BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **11**, No 3

Sofia • 2011

A Method to Construct a Decision Table from a Relation Scheme

Vu Duc Thi, Nguyen Long Giang

Institute of Information Technology, Vietnamese Academy of Science and Technology
Emails: vdthi@ioit.ac.vn nlgiang@ioit.ac.vn

Abstract: *The problem of constructing a data table which satisfies available knowledge, is one of the most important problems in the field of knowledge discovery and data mining. Based on some results in relational database theory, in this paper we introduce an algorithm to construct a decision table which satisfies a given relation scheme. In the worst case, the time complexity of the proposed algorithm is exponential in the number of attributes.*

Keywords: *Data mining, rough set theory, relational database, decision table.*

1. Introduction

In the consistent decision table $DS = (U, C \cup \{d\}, V, f)$, an attribute set B is called a reduct of C if B is a minimal set which satisfies the functional dependency $B \to \{d\}$. In relational databases, if r is a relation over an attribute set R then an attribute set B is called a minimal set of an attribute $a \in R$ if B is a minimal attribute set which satisfies the functional dependency $B \to \{a\}$. Therefore, when the decision table $DS = (U, C \cup \{d\}, V, f)$ can be considered as the relation r over the set of attributes $R = C \cup \{d\}$, the concept of reduct in DS is equivalent to the concept of minimal sets of the attribute $\{d\}$ over r. Given a relation scheme $s =< R, F >$ over an attribute set $R = C \cup \{d\}$, we have to

construct the decision table $DS = (U, C \cup \{d\}, V, f)$ such that the set of all its reductions is equivalent to the family of all minimal sets of the attribute *d* over *s*. This problem can be considered as the problem for constructing a data table which satisfies available functional dependencies. This is one of the most important problems in the field of knowledge discovery and data mining.

In this paper, we introduce an algorithm to construct a decision table from a given relation scheme. The algorithm is constructed based on some results concerning keys, antikeys and minimal sets of an attribute in [1, 2, 9]. In the worst case, it shows that the time complexity of the algorithm is exponential in the number of attributes. The paper is structured as follows. Section 2 presents some basic concepts in relational database and rough set theory. Section 3 proposes some basic algorithms in relational databases. Section 4 comes up with an algorithm to construct a decision table from a given relation scheme.

2. Basic concepts

2.1. Basic concepts of a relational database

In this section we briefly present the main concepts of the theory of relation database which will be needed in sequel. The concepts and facts given in this section can be found in [1-4, 8, 9].

Let $R = \{a_1, ..., a_n\}$ be a finite set of attributes. For each attribute a_i there is a nonempty set $D(a_i)$ of possible values of this attribute. An arbitrary finite subset of the Cartesian product $D(a_1) \times ... \times D(a_n)$ is called a relation over *R*. Clearly, a relation over *R* is a set of mappings $h: R \to \bigcup_{a \in R} D(a)$, where $h(a) \in D(a)$ for all $a \in R$.

Let $r = \{h_1, ..., h_m\}$ be a relation over $R = \{a_1, ..., a_n\}$. A functional dependency (*FD* for short) over *R* is a statement of the form $A \to B$, where $A, B \subseteq R$. *FD* $A \to B$ holds in a relation *r* over *R* if $(\forall h_i, h_j \in r)((\forall a \in A)(h_i(a) = h_j(a)) \Rightarrow (\forall b \in B)(h_i(b) = h_j(b)))$. Let $F_r = \{(A, B): A, B \subseteq R, A \to B\}$, F_r is called a full family of functional dependencies in *r*. Let *R* be a finite set and denote by *P*(*R*) its power set, we say that *F* is an *f*-family over *R* iif for all $A, B, C, D \subseteq R$ *:*

(1) $(A, A) \in F$,

(2) $(A, B) \in F, (B, C) \in F \Rightarrow (A, C) \in F$,

(3) $(A, B) \in F, A \subseteq C, D \subseteq B \Rightarrow (C, D) \in F$,

(4) $(A, B) \in F, (C, D) \in F \Rightarrow (A \cup C, B \cup D) \in F$.

Clearly, F_r is an *f*-family over *R*. It is known that if *F* is an arbitrary *f*-family over *R*, then there is a relation *r* such that $F_r = F$. F^+ is the set of all *FDs* which can be derived from *F* by rules (1)-(4).

A relation schema *s* is a pair $< R, F >$, where *R* is a set of attributes and *F* is a set of *FDs* on *R*. Denote $A^+ = \left\{ a \in R \middle| A \to \{a\} \in F^+ \right\}$, A^+ is called the *closure* of *A* on *s*. It is clear that $A \to B \in F^+$ iif $B \subseteq A^+$. Let *r* be a relation, $s = < R, F >$ be a relation scheme and $A \subseteq R$. Then *A* is a key of *r* (a key of *s*) if $A \to R\left(A \to R \in F^+\right)$. *A* is a minimal key of *r* (*s*), if *A* is a key of *r* (*s*) and any proper subset of *A* is not a key of *r*(*s*). Denote by $K_r\left(K_s\right)$ the set of all minimal keys of *r*(*s*). $K \subseteq P\left(R\right)$ is a Sperner-system if for any $K_1, K_2 \in K$ implies $K_1 \not\subset K_2$. Clearly, $K_r\left(K_s\right)$ are Sperner-systems.

Let K be a Sperner-system over *R* as the set of all minimal keys of *s* . We defined the set of antikeys of *K*, denoted by K^{-1}, as follows:

$$K^{-1} = \left\{ A \subset R : \left(B \in K\right) \Rightarrow \left(B \not\subset A\right) \right\} \text{ and if}$$

$$\left(A \subset C\right) \Rightarrow \left(\exists B \in K\right)\left(B \subseteq C\right).$$

It is easy to see that K^{-1} is the set of subsets of *R*, which does not contain the elements of K and which is maximal for this property. They are the maximal non-keys. Clearly, K^{-1} is also a Sperner-system.

Let *r* be a relation over *R*. Denote $E_r = \left\{ E_{ij} : 1 \le i < j \le |r| \right\}$, where $E_{ij} = \left\{ a \in R : h_i\left(a\right) = h_j\left(a\right) \right\}$. Then E_r is called an equality set of *r*. It is known [4] that for $A_r \subseteq R$, $A_r^+ = \cap E_{ij}$, if there exists $E_{ij} \in E_r : A \subseteq E_{ij}$, otherwise $A_r^+ = R$. In the next content we introduce some definitions about the family of all minimal sets of an attribute over a relation and a relation scheme.

Definition 1 [2]. Let $s = \left(R, F\right)$ be a relation scheme over *R* and $a \in R$. Set $K_a^s = \left\{ A \subseteq R : A \to \{a\}, \nexists B : \left(B \to \{a\}\right)\left(B \subset A\right) \right\}$. K_a^s is called a family of minimal sets of the attribute *a* over *s*.

Similarly, we define the family of minimal sets of an attribute over a relation.

Definition 2. Let *r* be a relation over *R* and $a \in R$. Set $K_a^r = \left\{ A \subseteq R : A \to \{a\}, \nexists B \subseteq R : \left(B \to \{a\}\right)\left(B \subset A\right) \right\}$. K_a^r is called a family of minimal sets of the attribute *a* over *r*.

It is clear that $R \notin K_a^s, R \notin K_a^r, \{a\} \in K_a^s, \{a\} \in K_a^r$ and K_a^s, K_a^r are Sperner systems over *R*.

2.2. Basic concepts of rough set theory

In this section we introduce some basic concepts in rough set theory [5, 6, 7]

A decision table is defined as $DS = \left(U, C \cup \{d\}, V, f\right)$, in which $U = \{u_1, u_2, ..., u_n\}$ is the finite & non-empty set of objects, $C = \{c_1, c_2, ..., c_m\}$ the set of condition attributes, *D* is the set of decision attributes and $C \cap D = \varnothing$, $V = \prod_{a \in C \cup D} V_a$ where V_a is the value range of the attribute *a*, $f : U \times \left(C \cup D\right) \to V$ is an information function, where $\forall a \in C \cup D, u \in U$, $f\left(u, a\right) \in V_a$ hold. Without loss of generality, suppose that *D* consists of the only one decision attribute *d* (in case *D* consists of many attributes then we assign an attribute to *D* by encoding). Therefore, from this time we consider the decision table $DS = \left(U, C \cup \{d\}, V, f\right)$, where $\{d\} \notin C$.

Every attribute subset $P \subseteq C \cup D$ determines an indiscernibility relation

$$IND\left(P\right) = \left\{\left(u, v\right) \in U \times U \,\middle|\, \forall a \in P, f\left(u, a\right) = f\left(v, a\right)\right\}.$$

IND(P) determines a partition of *U* which is denoted by U / P. Any element $\left[u\right]_P = \left\{v \in U \,\middle|\, \left(u, v\right) \in IND\left(P\right)\right\}$ in U / P is called an equivalent class.

With $B \subseteq C$ and $X \subseteq U$, *B-upper approximation* of X is the set $\overline{B}X = \left\{u \in U \,\middle|\, \left[u\right]_B \cap X \neq \varnothing\right\}$, and *B-lower approximation* of *X* is the set $\underline{B}X = \left\{u \in U \,\middle|\, \left[u\right]_B \subseteq X\right\}$, and *B-boundary* of X is the set $BN_B\left(X\right) = \overline{B}X \setminus \underline{B}X$ and *B-positive region* of *D* is the set $POS_B\left(D\right) = \bigcup_{X \in U/D} \left(\underline{B}X\right)$. A decision table *DS* is consistent iif $POS_C(D) = U$, in other words the functional dependency $C \to \{d\}$ is true. Conversely, *DS* is an inconsistent decision table and then $POS_C\left(\{d\}\right)$ is the maximum subset of *U* that the functional dependency $C \to \{d\}$ is true**.**

In rough set theory, P a w l a k [5] proposes the definition of a reduct, called the reduct based on a positive region.

Definition 3. Let $DS = \left(U, C \cup \{d\}, V, f\right)$ be a decision table. If $B \subseteq C$ satisfies

1) $POS_B(D) = POS_C(D)$,

2) $\forall b \in B, POS_{B-\{b\}}(D) \neq POS_C(D)$ then *B* is called a reduct of *C*.

If *DS* is a consistent decision table, *B* is an attribute reduction of *C* if *B* satisfies $B \to \{d\}$ and $\forall B' \subset B, B' \not\to \{d\}$. Let $RED\left(C\right)$ be the set of all

reducts of *C*. From Definition 2 and 3 we have $RED(C) = K_d^r - \{d\}$ where K_d^r is the family of all minimal sets of the attribute $\{d\}$ over $r = <U, C \cup \{d\}>$.

3. Basic algorithms in relational database

Let us give some basic algorithms in relational database that are used in the next section.

3.1. Algorithm for finding the set of antikeys

Algorithm 1 [1, 9]. Finding the set of antikeys K^{-1} from a given Sperner-system.

Input: Let $K = \{B_1, ..., B_m\}$ be a Sperner-system over *R*.

Output: K^{-1}.

Step 1. We set $K_1 = \{R - \{a\} : a \in B_1\}$. It is obvious that $K_1 = \{B_1\}^{-1}$.

Step q+1. (q<m). Assume that $K_q = F_q \cup \{X_1, ..., X_{tq}\}$, where $X_1, ..., X_{tq}$ are elements of K containing B_{q+1} and $F_q = \{A \in K_q : B_{q+1} \not\subseteq A\}$. For all $i\,(i = 1, ..., t_q)$ we construct the antikeys of $\{B_{q+1}\}$ on X_i in an analogous way as K_1, which are the maximal subsets of X_i not containing B_{q+1}. We denote them by $A_1^i, ..., A_{ri}^i$. Let

$$K_{q+1} = F_q \cup \{A_p^i : A \in F_q \Rightarrow A_p^i \not\subset A, 1 \le i \le t_q, 1 \le p \le r_i\}.$$

Finally, let $K^{-1} = K_m$.

Clearly, because K and K^{-1} are uniquely determined by one other, the determination of K^{-1}, based on our algorithm does not depend on the order of $B_1, ..., B_m$.

Computational complexity analysis of Algorithm 1

Denote by I_q the number of elements of K_q. According to [1, 9], the time complexity of the algorithm in the worst case is $O\left(|R|^2 \sum_{q=1}^{m-1} t_q u_q\right)$ where $u_q = I_q - t_q$ if $I_q > t_q$ and $u_q = 1$ if $I_q = t_q$.

R e m a r k s

Remark 1. In each step of the algorithm, K_q is obviously a Sperner-system. It is known [1, 9] that the size of an arbitrary Sperner-system over *R* can not be greater than $C_n^{[n/2]} \approx 2^{n+1/2} / \left(\prod .n^{1/2}\right)$ where $n = |R|$. Consequently, the worst-case time of the algorithm can not be more than exponential in the number of attributes.

Remark 2. In cases for which $I_q \le I_m \left(\forall q : 1 \le q \le m-1\right)$, the time complexity of the algorithm is not greater than $O\left(|R|^2 |K|\left|K^{-1}\right|^2\right)$. Thus, in these cases the algorithm finds K^{-1} in polynomial time in $|R|, |K|$ and $|K|^{-1}$. Especially, when $|K|, |K|^{-1}$ is small, this algorithm is effective.

3.2. Algorithm for finding the family of all minimal sets of an attribute over a relation scheme.

Algorithm 2 [2]. Finding a minimal set of the attribute *a*.
Input: Let $s = \left(R = \{a_1, ..., a_n\}, F\right)$ be a relational scheme, $a = \{a_1\}$.
Output: $A \in K_a^s$.

Step 1. We set $L(0) = R$.
Step i+1. $L(i+1) = L(i) - a_{i+1}$ if $L(i) - a_{i+1} \to \{a\}$,
$L(i+1) = L(i)$ otherwise.
Then $A = L(n)$.
According to [2], the computational complexity of Algorithm 2 is $O\left(|R|^2 |F|\right)$.

Algorithm 3 [2]. Finding a family of all minimal sets of the attribute *a*.
Input: Let $s = (R, F)$ be a relational scheme and $a \in R$.
Output: K_a^s.

Step 1. Set $L(1) = E_1 = \{a\}$.
Step i+1. If there are *C* and $A \to B$ such that $C \in L(i), A \to B \in F, \forall E \in L(i) \Rightarrow E \not\subseteq A \cup (C - B)$, then by Algorithm 2 construct an E_{i+1}, where $E_{i+1} \subseteq A \cup (C - B), E_{i+1} \in K_a^s$. We set $L(i+1) = L(i) \cup E_{i+1}$. In the converse case we set $K_a^s = L(i)$.

It is known [2] that the worst-case time complexity of this algorithm is $O\left(|R||F||K_a^s|\left(|R|+|K_a^s|\right)\right)$. Thus, the time complexity of this algorithm is polynomial in $|R|,|F|$ and $|K_a^s|$. According to Algorithm 3.1, the worst-case time of the algorithm cannot be more than exponential in the number of attributes. Clearly, if the number of elements of K_a^s for a relational scheme $s=(R,F)$ is polynomial in the size of *s*, then this algorithm is effective. Especially when $|K_a^s|$ is small.

4. Algorithm for constructing a decision table from a relation scheme

Problem. Given a relational scheme $s =< R, F >$ where $R = C \cup \{d\}$ and *F* is the set of functional dependencies over *R*. We have to construct the decision table $DS = (U, C \cup \{d\}, V, f)$ such that $RED(C) = K_d^s - \{d\}$ where K_d^s is the family of all minimal sets of the attribute *d* over *s* and $RED(C)$ is the set of all reducts of *C* in *DS*.

Algorithm 4. Constructing a decision table from a relation scheme.

Input: Let $s =< R, F >$ be a relation scheme, where $R = C \cup \{d\}$ and *F* is the set of functional dependencies over *R*.

Output: The decision table $DS = (U, C \cup \{d\}, V, f)$ such that $RED(C) = K_d^s - \{d\}$.

Step 1. From $s =< R, F >$, using Algorithm 3 we calculate K_d^s.

Step 2. From K_d^s, using Algorithm 1 we calculate $M_d = (K_d^s)^{-1}$. We assume that $M_d = \{A_1, A_2, ..., A_t\}$.

Step 3. We construct the decision $DS = (U, C \cup \{d\}, V, f)$ where $U = \{u_0, u_1, ..., u_t\}$, as follows:

- For all $c \in C$, we set $u_0(c) = 0$. Set $u_0(d) = 0$.
- For all $i\ (i = 1, ..., t)$, we set $u_i(c) = 0$ if $c \in A_i$; $u_i(c) = i$ otherwise. Set $u_i(d) = i$ for all $i\ (i = 1, ..., t)$.

In the next content, we prove $RED(C) = K_d^s - \{d\}$.

P r o o f: According to the method to construct the relation *r* we have $E_{1i} = A_{i-1}$ where $2 \le i \le t+1$ and $E_{ij} = A_{1i} \cap A_{1j}$ where $2 \le i < j \le t+1$, so

$E_{ij} = E_{1i} \cap E_{1j}$ or $E_{ij} \subset E_{1i}, E_{ij} \subset E_{1j}$ where $2 \le i < j \le t+1$. Therefore, the set $M = \{E_{1i} : 1 \le i \le t+1\}$ has the property $\{\forall A \in M \Rightarrow \nexists B \in M : A \subset B\}$. According to the definition of a maximal equality system M_r over *r*, we have $M_r = \{E_{1i} : 2 \le i \le t+1\}$. Hence

(5) $$M_r = M_d = \left(K_d^s\right)^{-1} = \{A_1, A_2, ..., A_t\}.$$

In the next content, we prove $M_r = \left(K_d^r\right)^{-1}$ where K_d^r is the family of all minimal sets of the attribute *d* over *r*.

1) For $A \in M_r$ we have $A^+ = A$, and *A* does not contain *d* so A^+ does not contain *d*, hence $A \to \{d\} \notin F^+$. Moreover, if there is a *B* such that $A \subset B$, according to the method to calculate the closure of an attribute set over a relation we have $B^+ = R$ and B^+ contains *d*, or $B \to \{d\} \in F^+$. According to the results in [2],

$$\left(K_d^r\right)^{-1} = \text{MAX}\left(F_r^+, d\right)$$

where

$$\text{MAX}\left(F_r^+, d\right) = \left\{A \subseteq R : A \to \{d\} \notin F^+, A \subset B \Rightarrow B \to \{d\} \in F^+\right\},$$

so we conclude $A \in \left(K_d^r\right)^{-1}$.

2) Conversely, if $A \in \left(K_d^r\right)^{-1}$ then obviously $A \ne R$. If there is a *B* such that $A \subset B$ and $A \to B$, then by the definition of antikeys we have $B \to \{d\}$ and $A \to \{d\}$. This is a contradiction. So there does not exist a *B* such that $A \subset B$ and $A \to B$, that is, $A^+ = A$ holds. Moreover, according to the definition of antikeys too, if there exists $B' \ne R$ such that $A \subset B'$, then $B' \to \{d\}$ or $\{d\} \subset B'^+$. Therefore, *A* is the maximal set which satisfies $A = A^+$ and *A* does not contain *d* (*i*)**.** On the other hand, over the relation *r* constructed for any $B \in M_r$ we have $B \ne R$, $B = B^+$ and *B* does not contain *d.* If there is a *D* such that $B \subset D$ then $D^+ = R$ or $\{d\} \subset D^+$. Therefore, M_r is the set of all maximal sets *B* which satisfies $B = B^+$ and *B* does not contain *d* (*ii*)**.** From 1) and 2) we can conclude $A \in M_r$.

From 1) and 2) we obtain

(6) $$M_r = \left(K_d^r\right)^{-1}.$$

From (5) and (6) we have $\left(K_d^r\right)^{-1} = M_d = \left(K_d^s\right)^{-1}$, or $K_d^r = K_d^s$. From the results of Definition 3 we have $RED(C) = K_d^s - \{d\}$.

Computational complexity analysis of Algorithm 4:

It is easy to see that the time complexity of *Step 1* computing K_d^s is the time complexity of Algorithm 3. The time complexity of *Step 2* computing $M_d = \left(K_d^s\right)^{-1}$ is the time complexity of Algorithm 1. Consequently, the worst-case time of the algorithm cannot be more than exponential in the number of attributes.

Example 1. Let $s =< R, F >$ be a relational scheme, where $R = \{a, b, c, d\}, C = \{a, b, c\}$ and the set of functional dependencies $F = \{\{a, c\} \to R, \{a\} \to \{a, b, d\}, \{b, c\} \to \{b, c, d\}\}$.

Using Algorithm 3, we compute $K_d^s = \{\{a\}, \{d\}, \{b, c\}\}$.

Using Algorithm 1, we compute $M_d = \left(K_d^s\right)^{-1} = \{\{b\}, \{c\}\}$.

As a result, the consistent decision table *DS* is constructed in Table 1.

Table 1

U	*A*	*b*	*c*	*d*
u_0	0	0	0	0
u_1	1	0	1	1
u_2	2	2	0	2

5. Conclusion

Based on some results concerning keys, antikeys and minimal sets of an attribute of J. Demetrovics and Vu Duc Thi in [1, 2, 9], we propose an algorithm to construct a decision table $DS = (U, C \cup \{d\}, V, f)$ from a given relation scheme $s =< R, F >$ where $R = C \cup \{d\}$. We prove that the set of all reducts in the obtained decision table is equivalent to the family of all minimal sets of the attribute $\{d\}$. In other words, the functional dependencies $B_i \to \{d\}$ must be satisfied over *s* where B_i is an attribute reduct of *C.* In the worst case the time complexity of the algorithm is exponential in the number of conditional attributes. The problem of constructing a decision table from a given relation scheme can be considered as a problem of constructing a data table which satisfies available knowledge. This is one of the important problems in the field of knowledge discovery and data mining.

Acknowledgments: This work was funded by the Vietnam's National Foundation for Science and Technology Development (NAFOSTED) via a research grant for fundamental sciences, Grant No 102.01-2010.09

R e f e r e n c e s

1. D e m e t r o v i c s, J., V. D. T h i. Relations and Minimal Keys. – Acta Cybernetica, Vol. **8**, 1988, No 3, 279-285.
2. D e m e t r o v i c s, J., V. D. T h i. Some Remarks on Generating Armstrong and Inferring Functional Dependencies Relation. – Acta Cybernetica, **12**, 1995, 167-180.
3. D e m e t r o v i c s, J., V. D. T h i. Describing Candidate Keys by Hyper-Graphs. – Computers and Artificial Intelligence, Vol. **18**, 1999, No 2, 191-207.
4. D e m e t r o v i c s, J., V. D. T h i. Some Computational Problems Related to Boyce-Codd Normal Form. – Annales Univ. Sci. Budapest. Sect. Comp., **19**, 2000, 119-132.
5. P a w l a k, Z. Rough Sets – Theoretical Aspects of Reasoning about Data. Dordrecht, Kluwer Academic Publishers, 1991.
6. P a w l a k, Z. Rough Set Theory and its Applications in Data Analysis. – Cybernetics and Systems, **29**, 1998, 661-688.
7. P a w l a k, Z. Rough Sets and Intelligent Data Analysis. – Information Sciences, Vol. **147**, 2002, Issues 1-4, 1-12.
8. T h i, V. D., N. H. S o n. Some Problems Related to Keys and the Boyce-Codd Normal Form. – Acta Cybernetica, **16**, 2004, 473-483.
9. T h i, V. D., N. H. S o n. On Armstrong Relations for Strong Dependencies. – Acta Cybernetica, **17**, 2006, No 3, 521-531.