is a reduct (an attributes subset) of *Mushroom*, the value 2, 3, 10, 11 and 20 correspond to the 2*nd*, 3*rd*, 10*th* 11*th* and 20*th* attribute in the data set *Mushroom*.

From the experimental results in Table 2-4, it is easy to see that, for data set *Mushroom*, the feature subset found by E-FSA is $Red = \{1, 5, 20, 21\}$, and the nearest reduct in Table 2 is the 45*th* reduct {5, 20, 21}. Comparing with the 45*th* reduct, there is one redundant feature in the feature subset *Red* and the found feature subset is obviously effective. And for data set *Winequality-white*, the feature subset is $Red = \{1, 2, 3, 5, 6, 7, 8, 9, 10, 11\}$, which is identical to the first reduct in Table 3,

**Table 1**
Data sets description.

| | Data sets | Samples | Attributes | Classes |
|---|---|---|---|---|
| 1 | Mushroom | 5644 | 22 | 2 |
| 2 | Winequality-white | 4898 | 11 | 9 |

**Table 2**
All reducts on Mushroom.

| No. | Reduct |
|---|---|
| 1 | 2,3,10,11,20 |
| 2 | 2,7,10,11,15,20 |
| ⋮ | ⋮ |
| 44 | 5,17,18,19,20 |
| 45 | 5,20,21 |
| ⋮ | ⋮ |
| 155 | 5,17,19,22 |
| 156 | 5,21,22 |

**Table 3**
All reducts on Winequality-white.

| No. | Reduct |
|---|---|
| 1 | 1,2,3,5,6,7,8,9,10,11 |
| 2 | 1,2,3,4,6,7,8,9,10,11 |

**Table 4**
Feature subset *Red* by using H-FSA.

| Data set | Feature subset |
|---|---|
| Mushroom | 1,5,20,21 |
| Winequality-white | 1,2,3,5,6,7,8,9,10,11 |

that is, the found feature subset is not only an appromaxited reduct but a reduct. Hence, one can conclude that algorithm E-FSA can find an effective approximate reduct, and the proposed algorithm E-FSA is feasible.

### 4.2. Efficiency analysis

In rough set theory, finding the minimal reduct of a decision table has been proved an NP-hard problem. Thus, many heuristic reduction algorithms have been developed, which can find a single reduct from a given decision table in a shorter time. To reduce computational time further, Qian et al. in [37](published in *Artificial Intellegence*) proposed an accelerated framework to accelerate a heuristic process of attribute reduction. Based on four kinds of representative reduction algorithms which are positive-region reduction [9–11], Shannon's entropy reduction [43,45], complementary entropy reduction [24,25] and combination entropy reduction [39], four kinds of feature selection accelerated algorithms (FSPA) were devised in [37]. Note that, "FSPA" is a uniform expression of the four kinds of accelerated algorithms, not a reduction algorithm. In Section 3.2 and 3.3, by using the accelerated framework in [37], we devised the algorithm E-FSA based on these four kinds of algorithms.

Because of that, algorithm E-FSA can find a single feature subset as well, we compare in this section the computational time of E-FSA with heuristic reduction algorithms. For convenience, among the many heuristic algorithms, we also select in this section above four kinds of representative heuristic algorithms to test the efficiency of E-FSA. Because of that above four kinds of algorithms have been accelerated in [37], we compare the computational time of E-FSA and FSPA based on those four kinds of algorithms in the experiments. In addition, there may be some difference between the feature subsets found by E-FSA and FSPA. Hence, we also compare the decision performance of the feature subsets according to four common evaluation measures, which are approximate classified precision, approximate classified quality, certainty measure and consistency measure.

Approximate classified precision and approximate classified quality, in rough set theory, were defined commonly to describe the precision of approximate classification [31].

**Definition 7** [31]**.** Let $S = (U, C \cup D)$ be a decision table and $U/D = \{X_1, X_2, \ldots, X_r\}$. The approximate classified precision of $C$ with respect to $D$ is defined as

$$AP_C(D) = \frac{|POS_C(D)|}{\sum_{i=1}^{r} |\overline{C}X_i|}. \tag{13}$$

**Definition 8** [31]**.** Let $S = (U, C \cup D)$ be a decision table. The approximate classified quality of $C$ with respect to $D$ is defined as

$$AQ_C(D) = \frac{|POS_C(D)|}{|U|}. \tag{14}$$

In rough set theory, by adopting reduction algorithms, one can get reducts for a given decision table. Then, based on a reduct, a set of decision rules can be generated from a decision table. We briefly recall the notions of decision rules, which will be used in the following development.

**Definition 9** [32,38]**.** Let $S = (U, C \cup D)$ be a decision table. $U/C = \{X_1, X_2, \ldots, X_m\}$, $U/D = \{Y_1, Y_2, \ldots, Y_n\}$ and $\cap Y_j \neq \emptyset$. $des(X_i)$ and $des(Y_j)$ are denoted the descriptions of the equivalence classes $X_i$ and $Y_j$, respectively. A decision rule induced by $C$ is formally defined as

$$Z_{ij} : des(X_i) \rightarrow des(Y_j), X_i \in U/C, Y_j \in U/D. \tag{15}$$

To evaluate the decision performance, certainty measure and support measure were introduced to evaluate a single decision rule and were not suitable for measuring a rule set [3,14]. For a rule set, two measures were introduced to measure the certainty and consistency in [32]. However, in [38], it has been pointed out that those two measures cannot give elaborate

depictions of the certainty and consistency for a rule set. To address this issue, certainty measure and consistency measure were proposed to evaluate the certainty and consistency of a set of decision rules [38], which has attracted considerable attention [47].

**Definition 10** [38]. Let $S = (U, C \cup D)$ be a decision table, $U/C = \{X_1, X_2, \ldots, X_m\}$, $U/D = \{Y_1, Y_2, \ldots, Y_n\}$, and $RULE = \{Z_{ij}|Z_{ij} : des(X_i) \to des(Y_j), X_i \in U/C, Y_j \in U/D\}$ . The certainty measure $\alpha$ of the decision rules on $S$ is defined as

$$\alpha(S) = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{|X_i \cap Y_j|^2}{|U||X_i|}. \tag{16}$$

**Definition 11** [38]. Let $S = (U, C \cup D)$ be a decision table, $U/C = \{X_1, X_2, \ldots, X_m\}$, $U/D = \{Y_1, Y_2, \ldots, Y_n\}$, and $RULE = \{Z_{ij}|Z_{ij} : des(X_i) \to des(Y_j), X_i \in U/C, Y_j \in U/D\}$. The consistency measure $\beta$ of the decision rules on $S$ is defined as

$$\beta(S) = \sum_{i=1}^{m}\frac{|X_i|}{|U|}\left[1 - \frac{4}{|X_i|}\sum_{j=1}^{n}\frac{|X_i \cap Y_j|^2}{|X_i|}(1 - \frac{|X_i \cap Y_j|}{|X_i|})\right]. \tag{17}$$

In the experiments, for the feature subsets found by E-FSA and FSPA, we compare their computational time, approximate classified precision(AP), approximate classified quality(AQ), certainty measure $\alpha$ and consistency measure $\beta$. Six UCI large-scale data sets are employed to test the algorithms, which are outlined in Table 5. In these six data sets, *Ticdata2000*, *Adult* and *Connect* are preprocessed by discretization using the data tool Rosetta.

The experimental results are reported in Tables 6-13. For convenience, positive-region reduction is represented by PR, Shannon's entropy reduction is represented by SCE, complementary entropy reduction is represented by LCE and combination entropy reductio is represented by CCE. Based on these four reduction algorithms, Tables 6, 8, 10, 12 show the feature subsets of E-FSA and FSPA and Tables 7, 9, 11, 13 show the comparison of computational time and the four evaluation measures.

According to above experimental results, it is easy to see from the Tables 6, 8, 10 and 12 that the feature subsets found by E-FSA and FSPA are relatively close. And from Tables 7, 9, 11 and 13, one can observe that the values for the four evaluation

**Table 5**
Description of data sets for efficiency.

| | Data sets | Samples | Attributes | Classes |
|---|---|---|---|---|
| 1 | Ticdata2000 | 5822 | 85 | 2 |
| 2 | Nursery | 12960 | 8 | 5 |
| 3 | Letter | 20000 | 16 | 26 |
| 4 | Adult | 45222 | 14 | 2 |
| 5 | Shuttle | 58000 | 9 | 7 |
| 6 | Connect | 67557 | 42 | 3 |

**Table 6**
Comparison of feature subsets based on PR.

| Data sets | FSPA | E-FSA |
|---|---|---|
| Ticdata2000 | 2,3,5,6,7,8,9,14,15,18,30, 39,43,44,45,47,48,49,52,54, 55,57,59,61,64,68,80,83 | 2,3,4,5,7,14,15,16,18,30,31, 38,39,43,44,45,47,48,49,52, 54,55,57,59,61,62,64,68,83 |
| Nursery | 1,2,3,4,5,6,7,8 | 1,2,3,4,5,6,7,8 |
| Letter | 3,4,8,9,10,11,12,13,14,15,16 | 1,2,4,7,8,9,10,11,12,13,15 |
| Adult | 1,2,3,4,7,8,11,13 | 1,2,3,4,6,7,11,13 |
| Shuttle | 1,2,3,5 | 1,2,3,5 |
| Connect | 1,2,3,4,5,7,8,9,10,11,13,14,15, 16,17,19,20,21,22,23,25,26,27,28, 29,31,32,33,34,36,37,38,39,41 | 1,2,3,4,5,7,8,9,10,11,13,14,15, 16,17,18,19,20,21,22,24,23,25,26, 27,31,32,33,34,35,37,38,39,40,41 |

**Table 7**
Comparison of evaluation measures and computational time based on PR.

| Data sets | FSPA | | | | | E-FSA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AQ | AP | $\alpha$ | $\beta$ | Time/s | AQ | AP | $\alpha$ | $\beta$ | Time/s |
| Ticdata2000 | 0.9792 | 0.9593 | 0.9901 | 0.9803 | 296.3750 | 0.9777 | 0.9563 | 0.9894 | 0.9789 | 140.4062 |
| Nursery | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 13.3594 | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 3.4218 |
| Letter | 1.0000 | 1.0000 | 0.9999 | 1.0000 | 112.6250 | 1.0000 | 1.0000 | 0.9999 | 1.0000 | 27.3906 |
| Adult | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 1811.5313 | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 80.2500 |
| Shuttle | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 712.25 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 48.0312 |
| Connect | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 116585.7031 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 2743.53125 |

**Table 8**
Comparison of feature subsets based on SCE.

| Data sets | FSPA | E-FSA |
|---|---|---|
| Ticdata2000 | 2,5,9,18,31,37,40,43,44, 45,47,48,49,54,55,57,58, 59,61,63,64,68,80,83 | 2,5,7,9,15,18,26,27,43, 44,45,47,48,49,52,54,55, 57,59,61,62,64,68,83 |
| Nursery | 1,2,3,4,5,6,7,8 | 1,2,3,4,5,6,7,8 |
| Letter | 2,3,4,8,9,10,11,12,13,14,15,16 | 1,2,3,4,5,6,8,9,10,11,12,13,15 |
| Adult | 1,2,3,4,7,8,11,13 | 1,2,3,4,6,7,11,13 |
| Shuttle | 1,2,3,5 | 1,2,3,5 |
| Connect | 1,2,3,4,5,7,8,9,10,11,13,14,15, 16,17,19,20,21,22,23,25,26,27,28, 30,31,32,33,34,35,37,38,39,41 | 1,2,3,4,5,7,8,9,10,11,13,14,15,16, 17,19,20,21,22,23,25,26,27,28,29 30,31,32,33,34,35,36,37,38,39,41 |

**Table 9**
Comparison of evaluation measures and computational time based on SCE.

| Data sets | FSPA | | | | | E-FSA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AQ | AP | $\alpha$ | $\beta$ | Time/s | AQ | AP | $\alpha$ | $\beta$ | Time/s |
| Ticdata2000 | 0.9792 | 0.9592 | 0.9901 | 0.9803 | 1043.8906 | 0.9773 | 0.9557 | 0.9893 | 0.9785 | 494.9218 |
| Nursery | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 187.9531 | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 51.3750 |
| Letter | 1.0000 | 1.0000 | 0.9999 | 1.0000 | 2740.2500 | 1.0000 | 1.0000 | 0.9999 | 1.0000 | 745.9843 |
| Adult | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 13467.5312 | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 1461.7031 |
| Shuttle | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 10153.1719 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1907.9687 |
| Connect | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 250924.1710 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 9096.250 |

**Table 10**
Comparison of feature subsets based on LCE.

| Data sets | FSPA | E-FSA |
|---|---|---|
| Ticdata2000 | 2,5,7,15,17,31,38,43, 44,45,47,48,49,54,55,57, 58,59,61,63,64,68,80,83 | 1,2,3,5,9,15,16,17,18,19,24, 30,31,38,39,43,44,45,47,48,49,52, 54,55,57,59,60,61,62,64,68,83 |
| Nursery | 1,2,3,4,5,6,7,8 | 1,2,3,4,5,6,7,8 |
| Letter | 1,2,4,5,8,9,10,11,12,13,15,16 | 1,2,4,5,7,8,9,10,11,12,13,15,16 |
| Adult | 1,2,3,4,7,8,11,13 | 1,2,3,4,6,7,11,13 |
| Shuttle | 1,2,3,9 | 1,2,3,9 |
| Connect | 1,2,3,4,5,7,8,9,10,11,13,14,15, 16,17,19,20,21,22,23,25,26,27,28, 30,31,32,33,34,35,37,38,39,41 | 1,2,3,4,5,7,8,9,10,11,13,14,15, 16,17,19,20,21,22,23,24,25,26,27,28, 30,31,32,33,34,35,37,38,39,40,41 |

**Table 11**
Comparison of evaluation measures and computational time based on LCE.

| Data sets | FSPA | | | | | E-FSA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AQ | AP | $\alpha$ | $\beta$ | Time/s | AQ | AP | $\alpha$ | $\beta$ | Time/s |
| Ticdata2000 | 0.9792 | 0.9592 | 0.9901 | 0.9803 | 1805.5625 | 0.9777 | 0.9563 | 0.9894 | 0.9789 | 892.3125 |
| Nursery | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 336.3125 | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 98.5937 |
| Letter | 1.0000 | 1.0000 | 0.9999 | 1.0000 | 5558.7813 | 1.0000 | 1.0000 | 0.9999 | 1.0000 | 1637.8750 |
| Adult | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 23847.4375 | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 2818.7187 |
| Shuttle | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 20228.3906 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 3916.0625 |
| Connect | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 350935.7188 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 15278.0937 |

**Table 12**
Comparison of feature subsets based on CCE.

| Data sets | FSPA | E-FSA |
|---|---|---|
| Ticdata2000 | 2,5,7,15,17,31,38,43, 44,45,47,48,49,54,55,57, 58,59,61,63,64,68,80,83 | 2,3,5,7,8,15,17,18,19,30, 31,39,43,44,45,47,48,49,52,54, 55,57,59,61,62,64,68,80,83 |
| Nursery | 1,2,3,4,5,6,7,8 | 1,2,3,4,5,6,7,8 |
| Letter | 2,4,5,7,8,9,10,11,12,13,15 | 2,3,5,6,7,8,9,10,11,12,13,15,16 |
| Adult | 1,2,3,4,7,8,11,13 | 1,2,3,4,6,7,11,13 |
| Shuttle | 1,2,3,8 | 1,2,3,8 |
| Connect | 1,2,3,4,5,7,8,9,10,11,13,14,15, 16,17,19,20,21,22,23,25,26,27,28, 30,31,32,33,34,35,37,38,39,41 | 1,2,3,4,5,7,8,9,10,11,13,14,15, 16,17,19,20,21,22,23,24,25,26,27, 28,30,31,32,33,34,35,37,38,39,41 |

**Table 13**
Comparison of evaluation measures and computational time based on CCE.

| Data sets | FSPA | | | | | E-FSA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AQ | AP | $\alpha$ | $\beta$ | Time/s | AQ | AP | $\alpha$ | $\beta$ | Time/s |
| Ticdata2000 | 0.9792 | 0.9592 | 0.9901 | 0.9803 | 1048.5781 | 0.9780 | 0.9570 | 0.9896 | 0.9792 | 437.3593 |
| Nursery | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 159.0938 | 0.9531 | 0.9104 | 0.9765 | 0.9531 | 51.2343 |
| Letter | 1.0000 | 1.0000 | 0.9999 | 1.0000 | 2610.3594 | 0.9999 | 1.0000 | 0.9999 | 0.9999 | 822.7968 |
| Adult | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 12568.5625 | 0.9997 | 0.9995 | 0.9998 | 0.9997 | 1451.5156 |
| Shuttle | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 10948.9218 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 2285.3906 |
| Connect | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 249955.3288 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 9103.6406 |

**Table 14**
Description of data sets for high-efficiency.

| | Data sets | Samples | Attributes | Classes |
|---|---|---|---|---|
| 1 | Poker-hand | 1025010 | 10 | 10 |
| 2 | Covtype | 581012 | 54 | 7 |

**Table 15**
Feature subsets and computational time on larger-scale data sets.

| Data sets | Feature subsets | Computational time/s |
|---|---|---|
| Poker-hand | 1,2,3,4,6,8,10 | 1251.859375 |
| Covtype | 1,2,3,4,5,6,7,8,9,10,11,13,15,16,17,18, 19,20,24,25,26,27,28,30,31,33,34,35, 36,37,38,40,43,44,45,46,47,49,52,53,54 | 23640.9375 |

measures of the feature subsets are very close, and even identical on some data sets. Whereas, the computational time of E-FSA is much shorter than that of FSPA. Namely, the performance and decision making of the feature subsets found by the two algorithms are very close whereas H-FSA is much faster. Hence, the experimental results indicate that, compared with FSPA, the algorithm E-FSA can find a valid feature subset(an approximate reduct) in a much shorter time.

### *4.3. Efficiency analysis for large-scale data sets*

From the experimental results in the previous subsections, one can see that the algorithm E-FSA can find an effective feature subset in a much shorter time. To further demonstrate the efficiency, we employ in this section two UCI very larger-scale data sets to conduct the experiment, which are outlined in Table 14. It should be pointed out that, by using some representative heuristic reduction algorithms including FSPA, these two data sets are too large in scale to get the feature subset within 100 h on a PC. In this section, we carry out the algorithm E-FSA on these two large-scale data sets and the experimental results are given in Table 15.

The experimental results indicate that, for those two very large-scale data sets, E-FSA can find their feature subsets within just 1251.859375 s (0.35 h) and 23640.9375 s (6.6 h) on a PC, respectively. Hence, algorithm E-FSA is efficient, especially for large-scale data sets.

## 5. Conclusions

At present, feature selection for large-scale data sets is still a challenging issue in the field of artificial intelligence. In this paper, with some concepts in statistics, an efficient rough feature selection algorithm has been proposed to deal with large-scale decision tables. The algorithm found a valid feature subset though dividing a large-scale table into small ones and fusing the feature selection results of small tables together. The experimental analysis shows that the proposed algorithm is feasible and efficient. Note that the proposed algorithm not only saves computational time, but also can handle some large-scale data sets that are very difficult to deal with on a PC because of the high computational time. It is our wish that the idea of dividing and fusing on data sets provides a new view and thinking on dealing with large-scale data sets in applications.

## Acknowledgment

This work was supported by National Natural Science Fund of China (Nos. 71031006, 60903110, 70971080), the National Key Basic Research and Development Program of China (973) (No. 2011CB311805), the Research Fund for the Doctoral Program of Higher Education (20101401110002), the Key Technologies R&D Program of Shanxi province (20110321027-01).

## References


[1] J.G. Bazan, Dynamic reducts and statistical inference, in: Proceedings of 5th International Conference Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96, Granada, Spain, 1996, pp. 1147–1151.
[2] J.G. Bazan, A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables, in: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery, Physica-Verlag, Heidelberg, 1998, pp. 321–365.
[3] I. Düntsch, G. Gediga, Uncertainty measures of rough set prediction, Artificial Intelligence 106 (1998) 109–137.
[4] S. Dick, A. Schenker, W. Pedrycz, A. Kandel, Regranulation: A granular algorithm enabling communication between granular worlds, Information Science 177 (2) (2007) 408–435.
[5] M. Dash, H. Liu, Feature selection for classification, Intelligent Data Analysis 1 (1997) 131–156.
[6] M. Dash, H. Liu, Consistency-based search in feature selection, Artificial Intelligence 151 (2003) 155–176.
[7] I. Guyon, A. Elisseeff, An introduction to variable feature selection, Machine Learning Research 3 (2003) 1157–1182.
[8] J.W. Guan, D.A. Bell, Rough computational methods for information systems, Artificial Intelligence 105 (1998) 77–103.
[9] J.W. Grzymala-Busse, An algorithm for computing a single covering, in: J.W. Grzymala-Busse (Ed.), Managing Uncertainty in Expert Systems, Kluwer Academic Publishers., Netherlands, 1991, pp. 66
[10] J.W. Grzymala-Busse, LERSla system for learning from examples based on rough sets, in: R. Slowinski (Ed.), Intelligent Decision Support: Handbook of Applications and A dvances of the Rough Set Theory, Kluwer Academic Publishers, Netherlands, 1992, pp. 3–18.
[11] X.H. Hu, N. Cercone, Learning in relational databases: a rough set approach, International Journal of Computational Intelligence 11 (2) (1995) 323–338.
[12] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recognition 40 (2007) 3509–3521.
[13] Q.H. Hu, D.R. Yu, Z.X. Xie, Information-preserving hybrid data reduction based on fuzzy-rough techniques, Pattern Recognition Letters 27 (5) (2006) 414–423.
[14] V.N. Huynh, Y. Nakamori, A roughness measure for fuzzy sets, Information Sciences 173 (2005) 255–275.
[15] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, IEEE Transactions on Knowledge and Data Engineering 16 (12) (2004) 1457–1471.
[16] R. Jensen, Q. Shen, Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches, IEEE Press/Wiley & Sons, Canada, 2008.
[17] J.P. Jia, Principles of Statistics, Fourth ed., China Renmin University Publishing, Beijing, 2009.
[18] Y.J. Jin, Z.F. Du, Y. Jiang, Sampling technique, China Renmin University Publishing, Beijing, 2008.
[19] G. Kader, M. Perry, Variability for Categorical Variables, Journal of Statistics Education 15 (2) (2007)
[20] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.
[21] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, Proceedings. AAAI 92 (1992) 129–134.
[22] M. Kryszkiewicz, P. Lasek, FUN: fast discovery of minimal sets of attributes functionally determining a decision attribute, Transactions on Rough Sets 9 (2008) 76–95.
[23] C.K. Lee, G.G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Information Processing and Management 42 (2006) 155–165.
[24] J.Y. Liang, K.S. Chin, C.Y. Dang, C.M. Yam Richid, A new method for measuring uncertainty and fuzziness in rough set theory, International Journal of General Systems 31 (4) (2002) 331–342.
[25] J.Y. Liang, Z.B. Xu, The algorithm on knowledge reduction in incomplete information systems, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (1) (2002) 95–103.
[26] M. Modrzejewski, Feature selection using rough set theory, in: Proceedings of European Conference on Machine Learning, 1993, pp. 213–226.
[27] D.Q. Miao, C. Gao, N. Zhang, Z.F. Zhang, Diverse reduct subspaces based co-training for partially labeled data, International Journal of Approximate Reasoning 52 (2011) 1103–1117.
[28] N.S. Nguyen, Approximate boolean reasoning: foundations and applications in data mining, Lecture Notes in Computer Science 3100 (2006) 334–506.
[29] J.X. Ni, Sampling survey, Guangxi Normal University Press, 2002.
[30] W. Pedrycz, G. Vukovich, Feature analysis through information granulation and fuzzy sets, Pattern Recognition 35 (2002) 825–834.
[31] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers., Boston, 1991.
[32] Z. Pawlak, Rough set theory and its applications in data analysis, Cybernetics and Systems 29 (1998) 661–688
[33] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Information Sciences 177 (1) (2007) 41–73.
[34] M. Perry, G. Kader, Variation as Unalikeability, Teaching Statistics 27 (2) (2005) 58–60.
[35] J. Qian, D.Q. Miao, Z.H. Zhang, W. Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, International Journal of Approximate Reasoning 52 (2011) 212–230.
[36] Y.H. Qian, J.Y. Liang, C.Y. Dang, Knowledge structure, knowledge granulation and knowledge distance in a knowledge base, International Journal of Approximate Reasoning 50 (1) (2009) 174–188.
[37] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artificial Intelligence 174 (2010) 597–618.
[38] Y.H. Qian, J.Y. Liang, D.Y. Li, H.Y. Zhang, C.Y. Dang, Measures for evaluating the decision performance of a decision table in rough set theory, Information Sciences 178 (2008) 181–202.
[39] Y.H. Qian, J.Y. Liang, Combination entropy and combination granulation in rough set theory, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 16 (2) (2008) 179–193.
[40] R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.
[41] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters 24 (2003) 833–849.
[42] A. Skowron, Extracting laws from decision tables: a rough set approach, Computational Intelligence 11 (1995) 371–388.
[43] D. Slezak, Approximate entropy reducts, Fundamenta Informaticae 53 (3 - 4) (2002) 365–390.
[44] D. Tian, X.J. Zeng, J. Keane, Core-generating approximate minimum entropy discretization for rough set feature selection in pattern classification, International Journal of Approximate Reasoning 52 (2011) 863–880.
[45] G.Y. Wang, H. Yu, D.C. Yang, Decision table reduction based on conditional information entropy, Chinese Journal of Computer 25 (7) (2002) 759–766.
[46] G.Y. Wang, J. Zhao, J.J. An, A comparative study of algebra viewpoint and information viewpoint in attribute reduction, Fundamenta Informaticae 68 (3) (2005) 289–301.
[47] W. Wei, J.Y. Liang, Y.H. Qian, F. Wang, C.Y. Dang, Comparative study of decision performance of decision tables induced by attribute reductions, International Journal of General Systems 39 (8) (2010) 813–838.
[48] S.X. Wu, M.Q. Li, W.T. Huang, S.F. Liu, An improved heuristic algorithm of attribute reduction in rough set, Journal of System Sciences and Information 2 (3) (2004) 557–562.
[49] Z.Y. Xu, Z.P. Liu, B.R. Yang, W. Song, A quick attribute reduction algorithm with complexity of $\max(O(|C||U|), O(|C|^2|U/C|))$, Chinese Journal of Computer 29 (3) (2006) 391–398.
[50] Y.Y. Yao, Probabilistic rough set approximations, International Journal of Approximate Reasoning 49 (2) (2008) 255–271.
[51] Y.Y. Yao, Y. Zhao, Discernibility matrix simplification for constructing attribute reducts, Information Sciences 179 (5) (2009) 867–882.
[52] Y.Y. Yao, Y. Zhao, Attribute reduction in decision-theoretic rough set models, Information Sciences 178 (17) (2008) 3356–3373.