Pergamon

Appl. Math. Lett. Vol. 7, No. 4, pp. 29–34, 1994
Copyright©1994 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0893-9659/94 $7.00 + 0.00

0893-9659(94)00046-8

Object Aggregation and Cluster Identification: A Knowledge Discovery Approach*

X.-H. Hu
Department of Computer Science, University of Regina
Regina, S.K., Canada S4S 0A2

(Received February 1994; accepted March 1994)

Abstract—A method for object aggregation and cluster identification has been proposed for knowledge discovery in databases. By integrating conceptual clustering and machine learning (especially learning-from-examples) paradigms, the method classifies the data into different clusters, extracts the characteristics of each cluster, and discovers knowledge rules based on the relationships among different clusters. Different kinds of knowledge rules, including hierarchical, equivalence and inheritance rules can be discovered efficiently.

Keywords—Knowledge discovery in databases, Conceptual clustering.

1. INTRODUCTION

An attribute-oriented method [1] was developed for discovering knowledge in relational databases, which integrates *learning from examples* techniques with database operations and extracts generalized data from actual data in the databases. A key to the success of the approach is the *attribute-oriented concept tree ascension for generalization.*

Previous studies on the method assume that the *pre-existence* of concept hierarchy information (provided by users, experts or data analysts). However, such information may not be available in many applications. It is important to discover data regularities in the absence of concept hierarchy information.

An algorithm which integrates conceptual clustering and machine learning, clusters data automatically, extracts characteristics for different classes and derives knowledge rules according to the relationships between different classes is presented.

2. APPROACHES TO CONCEPT CLUSTERING

Conceptual clustering, originally developed by Michalski and Stepp [2] as an extension to the process of numerical taxonomy, groups objects with common properties into clusters and extracts the characteristic of each cluster over a set of data objects. Currently, there are two views regarding conceptual clustering: one represents an extension to techniques of numerical taxonomy, whereas the other is a form of *learning-by-observations* or *concept formation* as distinct from methods of *learning-from-examples* or *concept identification.* The clustering algorithms which have been framed as extensions to numerical taxonomy techniques include CLUSTER/2 (see [2]) and COBWEB (see [3]); whereas those which can be viewed as an extension of *learning-by-observations* include HUATAO (see [4]) and Thought/KD1 (see [5]). We propose a technique

*The author is a member of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada, the Natural Sciences and Engineering Research Council, and the participation of PRECARN Associates, Inc.

which combines the advantages of both and discovers knowledge from databases by first clustering data using a numerical taxonomy, then extracting a characteristic feature for the cluster, and finally treating each cluster as a positive example as in *learning-from-examples* and using existing machine learning methods to derive knowledge rules.

3. KNOWLEDGE DISCOVERY BY CONCEPTUAL CLUSTERING

Our method is divided into three phases. Phase 1 uses a numerical taxonomy to classify the object set. Phase 2 assigns conceptual descriptions to object classes. Phase 3 finds the hierarchical, inheritance and domain knowledge based on different relationships among classes. For a numerical taxonomy, various measures of similarity have been proposed. Most of them are based on a Euclidean measure of distance between numerical attributes. Consequently, the algorithm works well only on numerical data. Many database applications use nonnumerical data. A new measure is proposed using the number of common attribute values in two data sets S_1 and S_2 as a similarity measurement, called *sim_value*(S_1, S_2). Notice that for any data set S, *sim_value*$(S, S) = 0$.

Algorithm [CDC]. Conceptual data clustering.
Input. A set of data stored in the relational table.
Output. A cluster hierarchy of the data set.
Method.

```
1. (Preliminary): Generalize attributes to a "desirable level [1]".
2. (Concept clustering): candidate_set := the data set obtained at Step 1.
       repeat for each pair of $S_1$ and $S_2$ in candidate_set, calculate $sim\_value(S_1, S_2)$.
        form clusters for the candidate_set based on a threshold for sim_value. (Note: The threshold varies for different candidate_sets and can be set by user/expert or determined by the analysis of sim_value distribution).
        remove redundant clusters.
       if there is a new cluster produced
       then form the hierarchy based on the new and untouched* clusters
       candidate_set := the new cluster ∪ the untouched clusters
       until candidate_set = $\phi$.
```

*Note: An untouched cluster is a cluster which is not a component of any newly formed cluster.

Three kinds of knowledge rules can be discovered from object classes:

(1) *hierarchical knowledge rules,*
(2) *the relationship between different attributes* and
(3) *inheritance knowledge rules.*

Given a set of data, suppose that the data is clustered into a hierarchy as illustrated in Figure 1 after phase 1. In Figure 1, H's denote the clusters in the hierarchy, $H_{i,j}$ is a subclass of H_i ($1 \leq i \leq k$, where k is the number of clusters in level 2). Let the conceptual descriptions assigned to these classes be $D_1, \ldots, D_k, D_{1,1}, D_{1,l}, \ldots, D_{k,1}, \ldots, D_{k,m}, \ldots$, and so on. The values of $k, l, \ldots, m$ depend on the actual data set.

For rule formation, there are three algorithms of knowledge discovery: *Hierarchical Knowledge Discovery (HKD)*, *Attribute Knowledge Discovery (AKD)*, and *Inheritance Knowledge Discovery (IKD)* (see [6]). For HKD, new rules are discovered by finding all of the possible implications between the descriptions of clusters in a cluster and those in its father cluster, namely $D_{i,j} \rightarrow D_i$. For AKD, the algorithm just looks for the characteristic description for each cluster, based on the relationship on different attribute values, then gives the result in terms of a logically equivalent form. For IKD, which is a modification of HKD, labels are used, which are either explicitly defined by users/experts in terms of domain knowledge or labels are produced automatically by the system.

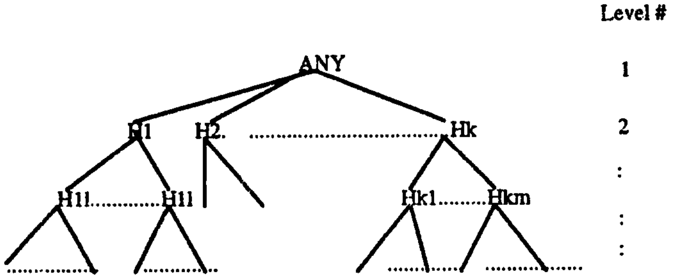


Figure 1. Conceptual hierarchy.

Table 1. The animal world.

#	Animal	Hair	Teeth	Eyes	Feathers	Feet	Eats	M	Flies	Swims
1	tiger	Y	pointed	forward	N	claw	meat	Y	N	Y
2	cheetah	Y	pointed	forward	N	claw	meat	Y	N	Y
3	giraffe	Y	blunt	side	N	hoof	grass	Y	N	N
4	zebra	Y	blunt	side	N	hoof	grass	Y	N	N
5	ostrich	N	N	side	Y	claw	grain	N	Y	N
6	penguin	N	N	side	Y	web	fish	N	N	N
7	albatross	N	N	side	Y	claw	grain	N	Y	Y
8	eagle	N	N	forward	Y	claw	meat	N	Y	N

Table 2. # of common attribute values after 1st iteration.

#	1	2	3	4	5	6	7	8
1	0	9	4	4	2	2	1	3
2	9	0	4	4	2	2	1	3
3	4	4	0	9	3	1	2	1
4	4	4	9	0	3	1	2	1
5	2	2	3	3	0	7	8	6
6	2	2	1	1	7	0	5	5
7	1	1	2	2	8	5	0	7
8	3	3	1	1	6	5	7	0

Cluster labeling plays an important role in knowledge discovery. The new rules discovered can be formed as

$D_1\&D_{i,j}\&\ldots\&D_{i,j,\ldots,k,l} \rightarrow \mathrm{LABEL}(H_{i,j,\ldots k,l})$, or

$\mathrm{LABEL}(H_{i,j,\ldots k})\&D_{i,j,\ldots k,l} \rightarrow \mathrm{LABEL}(H_{i,j,\ldots k,l})$

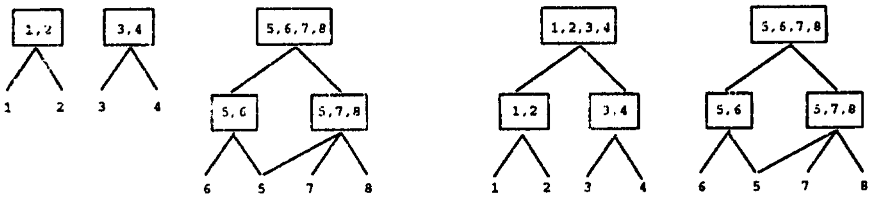
where the condition part of the rule consists of the conjunction of the description of the current cluster and the label of its father's cluster.

An example of this is given in the animal world depicted in Table 1, which is viewed as the data set passed the preliminary step.

The data in row 1 means that a tiger is a animal with hair, pointed teeth, forward eyes, claw feet, and no feathers, it gives milk and cannot fly, but can swim.

In Phase 1, the clustering algorithm CDC is applied to classify the data in Table 1. After the first iteration, the number of common attribute values between each pair of data is computed in Table 2. For example, the '9' in row 1, column 2 is computed by counting the number of common attributes between the data set in row 1 and row 2 of Table 1.

Suppose 6 is chosen as the threshold sim_value, the algorithm produces 8 clusters (1,2), (2,1), (3,4), (4,3), (5,6,7,8), (6,5), (7,5,8), (8,5,7). Thus, 5 distinct clusters (1,2), (3,4), (5,6,7,8), (5,6), (5,7,8) are formed after deleting redundant ones. A hierarchy is formed as depicted in Figure 2a.



(a) 1st iteration (b) 2nd iteration

Figure 2. Concept hierarchy.

	(1,2)	(3,4)	(5,6,7,8)
(1,2)	0	5	0
(3,4)	5	0	0
(5,6,7,8)	0	0	0

(a) iteration 2

	(1,2,3,4)	(5,6,7,8)
(1,2,3,4)	0	0
(5,6,7,8)	0	0

(b) iteration 3

Figure 3. # of common attribute values.

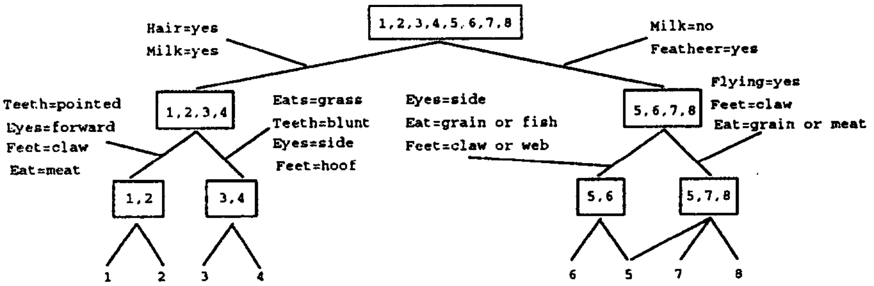


Figure 4. Conceptual hierarchy after 3rd iteration.

Next, the algorithm CDC is applied to (1,2), (3,4), (5,6,7,8). CDC calculates the similarity for the three clusters (1,2), (3,4), (5,6,7,8). The common attribute values are presented in Figure 3(a). Let 5 be the threshold value at this iteration. It results in the hierarchy shown in Figure 2(b).

Finally, the algorithm CDC is applied to (1,2,3,4), (5,6,7,8). After the third iteration, the common attribute values between these two clusters are presented in Figure 3(b) and the resultant conceptual hierarchy is illustrated in Figure 4. Notice that the characteristic descriptions of each cluster are the common values for all the data in the cluster.

In phase 3, the three Knowledge Discovery Algorithms HKD, AKD, and IKD (see [6]) are applied to the hierarchy depicted in Figure 4, respectively, resulting in three sets of rules as depicted in Tables 3(a), 3(b), and 4.

By substituting the labels by the names given by an expert as shown in Table 5, a set of meaningful rules can be obtained as shown in Table 6.

4. DISCUSSION AND CONCLUSION

Our method not only produces the clusters and their corresponding descriptions, but also discovers the hierarchical knowledge between different clusters based on their relationships and the inheritance knowledge. The method has the following advantages:

Table 3.

(a) Hierarchical knowledge rules

#	Knowledge rules discovered by HKD
1	Feet=hoof → Milk=yes
2	Teeth=pointed ∨ blunt → Milk=yes
3	Eat=grass → Milk=yes
4	Feet =hoof → Hair=yes
5	Teeth=pointed ∨ blunt → Hair=yes
6	Eat=grass → Hair=yes

(b) Equivalence rules

#	Knowledge rules discovered by AKD
1	Hair=yes ↔ Milk=yes
2	Feathers=yes ↔ Milk=no

Table 4. Inheritance knowledge rules.

#	Knowledge rules discovered by IKD
1	Label(1,2,3,4,5,6,7,8) ∧ (hair=yes ∨ Milk=yes) → Label(1,2,3,4)
2	Label(1,2,3,4,5,6,7,8) ∧ (Feathers=yes ∨ Milk=no) → Label(5,6,7,8)
3	Label(1,2,3,4) ∧ (Teeth=pointed ∨ Eyes=forward ∨ Feet=claw ∨ Eats=meat) → Label(1,2)
4	Label(1,2,3,4) ∧ (Teeth=blunt ∨ Eyes=side ∨ Feet=Hoof ∨ Eats=grass) → Label(3,4)

Table 5. Names list.

Labels given by system	Names given by expert or user
Label(1,2,3,4,5,6,7,8)	animals
Label(1,2,3,4)	mammals
Label(5,6,7,8)	birds
Label(1,2)	carnivorous mammals
Label(3,4)	ungulate
Label(5,6)	nonflying birds
Label(5,7,8)	meaningless cluster

Table 6. A set of meaningful rules after substitution.

#	After renaming the labels by experts or users
1	(Thing=animal) ∧ (hair=yes ∨ Milk=yes) → mammal
2	(Thing=animal) ∧ (Feathers=yes ∨ Milk=no) → bird
3	(Animal=mammal) ∧ (Teeth=pointed ∨ Eyes=forward ∨ Feet=claw ∨ Eats=meat) → carnivorous mammal
4	(Animal=mammal) ∧ (Teeth=blunt ∨ Eyes=side ∨ Feet=Hoof ∨ Eats=grass) → ungulate

(1) A hierarchy is discovered automatically without assistance. The number of clusters and the levels of the hierarchy are determined by the clustering algorithm; it is unlike the famous CLUSTER/2 in which the user must specify the number of final clusters and the initial seeds in the beginning.

(2) Objects are not assigned to clusters absolutely.

(3) All attributes are potentially significant.

(4) The threshold value has a big influence on whether or not an instance is admitted to a class. We can vary the threshold, obtain different hierarchy tables so the algorithm can generate different sets of rules to meet the needs of varied applications.

Our method is simple, but efficient, for discovering knowledge from a database, based on the observation of the cognitive process of human discovery, depending on the classification and abstraction of given data. A test of this method in a real world database will be reported in the future.

REFERENCES

1. J. Han, Y. Cai and N. Cercone, Knowledge discovery in databases: An attribute-oriented approach, *Proc. 18th Int'l. Conf. Very Large Data Bases*, Vancouver, B.C., Canada, pp. 547–559, (1992).
2. R. Michalski and R. Stepp, Automated construction of classifications: Conceptual clustering versus numerical taxonomy, *IEEE Trans. Pattern Analysis and Machine Intelligence* **5** (4), 396–409 (1983).
3. D. Fisher, Improving inference through conceptual clustering, *Proc. 1987 AAAI Conf.*, Seattle, WA, July 1987, pp. 461–465.
4. Y. Cheng and K.S. Fu, Conceptual clustering in knowledge organization, *IEEE Trans. Pattern Analysis and Machine Intelligence* **5** (9), 592–598 (1985).
5. J. Hong and C. Mao, Incremental discovery of rules and structure by hierarchical and parallel clustering, *Knowledge Discovery in Database*, (Edited by G. Piatetsky-Shapiro and W.J. Frawley), pp. 177–194, AAAI/MIT Press, (1991).
6. X. Hu, Conceptual clustering and concept hierarchy for knowledge discovery, MS Thesis, School of Computer Science, Simon Fraser University, Canada, (December 1992).