A Novel Approach of Rough Conditional Entropy-Based Attribute Selection for Incomplete Decision System

1. Introduction

Rough set theory, proposed by Pawlak [1–3], is an extension of set theory for the study of the intelligent systems characterized by uncertain, imprecise, incomplete, and inconsistent data. It has been proven to be an innovative and efficient mathematic tool, compared with other traditional data processing strategies like PCA, neural networks, SVM and so forth [4–7]. Unlike those methods, rough set theory allows knowledge discovering process to be conducted automatically by the data themselves without any dependence on the prior knowledge. By using the concepts of upper and lower approximations in rough set model to deal with the data, knowledge hidden in information systems could be discovered and expressed in the form of decision rules. Rough set methodology presents a novel paradigm to deal with uncertainty and then it has been successfully applied into feature selection [8], rule extraction [9, 10], uncertainty reasoning [11–13], decision evaluation [14], granular computing [15,16], and so on.

It has been known that Pawlak’s classical rough set model can only be used to tackle the problems of complete information systems [17]. Nevertheless, because of the error of data measuring, the impreciseness from the limitation of data acquisition manners and other probable factors, in the real database it is inevitable to meet the empty values, which stand for the inaccessible information in the database for the moment. In other words, the incomplete information systems with missing values often exist in practical knowledge acquisition. For now, two main approaches have been proposed to cope with incomplete information systems. One is indirect approach, which transforms an incomplete information system into a complete information system by eliminating objects with missing values or filling up missing values with processed data. The other is direct approach, which extends some basic notions in the classical rough set models [18,19]. In the past decade, with respect to different requirements, various extensions of the rough set models have been proposed, such as variable precision rough set models [20,21], rough set models based on tolerance relation [22,23], fuzzy rough set models [24], and weighted attributes rough set models [25].

In many fields such as data mining, machine learning, and pattern recognition, data sets containing huge numbers
2 Mathematical Problems in Engineering

of attributes can often be encountered. In such cases, feature selection, or attribute reduction as we know, is necessary. It is well known that irrelevant and redundant attributes in input attributes not only complicate the problem but also degrade solution accuracy [26, 27]. As a significant step among data preprocessing procedures, the main objective of attribute selection is to determine a minimal attribute subset, which is also called reduction, from a problem domain, which can retain a relatively high accuracy in replacing the original attributes.

In comparison with the study of attribute selection from complete information systems, scant effort has been made to develop the tolerance relation-based methods of attribute selection for incomplete information systems [28–31]. Furthermore, there is a lack of the study of attribute selection of incomplete decision systems, compared with the above. As an effective method for attribute selection, rough set can preserve the meaning of the attributes. The essence of rough set approach for attribute selection is to find a subset of attributes, which can predict the decision concepts as well as the original attribute set. So far, there have been some new widespread approaches which are usually considered to develop the classical rough set theory as follows: tolerance-based rough set model [32], covering rough set model [33, 34], and dominance-based rough set model [35, 36]. However, they still have some inherent disadvantages and are not suitable for attribute selection of incomplete decision systems.

As we know, finding the minimal reduct in an incomplete decision system is an NP-hard problem [37]. The general method of solving this kind of problems is adopting a heuristic search, which always depends on the measurements associated with the attributes [38]. However, little investigation has addressed the issue of measuring the uncertainty of knowledge of the tolerance relation-based rough set models in incomplete decision system until now. Hence, a further study on uncertainty measures applicable for evaluating the roughness and accuracy of a set in an incomplete decision system is of both theoretical and practical importance.

The main aim of this paper is to construct an effective uncertainty measure evaluating the roughness and accuracy of knowledge to find a heuristic attribute selection algorithm of incomplete decision systems. The rest of this paper is organized as follows. In Section 2, we briefly review some fundamental concepts concerning the main subject of this paper. Section 3 introduces the concept of entropy-based uncertainty measures and demonstrates several heuristic attribute selection algorithms of incomplete decision systems. Experimental comparisons and results are illustrated and analyzed in Section 4. Finally, some conclusions are presented in Section 5.

2. Preliminary

Classical rough set theory is originated by Qian et al. to deal with imprecise or vague concepts [39]. In the last decade, many generalized rough set models have been proposed and developed. In this section, we will only introduce some notions being used in this paper.

Pawlak’s classical rough set is to be defined at the first place. An information system in rough set theory is a pair \((U, A)\), where \(U = \{x_1, \ldots, x_n\}\) denotes a nonempty finite set of objects, called universe of discourse, and \(A = \{a_1, \ldots, a_m\}\) denotes a finite condition attribute set. With every attribute \(a \in A\) we associate a set \(V_a\), of its values, called the domain of \(a\). Then, a data pattern \((a_1(x), \ldots, a_m(x))\) can be defined by the object \(x\) and attributes from \(A\). A decision table, abbreviated as DT, is a special system with the form \((U, C \cup \{d\})\), where \(d \notin C\) denotes decision attribute. Let \(V_d\) denote the domain of decision attribute mapping \(d(x)\). For the application of pattern classification, the attribute set is just the feature set and the universe of the discourse may represent a training pattern set or a sign set of training pattern sets.

Let \(R\) be an equivalence relation on \(U\), which means relation \(R\) satisfies reflexivity, symmetry, and transitivity. Relation \(R\) generates a partition \(U/R = \text{IND}(R) \in \{[x]_R \mid x \in U\}\) on \(U\), where \(\text{IND}(R)\) denotes the equivalence classes, as well as indiscriminable class, generated by the equivalence relation \(R\). These are also called elementary sets of \(R\) in rough set theory. Let \(\emptyset\) denote the empty sets. For any \(X \subseteq U\), we can describe \(X\) by elementary sets of \(R\) and the two sets

\[
R_-(X) = \{[x]_R \mid [x]_R \subseteq X\},
\]

\[
R^-(X) = \{[x]_R \mid [x]_R \cap X \neq \emptyset\}
\]

which are called the lower and upper approximations of \(X\), respectively. Then, the concepts of positive region,

\[
\text{POS}(X) = R_-(X),
\]

negative region,

\[
\text{NEG}(X) = U - R^-(X),
\]

boundary region,

\[
\text{BN}(X) = R^-(X) - R_-(X)
\]

\[
= \{[x]_R \mid [x]_R \cap X \neq \emptyset\} - \{[x]_R \mid [x]_R \subseteq X\},
\]

and approximation measure,

\[
\alpha_R(X) = \frac{|R_-(X)|}{|R^-(X)|},
\]

are introduced, where \(X \neq \emptyset\). The lower approximation, which is equivalent to the positive region, is the complete set of the objects in the universe that can be unambiguously classified as belonging to the target set \(X\). In contrast, the upper approximation is the complete set of the objects that are possibly members of the target set \(X\). In other words, these objects cannot be positively classified as belonging to the complement of the target set \(X\), that is, \(\overline{X}\). Furthermore, the negative region contains the set of objects that can be definitely ruled out as the members of the target set \(X\). Finally, the approximation measure \(\alpha_R(X)\) is intended to capture
the degree of completeness of our knowledge about the target set $X$.

In most cases, some precise values of particular attributes in an information system are not known, which means missing or known partially. Then such a system is called an incomplete information system and is still denoted without any confusion by a pair $(U,A)$. As for an incomplete information system, a missing value $a(x)$ may be represented by the set of all the possible values for the corresponding attribute; that is, $a(x) = V_a$. Moreover, if $a(x)$ is known partially, for instance, which implies that $a(x)$ cannot be $b,c \in V_a$, then the value of $a(x)$ is specified as $V_a = \{b,c\}$.

In what follows, we only take the consideration of the incomplete information systems with missing values. In such a case, the special symbol "*" is to be used to indicate that the specific value of an attribute is missing. Let $S = (U,AT)$ be an incomplete information system. Each subset of attributes $A \subseteq AT$ determines a similarity relation

$$SIM(A) = \{(x,y) \in U \times U \mid \forall a \in A, a(x) = a(y) = * \}.$$  

(6)

The incomplete information system can be described by a set-valued information system with $a(x) = V_a$ when $a(x) = *$. In such a case, the similarity relation $SIM(A)$ can be equivalently defined as

$$SIM(A) = \{(x,y) \in U \times U \mid \forall a \in A, a(x) \cap a(y) \neq \emptyset \}.$$  

(7)

With the similarity relation $SIM(A)$, two objects are considered to be possibly indiscernible in terms of the values of attributes $A$. Furthermore, a similarity relation satisfies reflexive and symmetric, but may not be transitive, so it is one of the tolerance relations.

### 3. Rough Entropy-Based Uncertainty Measures

In this section, the concept of rough entropy is introduced to measure the uncertainty or roughness of knowledge in incomplete information systems. And then, some rough entropy-based uncertainty measures are presented in incomplete information systems and incomplete decision systems. Some important properties concerning the uncertainty measures are derived, respectively, and the relationship among them is discussed as well.

For a given information system, we need to assess its uncertainty or roughness for a target object or a target decision. A form of uncertainty measure, which is called rough entropy, has been mentioned in rough sets, rough relational databases, and information systems to calculate the roughness of knowledge. The following definition gives the description of the rough entropy in incomplete information systems.

### 3.1. Rough Entropy in IIS and IDS

Let $IIS = (U,A)$ denote an incomplete information system with $* \in V$ and $P \subseteq A$, and $U / SIM(P) = \{S_P(u_1), S_P(u_2), \ldots, S_P(u_{|U|})\}$. According to Shannon’s theory of information entropy, the rough entropy of knowledge $P$ on $U$ is denoted by

$$EN(P) = -\frac{1}{|U|} \sum_{i=1}^{|U|} \log_2 \frac{1}{|S_P(u_i)|},$$  

(8)

where $1/|S_P(u_i)|$ represents the probability of an element within the tolerance class $S_P(u_i)$ [40].

Let $IDS = (U,A)$ be an incomplete information system, with $P, Q \subseteq A$. If there exists a one-to-one onto function $h : U / SIM(P) \rightarrow U / SIM(Q)$ such that $|h(S_P(u_i))| = |Q_i(u_j)| = |S_Q(u_j)|$ for any $u_i \in U$, then $EN(P) = EN(Q)$. Therefore, the rough entropy of knowledge is invariant with respect to the different set of tolerance classes that are size isomorphic.

Let $IDS = (U,C \cup \{d\})$ denote an incomplete decision system with $* \in V_C$. For any subset of condition attributes $B \subseteq C$, a tolerance relation $T$, which is in generalized form of the former similarity relation, can be defined as

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

(9)

Then, the tolerance class of an object $x$ with respect to an attribute set $B$ is defined as

$$T_B(x) = \{y \in U \mid (x, y) \in T_B\}.$$  

(10)

Obviously, relation $T$ is reflexive and symmetric, but may not be transitive.

**Definition 1.** Let $IDS = (U, C \cup \{d\}), * \in V_C$ be an incomplete decision system, then the generalized decision $\partial_B = U \rightarrow V_d$ is defined as follows:

$$\partial_B(x) = \{u \mid y \in T_B(x) \land u = d(y)\}.$$  

(11)

**Definition 2.** Let $IDS = (U, C \cup \{d\}), * \in V_C$ be an incomplete decision system. Let $\partial_B = U \rightarrow V_d$ be the generalized decision function. If $|\partial_B(x)| = 1$ for any $x \in U$, then the incomplete decision is consistent, which implies that it is deterministic and definite, where $|\cdot|$ denotes the number of the elements of the set.

In what follows, we give an example, which was firstly shown by Kryszkiewicz [31], to demonstrate how tolerance relationship works in a specific incomplete decision system. Given an incomplete decision system shown in Table I, where $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$, $C = \{Price, Mileage, Size, Max-speed\} = \{P, M, S, X\}$
Table 1: An example of incomplete decision system.

<table>
<thead>
<tr>
<th>Car</th>
<th>Price</th>
<th>Mileage</th>
<th>Size</th>
<th>Max-speed</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>High</td>
<td>Full</td>
<td>Low</td>
<td>Good</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>*</td>
<td>Full</td>
<td>Low</td>
<td>Good</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>*</td>
<td>Compact</td>
<td>High</td>
<td>Poor</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>*</td>
<td>Full</td>
<td>High</td>
<td>Good</td>
</tr>
<tr>
<td>5</td>
<td>*</td>
<td>*</td>
<td>Full</td>
<td>High</td>
<td>Excellent</td>
</tr>
<tr>
<td>6</td>
<td>Low</td>
<td>High</td>
<td>Full</td>
<td>*</td>
<td>Good</td>
</tr>
</tbody>
</table>

and $d = \text{Acceleration}$, and $U/\text{IND}(d) = \{S_{\text{Good}}, S_{\text{Poor}}, S_{\text{Excel}}\}$.

By definition, we have

$$T_C(u_i) = \{u_i\}, \quad T_C(u_i \cap d_{\text{Good}}) = \{u_i\},$$

$$T_C(u_i \cap d_{\text{Poor}}) = T_C(u_i \cap d_{\text{Excellent}}) = \emptyset,$$

$$T_C(u_2) = \{u_2, u_6\}, \quad T_C(u_2 \cap d_{\text{Good}}) = \{u_2, u_6\},$$

$$T_C(u_3) = \{u_3\}, \quad T_C(u_3 \cap d_{\text{Poor}}) = \emptyset,$$

$$T_C(u_3 \cap d_{\text{Excellent}}) = \emptyset,$$

$$T_C(u_4) = \{u_4, u_5\}, \quad T_C(u_4 \cap d_{\text{Excellent}}) = \{u_4\},$$

$$T_C(u_5) = \{u_4, u_6\}, \quad T_C(u_5 \cap d_{\text{Excellent}}) = \{u_5\},$$

$$T_C(u_6) = \{u_2, u_6\}, \quad T_C(u_6 \cap d_{\text{Excellent}}) = \{u_2, u_6\},$$

$$T_C(u_6 \cap d_{\text{Poor}}) = \emptyset,$$

$$\frac{U}{\text{IND}(d)} = \{\{u_1, u_2, u_4, u_5\}, \{u_3\}, \{u_3\}\}.$$ (12)

Consequently, we get $|\partial_C(u_i)| = |\partial_C(-u_i)| = |\partial_C(u_j)| = |\partial_C(-u_j)| = 1$, which proves that the incomplete decision system is consistent.

**Definition 4.** Given a consistent incomplete decision system, $\text{IDS} = (U, C \cup \{d\})$ and $B \subseteq C$. Let $U/T_B = \{T_B(x_1), T_B(x_2), \ldots, T_B(x_U)\}$, $U/d = \{d_1, d_2, \ldots, d_m\}$. The conditional entropy of $B$ to $d$ is defined as follows:

$$EN(d \mid B) = -\sum_{i=1}^{U} p(T_B(u_i))$$

(13)

$$\times \sum_{j=1}^{[U/d]} p(d_j \mid T_B(u_i)) \log_2 p(d_j \mid T_B(u_i)),$$

where

$$p(T_B(u_i)) = \frac{\lbrack T_B(u_i) \rbrack}{\lbrack U \rbrack}, \quad i = 1, 2, \ldots, [U],$$

$$p(d_j \mid T_B(u_i)) = \frac{\lbrack T_B(u_i) \cap d_j \rbrack}{\lbrack T_B(u_i) \rbrack}, \quad j = 1, 2, \ldots, m.$$ (14)

Hence, we have

$$EN(d \mid B)$$

(15)

It is obvious that $EN(d \mid B) = 0$ when $T_B(x_i) \cap d_j = \emptyset$.

**Proposition 5.** Let $\text{IDS} = (U, C \cup \{d\})$ be a consistent incomplete decision system. Then, we have $EN(d \mid C) = 0$.

**Proof.** Since IDS is consistent, we have $\partial_C(x_i) = 1, x_i \in U$.

This means that $T_C(x_i) \subseteq d_j, d_j \in U/d, x_i \in U$.

Hence, we have

$$T_C(x_i) \cap d_j = \emptyset, \quad \forall d_j \in U/d, d_j \neq d_i.$$ (16)
Consequently, we know that

\[
\text{EN}(d \mid C) = \sum_{i=1}^{\lvert U \rvert} p(T_C(x_i)) \left( \sum_{j=1}^{\lvert d \rvert} p(d_j \mid T_C(x_i)) \log_2 p(d_j \mid T_C(x_i)) \right)
\]

\[
= -\sum_{i=1}^{\lvert U \rvert} \frac{|T_C(x_i)|}{\lvert U \rvert} \left( \sum_{j=1}^{\lvert d \rvert} \frac{|T_C(x_i) \cap d_j|}{|T_C(x_i)|} \log_2 \frac{|T_C(x_i) \cap d_j|}{|T_C(x_i)|} \right)
\]

\[
= -\sum_{i=1}^{\lvert U \rvert} \frac{|T_C(x_i)|}{\lvert U \rvert} \left( 0 + 0 + \cdots + 1 \log 1 + 0 + \cdots + 0 \right) = 0.
\]

(17)

**Proposition 6.** Let \( IDS = (U, C \cup \{d\}) \) be a consistent incomplete decision system; \( B \subseteq C \) is a relative reduct of \( C \) relative to decision attribute \( d \), and if only if

1. \( \text{EN}(d \mid C) = \text{EN}(d \mid B) \);
2. \( B' \subset B, \text{EN}(d \mid B') \neq \text{EN}(d \mid B) \).

**Proof.** (1) We first prove that if \( \text{EN}(d \mid C) = \text{EN}(d \mid B) \), \( \forall B' \subset B, \text{EN}(d \mid B') \neq \text{EN}(d \mid B) \) then \( B \subseteq C \) is a relative reduct.

By Proposition 5, we know that \( \text{EN}(d \mid C) = 0 \). Then
\( \text{EN}(d \mid B) = \text{EN}(d \mid C) = 0 \).

Since
\[
0 \leq \frac{|T_B(x_i)|}{\lvert U \rvert} \leq 1, \quad 0 \leq \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} \leq 1,
\]
we have
\[
\frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} \log_2 \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} \leq 0.
\]

At the same time, we know that \( \text{EN}(d \mid B) = 0 \) if and only if
\[
\frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} \log_2 \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} = 0, \quad \forall x_i, d_j.
\]

This implies that there exists only one \( d_j \in U/d \) such that \( T_B(x_i) \subseteq d_j \) and \( T_B(x_i) \cap d_j = 0, \forall d_r \in U/d, d_r \neq d_j \). This means that \( \partial_B(x_i) = 1, \forall x_i \in U \). Hence, we get \( \partial_B = \partial_C \).

From \( \forall B' \subset B, \text{EN}(d \mid B') \neq \text{EN}(d \mid B) \), we have \( \text{EN}(d \mid B') \neq 0 \). It follows that there at least exist \( x_i, d_j, d_r \) such that \( T_B(x_i) \cap d_j = 0 \) and \( T_B(x_i) \cap d_r = 0 \). This means that \( \partial_B(x_i) \geq 2 \). Hence, \( \partial_B \neq \partial_B \).

(2) We then prove that if \( B \subseteq C \) is a relative reduct, then \( \text{EN}(d \mid C) = \text{EN}(d \mid B), \forall B' \subset B, \text{EN}(d \mid B') \neq \text{EN}(d \mid B) \).

In this case, we get \( \partial_B = \partial_C \); that is, \( \lvert \partial_B(x_i) \rvert = \lvert \partial_C(x_i) \rvert = 1, \forall x_i \in U \). It follows that, \( \forall x_i \in U \), there exists \( d_j \in U/d \) such that \( T_B(x_i) \subseteq d_j \) and \( T_B(x_i) \cap d_r = 0, \forall d_r \in U/d, d_r \neq d_j \).

It follows that
\[
\frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} \log_2 \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} = 1 \log 1 = 0, \quad \forall x_i, d_j.
\]

Therefore, this means that
\[
\sum_{j=1}^{\lvert d \rvert} \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} \log_2 \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} = 0, \quad \forall x_i.
\]

Consequently, we know
\[
\text{EN}(d \mid B) = -\sum_{i=1}^{\lvert U \rvert} p(T_B(x_i)) \sum_{j=1}^{\lvert d \rvert} p(d_j \mid T_B(x_i)) \log_2 p(d_j \mid T_B(x_i))
\]

\[
= -\sum_{i=1}^{\lvert U \rvert} \frac{|T_B(x_i)|}{\lvert U \rvert} \sum_{j=1}^{\lvert d \rvert} \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|} \log_2 \frac{|T_B(x_i) \cap d_j|}{|T_B(x_i)|}
\]

\[
= -\sum_{i=1}^{\lvert U \rvert} \frac{|T_B(x_i)|}{\lvert U \rvert} \cdot 0 = 0.
\]

(23)

For \( \forall B' \subset B \), we know that \( \partial_B \neq \partial_B \); that is, there at least exists \( x_i \in U \) such that \( \partial_B(x_i) \neq \partial_B(x_i) \). It follows that \( \lvert \partial_B(x_i) \rvert > \lvert \partial_B(x_i) \rvert = 1 \). Thus, there at least exist \( d_j \in U/d \) and \( d_r \in U/d \), \( d_r \neq d_j \) such that
\[
T_B(x_i) \cap d_r \neq 0,
\]

\[
T_B(x_i) \cap d_j \neq 0.
\]

Then, we get \( \text{EN}(d \mid B') \neq \text{EN}(d \mid B) \).

From parts (1) and (2), we finally prove that the theorem holds.

\( \square \)

### 4. Attribute Selection Approaches Based on Rough Conditional Entropy for IDS

Two important steps are contained in the procedure of attribute selection: evaluation of a candidate attribute subset and search strategy through the attribute space, which are all for finding the most significant attributes, that is, relative reduct. Therefore, we use the conditional entropy discussed previously in Section 3 to evaluate the attribute subset and a measurement is defined as follows.

**Definition 7.** Given a consistent decision system \( IDS = (U, C \cup \{d\}) \), let \( B \subseteq C \) and \( a \in C - B \). Then, the significance of attribute relative to \( B \) is defined as
\[
\text{SIG}(a, B, d) = \text{EN}(d \mid B) - \text{EN}(d \mid B \cup \{a\}).
\]

This definition describes the increment of discernibility power relative to the decision caused by involving attribute \( a \).
It implies that the larger the difference between $\text{EN}(d | B)$ and $\text{EN}(d | B \cup \{a\})$ is, the more significant the specific condition attribute $a$ is for condition attribute subset $B$. Thus, it can be used as a new measurement for attribute selection in incomplete decision system. According to this new measurement, three attribute selection approaches based on different search strategies are proposed, respectively, in the following subsections.

4.1. Breadth-First: Exhaustive Search. Breadth-first is one of earliest feature selection or attribute selection algorithms in machine learning area. It begins with an empty attribute set and carries out search process with breadth-first strategy until it finds a minimal subset that satisfies stop criterion. Since the Breadth-first algorithm adopts an exhaustive search strategy, it can guarantee an optimal solution [41]. Then, we present the Breadth-first approach for attribute selection of incomplete decision system as shown in Algorithm 1.

Example 8. Given the incomplete decision system shown in Table 2, we have

$$
C = \{ P, M, S, X \},
$$

$$
\frac{U}{d} = \{ \{u_1, u_2, u_4, u_5\}, \{u_2\}, \{u_3\} \},
$$

$$
T_C(u_1) = T_C(u_2) = T_C(u_6) = \{u_1, u_2, u_6\}, \quad (26)
$$

$$
T_C(u_3) = \{u_3\},
$$

$$
T_C(u_4) = \{u_4\},
$$

$$
T_C(u_5) = \{u_5\}.
$$

According to the previous definition of rough conditional entropy, we can simply get

$$
\text{EN}(d | C) = -\left[ 3 * \frac{3}{6} * \left( \frac{3}{3} \log_2 \frac{3}{3} \right) + \frac{1}{6} * (1 \log_2 1) \right] - \frac{1}{6} * (1 \log_2 1) = 0.
$$

(27)

Consequently, we can deduce other conditional entropy values with respect to the different condition attributes in the same way as follows:

$$
T_P(u_1) = T_P(u_4) = \{u_1, u_2, u_3, u_4, u_6\},
$$

$$
T_P(u_2) = T_P(u_3) = T_P(u_6) = \{u_1, u_2, u_3, u_4, u_5, u_6\},
$$

$$
T_P(u_5) = \{u_2, u_3, u_5, u_6\},
$$

$$
\text{EN}(d | \{P\}) = -\left[ 2 * \frac{5}{6} * \left( \frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right) \right.
$$

$$
+ \frac{3}{6} * \left( \frac{4}{6} \log_2 \frac{4}{6} + \frac{1}{6} \log_2 \frac{1}{6} + \frac{1}{6} \log_2 \frac{1}{6} \right) \right]
$$

$$
= 5.9581,
$$

$$
T_M(u_1) = T_M(u_2) = T_M(u_3) = T_M(u_4) = T_M(u_5) = \{u_1, u_2, u_3, u_4, u_5, u_6\},
$$

$$
\text{EN}(d | \{M\}) = -\left[ 6 * \frac{6}{6} * \left( \frac{4}{6} \log_2 \frac{4}{6} + \frac{1}{6} \log_2 \frac{1}{6} + \frac{1}{6} \log_2 \frac{1}{6} \right) \right]
$$

$$
= 7.5098,
$$

$$
T_S(u_1) = T_S(u_2) = T_S(u_3) = T_S(u_4) = T_S(u_5) = \{u_3\},
$$

$$
\text{EN}(d | \{S\}) = -\left[ \frac{5}{6} * \left( \frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right) + \frac{1}{6} \log_2 1 \right]
$$

$$
= 3.0080,
$$

$$
T_X(u_1) = T_X(u_2) = T_X(u_3) = \{u_1, u_2, u_6\},
$$

$$
T_X(u_4) = \{u_4\},
$$

$$
\text{EN}(d | \{X\}) = -\left[ \frac{3}{6} * \left( \frac{3}{3} \log_2 \frac{3}{3} \right) + \frac{3}{6} \right]
$$

$$
* \left( \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{3} \log_2 \frac{1}{3} \right) \right]
$$

$$
= 2.3774.
$$

(28)

Then, we calculate the rough conditional entropy values with respect to different combinations of two condition attributes:

$$
T_{P,M}(u_1) = T_{P,M}(u_4) = \{u_1, u_2, u_3, u_4, u_6\},
$$

$$
T_{P,M}(u_2) = T_{P,M}(u_5) = \{u_1, u_2, u_3, u_4, u_5, u_6\},
$$

$$
T_{P,M}(u_3) = \{u_2, u_3, u_5, u_6\},
$$

$$
\text{EN}(d | \{P, M\}) = -\left[ \frac{5}{6} * \left( \frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right) \right]
$$
Input: An incomplete decision system \((U, C \cup \{d\})\).
Output: An attribute selection result \(R\).
(1) For every size = 0 to \(|C|\)
(2) For all subset SelectAttr with \(|\text{SelectAttr}| = \text{size}\)
(3) If \(\text{EN}(d \mid \text{SelectAttr}) \neq \text{EN}(d \mid C)\), go to Step 2, otherwise return \(R = \text{SelectAttr}\)
(4) End
(5) End

Algorithm 1: Breadth-first.

Table 2: Another example of incomplete decision system.

<table>
<thead>
<tr>
<th>Car</th>
<th>Price</th>
<th>Mileage</th>
<th>Size</th>
<th>Max-speed</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>High</td>
<td>Full</td>
<td>Low</td>
<td>Good</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>*</td>
<td>Full</td>
<td>Low</td>
<td>Good</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>*</td>
<td>Compact</td>
<td>High</td>
<td>Poor</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>*</td>
<td>Full</td>
<td>High</td>
<td>Good</td>
</tr>
<tr>
<td>5</td>
<td>Low</td>
<td>*</td>
<td>Full</td>
<td>High</td>
<td>Excellent</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>High</td>
<td>Full</td>
<td>Low</td>
<td>Good</td>
</tr>
</tbody>
</table>

\[
\text{EN}(d \mid \{P,S\}) = -\left[ 2 \times \frac{5}{6} \times \frac{4}{5} + \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{3} \log_2 \frac{1}{3} \right] + 2 \times \frac{5}{6} \times \frac{4}{5} + \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{6} \times \left( 1 \log_2 1 \right) + \frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3} \right] = 5.9581,
\]

\[
T_{PS}(u_1) = T_{PS}(u_4) = \{u_1, u_2, u_4, u_6\},
\]

\[
T_{PS}(u_2) = T_{PS}(u_5) = \{u_1, u_2, u_4, u_5, u_6\},
\]

\[
T_{PS}(u_3) = \{u_3\},
\]

\[
T_{PS}(u_5) = \{u_2, u_5, u_6\},
\]

\[
\text{EN}(d \mid \{M,S\}) = -\left[ 5 \times \frac{5}{6} \times \frac{4}{5} + \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{6} \times (\log_2 1) \right] = 3.0080,
\]

\[
T_{MS}(u_1) = T_{MS}(u_2) = T_{MS}(u_4) = \{u_1, u_2, u_6\},
\]

\[
T_{MS}(u_3) = T_{MS}(u_5) = \{u_3, u_4, u_5\},
\]

\[
T_{MS}(u_5) = \{u_3\},
\]

\[
\text{EN}(d \mid \{M,X\}) = -\left[ 3 \times \frac{5}{6} \times \frac{3}{5} + \frac{1}{3} \log_2 \frac{1}{3} + \frac{1}{3} \log_2 \frac{1}{3} \right] = 2.3774,
\]

\[
T_{SX}(u_1) = T_{SX}(u_2) = T_{SX}(u_6) = \{u_1, u_2, u_6\},
\]

\[
T_{SX}(u_3) = \{u_3\},
\]

\[
T_{SX}(u_4) = T_{SX}(u_5) = \{u_4, u_5\},
\]
Mathematical Problems in Engineering

Finally, we calculate the rough conditional entropy values with respect to different combinations of three condition attributes:

\[
T_{P,M,S}(u_1) = T_{P,M,S}(u_4) = \{u_1, u_2, u_4, u_6\},
\]

\[
T_{P,M,S}(u_2) = T_{P,M,S}(u_5) = \{u_1, u_2, u_4, u_5\},
\]

\[
T_{P,M,S}(u_3) = \{u_3\},
\]

\[
T_{P,M,S}(u_5) = \{u_2, u_3, u_6\},
\]

\[
\text{EN}(d | \{P, M, S\}) = -\left[ 2 \ast \frac{4}{6} \ast \left( \frac{4}{5} \log_2 \frac{4}{5} \right) \right.
\]

\[
+ 2 \ast \frac{5}{6} \ast \left( \frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right)
\]

\[
+ \frac{3}{6} \ast \left( \frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3} \right) \left( 30 \right)
\]

\[
= 1.6624,
\]

\[
T_{P,M,X}(u_1) = T_{P,M,X}(u_2) = T_{P,M,X}(u_6) = \{u_1, u_2, u_6\},
\]

\[
T_{P,M,X}(u_3) = \{u_3, u_4, u_3\},
\]

\[
T_{P,M,X}(u_4) = \{u_3, u_4\},
\]

\[
T_{P,M,X}(u_5) = \{u_3, u_5\},
\]

\[
\text{EN}(d | \{P, M, X\}) = -\left[ 3 \ast \frac{3}{6} \ast \left( \frac{3}{5} \log_2 \frac{3}{5} \right) + \frac{1}{6} \ast \left( \log_2 \frac{1}{6} \right) \right.
\]

\[
+ \frac{2}{6} \ast \left( \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right) \left( 31 \right)
\]

\[
= 1.4591,
\]

\[
T_{P,S,X}(u_1) = T_{P,S,X}(u_2) = T_{P,S,X}(u_6) = \{u_1, u_2, u_6\},
\]

\[
T_{P,S,X}(u_3) = \{u_3\},
\]

\[
T_{P,S,X}(u_4) = \{u_4\},
\]

\[
T_{P,S,X}(u_5) = \{u_5\},
\]

\[
\text{EN}(d | \{P, S, X\})
\]

\[
= -\left[ 3 \ast \frac{3}{6} \ast \left( \frac{3}{5} \log_2 \frac{3}{5} \right) + \frac{1}{6} \ast \left( \log_2 \frac{1}{6} \right) \right.
\]

\[
+ \frac{1}{6} \ast \left( \log_2 \frac{1}{6} \right) \right]
\]

\[
= 0.6667.
\]

Since \( \text{EN}(d | \{P, S, X\}) = \text{EN}(d | C) = 0 \), we obtain the desired attribute set of selected \( \{P, S, X\} \), which is a relative reduct of the original condition attribute set \( C \). It also means that the search procedure ends at this step and it is not necessary to calculate the entropy value of the last combination of three attributes \( \{M, S, X\} \) anymore. The detailed search procedure is illustrated in Figure 1.

### 4.2. Depth-First: Heuristic Search

We can also use heuristic search or greed search to find attribute reduction. At the very beginning, the candidate attribute subset is empty. Then, a new attribute which can maximize the significance measure is added to the selected attribute subset each time, until the stop criterion is satisfied. Depth-first algorithm is fast, close to optimal, and deterministic [42]. Here, we present the Depth-first approach for incomplete decision system, as shown in Algorithm 2.

**Example 9.** Given the incomplete decision system shown in Table 2, we have

\[
\text{SelectAttr} = \emptyset,
\]

\[
\text{unSelectAttr} = C = \{P, M, S, X\},
\]

\[
U \div d = \{[u_1, u_2, u_4, u_6], [u_3], [u_5]\},
\]

\[
\text{EN}(d | \{P\})
\]

\[
= -\left[ 2 \ast \frac{5}{6} \ast \left( \frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right) \right.
\]

\[
+ \frac{3}{6} \ast \left( \log_2 \frac{2}{3} + \log_2 \frac{1}{3} \right) \left( 32 \right)
\]

\[
= 5.9581,
\]

\[
\text{EN}(d | \{M\})
\]

\[
= -\left[ 6 \ast \frac{6}{6} \ast \left( \frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right) \right.
\]

\[
+ \frac{4}{6} \ast \left( \log_2 \frac{2}{4} + \log_2 \frac{1}{4} \right) \left( 33 \right)
\]

\[
= 7.5098,
\]

\[
\text{EN}(d | \{S\})
\]

\[
= -\left[ 5 \ast \frac{5}{6} \ast \left( \frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right) + \frac{1}{6} \ast \left( \log_2 \frac{1}{6} \right) \right]
\]

\[
= 3.0080,
\]
Input: An incomplete decision system \((U, C \cup \{d\})\).
Output: A selected attribute subset \(R\).

1. Initialize \(\text{SelectAttr} = \emptyset\), \(\text{unSelectAttr} = C\).
2. For every attribute \(a_i \in \text{unSelectAttr}\), \(\text{SelectAttr} = \text{SelectAttr} \cup \{a_i\}\), calculate the tolerance class of object \(u_i\), \(T_{\text{Attr}}(u_i)\).
3. Calculate the conditional entropy \(\text{EN}(d | \text{SelectAttr}) = \text{EN}(d | \text{SelectAttr} \cup \{a_i\})\).
4. Choose the attribute \(a\) which minimizes \(\text{EN}(d | \text{SelectAttr} \cup \{a\})\), that is choose the attribute with the maximum significance measure \(\text{SIG}(a, \text{SelectAttr}, d)\).
5. \(\text{SelectAttr} = \text{SelectAttr} \cup \{a\}\), \(\text{unSelectAttr} = \text{unSelectAttr} - \{a\}\).
6. If \(\text{EN}(d | \text{SelectAttr}) \neq \text{EN}(d | C)\), go to Step 2, otherwise \(R = \text{SelectAttr}\).
7. End.

**Algorithm 2: Depth-first.**

\[
\text{EN}(d | \{X\}) = - \left[ 3 \cdot \frac{3}{6} \cdot \left( 3 \cdot \log_2 \frac{3}{3} \right) + 3 \cdot \frac{3}{6} \cdot \left( \frac{1}{3} \cdot \log_2 \frac{1}{3} + \frac{1}{3} \cdot \log_2 \frac{1}{3} + \frac{1}{3} \cdot \log_2 \frac{1}{3} \right) \right] = 2.3774.
\]

Since \(\text{EN}(d | \{X\})\) has the minimum conditional entropy value, we choose the attribute \(X\) as one of selected attributes. Therefore, we can get

\[
\text{SelectAttr} = \text{SelectAttr} \cup \{X\} = \{X\},
\]

\[
\text{unSelectAttr} = \text{unSelectAttr} - \{X\} = \{P, M, S\}.
\]

Since \(\text{EN}(d | \{X\}) \neq \text{EN}(d | C)\), we still need to add more attributes to \(\text{SelectAttr}\). On the basis of the rule of heuristic search, we only need to test and compare the conditional entropy values of the subsets in which the condition attribute \(X\) is included:

\[
\text{EN}(d | \{P, X\}) = - \left[ 3 \cdot \frac{3}{6} \cdot \left( 3 \cdot \log_2 \frac{3}{3} \right) + 3 \cdot \frac{3}{6} \cdot \left( \frac{1}{3} \cdot \log_2 \frac{1}{3} + \frac{1}{3} \cdot \log_2 \frac{1}{3} + \frac{1}{3} \cdot \log_2 \frac{1}{3} \right) \right] = 2.3774.
\]

As previously mentioned, since \(\text{EN}(d | \{S, X\})\) is the minimum, we choose the attribute \(S\) as another selected attribute. Thus we have

\[
\text{SelectAttr} = \text{SelectAttr} \cup \{S\} = \{S, X\},
\]

\[
\text{unSelectAttr} = \text{unSelectAttr} - \{S\} = \{P, M\}.
\]
Input: An incomplete decision system \((U, C \cup \{d\})\) and MaxTries.
Output: A selected attribute subset \(R\).

1. Initialize SelectAttr = \(C\).
2. For \(j = 0\) to MaxTries
3. Randomly choose a subset of condition attributes \(S_j\).
4. If \(\text{EN} (d \mid S_j) = \text{EN} (d \mid C)\) and \(|S_j| < |\text{SelectAttr}|\)
5. SelectAttr = \(S_j\).
6. End
7. \(R = \text{SelectAttr}\)

Algorithm 3: LVF.

Since \(\text{EN}(d \mid \{S, X\}) \neq \text{EN}(d \mid C)\), it is still necessary to choose more attributes to add to the selected attributes in which attributes \(S, X\) are definitely included:

\[
\text{EN}(d \mid \{P, S, X\}) = -\left[ 3 \cdot 3^3 \cdot \left( \frac{1}{3} \log_2 \frac{3}{3} \right) + \frac{1}{6} \cdot (1 \log_2 1) \right] = 0,
\]

\[
\text{EN}(d \mid \{M, S, X\}) = -\left[ 3 \cdot 3^3 \cdot \left( \frac{1}{3} \log_2 \frac{3}{3} \right) + \frac{1}{6} \cdot (1 \log_2 1) \right] + 2 \cdot \frac{2}{6} \cdot \left( \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right) = 0.6667.
\]

Obviously, since \(\text{EN}(d \mid \{P, S, X\})\) is the minimum, we choose the attribute \(P\). Thus, we have

\[
\text{SelectAttr} = \text{SelectAttr} \cup \{P\} = \{P, S, X\},
\]

\[
\text{unSelectAttr} = \text{unSelectAttr} - \{P\} = \{M\}.
\]

And since \(\text{EN}(d \mid \{P, S, X\}) = \text{EN}(d \mid C) = 0\), the stop criterion is satisfied and the algorithm terminates. Hence, the final result of attributes selection is \(R = \{P, S, X\}\). The detailed procedure of the Depth-first algorithm is shown in Figure 2.

4.3. LVF: Probabilistic Search. Las Vegas algorithm is new for attribute subset selection and can make probabilistic choices of subsets in an optimal search. Las Vegas Filter, which is abbreviated as LVF, is a probabilistic algorithm where probabilities of generating any subset are equal [43]. In this paper, we use the investigated conditional entropy as LVF’s evaluation measurement. It generates attribute subsets randomly with equal probability and records the minimal size of attributes subset satisfying the stop criterion of maximum tries times. LVF is fast and efficient in reducing the number of candidate features in the early stages and can produce optimal solutions if the computing resources permit. The LVF approach for attribute selection in incomplete decision system is given, as shown in Algorithm 3.

5. Experiments

In this section, the performances of our attribute selection algorithms given in previous section are demonstrated and compared. Several real-life incomplete data sets from UCI Repository of Machine Learning Database at the University of California are used in our experiments. These experiments are performed on a personal computer with Windows 7, Intel (R) Core (TM) i3 CPU 2.13 GHz, and 4 GB RAM. The objective of these experiments is to evaluate the effectiveness and efficiency of the previous algorithms. The summary and statistic of the experimental data sets are shown in Table 3 and Figure 3, respectively.

Since some incomplete data sets contain continuous condition attribute values, we conduct a discretization preprocess to turn these continuous values into discrete ones before carrying out attribute selection. The aim of this step is to compress the data and reduce the time consumption of subsequent attribute selection. The running time of each algorithm is average CPU time, expressed in seconds.

In Depth-first algorithm, we terminate the program when it runs beyond 5000 seconds. Moreover, we set the parameter MaxTries in LVF algorithm to variant values for different incomplete data sets, according to their sizes. The running time and the size of attribute selection results are highly concerned with the choice of the parameter MaxTries. It means that when MaxTries grows, the running time of LVF approach increases linearly and the size of selected attribute subset decreases. Both of them for each attribute selection approach are shown in Table 4. The running time of each approach is the average CPU time, expressed in seconds. We can easily find that Breadth-first approach takes much more time, even more than 5000 seconds, to obtain an attribute reduction, compared with the other two approaches. Three data sets out of six are too large in scale to calculate in limited time. Furthermore, it is also easy to be observed in Table 4 that Depth-first approach tends to select fewer attributes than LVF approach. In other words, the size of attribute subset selected by Depth-first approach intends to be smaller than that of attribute subset selected by LVF approach. And Depth-first approach consumes less time than the other two approaches in most instances. The time consumptions of Breadth-first approach, Depth-first approach, and LVF approach for attribute selection are \(O(2^{|C| \cdot |U|^2})\), \(O(|C|^4 |U|^2)\), and \(O(\text{MaxTries} \cdot |C| \cdot |U|^2)\), respectively, where MaxTries, |C|,
SelectAttr = 0

\{P\} \leftrightarrow 5.9581 \quad \{M\} \leftrightarrow 7.5098 \quad \{S\} \leftrightarrow 3.0080 \quad \{X\} \leftrightarrow 2.3774

\{P, X\} \leftrightarrow 1.4591 \quad \{M, X\} \leftrightarrow 2.3774 \quad \{S, X\} \leftrightarrow 0.6667 \quad \{M, S, X\} \leftrightarrow 0.6667

\{P, S, X\} \leftrightarrow 0

\text{EN}\left( d \mid \{P, S, X\} \right) = \text{EN}\left( d \mid C \right) = 0

\textbf{Figure 2:} The flow chart of the Depth-first algorithm.

\textbf{Figure 3:} The statistical result of experimental data sets.
Table 3: The summary of experimental data sets.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Instances</th>
<th>Condition attributes</th>
<th>Missing values</th>
<th>Decision classes</th>
<th>Incomplete rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lung cancer</td>
<td>32</td>
<td>56</td>
<td>5</td>
<td>3</td>
<td>0.28</td>
</tr>
<tr>
<td>Large soybean</td>
<td>307</td>
<td>35</td>
<td>712</td>
<td>19</td>
<td>6.63</td>
</tr>
<tr>
<td>Dermatology</td>
<td>366</td>
<td>34</td>
<td>8</td>
<td>6</td>
<td>0.06</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>699</td>
<td>10</td>
<td>16</td>
<td>2</td>
<td>0.23</td>
</tr>
<tr>
<td>Credit</td>
<td>690</td>
<td>15</td>
<td>71</td>
<td>2</td>
<td>0.69</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>155</td>
<td>19</td>
<td>167</td>
<td>2</td>
<td>5.67</td>
</tr>
</tbody>
</table>

Table 4: Size of selected attribute subsets and running time.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Breadth-first</th>
<th>Depth-first</th>
<th>LVF</th>
<th>Full attr.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size</td>
<td>Time (s)</td>
<td>Size</td>
<td>Time (s)</td>
</tr>
<tr>
<td>Lung cancer</td>
<td>4</td>
<td>227.59</td>
<td>4</td>
<td>0.68</td>
</tr>
<tr>
<td>Large soybean</td>
<td>N/A</td>
<td>&gt;5000</td>
<td>14</td>
<td>80.99</td>
</tr>
<tr>
<td>Dermatology</td>
<td>N/A</td>
<td>&gt;5000</td>
<td>6</td>
<td>58.22</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>5</td>
<td>367.37</td>
<td>5</td>
<td>51.59</td>
</tr>
<tr>
<td>Credit</td>
<td>N/A</td>
<td>&gt;5000</td>
<td>7</td>
<td>89.94</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>5</td>
<td>349.07</td>
<td>6</td>
<td>6.07</td>
</tr>
</tbody>
</table>

Table 5: Performance of attribute selection algorithms with SVM-RBF classifier.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Classification accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Breadth-first</td>
</tr>
<tr>
<td>Lung cancer</td>
<td>66.92</td>
</tr>
<tr>
<td>Large soybean</td>
<td>N/A</td>
</tr>
<tr>
<td>Dermatology</td>
<td>N/A</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>97.11</td>
</tr>
<tr>
<td>Credit</td>
<td>N/A</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>83.75</td>
</tr>
</tbody>
</table>

and $|U|$ denote the MaxTries parameter in LVF approach, the total numbers of the condition attributes, and instances in incomplete data sets, respectively. The relationships between incomplete data sets and the number of selected attributes, the running time of attribute selection, are illustrated in Figures 4 and 5.

The final part of our experiments is to compare and evaluate the efficiency of the proposed algorithms in practical classification tasks. For each of the six data sets in Table 5, we employ the SVM-RBF classifier, which is one of the most frequently used classifiers. We also apply the 10-fold cross-validation method to estimate the classification accuracy with respect to the reducts generated by the proposed algorithms. In each fold, the redundant attributes from the current training set are removed at the beginning, according to the proposed algorithms. Then, the test set is classified by using the rules generated from the training set. The final results of classification accuracies are shown in Table 5. It can be seen that, by using attribute selection algorithms, the classification accuracies for incomplete data sets are all raised in different degrees, compared with the classification accuracy for the original full attributes. It also can be noticed that the Depth-first algorithm exhibits the highest classification accuracy on each incomplete data set. Therefore, the experimental results demonstrate that the proposed algorithms are effective for attribute selection tasks in application domains.

6. Conclusion

In this paper, a rough conditional entropy-based attribute selection approach is proposed to evaluate the significance of condition attributes and find the minimal reduct in incomplete decision systems. By this measure, three types of attribute selection approaches, including the exhaustive search strategy approach Breadth-first, the heuristic search strategy approach Depth-first, and the probabilistic search approach LVF, are constructed. To evaluate the effectiveness of the introduced approaches, experiments on several real-life incomplete data sets are conducted. The experimental results suggest that Depth-first and LVF approaches are practical for attribute selection for classification of high-dimensional data with thousands of condition attributes, and they can efficiently enhance classification accuracy with predominant attributes. However, the process of examining exhaustively all combinations of condition attributes for finding the optimal one is an NP-hard problem. So far, it still cannot be easily calculated by our approaches if there are hundreds of thousands of condition attributes in a complex incomplete decision system. Therefore, for large data sets, to reduce the time consumption of the process of attribute selection, more applicable approaches such as parallel heuristic algorithms are desirable for incomplete decision systems with large scale. This issue needs to be investigated in the future.
Conflict of Interests

The authors would like to declare that there is no conflict of interests regarding the publication of the paper.

Acknowledgments

The authors would like to thank the anonymous reviewers sincerely for their insightful comments and constructive effort on this investigation. Moreover, the authors thank the UCI Repository of Machine Learning Database at the University of California for providing the experiment data sets. This work is supported by the National Natural Science Foundation of China (no. 61074176).

References

Mathematical Problems in Engineering


