Handling Queries in Incomplete CKBS through Knowledge Discovery

Zbigniew W. Raś

University of North Carolina, Dept. of Comp. Science, Charlotte, N.C. 28223, USA
Polish Academy of Sciences, Dept. of Comp. Science, 01-237 Warsaw, Poland
email: ras@uncc.edu or ras@wars.ipipan.waw.pl

Abstract. In this paper, we propose a new query answering system for an incomplete Cooperative Knowledge-Based System (CKBS). CKBS is a collection of autonomous knowledge-based systems called agents which are capable of interacting with each other. In the first step of the query processing strategy, the contacted site of CKBS will identify all locally incomplete attributes used in a query. An attribute is locally incomplete if there is an object in a local information system with an incomplete information on this attribute. The values of all locally incomplete attributes are treated as concepts to be learned at other sites of CKBS (see [6]). Rules discovered at all these sites are sent to the site contacted by the user and used locally by the query answering system to replace an incomplete information by values provided by the rules.

In the second step of the query processing strategy, an incomplete information is removed from the local information system in a maximal number of places. Next, the query answering system finds the answer to a user query in a usual way (similar to CKBS query answering system).

Key Words: incomplete information system, cooperative query answering, rough sets, multi-agent system, knowledge discovery.

1 Introduction

By a cooperative knowledge-based system ($CKBS$) we mean a collection of autonomous knowledge-based systems called agents (sites) which are capable of interacting with each other. Each agent is represented by an information system (either complete or incomplete) and a collection of rules called a knowledge base. Any site of $CKBS$ can be a source of a local or a global query. By a local query for a site i (or i-reachable query) we mean a query entirely built from attributes which are complete and local at site i. Local queries need only to access an information system of the site where they were issued and they are completely processed on the system associated with that site. In order to resolve a global query for a site i (built from attributes not necessarily complete or local at site i) successfully, we have to access an information system at more than one site of $CKBS$ and discover rules describing attributes (used in a query) which are either not complete or not local at the site i. Rules discovered by neighbors of i are sent to the site i and used locally by the query answering system to replace some of the incomplete vales in a local information system by values provided by the rules. After the process of removing as many incomplete vales as possible in the information system of site i, the query answering system finds the answer to a user query in a usual way (similarly to CKBS query answering system).

There is a number of strategies which allow us to find rules describing decision attributes in terms of classification attributes. We should mention here such systems like $LERS$ (developed by J. Grzymala-Busse), $DQuest$ (developed by W. Ziarko), $AQ15$ (developed by R. Michalski) or rules discovery system based on discriminant functions proposed by A. Skowron (see [8]). Most of these strategies have been developed under the assumption that the database part of KBS is complete. Problem of inducing rules from attributes with incomplete values was discussed in ([2], [3], [4]). Our strategy shows how to compute such rules with certainty factors not necessarily equal to 1 and next how to use them to make local information system more complete. The Chase algorithm presented for instance in [1] is using dependencies to make a database more complete. We use rules learned at remote sites to achieve a similar goal.

2 Basic definitions

In this section, we introduce the notion of an information system, distributed information system, a knowledge base, and $s(i)$-queries which can be processed locally at site i.

By an information system ([5], [4]) we mean a structure $S = (X, A, V, f)$, where X is a finite set of objects, A is a finite set of attributes (or properties), V is the set-theoretical union of domains of attributes from A, and f is a classification function which describes objects in terms of their attribute values. We assume that:

- $V = \bigcup\{V_a : a \in A\}$ is finite,
- $V_a \cap V_b = \emptyset$ for any $a, b \in A$ such that $a \neq b$,
- $f : X \times A \longrightarrow 2^V$ where $f(x, a) \in 2^{V_a} - \{\emptyset\}$ for any $x \in X$, $a \in A$.

If $f(x, a) = V_a$, then the value of the attribute a for the object x is unknown. We will call system S incomplete if there is $a \in A$, $x \in X$ such that $card(f(x, a)) \geq 2$. Also, if $card(f(x, a)) \geq 2$, then the attribute a is called incomplete. Otherwise system S as well as the attribute a are called complete. The set of all incomplete attributes in S we denote by $In(A)$ and the set $\bigcup\{V_a : a \in In(A)\}$ by $In(V)$. For simplicity reason any complete or incomplete information system will be called, in this paper, an information system.

Let $S_1 = (X_1, A_1, V_1, f_1)$, $S_2 = (X_2, A_2, V_2, f_2)$ be information systems.

- S_2, S_1 are consistent if $f_1(x, a) \subseteq f_2(x, a)$ or $f_2(x, a) \subseteq f_1(x, a)$ for any $a \in A_1 \cap A_2$, $x \in X_1 \cap X_2$.

By a distributed information system [8] we mean a pair $DS = (\{S_i\}_{i \in I}, L)$ where:

- $S_i = (X_i, A_i, V_i, f_i)$ is an information system for any $i \in I$,
- L is a symmetric, binary relation on the set I,
- I is a set of sites.

System DS is called incomplete, if $(\exists i \in I)[S_i$ is *incomplete*].

Systems S_i, S_j (sites i, j) are called neighbors in DS if $(i,j) \in L$. The transitive closure of L in I is denoted by $L^\star$.

A distributed information system $DS = (\{S_i\}_{i \in I}, L)$ is consistent if:
$(\forall i)(\forall j)(\forall x \in X_i \cap X_j)(\forall a \in A_i \cap A_j)$
$[(x,a) \in Dom(f_i) \cap Dom(f_j) \longrightarrow f_i(x,a) \subseteq f_j(x,a)$ or $f_j(x,a) \subseteq f_i(x,a)]$.

By a set of $s(i)$-terms we mean a least set T_i such that:

- $\mathbf{0}, \mathbf{1} \in T_i$,
- $(a,w) \in T_i$ for any $a \in A_i$ and $w \in V_{ia}$,
- if $t_1, t_2 \in T_i$, then $(t_1 + t_2), (t_1 \star t_2), \sim t_1 \in T_i$.

We say that:

- $s(i)$-term t is *atomic* if it is of the form (a,w) or $\sim (a,w)$ where $a \in B_i \subseteq A_i$ and $w \in V_{ia}$
- $s(i)$-term t is *positive* if it is of the form $\prod\{(a,w) : a \in B_i \subseteq A_i$ and $w \in V_{ia}\}$
- $s(i)$-term t is *primitive* if it is of the form $\prod\{t_j : t_j$ is atomic $\}$
- $s(i)$-term is in *disjunctive normal form* (DNF) if $t = \sum\{t_j : j \in J\}$ where each t_j is primitive.

By a query for a site i ($s(i)$-query) we mean any element in T_i which is in DNF.

Before we give the interpretation of $s(i)$-queries, we introduce the notion of X-algebra. So, let us assume that X is a set of objects. By an X-algebra we mean a sequence $(\mathbf{P}, \bigoplus, \bigotimes, \neg)$ where:

- $\mathbf{P} = \{P_i : i \in J\}$ where $P_i = \{(x, p_{<x,i>}) : p_{<x,i>} \in [0,1]$ & $x \in X\}$,
- $P_i \bigotimes P_j = \{(x, p_{<x,i>} \cdot p_{<x,j>}) : x \in X\}$,
- $P_i \bigoplus P_j = \{(x, max(p_{<x,i>}, p_{<x,j>})) : x \in X\}$,
- $\neg P_i = \{(x, 1 - p_{<x,i>}) : x \in X\}$,
- $\mathbf{P}$ is closed under the above three operations.

Theorem 1. Let $P_i, P_j, P_k \in \mathbf{P}$. Then:

- $(P_i \bigotimes P_j) \bigotimes P_k = P_i \bigotimes (P_j \bigotimes P_k)$,
- $(P_i \bigoplus P_j) \bigoplus P_k = P_i \bigoplus (P_j \bigoplus P_k)$,
- $(P_i \bigoplus P_j) \bigotimes P_k = (P_i \bigotimes P_k) \bigoplus (Pj \bigotimes P_k)$,
- $P_i \bigotimes P_j = P_j \bigotimes P_i$,
- $P_i \bigoplus P_j = P_j \bigoplus P_i$,

- $P_i \bigoplus P_i = P_i$.

Let $DS = (\{S_j\}_{j \in I}, L)$ be a distributed information system where $S_j = (X_j, A_j, V_j, f_j)$ and $V_j = \bigcup\{V_{ja} : a \in A_j\}$, for any $j \in I$. By a standard interpretation of $s(i)$-queries in DS we mean a partial function M_i, from the set of $s(i)$-queries into X_i-algebra, defined as follows:

- $Dom(M_i) \subseteq \mathbf{T}_i$,
- $M_i((a,w)) = \{(x,p) : x \in X_i \ \& \ w \in f_i(x,a) \ \& \ p = 1/card(f_i(x,a))\}$ for any $w \in V_i$,
- $M_i(\sim (a,w)) = \neg M_i((a,w))$
- for any atomic term $t_1(a) \in \{(a,w), \sim (a,w)\}$ and any primitive term $t = \prod\{s(b) : (s(b) = (b, w_b) \text{ or } s(b) = \sim (b, w_b)) \ \& \ (b \in B_i \subset A_i) \ \& \ (w_b \in V_{ib})\}$ we have

$$M_i(t \star t_1(a)) = M_i(t) \bigotimes M_i(t_1) \text{ if } a \notin B_i$$
$$M_i(t \star t_1(a)) = \emptyset \text{ if } a \in B_i \text{ and } t_1(a) \neq s(a),$$
$$M_i(t \star t_1(a)) = M_i(t) \text{ if } a \in B_i \text{ and } t_1(a) = s(a).$$

- for any $s(i)$-terms t_1, t_2

$$M_i(t_1 + t_2) = M_i(t_1) \bigoplus M_i(t_2).$$

By (k,i)-rule in $DS = (\{S_j\}_{j \in I}, L)$, $k, i \in I$, we mean a pair (t,c) such that:

- either $c \in In(V_i) \cap V_k$ or $c \in V_k - V_i$,
- t is a positive $s(k)$-term which belongs to $\mathbf{T}_k \cap \mathbf{T}_i$,
- if $(x, p1) \in M_k(t)$ then $(\exists p2)[(x, p2) \in M_k(c)]$.

An object x satisfies a rule $r = (t,c)$ with a certainty p at site k, if $p = p1 \cdot p2$, $(x, p1) \in M_k(t)$, and $(x, p2) \in M_k(c)$.

We say that (k,i)-rule (t,c) is in k-optimal form if there is no other subterm $t1 \in \mathbf{T}_k \cap \mathbf{T}_i$ of $s(k)$-term t, such that: if x satisfies rule (t,c) with certainty p, then x satisfies rule $(t1, c)$ with the same or higher certainty.

Let $X = \{x_i : 1 \leq i \leq n\}$ and x_i satisfies the rule $r = (t,c)$ with a certainty p_i at site k for any $i \in \{1, 2, ..., n\}$. We say that r has certainty p, if
$p = [\Sigma\{p_i : p_i \neq 0 \ \& \ 1 \leq i \leq n\}]/[card\{i : p_i \neq 0 \ \& \ 1 \leq i \leq n\}]$.

By a knowledge base D_{ki} we mean any set of (k,i)-rules satisfying the condition below:

$$\text{if } (t,c) \in D_{ki} \text{ then } (\exists t_1)(t_1, \sim c) \in D_{ki}.$$

We say that a knowledge base D_{ki} is in k-optimal form if all its rules are in k-optimal form.

In [6] we proposed an algorithm to construct a knowledge base D_{ki} in k-optimal form. Let us assume that $L(D_{ki}) = \{(t,c) \in D_{ki} : c \in In(V_i)\}$. The algorithm, given below, converts system S_i in DS to a new more complete information system $Chase(S_i)$.

```
Algorithm $Chase(S_i, In(A_i), L(D_{ki})$;
    Input system $S_i = (X_i, A_i, V_i, f_i)$, set of incomplete attributes
    $In(A_i) = \{a_1, a_2, ..., a_k\}$, and a set of rules $L(D_{ki})$
    Output a system $Chase(S_i)$.
    begin
    $j := 1$;
    while $j \leq k$ do
            for all $c \in V_{a_j}$ do
                    while there is $x \in X_i$ and a rule $(t, c) \in L(D_{ki})$
                    such that $x \in M_i(t)$ and $card(f_i(x, a_j)) \neq 1$ do
                    $f_i(x, a_j) := c$ ;
            $j := j + 1$
    $Chase(S_i) := S_i$
    end
```

By a standard chase-interpretation $\hat{M}_i$ of $s(i)$-queries in a distributed system DS, we mean the standard interpretation M_i of $s(i)$-queries in a distributed information system $Chase_i(DS) = (\{\hat{S}_j\}_{j \in I}, L)$, where:

- $\hat{S}_j = S_j$ if $j \neq i$,
- $\hat{S}_j = Chase(S_j)$ if $j = i$.

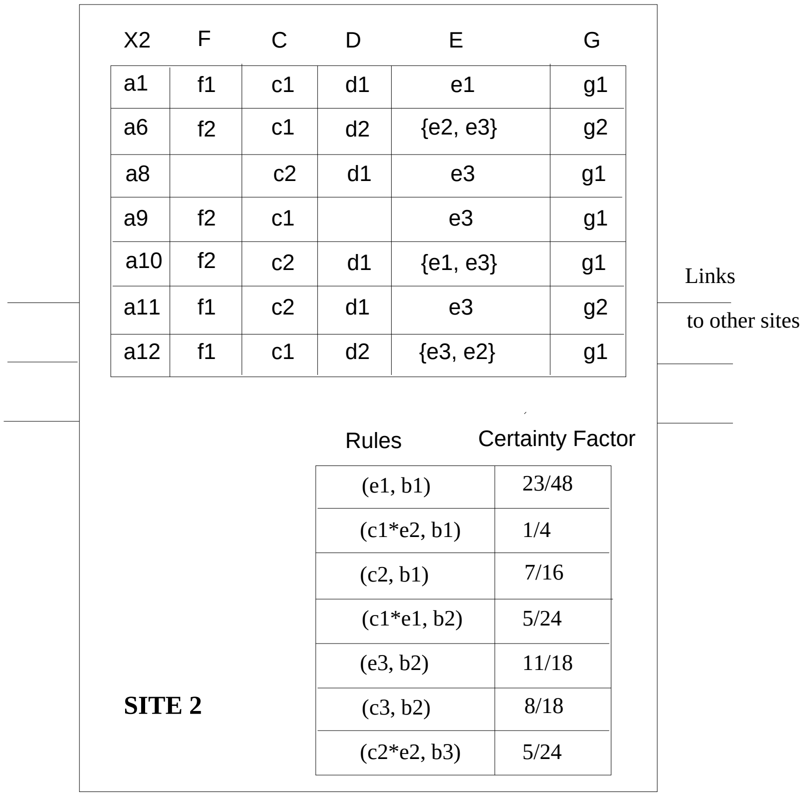
3 Cooperative knowledge-based system

In this section, we define a Cooperative Knowledge Based System ($CKBS$) and introduce the notion of its consistency. We also give an example of CKBS.

Let $\{D_{ki}\}_{k \in K_i}$, $K_i \subseteq I$, be a collection of knowledge bases where D_{ki} was created at site $k \in I$ for any $k \in K_i$ and $D_i = \bigcup\{D_{ki} : k \in K_i\} \cup R_i$. By R_i we mean a set of rules (t, c) created by an expert and stored at site i. Additionally, we assume here that t is an $s(i)$-term. System $(\{(S_i, D_i)\}_{i \in I}, L)$, introduced in ([6], [7]), is called a cooperative knowledge-based system ($CKBS$).

Rules $(t1, w1) \in D_{ki}$, $(t2, w2) \in D_{ni}$ are consistent at Site i if $At(w1) \neq At(w2)$ or $w1 = w2$ or $M_i(t1 \star t2) = \emptyset$. Otherwise, we call them possibly inconsistent. We say that the knowledge base D_i is consistent at Site i if any two rules in D_i are consistent at Site i. Similarly, we say that the cooperative knowledge based system $DS = (\{(S_i, D_i)\}_{i \in I}, L)$ is consistent if D_i is consistent at Site i for any $i \in I$.

Figure 1 gives an example of $CKBS$. Rules in the knowledge base of Site 2 have been computed at another site of $CKBS$. It can be easily checked that these rules are consistent at Site 2.



X2	F	C	D	E	G
a1	f1	c1	d1	e1	g1
a6	f2	c1	d2	{e2, e3}	g2
a8		c2	d1	e3	g1
a9	f2	c1		e3	g1
a10	f2	c2	d1	{e1, e3}	g1
a11	f1	c2	d1	e3	g2
a12	f1	c1	d2	{e3, e2}	g1

Rules	Certainty Factor
(e1, b1)	23/48
(c1*e2, b1)	1/4
(c2, b1)	7/16
(c1*e1, b2)	5/24
(e3, b2)	11/18
(c3, b2)	8/18
(c2*e2, b3)	5/24

Fig. 1. Site 2 of CKBS

4 Query Language and Its Interpretation.

In this section we introduce a query language and propose its optimistic interpretation in a $Site(i)$ of $CKBS$. A formal system for handling queries in $CKBS$ will be presented in a separate paper.

Standard chase-interpretation $\hat{M}_i$, introduced in Section 2, shows how to interpret $s(i)$-queries in a $Site(i)$ of $CKBS$. The question of interpreting DNF queries built from values of attributes belonging to a superset of V_i in $Site(i)$ remains open. Such queries are called global for a Site i. Their standard interpretation at Site i of a cooperative knowledge based system $(\{(S_j, D_j)\}_{j\in I}, L)$, where $D_j = \bigcup\{D_{nj} : n \in K_j\}$, $S_i = (X_i, A_i, V_i, f_i)$ is proposed. To simplify our notation, we write S instead of S_i ,we write w instead of and atomic term (a, w) and assume that $V = V_i = \bigcup\{V_{ia} : a \in A_i\}$ and $C_S = \bigcup\{V_j : j \in I\} - V$. Elements in C_S are called concepts at site i.

By a query language $L(S, C_S)$ we mean a sequence (A, T, F), where A is an alphabet, T is a set of DNF terms (queries), and F is a set of atomic formulas.

The alphabet A of $L(S, C_S)$ contains:

- constants: w where $w \in V_i \cup C_S$
- constants: $\mathbf{0}, \mathbf{1}$
- functors: $+$, $\star$, $\sim$
- predicate: $=$
- auxiliary symbols: (,).

The set of terms T is a least set such that:

- constants $\mathbf{0}, \mathbf{1}$ are terms,
- if w is a constant, then $w, \sim w$ are terms,
- if t_1, t_2 are terms, then $t_1 \star t_2$ is a term.

The set of DNF terms is a least set such that:

- if t is a term, then t is a DNF term,
- if t_1, t_2 are DNF terms, then $t_1 + t_2$ is a DNF term.

Parentheses are used, if necessary, in the obvious way. As will turn out later, the order of a sum or product is immaterial. So, we will abbreviate finite sums and products as $\sum\{t_j : j \in J\}$ and $\prod\{t_j : j \in J\}$, respectively.

The set of atomic formulas F is a least set such that:

- if t_1, t_2 are DNF terms, then $(t_1 = t_2)$ is an atomic formula.

Let $\hat{M}_i$ be a standard chase-interpretation of local $s(i)$-queries in $DS = (\{S_j)\}_{j \in I}, L)$. By a standard interpretation of DNF queries and atomic formulas from $L(S, C_S)$ in S-consistent cooperative knowledge based system $(\{S_j, \{D_{kj}\}_{k \in K_j}\}_{j \in I}, L)$, where $S = (X_i, A_i, V_i, f_i)$ and $V_i = \bigcup\{V_{ia} : a \in A_i\}$, we mean a partial function N_i from the set of DNF queries into X_i-algebra $(\mathbf{P}, \bigoplus, \bigotimes, \neg)$ such that:

(1) for any $w \in V_{ia}$,
$N_i(w) = \hat{M}_i(a, w)$, $N_i(\sim w) = \neg N_i(w)$

(2) if $w \in C_S$,
$N_i(w) = max(\{(x,p) : x \in X_i \;\&\; (\exists n \in K_i)(\exists p > 0)(\exists t)[(t, w) \in D_{ni} \;\&\; (x,p) \in \hat{M}_i(t)]\})$,
$N_i(\sim w) = max(\{(x,p) : x \in X_i \;\&\; (\exists n \in K_i)(\exists p > 0)(\exists t)[(t, \sim w) \in D_{ni} \;\&\; (x,p) \in \hat{M}_i(t)]\})$ where $(x,p) \in max(D)$ iff $\sim (\exists q > p)((x,p) \in D \;\&\; (x,q) \in D)$

(3) $N_i(\mathbf{0}) = N_i(\sim \mathbf{1}) = \emptyset$, $N_i(\mathbf{1}) = N_i(\sim \mathbf{0}) = X_i$

(4) for any terms t, w
$N_i(t \star w) = N_i(t) \bigotimes N_i(w)$, $N_i(t \star (\sim w)) = N_i(t) \bigotimes N_i(\sim w)$

(5) for any DNF terms t_1, t_2
$N_i(t_1 + t_2) = N_i(t_1) \cup N_i(t_2)$,

(6) for any DNF terms t_1, t_2
$N_i((t_1 = t_2)) =$ (if $N_i(t_1) = N_i(t2)$ then T else F)
(T stands for True and F for False)

From the point of view of site i, the interpretation N_i represents a pessimistic approach to query evaluation. If (x, p) belongs to the response of a query t, it means that x satisfies the query t with a confidence not less than p.

5 Conclusion

We have proposed a new query answering system (QAS) for an incomplete cooperative knowledge based system (CKBS). The Chase algorithm based on rules discovered at remote sites helps to make the data at the local site more complete and the same improve the previous QAS (see [6]) for CKBS.

References

1. Atzeni, P., DeAntonellis, V., "Relational database theory", The Benjamin Cummings Publishing Company, 1992
2. Grzymala-Busse, J., *On the unknown attribute values in learning from examples*, Proceedings of ISMIS'91, LNCS/LNAI, Springer-Verlag, Vol. 542, 1991, 368-377
3. Kodratoff, Y., Manago, M.V., Blythe, J."Generalization and noise", in *Int. Journal Man-Machine Studies*, Vol. 27, 1987, 181-204
4. Kryszkiewicz, M., Rybinski, H., *Reducing information systems with uncertain attributes*, Proceedings of ISMIS'96, LNCS/LNAI, Springer-Verlag, Vol. 1079, 1996, 285-294
5. Pawlak, Z., "Rough sets and decision tables", in *Proceedings of the Fifth Symposium on Computation Theory*, Springer Verlag, Lecture Notes in Computer Science, Vol. 208, 1985, 118-127
6. Ras, Z.W., Joshi, S., "Query approximate answering system for an incomplete DKBS", in *Fundamenta Informaticae Journal*, IOS Press, Vol. XXX, No. 3/4, 1997, 313-324
7. Ras, Z.W., "Cooperative knowledge-based systems", in *Intelligent Automation and Soft Computing*, AutoSoft Press, Vol. 2, No. 2, 1996, 193-202
8. Skowron, A., "Boolean reasoning for decision rules generation", in *Methodologies for Intelligent Systems, Proceedings of the 7th International Symposium on Methodologies for Intelligent Systems*, (eds. J. Komorowski, Z. Ras), Lecture Notes in Artificial Intelligence, Springer Verlag, No. 689, 1993, 295-305

This article was processed using the LaTeX macro package with LLNCS style