Learning maximally general rules from examples

Tzung-Pei Hong, Tzu-Ting Wang and Been-Chian Chien
Department of Information Management
I-Shou University
Kaohsiung, 84008, Taiwan, R.O.C.
e-mail: {tphong, m873204m, cbc}@isu.edu.tw

ABSTRACT

The rough-set theory, proposed by Pawlak in 1982, can serve as a new mathematical tool to deal with data classification problems. It adopts the concepts of equivalence classes to partition the training instances according to some criteria. In this paper, we propose a new algorithm to deal with the problem of producing a set of maximally general rules for coverage of training examples using rough sets. The proposed method first calculates the lower approximations and the upper approximations. An iterative induction process is then adopted to find the maximally general rules. The rules derived can then be used to build a prototype knowledge base.

Keywords: knowledge acquisition, rough set, data mining, certain rule, possible rule.

1. Introduction

Although a wide variety of expert systems have been built, a development bottleneck occurs in knowledge acquisition. Building a large-scale expert system often involves creating and extending a large knowledge base over the course of many months or years [1][12]. Shortening the development time is then the most important factor for the success of expert systems. Machine learning techniques have thus been developed to ease the knowledge-acquisition bottleneck. Among machine learning approaches, deriving inference rules from training examples is the most common [5][7][8][13]. Given a set of examples and counterexamples of a concept, the learning program tries to induce general rules that describe all of the positive training instances and none of the counterexamples. If the training instances belong to more than two classes, the learning program tries to induce general rules for describing each class. Machine learning then provides a feasible way to build a prototype expert system.

Recently, the rough-set theory has been used in reasoning and knowledge acquisition for expert systems [3][9]. It was proposed by Pawlak in 1982 [10] with the concept of equivalence classes as its basic principle. Several applications and extensions of the rough-set theory have also been proposed [2][6][9][14]. In this paper, we deal with the problem of producing a coverage set of maximally general rules from training examples. A new learning algorithm based on the rough-set theory is proposed to induce maximally general rules.

The remainder of this paper is organized as follows. The rough-set theory is reviewed in Sections 2. The notation used in this paper is described in Section 3. In Section 4, a new learning algorithm based on the rough-set theory is proposed to induce maximally general rules from training examples. An example is then given to illustrate the proposed algorithm in Section 5. Conclusions are finally given in Section 6.

2. Review of the rough-set theory

The rough-set theory, proposed by Pawlak in 1982 [10][11], can serve as a new mathematical tool for dealing with data classification problems. It adopts the concept of equivalence classes to partition training instances according to some criteria. Two kinds of partitions are formed in the mining process:
lower approximations and upper approximations, from which certain and possible rules can easily be derived.

Formally, let \( U \) be a set of training examples (objects), \( A \) be a set of attributes describing the examples, \( C \) be a set of classes, and \( V_i \) be a value domain of an attribute \( A_j \). Also let \( v_j^{(i)} \) be the value of attribute \( A_j \) for the \( i \)-th object \( \text{Obj}(i) \). When two objects \( \text{Obj}(i) \) and \( \text{Obj}(k) \) have the same value of attribute \( A_j \), \( v_j^{(i)} = v_j^{(k)} \), \( \text{Obj}(i) \) and \( \text{Obj}(k) \) are said to have an indiscernibility relation (or an equivalence relation) on attribute \( A_j \). Also, if \( \text{Obj}(i) \) and \( \text{Obj}(k) \) have the same values for each attribute in subset \( B \) of \( A \), \( \text{Obj}(i) \) and \( \text{Obj}(k) \) are also said to have an indiscernibility (equivalence) relation on attribute set \( B \). These equivalence relations thus partition the object set \( U \) into disjoint subsets, denoted by \( U/B \), and the partition including \( \text{Obj}(i) \) is denoted \( B(\text{Obj}(i)) \).

Example 1: Table 1 shows a data set containing seven objects \( U = \{ \text{Obj}(1), \text{Obj}(2), \ldots, \text{Obj}(7) \} \), two attributes \( A = \{ \text{Systolic Pressure} (\text{SP}), \text{Diastolic Pressure} (\text{DP}), \text{Blood Pressure} (\text{BP}) \} \), and a class set \( C \). The attributes and the class set each have three possible values: \{Low (L), Normal (N), High (H)\}.

Since \( \text{Obj}(1) \) and \( \text{Obj}(3) \) have the same attribute value (N) for attribute SP, they share an indiscernibility relation and thus belong to the same equivalence class for SP. The equivalence partitions for singleton attributes can be derived as follows:

<table>
<thead>
<tr>
<th>Object</th>
<th>Systolic Pressure (SP)</th>
<th>Diastolic Pressure (DP)</th>
<th>Blood Pressure (BP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{Obj}^{(1)}</td>
<td>L</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>\text{Obj}^{(2)}</td>
<td>H</td>
<td>N</td>
<td>H</td>
</tr>
<tr>
<td>\text{Obj}^{(3)}</td>
<td>N</td>
<td>H</td>
<td>N</td>
</tr>
<tr>
<td>\text{Obj}^{(4)}</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>\text{Obj}^{(5)}</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>\text{Obj}^{(6)}</td>
<td>N</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>\text{Obj}^{(7)}</td>
<td>L</td>
<td>N</td>
<td>L</td>
</tr>
</tbody>
</table>

\( U/[\text{SP}] = \{ \\{ \text{Obj}^{(2)}, \text{Obj}^{(5)} \}, \{ \text{Obj}^{(3)}, \text{Obj}^{(6)} \}, \{ \text{Obj}^{(1)}, \text{Obj}^{(4)} \} \}, \) and \( U/[\text{DP}] = \{ \{ \text{Obj}^{(1)}, \text{Obj}^{(2)}, \text{Obj}^{(7)} \}, \{ \text{Obj}^{(3)} \} \}{\text{Obj}^{(4)}}, \{ \text{Obj}^{(5)} \} \{ \text{Obj}^{(6)} \} \} \) is denoted \( B(\text{Obj}(i)) \).

The sets of equivalence classes for subset \( B \) are referred to as \( B \)-elementary sets. Also, \( \{ \text{SP} \}\{ \text{Obj}(1) \} = \{ \text{SP} \}\{ \text{Obj}(3) \} = \{ \text{Obj}^{(1)}, \text{Obj}^{(3)} \} \).

The rough-set approach analyzes data according to two basic concepts, namely the lower and the upper approximations of a set. Let \( X \) be an arbitrary subset of the universe \( U \), and \( B \) be an arbitrary subset of attribute set \( A \). The lower and the upper approximations for \( B \) on \( X \), denoted \( B(X) \) and \( B^*(X) \) respectively, are defined as follows:

\[ B(X) = \{ x \mid x \in U, B(x) \subseteq X \} \] and

\[ B^*(X) = \{ x \mid x \in U \text{ and } B(x) \cap X \neq \emptyset \} \]

3. Notation

The following notation is used in this paper.

\( U \): the universe of all objects;
\( n \): the total number of training examples (objects) in \( U \);
\( \text{Obj}^{(i)} \): the \( i \)-th training example (object), \( 1 \leq i \leq n \);
\( A \): the set of all attributes describing \( U \);
\( m \): the total number of attributes in \( A \);
\( A_j \): the \( j \)-th attribute, \( 1 \leq j \leq m \);
\( v_j^{(i)} \): the value of \( A_j \) for \( \text{Obj}^{(i)} \);
\( B \): an arbitrary subset of \( A \);
\( B^*(X) \): the lower approximation for \( B \) on \( X \);
\( B^*(X) \): the upper approximation for \( B \) on \( X \);
\( C \): the set of classes to be determined;
\( c \): the total number of classes in \( C \);
\( x \): the \( i \)-th class, \( 1 \leq i \leq c \).

4. Learning maximally general rules based on rough sets

In this section, we propose a learning algorithm, based on rough sets, for inducing a coverage set of maximally general rules from training examples. The details of the proposed learning algorithm are described as follows.

The rough-set-based learning algorithm for learning maximally general rules:
Input: A data set with $n$ objects belonging to $c$ classes, each object having $m$ attribute values.

Output: A set of maximally general rules.

Step 1: Partition the object set into disjoint subsets according to class labels. Denote each set of objects belonging to the same class $C_l$ as $X_l$.

Step 2: Find the elementary sets of the singleton attributes.

Step 3: Initialize $l = 0$, where $l$ is used to count the number of the class being processed.

Step 4: Set $l = l + 1$.

Step 5: Set $q = 0$, where $q$ is used to count the number of attributes being processed.

Step 6: Set $q = q + 1$.

Step 7: Compute the lower approximation of each subset $B$ with $q$ attributes for class $X_l$ as:

$$B_l(X_l) = \{ (B_l(x)) \mid x \in U, B_l(x) \subseteq X_l, l \leq k \leq |B(x)| \}.$$ 

Step 8: Choose the $B'_l(x)$ with the maximum value of $|B'_l(x)|$; put it in the certain rule set.

Step 9: Remove the elements in $B'_l(x)$ from any $B_l(x)$ and from $X_l$.

Step 10: Repeat Steps 8 and 9 until all $B_l(x)$s are empty.

Step 11: If $X_l$ is empty, go to Step 19; if $X_l$ is not empty and $q \neq m$, go to Step 6; otherwise, do the next step.

Step 12: Compute the upper approximation of each singleton $B$ (with 1 attribute) for class $X_l$ as:

$$B'(X_l) = \{ (B_l(x)) \mid x \in U, B_l(x) \cap X_l \neq \emptyset, l \leq k \leq |B(x)| \}.$$ 

Step 13: Calculate the plausibility measure of each $B_l(x)$ as:

$$p(B_l(x)) = \frac{|B_l(x) \cap \text{original } X_l|}{|B_l(x)|},$$

where original $X_l$ means the object set of class $l$ in Step 1.

Step 14: Choose the subset $B'_l(x)$ with the maximum plausibility measure; put it in the possible rule set.

Step 15: Set $X'_l = X_l$ and remove the elements in $B'_l(x)$ from any $B_l(x)$ and from $X_l$.

Step 16: If $B_l(x)$ contains no elements in $X_l$, remove it from the upper approximation.

Step 17: Recursively check whether $B'_l(x)$'s specialization can generate a set of new $B''_l(x)$ covering the elements in $(B'_l(x) \cap X_l)$ with plausibility measure larger than that of $B'_l(x)$. Replace $B'_l(x)$ with $B''_l(x)$ in the possible rule set.

Step 18: Repeat Steps 14 to 17 until $X_l$ is empty.

Step 19: Repeat Steps 4 to 18 until $l = c$.

Step 20: Derive the certain rules from the certain rule set.

Step 21: Derive the possible rules from the possible rule set.

After Step 21, a set of maximally general certain and possible rules for covering the training examples can be derived, and can serve as meta-knowledge concerning the given data set.

5. An example

In this section, an example is given to show how the proposed algorithm can be used to generate maximally general certain and possible rules from training examples. Assume that the seven objects in Table 1 are to be learned. The proposed learning algorithm processes this data set as follows.

Step 1: Since three classes exist in the data set, three partitions are formed as follows:

$$X_H = \{ \text{Obj}^{(2)}, \text{Obj}^{(3)}, \text{Obj}^{(6)} \},$$

$$X_N = \{ \text{Obj}^{(1)}, \text{Obj}^{(5)} \},$$

$$X_L = \{ \text{Obj}^{(4)}, \text{Obj}^{(7)} \}.$$ 

Step 2: The elementary sets of the singleton attributes $SP$ and $DP$ are found as follows:

$$U \setminus \{SP\} = \{ (\text{Obj}^{(2)}, \text{Obj}^{(3)}, \text{Obj}^{(5)}, \text{Obj}^{(6)}) \},$$

$$U \setminus \{DP\} = \{ (\text{Obj}^{(1)}, \text{Obj}^{(5)}, \text{Obj}^{(7)}) \}.$$ 

Step 3: $l$ is set to 0.

Step 4: $l = l + 1$, where $l$ is used to count the identification number of the class being processed.

Step 5: $q$ is set to 0.
Step 6: $q = q+1=1$, where $q$ is used to count the number of attributes being processed.

Step 7: The lower approximation of a single attribute ($q=1$) for class $X_H$ is first calculated. Since the three objects $Obj(2), Obj(5), Obj(6)$ are included in $X_H$, thus:

$SP(X_H) = \{(Obj(2), Obj(5))\}$,
$DP(X_H) = \emptyset$.

Step 8: Since the subset $\{Obj(2), Obj(5)\}$ has the maximum number of elements in $SP(X_H)$ and $DP(X_H)$, it is first put in the certain rule set.

Step 9: The elements in $\{Obj(2), Obj(5)\}$ are then removed from $SP(X_H)$, $DP(X_H)$ and $X_H$. Thus:

$SP(X_H) = \emptyset$,
$DP(X_H) = \emptyset$, and
$X_H = \{Obj(6)\}$.

Step 10: Steps 8 and 9 are repeated until all lower approximations are empty. In this case, since both $SP(X_H)$ and $DP(X_H)$ are empty, the next step is executed.

Step 11: Since $X_H$ is still not empty and $q (=1) \neq m (=2)$, Steps 6 to 10 are executed again. $q = 1+1 = 2$. The lower approximation of two attributes ($q=2$) for class $X_H$ ($\{Obj(6)\}$) is calculated as:

$SPDP(X_H) = \emptyset$.

Therefore, no other subsets of lower approximations are put in the certain rule set. Since $X_H$ is still not empty and $q (=2) = m (=2)$, Step 12 is then executed to find possible rules.

Step 12: The upper approximation of each singleton attribute for class $X_H$ is calculated. Since only $\{Obj(6)\}$ is included in $X_H$ after Step 12, thus:

$SP^*(X_H) = \{(Obj(3), Obj(6))\}$, and
$DP^*(X_H) = \{(Obj(3), Obj(5), Obj(6))\}$.

Step 13: The plausibility measure of the subset $SP_N(x) = \{Obj(3), Obj(6)\}$ is calculated as:

$p(SP_N(x)) = \frac{1}{2} = 0.5$.

Similarly, the plausibility measure of the subset $DP_H(x) = \{Obj(3), Obj(5), Obj(6)\}$ is calculated as:

$p(DP_H(x)) = \frac{1}{3} = 0.33$.

Step 14: Since $SP_N(x) = \{Obj(3), Obj(6)\}$ has the maximum plausibility measure, it is put in the possible rule set.

Step 15: $X'_H = X_H = \{Obj(6)\}$; $Obj(3)$ and $Obj(6)$ are removed from $SP^*(X_H)$, $DP^*(X_H)$ and $X_H$. Thus:

$SP^*(X_H) = \emptyset$,
$DP^*(X_H) = \{Obj(3)\}$, and
$X_H = \emptyset$.

Step 16: Since $\{Obj(3)\}$ contains no elements in $X_H$, it is removed from the upper approximation. Thus:

$SP^*(X_H) = \emptyset$,
$DP^*(X_H) = \emptyset$, and
$X_H = \emptyset$.

Step 17: $SP_N(x) \cap X'_H = \emptyset$. $SP_N(x) (= \{Obj(3), Obj(6)\})$ is then specialized by the attribute $DP$ as:

$SP_N,DP_N(Obj(6)) = \emptyset$,
$SP_N,DP_N(Obj(3)) = \emptyset$, and
$SP_N,DP_N(Obj(6)) = \{\{Obj(3), Obj(6)\}\}$.

The plausibility measure of the subset $SP_N,DP_H(Obj(6))$ is calculated as:

$p(SP_N,DP_H(Obj(6))) = \frac{1}{2} = 0.5$.

Since $p(SP_N,DP_H(Obj(6))) \leq p(DP_H(Obj(6)))$, $(SP_N,DP_H)$ does not replace $SP_N$ in the possible rule set.

Step 18: Since $X_H$ is empty, Step 19 is executed.

Step 19: Steps 4 to 17 are repeated for finding the certain and possible rule sets for classes $X_N$ and $X_L$. The final results are shown as follows:

Certain rule sets:

$X_H = \{Obj(2), Obj(5)\}$,
$X_N = \emptyset$, and
$X_L = \{Obj(6)\}$.

Possible rule sets:

$X_H = \{Obj(3), Obj(6)\}$,
$X_N = \{Obj(3), Obj(5), Obj(6)\}$, and
$X_L = \{Obj(3), Obj(6)\}$.

Step 20: The certain rules derived from the certain rule set are shown as follows:

1. If Systolic Pressure = High Then Blood Pressure = High.
2. If Diastolic Pressure = Low Then Blood Pressure = Low.

Step 21: The possible rules derived from the possible rule set are shown as follows:
1. If Diastolic Pressure = High Then Blood Pressure = High,
   with plausibility = 0.5.
2. If Systolic Pressure = Normal Then Blood Pressure = Normal,
   with plausibility = 0.5.
3. If Systolic Pressure = Low and Diastolic Pressure = Normal Then Blood Pressure = Low,
   with plausibility = 0.5.

These certain and possible rules can then serve as meta-knowledge concerning the given training set.

6. Conclusions

In this paper, we have proposed a new learning algorithm that can discover maximally-general certain and possible rules from training examples. The proposed method adopts the concepts of rough sets to partition the training instances according to some criteria. It first calculates the lower approximations and the upper approximations. It then adopts an iterative process to find a coverage set of maximally general certain and possible rules. The proposed method can then be used to build a prototype knowledge base.

References