Introduction to Rough Sets Theory

Introduction

Rough sets theory was first introduced by Pawlak in the 1980's [1]. An early application of rough sets theory to knowledge discovery systems was introduced to identify and remove redundant variables, and to classify imprecise and incomplete information. Reduct and core are the two important concepts in rough sets theory. Based on Pawlak's book [1], we explain the basic concepts in rough sets theory in the following.

1. Decision Table

A data set can be represented as a decision table, which is used to specify what conditions lead to decisions. A decision table is defined as \( T = (U, C, D) \), where \( U \) is the set of objects in the table, \( C \) is the set of the condition attributes and \( D \) is the set of the decision attributes. We show an example of a decision table in the following Table 1.

<table>
<thead>
<tr>
<th>ID</th>
<th>make</th>
<th>model</th>
<th>cyl</th>
<th>door</th>
<th>displac</th>
<th>compress</th>
<th>power</th>
<th>trans</th>
<th>weight</th>
<th>Mileage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>USA</td>
<td>6</td>
<td>2</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>auto</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>USA</td>
<td>6</td>
<td>4</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>manual</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>USA</td>
<td>4</td>
<td>2</td>
<td>small</td>
<td>high</td>
<td>medium</td>
<td>auto</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>USA</td>
<td>4</td>
<td>2</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>manual</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>USA</td>
<td>4</td>
<td>2</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>manual</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>USA</td>
<td>6</td>
<td>4</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>auto</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>USA</td>
<td>4</td>
<td>2</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>auto</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>USA</td>
<td>4</td>
<td>2</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>manual</td>
<td>light</td>
<td>high</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Japan</td>
<td>4</td>
<td>2</td>
<td>small</td>
<td>high</td>
<td>low</td>
<td>manual</td>
<td>light</td>
<td>high</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Japan</td>
<td>4</td>
<td>2</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>manual</td>
<td>medium</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Japan</td>
<td>4</td>
<td>2</td>
<td>small</td>
<td>high</td>
<td>high</td>
<td>manual</td>
<td>medium</td>
<td>high</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Japan</td>
<td>4</td>
<td>2</td>
<td>small</td>
<td>medium</td>
<td>low</td>
<td>manual</td>
<td>medium</td>
<td>high</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Japan</td>
<td>4</td>
<td>2</td>
<td>small</td>
<td>high</td>
<td>medium</td>
<td>manual</td>
<td>medium</td>
<td>high</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>USA</td>
<td>4</td>
<td>2</td>
<td>small</td>
<td>high</td>
<td>medium</td>
<td>manual</td>
<td>medium</td>
<td>high</td>
<td></td>
</tr>
</tbody>
</table>

This table shows an artificial data set about the cars [2]. The mileage of a car is related to the model of the car, the number of cylinders, the number of doors, the displacement, the compression, the power, the transmission, and the weight of the car.
Table 1 can be used to decide whether a car has a high or medium mileage according to its features (e.g., the model, the transmission and the weight). For example, the first row of this table specifies that a USA car, with 6 cylinders, 2 doors, medium displacement, high compression, high power, automatic transmissions, and medium weight, has a medium mileage.

The rows in this table are called the objects, and the columns in this table are called attributes [3]. Condition attributes are the features of a car related to its mileage; therefore, \( C = \{\text{make}_\text{model}, \text{cyl}, \text{door}, \text{displace}, \text{compress}, \text{power}, \text{trans}, \text{weight}\} \). Mileage is the decision attribute; therefore, \( D = \{\text{Mileage}\} \). There are 14 objects in this data, and there do not exist missing attribute values.

Here we only look at the situation when the value of the decision attributes is binary. And we will not discuss the situation when the condition attributes have missing values.

The reduct and the core are important concepts in rough sets theory. Reduct sets contain all the representative attributes from the original data set. A reduct contains a subset of condition attributes that are sufficient to classify the decision table. A reduct may not be unique. The core is contained in all the reduct sets, and it is the necessity of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

### Table 2 Multiple Reducts for the Artificial Car Data Set

<table>
<thead>
<tr>
<th>No.</th>
<th>Reduct Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{ make_model, compress, power, trans }</td>
</tr>
<tr>
<td>2</td>
<td>{ make_model, cyl, compress, trans }</td>
</tr>
<tr>
<td>3</td>
<td>{ make_model, displace, compress, trans }</td>
</tr>
<tr>
<td>4</td>
<td>{ make_model, cyl, door, displace, trans, weight }</td>
</tr>
</tbody>
</table>

Table 2 shows the reducts of the car data set generated by the ROSETTA software [4]. For example, a reduct can be a set of condition attributes containing \{the model, the compression, the power and the transmissions\} of a car. With this reduct, all the 14 objects can be correctly classified completely (according to their mileage type). A subset of \{ make_model, cyl \} is not a reduct of this car data, because with only these two attributes one cannot fully classify all the objects; in addition, there exists redundancy and contradictions. For example, in Table 1, with a subset of \{ make_model, cyl \}, we cannot classify object No.7 and No.8. They both describe USA cars with 4 cylinders, but they have different mileage.

A reduct is often used in the attribute selection process to reduce unnecessary attributes towards decision making applications. According to the reduct No.1 in Table 2, one can generate a rule, i.e., a USA car with high compression, high power and automatic transmission has medium mileage, which is more succinct than a rule specifying that a USA car with 6 cylinders, 2 doors, medium displacement, high compression, high power, automatic transmissions and medium weight, has medium mileage.
Core attributes are the essential information in a data set. The core attributes are contained by all the reducts. From Table 2, we can see the intersection of all the listed reducts is as follows.

\[ \{ \text{make\_model, trans} \} \]

This set contains the core attributes. Core attributes can be obtained by the core generation algorithms proposed by Hu et al. [3], discussed in greater detail in the following section 2.

2. Rough Sets based Knowledge Discovery Systems

We briefly survey current rough sets based knowledge discovery systems. We discuss the individual functions of each system based on general characteristics, such as the data sets, the preprocessing tasks, the related rough sets tasks, the rule generations and so on.

- **ROSETTE** ROSETTA [4] is freely distributed. Downloadable versions for both the Windows and Linux operating systems are available. The software supports the complete data mining process, from data preprocessing, including handling incomplete data, data discretization, generating reduct sets which contain essential attributes for the given data set, to classification, rule generation, and cross validation evaluation. Some discretization and reducts generation packages are from the RSES library [5].

- **RSES2.2** RSES[5] stands for Rough Set Exploration System. There are downloadable versions for both the Windows and Linux operating systems. It is still maintained and being developed. The system supports data preprocessing, handling incomplete data, data decomposition, reducts generation, classification, and cross validations.

- **ROSE2** ROSE[6] stands for Rough Sets Data Explorer. This software is designed to process data with large boundary regions. The software supports data preprocessing, data discretization, handling missing values, core and reducts generation, classifications and rule generation, as well as evaluations. This software provides not only the classical rough set model, but also the variable precision model, which is not provided by [4] and [5].

- **LERS** LERS [7] stands for Learning from Examples based on Rough Sets. It is not publicly available. The system was designed especially to process missing values of attributes and inconsistency in the data set. Certain rules and possible rules are both extracted based on the lower and upper approximations.
In addition to the rough sets based systems mentioned above, there are other available knowledge discovery systems based on the methodologies of rough sets such as DBROUGH [8] and GROBIAN [9].

3. Current Reduct Generation and Core Generation Approaches

As discussed earlier, a reduct of a decision table is a set of condition attributes that is sufficient to define the decision attributes. A reduct does not contain redundant attributes towards a classification task. It is often used in the attribute selection process to reduce the redundant attributes, and to reduce the computation cost for rule generations. There may exist more than one reduct for each decision table. Finding all the reduct sets for a data set is NP-hard [10]. Approximation algorithms are used to obtain reduct sets [11]. The intersection of all the possible reducts is called the core. The core is contained in all the reduct sets, and it is the essential part of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

Previously, many research efforts on designing reduct generation and core generation approaches have been proposed. In this section, we summarize a few current algorithms and software that are commonly used.

3.1 ROSETTE For reduct generation in ROSETTA GUI version 1.4.41., the software provides Genetic reducer, Johnson reducer, Holte1R reducer, Manual reducer, Dynamic reducer, RSES Exhaustive reducer and so on. Genetic reducer is an approximation algorithm based on a genetic algorithm for multiple reducts generation. The Johnson reducer generates only a single reduct with minimum number of attributes.

3.2 RSES RSES provides a genetic algorithm to control the number of reducts generated, which is appropriate for larger data sets to only generate representative reducts.

Note that there are other reduct generation approaches provided by some other software such as ROSE2 [6]. In ROSE2 software, there are three reduct generation functions, the “lattice search”, “heuristic search” and “manual search” approaches. The “lattice search” approach for reduct generation is used when the expected number of reducts are rather small. The other two reduct generations support larger datasets although they require domain experts' knowledge on the data for choosing attributes.

3.3 Hu’s Reduct and Core Generation Hu et al. [3] proposed a new rough set model based on database operations such as cardinality and projection. By combining a relational algebra with the rough sets theory, the approach is designed to increase the efficiency of the core and reduct computation. A reduct is redefined based on the database operations.

The reduct is defined to be a subset \( \text{REDU} \subseteq \text{C} \) of condition attributes with respect to the decision attribute \( D \) where \( \text{REDU} \) is a minimum subset of attributes that has the same classification power as the entire condition attributes.
Let $K(REDU,D)$ be the proportion of the data instances in the decision table that can be classified. $K$ is also defined to be the degree of dependency between $REDU$ and the decision attribute $D$, and is the stopping criteria for the algorithm, as shown in Eq.3.3.1. $Card$ denotes the count operation in databases, and $\Pi$ denotes the projection operation in databases.

$$K(REDU,D) = \frac{Card(\Pi(REDU + D))}{Card(\Pi(C + D))} \quad (3.3.1)$$

A measure of merit value is defined to evaluate the effect of each condition attribute on the decision attribute $D$. For a condition attribute $C_i \in C$, the merit of $C_i$ can be calculated by

$$Merit(C_i,C,D) = 1 - \frac{Card(\Pi(C - \{C_i\} + D))}{Card(\Pi(C + D))} \quad (3.3.2)$$

During the reduct generation, the condition attribute with the highest merit value at the moment is included in the reduct. In case multiple highest merit values exist, the condition attribute with the least combination with other attributes in the current reduct is selected. The algorithm iterates until the minimum set of attributes which is as representative as the entire condition attributes is obtained. The reduct generation algorithm is shown in Algorithm 1. The reduct generation is designed to guarantee that the generated reduct will have the minimum number of attributes.

---

**Algorithm 1: Hu’s Reduct Generating Algorithm**

**input**: Decision table $T = (U,C,D)$, $C$ is the condition attributes set; $D$ is the decision attribute set.

**output**: $REDU$, reduct of $C$.

1. Core Generation Algorithm to generate Core;
2. $REDU = Core$;
3. $AR = C - REDU$;
4. for each attribute $C_i \in AR$ do
   5. $Merit(C_i,C,D) = 1 - \frac{Card(\Pi(C - \{C_i\} + D))}{Card(\Pi(C + D))}$
   6. end
5. maximum $Merit(C_i,C,D)$;
   /* In case there are several attributes with the same merit value, choose the attribute which has the least number of combinations with those attributes in $REDU$. minimum $Card(\Pi(C_i + REDU))/*$*/
6. $REDU = REDU + \{C_i\}$, $AR = AR - \{C_i\}$;
7. if $K(REDU,D) = 1$ then return $REDU$;
8. else go to Step 4

Recall that the core represents the most important information of the original data set, all reducts contain the core.
Since it is infeasible to obtain the core attributes by intersecting all the possible reducts, other approaches are proposed to generate the core attributes. Hu et al. [3] introduced a core generation algorithm based on rough sets theory and efficient database operations, without generating reducts. The algorithm is shown in Algorithm 2, where $C$ is the set of condition attributes, and $D$ is the set of decision attributes.

### Algorithm 2: Core Generating Algorithm

**input**: Decision table $T = (C, D)$, $C$ is the condition attributes set; $D$ is the decision attribute set.

**output**: $Core$, Core attribute set.

1. $Core \leftarrow \emptyset$;
2. for each condition attribute $A \in C$ do
3. \hspace{1em} if $\text{Card}(\Pi(C - \{A\} + D)) \cdot \text{Card}(\Pi(C - \{A\}))$ then
4. \hspace{2em} $Core = Core + \{A\}$;
5. \hspace{1em} end
6. end
7. return $Core$

This algorithm is developed to consider the effect of each condition attribute on the decision attribute. The intuition is that, if the core attribute is removed from the decision table, the rest of the attributes will bring different information to the decision making. A theoretical proof of this algorithm is provided in [3]. The algorithm takes advantage of efficient database operations such as count and projection. Since the attributes of the core are contained in any reduct sets of a data set, this algorithm also provides an evaluation to justify the correctness of the reduct sets.

There are other reduct generation approaches such as the QuickReduct algorithm, which was first applied in information retrieval systems to reduce the dimensions of the input text data [12]. The algorithm uses the same stopping criteria of the degree of dependency as Eq. 3.3.1 to select a reduct. Comparing to Hu's reduct generation, this algorithm initializes the reduct set with an empty set, whereas for Hu's approach the reduct set is initialized to be the core set. Note that Hu's reduct generation and QuickReduct generation do not always produce the minimum reduct [13]. Recent research [14] indicates an addition-only strategy normally will not produce a minimum reduct.

### References


