
ELSEVIER

Journal of Information Sciences 107 (1998) 169–176

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

A rough set approach to attribute generalization in data mining

Chien-Chung Chan [1]

Department of Mathematical Sciences, University of Akron, Akron, OH 44325-4002, USA

Received 1 September 1996; accepted 10 July 1997

Abstract

This paper presents a method for updating approximations of a concept incrementally. The results can be used to implement a quasi-incremental algorithm for learning classification rules from very large data bases generalized by dynamic conceptual hierarchies provided by users. In general, the process of attribute generalization may introduce inconsistency into a generalized relation. This issue is resolved by using the inductive learning algorithm, LERS based on rough set theory.

Keywords: Rough sets; Data mining; Inductive learning

1. Introduction

In inductive machine learning and data mining from very large data bases, it is well known that background knowledge can be used as an effective guidance for extracting useful and interesting information from the data. When using relational data bases as sources of data mining, it has been shown in [1,2] that conceptual hierarchies defined on the domains of attributes can be used to reduce source relations into generalized relations, thus effective data mining can be accomplished. Conceptual hierarchies usually vary based on users' views and interests, therefore, it is important to handle dynamic conceptual hierarchies effectively.

[1] E-mail: chan@cs.uakron.edu.

0020-0255/98/$19.00
PII: S0020-0255(97)10047-0

The use of conceptual hierarchies to generalize relations is similar to the situation of discretizing attributes with continuous domains. In general, a generalized table may be inconsistent. Thus, a data mining tool must include a mechanism to deal with inconsistent data.

Some data mining tasks from the rough sets perspective have been discussed in [3]. Our focus here is on the task of generating classification rules from data. Based on rough sets [4] and the concept of lower and upper boundary sets [5], we introduce a method for updating approximations by considering adding and deleting one attribute at a time. When a generalization is applied to an attribute, we can use the method to update approximations by deleting the original attribute first, followed by inserting the generalized one using information of current approximations. This feature can support incremental updating of approximations, which is essential to dealing with dynamic attribute generalization.

To handle inconsistent data, we use the inductive learning algorithm LERS [6,7] as a rule generator. Thus, the proposed algorithm can be used to learn minimal discriminant rules from data bases in light of dynamic conceptual hierarchies and inconsistency.

In the following section, we introduce terms and definitions to be used in the paper. In Section 3, we present results that can be used for updating approximations using one attribute at a time. A quasi-incremental algorithm for learning classification rules is outlined in Section 4. Section 5 concludes the paper.

2. Terms and definitions

A *decision table* is a collection U of objects that are described by a finite set A of attributes. One attribute in A is designated as a *decision attribute*, and the rest of the attributes are called *condition attributes*. An *approximation space* is a pair (U, R) where R is an equivalence relation defined on U. A partially ordered set of equivalence relations defined on the domain of an attribute is called a *conceptual hierarchy*. We also call the equivalence relations in a conceptual hierarchy *attribute generalizations*. Given an approximation space (U, R), for any subset X of U, X can be described by a pair of sets, *lower approximation* of X and *upper approximation* of X, denoted as $\underline{R}X$ and $\bar{R}X$ respectively. A subset X of U is definable in (U, R) if and only if $\underline{R}X = X = \bar{R}X$. The *lower boundary* of X in (U, R) is defined as $\underline{\Delta}X = X - \underline{R}X$ and the *upper boundary* of X in (U, R) is defined as $\bar{\Delta} = \bar{R}X - X$. Thus, a subset X is definable in (U, R) if and only if $\underline{\Delta}_R X = \varnothing = \bar{\Delta}_R X$. For any subset X of U, the lower and upper approximations of X are always definable in an approximation space.

3. Updating approximations incrementally

In this section, we consider the problem of updating approximations of a subset X of U in terms of adding and removing one attribute at a time. The concept of boundary sets were introduced in [5] where it has been used as a tool for learning rules from examples. In the following, boundary sets are used to update approximations of a subset X incrementally.

Proposition 3.1. *Let a be an attribute in A, and a is not in P. The lower approximation of X by adding a to P can be updated in terms of $\underline{P}X$, $\underline{\Delta}_P X$, $\underline{\{a\}}X$, and $\underline{\Delta}_{\{a\}}X$ as*

$$\underline{P \cup \{a\}}X = \underline{P}X \cup \underline{\{a\}}X \cup Y,$$

where $Y = \{x \text{ in } \underline{\Delta}_P X \cap \underline{\Delta}_{\{a\}}X \mid \bigcap_{b \in P \cup \{a\}} [x]_b \subseteq X\}$.

Proof. Let X be a subset of U and x be an example in U such that $x \in \underline{P \cup \{a\}}X$. If x is not in $\underline{P}X \cup \underline{\{a\}}X$, then x must be in Y. Because x is not in $\underline{P}X \cup \underline{\{a\}}X$ if and only if x is in $\underline{\Delta}_P X \cap \underline{\Delta}_{\{a\}}X$, and $x \in \underline{P \cup \{a\}}X$ if and only if $\bigcap_{b \in P \cup \{a\}} [x]_b \subseteq X$. Therefore, we have x is in Y. □

Proposition 3.2. *Let a be an attribute in P. The lower approximation of X by removing a from P can be updated in terms of $\underline{P}X$ and $\underline{\Delta}_{P-\{a\}}X$ as*

$$\underline{P - \{a\}}X = \underline{P}X - \underline{\Delta}_{P-\{a\}}X,$$

where

$$\underline{\Delta}_{P-\{a\}}X = \{x \text{ in } \bigcap_{b \in P-\{a\}} \underline{\Delta}_{\{b\}}X \mid \bigcap_{b \in P-\{a\}} [x]_b \not\subset X\}.$$

Note that attribute a is redundant when $\underline{\Delta}_{P-\{a\}}X(\underline{P}X) = \emptyset$.

Proof. In general, we have $\underline{P - \{a\}}X \subseteq \underline{P}X$. In terms of lower boundary sets, we have $\underline{\Delta}_P X \subseteq \underline{\Delta}_{P-\{a\}}X$. The contribution of an attribute a to the lower approximation of X by P can be characterized by the set $\underline{\Delta}_{P-\{a\}}X - \underline{\Delta}_P X = \{x \text{ in } U \mid x \in \underline{\Delta}_{P-\{a\}}X \text{ and } x \notin \underline{\Delta}_P X\}$. Therefore, the effect of removing attribute a from P to the lower approximation of X is $\underline{P - \{a\}}X = \underline{P}X - (\underline{\Delta}_{P-\{a\}}X - \underline{\Delta}_P X) = \underline{P}X - \underline{\Delta}_{P-\{a\}}X - \underline{\Delta}_P X$, which can be simplified as $\underline{P}X - \underline{\Delta}_{P-\{a\}}X$, because $\underline{P}X \cap \underline{\Delta}_P X = \emptyset$. □

Proposition 3.3. *Let a be an attribute in A, and a is not in P. The upper approximation of X by adding a to P can be updated in terms of $\bar{\Delta}_P X$ as*

$$\overline{P \cup \{a\}}X = X \cup (\bar{\Delta}_P X - Z)$$

where Z denotes the set of extra objects that is definable by adding attribute a to P and it is defined as

$$Z = \{x \text{ in } \bigcap_{b \in P \cup \{a\}} \bar{\Delta}_{\{b\}} X \mid \bigcap_{b \in P \cup \{a\}} [x]_b \subseteq \bigcap_{b \in P \cup \{a\}} \bar{\Delta}_{\{b\}} X\}.$$

Proof. Let $x \in \overline{P \cup \{a\}} X$ and $x \notin X$. Then x is in $\bar{\Delta}_{p \cup \{a\}} X$ from the definition of upper boundary sets. This implies that x is in $\bar{\Delta}_p X$ and $\bigcap_{b \in P \cup \{a\}} [x]_b \cap X \neq \emptyset$. Because $\left(\bigcap_{b \in P \cup \{a\}} \bar{\Delta}_{\{b\}} X\right) \cap X = \emptyset$. Therefore $\bigcap_{b \in P \cup \{a\}} [x]_b$ is not a subset of $\bigcap_{b \in P \cup \{a\}} \bar{\Delta}_{\{b\}} X$. Thus, x is not in Z. Therefore, x is in $\bar{\Delta}_p X - Z$. □

Proposition 3.4. *Let a be an attribute in P. The upper approximation of X by removing a from P can be updated in terms of $\bar{\Delta}_p X$ as*

$$\overline{P - \{a\}} X = X \cup \bar{\Delta}_p X \cup Z'$$

where $Z' = \left\{x \text{ in} \bigcap_{b \in P - \{a\}} \bar{\Delta}_{\{b\}} X \mid \bigcap_{b \in P - \{a\}} [x]_b \not\subset \bigcap_{b \in P - \{a\}} \bar{\Delta}_{\{b\}} X\right\}$.

Proof. Let $x \in \overline{P - \{a\}} X$ and $x \notin X$, then x is in $\bar{\Delta}_{p - \{a\}} X$ by definition. In general, we have $\bar{\Delta}_p X \subseteq \bar{\Delta}_{p - \{a\}} X$. Therefore, if $x \in \overline{P - \{a\}} X$ and $x \notin X$ and $x \notin \bar{\Delta}_p X$, then x must be in Z', because $\bigcap_{b \in P - \{a\}} [x]_b \subseteq \bigcap_{b \in P - \{a\}} \bar{\Delta}_{\{b\}} X$ if and only if $x \in \overline{P - \{a\}} X$. This would contradict the assumption that $x \in \overline{P - \{a\}} X$. □

Example. We use Table 1 to illustrate the above results. For simplicity, we will use an attribute name to denote a singleton set of attribute.

From Table 1 the partitions generated by single attributes are:

$$a^* = \{\{e1, e2, e3, e4\}, \{e5, e6\}, \{e7, e8\}\},$$
$$b^* = \{\{e1, e3\}, \{e2, e4, e5, e6\}, \{e7, e8\}\},$$
$$c^* = \{\{e1, e3, e5, e6\}, \{e2, e4\}, \{e7, e8\}\},$$
$$d^* = \{\{e1, e2, e3, e4, e5, e6\}, \{e7, e8\}\}.$$

Let $X = \{e1, e2, e5, e6\}$.

Table 1
A decision table with attributes $A = \{a, b, c, d\}$ and $U = \{e1, \ldots, e8\}$

Example	a	b	c	d
$e1$	0	L	0	L
$e2$	0	R	1	L
$e3$	0	L	0	L
$e4$	0	R	1	L
$e5$	1	R	0	L
$e6$	1	R	0	L
$e7$	2	S	2	H
$e8$	2	S	2	H

Then the lower approximations of X by single attributes are

$$\underline{a}X = \{e5, e6\},$$
$$\underline{b}X = \{e1, e3\},$$
$$\underline{c}X = \{e1, e3, e5, e6\},$$
$$\underline{d}X = \varnothing$$

The lower boundaries of X by single attributes are

$$\underline{\Delta}_a X = X - \underline{a}X = \{e1, e3\},$$
$$\underline{\Delta}_b X = X - \underline{b}X = \{e5, e6\},$$
$$\underline{\Delta}_c X = X - \underline{c}X = \varnothing,$$
$$\underline{\Delta}_d X = X - \underline{d}X = \{e1, e3, e5, e6\}.$$

In the following, we consider updating lower approximations of X by adding and removing one attribute.

3.1. Adding a new attribute

Let $P = \{b\}$, so $\underline{P}X = \{e1, e3\}$ and $\underline{\Delta}_P X = \{e5, e6\}$.

Let $R = P \cup \{d\}$. To compute $\underline{R}X$, we first compute the set Y as $Y = \{X \text{ in } \underline{\Delta}_P X \cap \underline{\Delta}_{\{d\}} X \mid \bigcap_{b \in P \cup \{d\}} [x]_b \subseteq X\} = \varnothing$.

From $\underline{\Delta}_P X \cap \underline{\Delta}_d X = \underline{\Delta}_b X \cap \underline{\Delta}_d X = \{e5, e6\}$ and

$$[e5]_b \cap [e5]_d = \{e2, e4, e5, e6\} \not\subset X \text{ and}$$
$$[e6]_b \cap [e6]_d = \{e2, e4, e5, e6\} \not\subset X.$$

Therefore, both $e5$ and $e6$ are not in Y.

Now we compute $\underline{R}X$ by

$$\underline{R}X = \underline{P}X \cup \underline{d}X \cup Y = \underline{b}X \cup \underline{d}X \cup Y = \{e1, e3\}.$$

3.2. Removing an attribute

Next, we show how to update lower approximation of X when attribute b is removed from $R = \{b, d\}$.

From the above and Proposition 3.2, we have $\underline{R}X = \{e1, e3\}$, $\underline{\Delta}_R X = \{e5, e6\}$, and $Y = \underline{\Delta}_{R-\{b\}} X = \{e1, e3, e5, e6\}$. Therefore, we have $\underline{R - \{b\}}X = \underline{R}X - Y = \{e1, e3\} - \{e1, e3, e5, e6\} = \varnothing$.

In the following, we consider updating upper approximations by adding and removing one attribute at a time.

3.3. Updating upper approximations

From the above table, the upper approximations of X by single attributes are all equal, namely,

$$\bar{a}X = \bar{b}X = \bar{c}X = \bar{d}X = \{e1, e2, e3, e4, e5, e6\}.$$

Therefore, the upper boundaries of X by single attributes are also equal. We have

$$\bar{\Delta}_a X = \bar{\Delta}_b X = \bar{\Delta}_c X = \bar{\Delta}_d X = \{e2, e4\}.$$

3.4. Adding a new attribute

Let $P = \{b\}$. We have $\bar{P}X = \bar{b}X = \{e1, e2, e3, e4, e5, e6\}$ and $\bar{\Delta}_P X = \bar{\Delta}_b X = \{e2, e4\}$.

Let $R = P \cup \{d\}$. To compute $\bar{R}X$, we first compute Z as

$$Z = \{x \text{ in } \bar{\Delta}_{\{b\}}X \cap \bar{\Delta}_{\{d\}}X \mid [x]_b \cap [x]_d \subseteq \bar{\Delta}_{\{b\}}X \cap \bar{\Delta}_{\{d\}}X\} = \varnothing.$$

Then, update the upper approximation of X by $P \cup \{d\}$ as

$$\bar{R}X = X \cup (\bar{\Delta}_P X - Z) = \{e1, e2, e3, e4, e5, e6\}.$$

3.5. Removing an attribute

Next, we show how to update upper approximation of X when attribute b is removed from $R = \{b, d\}$.

From the above and Proposition 3.4, we have $\bar{R}X = \{e1, e2, e3, e4, e5, e6\}$ and $\bar{\Delta}_R X = \{e2, e4\}$.

Next, we compute Z' as

$$Z' = \{x \text{ in } \bar{\Delta}_{\{d\}}X \mid [x]_d \not\subset \bar{\Delta}_{\{d\}}X\} = \{e2, e4\}.$$

Now, the upper approximation of X by $R - \{b\}$ is updated as

$$\overline{R - \{b\}}X = X \cup \bar{\Delta}_R X \cup Z' = \{e1, e2, e3, e4, e5, e6\}.$$

In summary, the above results show that we can update approximations incrementally by using the boundary sets of single attributes and the intersection of the sets denoted by corresponding attribute–value pairs. This can be used to implement a data mining tool that is capable of dealing with dynamic conceptual hierarchies provided by users.

4. Learning classification rules from data

Based on the results in Section 3, we propose a top–down algorithm for learning classification rules from data. The algorithm uses LERS learning algorithm to generate rules, therefore, the learned rules are disjunctive minimal discriminant descriptions of target classes. When a table is inconsistent, the algorithm learn certain rules from lower approximations and possible rules

from upper approximations [8]. For consistent tables, we have only one set of rules.

Inputs: 1. A decision table with attribute set A and a decision attribute d in A.
2. Attribute generalizations provided by users.
Outputs: certain and possible classification rules for classes in the decision attribute.

```
begin
    From the decision table,
    for each class X_i in the decision attribute do
  begin
      Compute lower and upper approximations generated by single
      condition attributes;
      Compute lower and upper boundary sets by single condition attributes;
      Find a cover R for the set of condition attributes;
      Compute lower and upper approximations generated by R;
      Compute lower and upper boundary sets generated by R;
  end;
  repeat
        Get a generalization g_i for attribute a provided by the user;
    if attribute a is in R
      then
      begin
        Update approximations and boundary sets by R − {a};
        Update approximations and boundary sets by R ∪ {a} based on g_i;
      end;
      else
        Update approximations and boundary sets by R ∪ {a};
    Find a cover for the new set of attributes;
    if rules desired
    then generate rules by LERS;
  until terminated by user;
end.
```

The concept of cover of attributes was introduced in [9]. An algorithm for finding covers can be found in [5]. Algorithms related to the LERS family of learning programs can be found in [6,7].

5. Conclusions

Conceptual hierarchies have been used to generalize very large data bases in order to support effective data mining tasks. These generalizations usually are

user-dependent, therefore, they are dynamic in nature. In addition, the process of generalizations may introduce inconsistency into generalized data. Data inconsistency can be handled effectively by approximations in rough set theory. In this paper, we have presented a method for updating approximations incrementally which can be used as an effective tool to deal with dynamic attribute generalizations. Combining the proposed method and the LERS inductive learning algorithm, we have given a quasi-incremental algorithm for learning classification rules from data bases.

References

[1] Cai Yandong, N. Cercone, J. Han, Attribute-oriented induction in relational databases, in: G. Piatetsky-Shapiro, W.J. Frawley (Eds.), Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, MA, 1991, pp. 213–228.

[2] J. Han, Y. Fu, Attribute-oriented induction in data mining, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 399–421.

[3] V.V. Raghavan, H. Sever, The state and rough sets for databse mining applications, in: 23rd Annual Computer Science Workship on Rough Sets and Database Mining, Nashville, Tennessee, 2 March 1995, pp. 1–11.

[4] Z. Pawlak, Rough sets, Int. J. Comput. Inform. Sci. 11 (1982) 341–356.

[5] C.-C. Chan, J.W. Grzymala-Busse, Rough-set boundaries as a tool for learning rules from examples, in: Proceedings of Fourth International Symposium on Methodology for Intelligent Systems, 12–14 October 1989, pp. 281–288.

[6] J.W. Grzymala-Busse, The LERS family of learning systems based on rough sets, in: Proceedings of the Third Midwest Artificial Intelligence and Cognitive Science Society Conference, 12–14 April 1991, pp. 103–107.

[7] C.-C. Chan, Incremental learning of production rules from examples under uncertainty: A rough set approach, International Journal of Software Engineering and Knowledge Engineering 1 (4) (1991) 439–461.

[8] J.W. Grzymala-Busse, Knowledge acquisition under uncertainty: A rough set approach, J. Intell. Robotic Syst. 1 (1988) 3–16.

[9] J.W. Grzymala-Busse, Managing Uncertainty in Expert Systems, Kluwer Academic Publishers, Dordrecht, 1991.