The Third AFSS(1998), 122-127

Integrated Method for Knowledge Discovery in Databases

Hong Chung*, Kyoung Oak Choi**, Hwan Mook Chung**

* Department of Computer Engineering, Keimyung University, Taegu, Korea
**Faculty of Electronics & Information Engineering, Catholic Univ. of Taeguhyosung

ABSTRACT

This paper suggests an integrated method for discovering knowledge from a large database. Our approach applies an attribute-oriented concept hierarchy ascension technique to extract generalized data from actual data in databases, induction of decision trees to measure the value of information, and knowledge reduction of rough set theory to remove dispensable attributes and attribute values. The integrated algorithm first reduces the size of database for the concept generalization, reduces the number of attributes by way of eliminating condition attributes which have little influence on decision attribute, and finally induces simplified decision rules removing the dispensable attribute values by analyzing the dependency relationships among the attributes.

1. Introduction

Recently, many researches are being progress which extract available information from the databases. It is called KDD (Knowledge Discovery in Databases) or data mining[2]. As we can see it from the recent workshops[7,8,12,13] or various papers[1,9,10, 14], the theory about KDD makes researchers be much interested in machine learning, intelligent databases, knowledge acquistion, and so on.

There are many techniques for KDD. Of them, many researches have been discussing attribute-oriented induction suggested by Han and Cercone[3], induction of decision trees introduced by Quinlan[11], and knowledge reduction of rough set introduced by Pawlak[5]. But attribute-oriented induction method does not analyze data dependency relationships among attributes, and so the generated rules may not be concise since they may contain some redundant information or unnecessary constraints[4]. Thus, it is needed to perform comprehensive analysis of properties of the generalized data prior to the generation of rules. Induction of decision trees can be simple through representing the data dependency of the attributes as a tree graph, but it is difficult to apply the method to large databases which have many attributes and tuples. On the other hand, the rough set technique provides the necessary tools to analyzing the set of attributes globally, but it may not be feasible to apply directly to a large database because of its computational complexity, which is NP-hard[4]. Therefore, it is necessary to reduce the number of attributes and tuples in very large databases prior to the knowledge discovery.

In this paper, we suggest an integrated method which mixes attribute-oriented concept hierarchy for generalizing databases, measurement of the value of information used in induction of decision trees, and knowledge reduction method of rough set theory for getting higher level of information from lower

level of data in databases and inducing the most simplified decision rules through eliminating the dispensable attributes.

2. Generalization of Database

A large database usually contains a huge set of distinct attribute values. To obtain a simple and concise scheme and derive decision rules for each class, we should first generalize the primitive data instances to higher level concepts. This is realized through attribute-oriented generalization on the task-relevant relation.

Concept hierarchies are a collection of generalization relations[3]. A generalization relation is defined to be a relation between a set of attribute values and a single value which is more general one. A generalization relation can be represented by $\{A_1,A_2,...,A_k\}\subset B$, in which B is a generalization of each A_i, for $1\le i\le k$. All concept herarchies for a database are obtained from domain experts and stored together in a concept hierarchy table.

Suppose we have a basic database of used-car in Table 1.

Table 1: Used-Car Database

No	Model	Size	Type	Transmitter	Year	Odometer
1	Sonata	Medium	DOHC	Auto	90	156000
2	Lanos	Small	DOHC	Auto	94	100000
3	Concord	Medium	SOHC	Manual	88	256000
4	Grandeur	Large	SOHC	Auto	88	200000
5	Sonata	Medium	SOHC	Manual	89	163000
6	Leganza	Medium	DOHC	Auto	95	111000
7	Credos	Medium	SOHC	Manual	89	180000
8	Avante	Small	DOHC	Auto	92	120000
9	Prince	Medium	SOHC	Auto	90	85000
10	Concord	Medium	SOHC	Manual	90	130000
11	Nubira	Small	DOHC	Auto	94	89000
12	Pride	Small	DOHC	Manual	91	175000
13	Sonata	Medium	DOHC	Auto	90	160000
14	Potentia	Large	SOHC	Manual	90	211000
15	Credos	Medium	SOHC	Auto	93	195000
16	Avante	Small	DOHC	Auto	93	175000

We define the concept hierachy table for the used-car as the followings:

Model:
{Sonata, Grandeur, Avante} ⊂ Hyundai
{Pride, Concord, Potentia, Credos} ⊂ Kia
{Leganza, Lanos, Prince, Nubira} ⊂ Daewoo
Year
{..90} ⊂ Old
{91..93} ⊂ Medium
{94..} ⊂ New
Odometer
{..120000} ⊂ Short
{121001..190000} ⊂ Medium
{191001..} ⊂ Long

The generalization is performed by ascending the concept heirachy(i.e. substituting the values of the attribute in each tuple by its corresponding higher level concept) if there is a higher level concept. Attribute which is key of relation can not be generalized because there is no higher level of concepts in the hierarchies .

In Table 1, *Model* is generalized to car maker and *Year* and *Odometer* are generalized to grade. As *Size*, *Type*, and *Transmitter* have no higher level of concept, they remain. The level of generalization is depends on the concept hierarchies of each application. The first attribute, *No*, which is the key of the relations should be removed because of having no higher level concept.

As a result of generalization of the lower-level databases, the duplicate tuples are made. If we remove the duplicates, the size of database would be reduced. To represent these as a type of knowlege representation as one in Table 2, we substitute numeric symbols for each attribute value in the following for the convenience of expression.

Model = {Kia,Hyundai,Daewoo} = {1,2,3}
Size = {Large,Medium,Small} = {1,2,3}
Type = {SOHC,DOHC} = {1,2}
Transmitter = {Auto,Manual} = {1,2}
Year = {Old,Medium,New} = {1,2,3}
Odometer = {Long,Medium,Short} = {1,2,3}

We regard *Model*, *Size*, *Type*, *Transmitter* and *Year*(representing as a, b, c, d and e) as condition attributes, and *Odometer*(representing as f) as a decision attribute. Hereafter, we use the term example in substitute for tuple.

Table 2

	a	b	c	d	e	f
1	1	2	1	2	1	1
2	2	1	1	1	1	1
3	1	1	1	2	1	1
4	1	2	1	1	2	1
5	2	2	1	2	1	2
6	2	2	2	1	1	2
7	1	2	1	2	1	2
8	1	3	2	2	2	2
9	2	3	2	1	2	2
10	3	3	2	1	3	3
11	3	2	2	1	3	3
12	2	3	2	1	2	3
13	3	2	1	1	1	3

As a result, through the ascension of the concept hierachy, lower level of database in Table 1 is simplified to a set of examples in Table 2. Below are given normal forms[5] of decision rules considered in Table 2.

```
a1b2c1d2e1va2b1c1d1e1va1b1c1d2e1
      va1b2c1d1e2  --> f1
a2b2c1d2e1va2b2c2d1e1va1b2c1d2e1
      va1b3c2d2e2va2b3c2d1e2 --> f2
a3b3c2d1e3va3b2c2d1e3va2b3c2d1e2
      va3b2c1d1e1 --> f3
```

These rules may contain some redundant information and unnecessary constraints since data dependency relationships among attributes are not analyzed in these rules. And it is required to simplify the rules because they are too complex to deal with.

3. Reduction of Condition Attributes

For the reduction of condition values which influence very little on a decision attribute, we use measurement of the value of information. Inductive method of decision trees generates decision rules which distinguish the concepts from the examples given[11]. The concepts which classifies examples are classes, and they are described as some attributes. A set of examples consists of attributes, and each of the attributes has value. The method of rule generation, if a set of examples K is not classified by only one class, selects one attribute and separates K into K_1, K_2, ..., K_n depending on the attribute values. This K_i is a subset of the set K, which the attribute takes *i*th attribute value. At this time, selected attribute forms a root node. In selecting an attribute, it is desired that the height of the tree should be as short as possible, i.e. it is desirable to select an attribute which has the most discrimination in the examples among the attributes of the table. For the measurement of degree of discrimination among the attributes, Quinlan[11] suggested information theoretic measure which determines the complexity or simplicity of information.

Value of information that a set of examples K holds, might be expressed as the following entropy:

$$M(K) = \sum_{i=1}^{m} P_i \log_2(1/P_i) = -\sum_{i=1}^{m} P_i \log_2 P_i$$

where, Pi is the ratio of class Ki in the set of examples K.

In case that a set of examples K having value of information M(K) is separated into sets of lower level of examples by an attribute X, if value of information B(K,X) is smaller than the M(K), the gaining value of information is M(K)-B(K,X), the difference between the value of information of original table and the separated decision trees owing to attribute X.

When an attribute Xj, j=1,...,m has |Xj| classes and the class is Ki, i=1,...,m, the value of information seperated by using attribute Xj, is as following.

$$B(K,Xj) = \sum_{i=1}^{|Xj|} Wi * M(Si)$$

where, M(Si) is value of information of the lower level of set of examples Si if an attribute Xj has *i*th class value, and Wi is weight as following:

Wi = number of examples in Si / number of examples in K

For generating decision trees in the table of examples in Table 2, there are five cases according to which attribute we select for root attribute. Suppose *f* is decision attribute, value of information in the set of examples is calculated as:

$$M(K) = -(4/13)\log_2(4/13)-(5/13)\log_2(5/13) -(4/13)\log_2(4/13) = 1.577$$

In case of partitioning attribute *a* selected by lower level set of examples,

```
          a
---------------
1       2       3
12121 1 21111 1 33213 3
11121 1 22121 2 32213 3
12112 1 22211 2 32111 3
12121 2 23212 2
13222 2 23212 3
```

- Value of information in lower level set with class 1:
  $M(S_1) = -(3/5)\log_2(3/5)-(2/5)\log_2(2/5) = 0.971$
- Value of information in lower level set with class 2:
  $M(S_2) = -(1/5)\log_2(1/5)-(3/5)\log_2(3/5) -(1/5)\log_2(1/5) = 1.371$
- Value of information of lower level set with class 3:
  $M(S_3) = -(3/3)\log_2(3/3) = 0$

Therefore,

- B(K,'a') = 0.971*(5/13)+1.371*(5/13) +0*(3/13) = 0.901
- M(K)-B(K,'a') = 1.577-0.901 = 0.676

Value of information gained of *b*, *c*, *d*, and *e* could be calculated in the same way, which results in 0.431, 0.373, 0.260, and 0.334 respectively. Suppose that the value of information below 0.35 is trivial and could be eliminated, attributes *d* and *e* are candidates for elimination. In other words, we can make sense that two condition attributes, *Transmitter* and *Year* are the least influencing attributes to the decision attribute, *Odometer*. If we remove duplicates examples except one from the table, Table 2 is simplified to be Table 3.

Table 3

	a	b	c	f
1	1	2	1	1
2	2	1	1	1
3	1	1	1	1
4	2	2	1	2
5	2	2	2	2
6	1	2	1	2
7	1	3	2	2
8	2	3	2	2
9	3	3	2	3
10	3	2	2	3
11	2	3	2	3
12	3	2	1	3

Normal forms[5] of decision rules considered in Table 3 are shown below.

```
a1b2c1va2b1c1va1b1c1 --> f1
a2b2c1va2b2c2va1b2c1
    va1b3c2va2b3c2 --> f2
a3b3c2va3b2c2va2b3c2va3b2c1 --> f3
```

The dependency relationships among the attribute values are not analyzed in these rules, so they could contain some redundant information.

4. Elimination of Attribute Values

The rough set theory was introduced by Pawlak[6]. It is used for formal reasoning with uncertain information, machine learning, knowledge discovery, and representation and reasoning about imprecise knowledge.

Let U be the universe of discourse, and let R be an equivalence relation on U. The pair A=(U,R) is called an approximation space. If x,y∈U and (x,y)∈R, x and y are said to be indiscernable in A. Each equivalence class of the relation R is called an elementary set in A. A finite union of elementary sets in A is called a composed definable set or simply composed set in A.

Let X be a subset of U. The least composed set in A containing X is called upper approximation of X in A, denoted by R_UX; the greatest composed set in A contained in X is called lower approximation of X in A, denoted by R_LX:

lower approximation of X in A:
$$R_LX = \{x \in U \mid [x]_R \subseteq X\}$$
upper approximation of X in A:
$$R_UX = \{x \in U \mid [x]_R \cap X \neq \emptyset\}$$

where, for any element x of U, $[x]_R$ represents the equivalence class of x in relation R.

The set R_LX is the set of all elements of U which can be with certainty classified as element of X in the relation R; Set R_UX is the set of elements of U which can be possible classified as elements of X.

We shall also employ the following denotations:

$POS_R(X)=R_LX$, R-positive region of X
$NEG_R(X)=U-R_UX$, R-negative region of X
$BN_R(X)=R_UX-R_LX$, R-boundary region of X

The positive region $POS_R(X)$ or the lower approximation of X is the collection of those examples which can be classified with full certainty as members of the set X using relation R. Similarly, the negative region $NEG_R(X)$ is the collection of examples with which it can be determined without any ambiguity. The boundary region $BN_R(X)$ is, in a sense, the undecidable area of the universe.

In reduction of knowledge the basic role is played, in rough sets theory[5], by two fundamental concepts, a *reduct* and *core*. Intuitively, a reduct of knowledge is its essential part, which suffices to define all basic concepts occuring in the considered knowledge, whereas the core is in a certain sense its most important part. The set of all indispensable attributes in condition attributes C, with respect to decision attribute D, is called core of C, and it is defined as:

$$CORE(C,D) = \{a \in C \mid POS_C(D) \neq POS_{C-\{a\}}(D)\}$$

The following is an important property establishing the relationship between the core and reducts[5].

CORE(C,D) = ∩RED(C,D)

BcC is defined as reduct in knowledge system if B is independent with respect to D and $POS_C(D)=POS_B(D)$, i.e. reduct B is the essential part which discerns the decision rules by the knowledge systems.

Let F={X1,...,Xn}, be a family of sets. We say that Xi is dispensable if ∩(F-{Xi})=∩F; otherwise it is indispensable.

In order to reduce superfluous values of condition attributes, we have first to compute core values of condition attributes in every example. The family of equivalence sets of example 1 in Table 3 is the following.

F = {[1]a,[1]b,[1]c}
= {{1,3,6,7},{1,4,5,6,10,12},{1,2,3,4,6,12}}

In order to find dispensable categories, we have to drop one category at a time and check whether the intersection of remaining categories is still included in the decision category [1]f={1,2,3}, i.e.

[1]b∩[1]c = {1,4,5,6,10,12}∩{1,2,3,4,6,12}
= {1,4,6,12}
[1]a∩[1]c = {1,3,6,7}∩{1,2,3,4,6,12} = {1,3,6}
[1]a∩[1]b = {1,3,6,7}∩{1,4,5,6,10,12} = {1,6}

All are cores because *a*, *b*, *c* are not included in [1]f={1,2,3}. Therefore, the core values are a(1)=1, b(1)=2, c(1)=1. Similarly we can compute remaining core values of condition attributes in every example and the final result is presented in Table 4.

Table 4

	a	b	c	f
1	1	2	1	1
2	-	1	-	1
3	-	1	-	1
4	2	2	-	2
5	2	2	-	2
6	1	2	1	2
7	1	-	-	2
8	2	3	2	2
9	3	-	-	3
10	3	-	-	3
11	2	3	2	3
12	3	-	-	3

Having computed core values of condition attributes, we can compute the reducts of attribute values. We have first to examine whether the core values themselves could be the reduct values.

- Example 1, 6, 8, 11: all attributes are cores, and there is no reduct.
- Example 2: rule [2]b-->[2]f is true, and itself is reduct.
- Example 3: rule [3]b-->[3]f is true, and itself is reduct.
- Example 4: rule [4]a∩[4]b-->[4]f is true, and itself is reduct.
- Example 5: rule [5]a∩[5]b-->[5]f is true, and itself is reduct.

- Example 7: rule [7]a-->[7]f is not true. In order to compute reducts of family F={[7]a,[7]b,[7]c}={{1,3,6,7},{7,8,9,11},{5,7,8,9,10,11}}, we have to find all subfamilies G⊆F such that ∩G⊆[7]f={4,5,6,7,8}. There are three following subfamilies of F,
[7]b∩[7]c={7,8,9,11}∩{5,7,8,9,10,11}={7,8,9,11}
[7]a∩[7]c={1,3,6,7}∩{5,7,8,9,10,11}={7}
[7]a∩[7]b={1,3,6,7}∩{7,8,9,11}={7}
and only two of them,
[s7]a∩[7]c = {7} ⊆ [7]f
[7]a∩[7]b = {7} ⊆ [7]f
are reducts of the family F. Hence we have two value reducts: a(7)=1 and b(7)=3, and a(7)=1 and c(7)=2. This means that the attribute values of attribute *a* and *b* or *a* and *c* are characteristic for class 2.

- Example 9: rule [9]a-->[9]f is true, and itself is reduct.
- Example 10: rule [10]a-->[10]f is true, and itself is reduct.
- Example 12: rule [12]a-->[12]f is true, and itself is reduct.

Here, after duplicate examples are removed, we can list value reducts for all decision rules in Table 5.

Table 5

	a	b	c	f
1	1	2	1	1
2	x	1	x	1
4	2	2	x	2
6	1	2	1	2
7	1	3	x	2
7′	1	x	2	2
8	2	3	2	2
9	3	x	x	3
11	2	3	2	3

As we can see from the table in row 7, there are two reducts of condition attributes: a1b3 and a1c2. Therefore, the representation of Table 5 using decision rule is the following.

decision class 1: a1b2c1vb1 --> f1
decision class 2: a2b2va1b2c1va1b3 --> f2
and a2b2va1b2c1va1c2 --> f2

decision class 3: a3va2b3c2 --> f3

The combined forms of these rules are two sets of decision rules as shown below.

1) a1b2c1vb1 --> f1
   a2b2va1b2c1va1b3 --> f2
   a3va2b3c2 --> f3
2) a1b2c1vb1 --> f1
   a2b2va1b2c1va1c2 --> f2
   a3va2b3c2 --> f3

We represent the first set of the rules to knowledge discovered from used-car database as following:

Model=Kia and Size=Medium
  and Type=SOHC or Size=Large
      --> Odometer=Long
Model=Hyundai and Size=Medium
  or Model=Kia and Size=Medium
  and Type=SOHC or Model=Kia
  and Size=Small
      --> Odometer = Medium
Model=Daewoo or Model=Hyundai
  and Size=Small and Type=DOHC
      --> Odometer = Short

5. Conclusion

In this paper, to discover knowledge from a large database, we first reduce the size of database by making it generalized conceptually, reduce the number of attributes by eliminating condition attributes which have little influence on decision attribute, and induce simplified decision rules using the way of removing the dispensable attribute values by analyzing the dependency relationships among the attributes. The method is the integrated one which mixes and applies the principle features from the various methods of knowledge discovery, i.e. attribute-oriented induction, induction of decision trees, knowledge reduction of rough set theory.

The generated knowledge rules are simpler and more expressive than other methods. And the knowledge induced by concept generalization is expressed as high level abstraction.

References

[1] N. Cercone and M. Tsuchiya, Special Issue on Learning and Discovery in Databases, *IEEE Transactions on Knowledge and Data Engineering* 5, 1993

[2] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus, "Knowledge Discovery in Databases: An Overview", *AI Magazine*, Fall, 1992

[3] J. Han, Y. Cai, and N. Cercone, "Knowledge Discovery in Databases: An Attribute-Oriented Approach," *Proceeding of the 18th Conference on Very Large Data Bases*, Vancouver, Canada, PP. 340-355, 1992

[4] X. Hu, N. Cercone, and J. Han, "An Attribute-Oriented Rough Set Approach for Knowledge in Databases," *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, Alberta, Canada, 12-15 October 1993

[5] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer, 1991

[6] Z. Pawlak, "Rough Sets," *International Journal of Computer and Information Science*, 11, pp.341-356, 1982

[7] G. Piatetsky-Shaprio, Report on AAA-91 Workshop on Knowledge Discovery in Databases, *IEEE Expert*, October, 1991

[8] G. Piatetsky-Shaprio, KDD-93 : *Proceeding of AAA-93 Workshop on Knowledge Discovery in Databases*, AAAI Press 1993

[9] G. Piatetsky-Shaprio and C. Matheus, Knowledge Discovery Workbench for Exploring Business Databases, *Internal J. of Intelligent Systems*, September, 1992

[10] G. Piatetsky-Shaprio, Special Issue on Knowledge Discovery in Databases, *J. of Intelligent Information Systems* 3, December, 1994

[11] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning* 1, pp.81-106, 1986

[12] W. Ziarko, *Proceedings of the Int. Workshop on Rough Sets and Knowledge Discovery*, Banff, Canada, 1993

[13] J, Zytkow, *Proceedings of the Machine Discovery Workshop*, Aberdeen, Scotland, July, 1992,

[14] J. Zytkow, Special Issue on Machine Discovery, *Machine Learning* 12, 1993