A New Method for Constructing Decision Tree Based on Rough Set Theory

Wejun Wen
College of Information Science & Technology, Qingdao University of Science & Technology, QingDAO, China
Email: w_wjun@163.com

Abstract—Decision tree induction algorithms have been used for classification in a wide range of application domains. In the process of constructing a tree, the criteria of selecting test attributes will influence the classification ability of the tree. The paper presents a new method of constructing decision trees based on rough set that uses a core attribute and a discernibility matrix to select the attribute with the maximal contribution to classification. It overcomes the anomalies associated with ID3 while providing superior classification.

II. BASIC CONCEPTS OF ROUGH SET THEORY

Definition 2.1. In rough set theory, an information system is defined as \( S = (U,Q,V,f) \), where \( U \) is a finite set of objects called the universe, \( Q \) is a finite set of attributes, \( V = V_a \), \( \forall q \in Q \), \( V_q \) is a value of the attribute \( q \), and \( f: U \times Q \rightarrow V \) is the total decision function (also called the information function) such that \( f(x,q) \in V_q \) for every \( q \in Q, x \in U \). A decision table is an information system where \( Q = C \cup D \). \( C \) is the set of conditional attributes and \( D \) is the set of decision attributes. In the information system, a subset \( A \subset Q \) is called the indiscernibility relationship, denoted by \( \text{IND}(A) \), which is defined as

\[
\text{IND}(A) = \{(x,y) \in U \times U | \forall a \in A, f(x,a) = f(y,a)\}
\]

(2.1)

\( \text{IND}(A) \) is an equivalence relationship that partitions \( U \) into equivalence classes, the sets of objects that are indiscernible with respect to \( A \). Sets of such partitions are denoted by \( U/\text{IND}(A) \).

Definition 2.2. A reduct is the minimum set of attributes that preserves the indiscernibility relationship. The relative reduct of the attribute set \( P \), \( P \subset Q \) is called the reduct of \( Q \) denoted by \( \text{RED}_v(P) \) if \( P \) is optimal among all subsets of \( Q \). The intersection of all reducts of \( Q \) is called the core of \( Q \), denoted by \( \text{CORE}(Q) \).

If \( a \in P \) and \( a \notin \text{CORE}(Q) \), then the decision performance of the original system will remain unchanged if attribute \( a \) is deleted from \( P \). Otherwise, the decision performance of the original system will change. The reduct and the core make the attribute core set very important in decision making, and we can use it to create simpler rules for an information system.

Definition 2.3. Skowron and Rauszer[10] proposed the discernibility matrix as a way to represent knowledge. Let \( S = (U,Q,V,f) \) be an information system with \( U = \{x_1, x_2, ... , x_n\} \). For the discernibility matrix of \( S \), denoted\( (S) \), the \( n \times n \) matrix is defined as

\[
(e_{ij}) = \{a \in Q : a(x_i) \neq a(x_j)\} \text{ for } i, j = 1, 2, 3,...n
\]

(2.2)
Thus, $c_{ij}$ is the set of all attributes that discern objects $x_i$ and $x_j$. In a discernibility matrix, the diagonal elements are because $c_{ii} = c_i$. Therefore, the upper triangular part can be omitted.

**Definition 2.4.** Approximation in rough set theory is another major concept. $Q = C \cup D$ is called the attribute set, with $A \subset Q$, $X \subset U$ is the object set of interest. The lower approximation of $X$ is defined as

$$A(X) = \bigcup \{ Y \in U / A : Y \subset X \neq \emptyset \}$$

(2.3)

The upper approximation of $X$ is

$$\overline{A}(X) = \bigcup \{ Y \in U / A : Y \cap X = \emptyset \}$$

(2.4)

The boundary region is

$$BN_a = A(X) - \overline{A}(X) .$$

(2.5)

### III. ALGORITHM: ROUGH SET-BASED DECISION TREE USING A CORE AND ENTITY COMPARISON

In this section, we propose a new method to create a decision tree using a rough set-based method. We use a discernibility matrix to find the core attribute, and compare objects to select the attributes that contribute most to the classification.

Previous studies on decision trees using rough sets have included other concepts such as entropy. While our work also uses the concept of rough set theory, we consider attributes that are distinguishable between objects. Our underlying concept is that the more unique an attribute is, the greater its contribution to classification will be. Table 1 helps explain the criteria for selecting attributes that contribute maximally to the classification. It shows four cases that resulted from comparing two objects for the condition attributes value and the class.

Let $C$ be a condition attribute set, $C = \{c_1, c_2, ... c_n\}$, and $D$ be a decision attribute set, $D = \{d_1, d_2, ... d_l\}$.

**TABLE I. COMPARISON OF TWO OBJECTS**

<table>
<thead>
<tr>
<th>Condition case of object $x_i$</th>
<th>Condition attributes value</th>
<th>Decision attributes value</th>
<th>Judgment of condition attribute $c_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>comparison of object $x_i$</td>
<td>same</td>
<td>same</td>
<td>positive</td>
</tr>
<tr>
<td>and object $x_j$</td>
<td>2 same</td>
<td>different</td>
<td>negative</td>
</tr>
<tr>
<td></td>
<td>3 different</td>
<td>same</td>
<td>negative</td>
</tr>
<tr>
<td></td>
<td>4 different</td>
<td>different</td>
<td>positive</td>
</tr>
</tbody>
</table>

Next, we describe our method, which uses core attributes, a discernibility matrix, and a contribution function instead of an entropy function.

The discernibility matrix of an information system is an $n \times n$ matrix with entries $c_{ij}$ defined as

$$c_{ij} = \{ a \in Q : a(x_i) \neq a(x_j) \}$$

if $d(x_i) \neq d(x_j)$ for $i, j = 1, 2, 3, ... n$, where $a$ is a condition attribute and $d$ is a decision attribute. In this case, the condition attribute $a$ is judged a positive case. The discernibility matrix suggested by Skowron and Rauszer\[8\] presents the set of relative reducts, i.e. the family of all minimal conditional attribute sets sufficient to discern between objects from different decision classes in the case of consistent decision tables. Our suggested algorithm uses Skowron and Rauszer’s\[9\] idea but also considers relationships between objects in the same class. We define this as

$$c_{ij} = \{ a \in A : a(x_i) \neq a(x_j) \}$$

if $d(x_i) = d(x_j)$ for $i, j = 1, 2, 3, ... n$, this is judged to be a negative case.

Entry $c_{ij}$ is the set of all condition attributes that discern objects $x_i$ and $x_j$. Based on the above discussion, our classification contribution function is defined as follows:

1. **Positive case**

   $$CC_{p}(a_i) = \frac{\sum_{j=1}^{n} I(c_i \cap a_j)}{n(c_i)} , \text{ where } c_i \cap a_j \neq \emptyset \quad (3.1)$$

   where $I(c_i \cap a_j)$ is the index function of $(c_i \cap a_j)$. If condition attribute $a_i$ is an element of $c_i$, then $I(c_i \cap a_j) = 1$; otherwise it is 0. $CC_p$ denotes the classification contribution with the positive case.

2. **Negative case**

   $$CC_{n}(a_i) = -\sum_{j=1}^{n} \frac{I(c_i \cap a_j)}{n(c_i)} , \text{ where } c_i \cap a_j \neq \emptyset \quad (3.2)$$

   $CC_n$ denotes the classification contribution with the negative case.

3. **Classification contribution function**

   The classification contribution is the sum of the positive and negative cases for the condition attribute $a_i$.

   $$CC_{a_i} = \left[ -\sum_{j=1}^{n} \frac{I(c_i \cap a_j)}{n(c_i)} \cdot \left[ d(x_i) \neq d(x_j) \right] \right] - \left[ \sum_{j=1}^{n} \frac{I(c_i \cap a_j)}{n(c_i)} \cdot \left[ d(x_i) = d(x_j) \right] \right]$$

   where $c_i \cap a_j \neq \emptyset$. \( (3.3) \)

From (3.3), we can select the attribute with the maximum $CC_{a_i}$. That attribute makes the maximal contribution to the classification.

In the rough set view, three cases appear in the discernibility matrix. The first case has no core attributes, the second case has only one core attribute, and third case has more than one core attribute.

Based on the above discussion, our algorithm generates a decision tree in the following way:

1. Make the discernibility matrix. The three cases are the following:

   - Case (a-1) involving no core attributes,
   - Case (b-1) involving only one core attribute, and
   - Case (c-1) involving more than one core attribute.

2. Node condition:
Case (a-2): Measure the classification contribution \( CC_{a} \) for each attribute in the reduct set and select the condition attribute with the maximal value of \( CC_{a} \) as a node.

Case (b-2): Select the core attribute as a node.

Case (c-2): Measure the classification contribution \( CC_{a} \) for each core attribute and select the condition attribute with the maximal value of \( CC_{a} \) as a node.

(3) Select each expanding attribute as a node for each level.

Repeat steps 1 and 2 recursively until all objects in a node belong to the same class.

IV. EXAMPLE ANALYSIS

We can use Table II to make a discernibility matrix.

TABLE II. SET OF EXAMPLE

<table>
<thead>
<tr>
<th>No</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

The core can be defined as the set of all single-element entries of the discernibility matrix. It reveals two core attributes \{C,D\}. We can calculate the classification contribution of C and D as follows:

\[
CC_{c}(C) = \left(-1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \right) = -8.17
\]

\[
CC_{c}(D) = \left(-1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \right) = -8.17
\]

\[
CC_{c} = 6.58 - 8.17 = -1.59
\]

The classification contribution of C is the sum of \( CC_{c}(C) \) and \( CC_{c}(C) \).

\[
CC_{c} = 6.58 - 8.17 = -1.59
\]

Using the same method, we can calculate the classification contribution of the condition attribute D as follows:

\[
CC_{c}(D) = 12.5,
\]

\[
CC_{p}(D) = 7.08,
\]

When \( CC_{c} \) is greater than \( CC_{p} \), attribute C is selected as the root node based on the criteria of the classification contribution function. Other attributes are tested on branches C=2 and C=3, respectively, and if C=1, then d=1. This produces the subtree shown in Fig. 1.

![Figure 1. The generated subtree.](image1)

We can apply the same process for subsets C=2 and C=3 until they satisfy the node condition. The resulting decision tree for this example is shown in Fig. 2.

![Figure 2. Decision tree construction using the proposed algorithm.](image2)

This simple example clarifies how our algorithm achieves the results in a decision tree with fewer leaves than a tree constructed using an ID3.

V. CONCLUSIONS

In this study, we used rough set theory to select attributes as nodes of a decision tree at each level of expansion. We formally proved that the problems involved in constructing an optimal classification decision tree are NP-complete. We presented a proposed classification decision tree algorithm based on core attributes and reduct to provide an algorithm that is heuristically superior to ID3 and C4.5. In the rough set view, the core and reduct are essential attributes for classification. Our proposed method uses these attributes together to overcome the anomalies common in entropy-based decision tree construction, while providing superior classification. In most cases, we found that our proposed algorithm was more effective than the ID3 algorithm and provided better accuracy than both ID3 and C4.5.

The time to compute the reduct set can be prohibitively long when the decision table has many attributes or different values of attributes and objects. The reason for this is that in general, the size of the reduct set is
exponential with respect to the size of the table, and the problem of computing of a minimal reduct is NP-hard.

Bazan et al. suggest a more efficient method for reducing reduct computation. Further work is required to include this reduction in our proposed method.

REFERENCES
