Distributed Information Systems and Data Security Problem

AGNIESZKA DARDZINSKA-GLEBOCKA
Department of Computer Science
Bialystok Technical University
Wiejska 45A, 15-351 Bialystok
POLAND

Abstract: - In this work we present a new rule-discovery method for Distributed Information System (*DIS*). Distributed Information System is the system that connects a number of information systems using network communication technology. This communication can be driven by request for knowledge needed to predict for maximal optimization what values should replace some null values. In this work we recall the notion of a distributed information system to talk about handling semantic inconsistencies between sites. Semantic inconsistencies are due to different interpretations of attributes and their values on the concepts level among sites. Different interpretations can be also linked with a different way of treating null values among sites. Some attributes might be just hidden because of the security reason. In such case we have to be certain that the missing data can not be reconstructed from the available data by any known null value imputation method and that some information in information system can not be uncovered as well. Assuming that one attribute is hidden at one of the sites of distributed information system we will try to reconstruct this attribute. In this work we will also show which values have to be hidden from users to guarantee that the hidden attribute can not be reconstructed.

Key-Words: - Information System, Distributed Information System, Incomplete Information System, Null Value, Chase algorithm.

1 Introduction

Distributed Information System (*DIS*) is a system that connects a number of information systems using network communication technology. In this paper, we assume that these systems are autonomous and incomplete.

Definition 1
By an Information System we mean a triple *S=(X,A,V)* where:
- *X* is a nonempty, finite set of objects;
- *A* is a nonempty, finite set of attributes;
- $V = \bigcup\{V_a : a \in A\}$ is a set of values of attributes, where V_a is a set of values of attribute *a*, for any $a \in A$.

Additionally we assume that:
- $V_a \cap V_b = \varnothing$ for any $a, b \in A$ such that $a \neq b$,
- $a : X \to V_a$ is a function for every $a \in A$.

We assume also, that the sum of the weights assigned to the attribute values has to be equal 1 in one tuple. In a case when we have empty space in one tuple, we understand that there can be all the values of attribute (from the domain of a given attribute) with equal weights. The definition of an information system of type λ and distributed information system (*DIS*) used in this paper was given in [7]. The type λ is used to check the weights assigned to values of attributes by *Chase* algorithm [7], if they are greater than or equal to threshold λ. If the weight assigned by *Chase* to one of the attribute values is less than the given value λ, then this attribute value is ruled out. Semantic inconsistencies among sites are due to different interpretations of attributes and their values among sites (for instance one site can interpret the concept e.g. *beautiful* differently than another one). Ontology [1], [3], [4], [5], [9], [10], [11], [12] is understood as a set of terms of a particular information domain and the relationships between them. If two information systems agree on the ontology associated with attribute *beautiful* and its values, then such attribute can be used as a kind of semantical bridge between these systems. Different interpretations are also due to the way each site is handling null values. Null value replacement by a value predicted by statistical or some rule-based methods is quite common before queries are answered by *QAS*. In [7], the notion of *rough semantics* and a method of its construction was proposed. The rough semantics can be used to optimize the model and handle semantic inconsistencies among sites due to different interpretations of incomplete values.

2 Query Processing with Incomplete Data

Let us start with the definition of partially incomplete information system S.

Definition 2

By a partially incomplete Information System *S=(X,A,V)* of type λ we mean the incomplete information system, with three conditions:

$(\forall x \in X)(\forall a \in A)$ $a_S(x)$ is defined;

$$[(a_S(x) = \{(a_i, p_i) : 1 \le i \le m)\}) \to \sum_{i=1}^{m} p_i = 1]\}$$

$$[(a_S(x) = \{(a_i, p_i) : 1 \le i \le m)\}) \to (\forall i)(p_i \ge \lambda)\}$$

Now, let us assume that S_1, S_2 are partially incomplete information systems, both of type λ. The same objects from the set of objects X are stored in both systems and the same attributes from the set of attributes A are used to describe them. The meaning and granularity of values of attributes from A in both systems S_1, S_2 is also the same. Additionally, we assume that

$a_{S_1}(x) = \{(a_{1i}, p_{1i}) : 1 \le m_1\}$ and

$a_{S_2}(x) = \{(a_{2i}, p_{2i}) : 1 \le m_2\}$.

We say that containment relation Ψ holds between S_1 and S_2, if the following two conditions hold:

$$(\forall x \in X)(\forall a \in A)[card(a_{S_1}(x)) \ge card(a_{S_2}(x))]$$

$$(\forall x \in X)(\forall a \in A)[[card(a_{S_1}(x)) = card(a_{S_2}(x))]] \to [\sum_{i \ne j} | p_{2i} - p_{2j} | > \sum_{i \ne j} | p_{1i} - p_{1j} |]]$$

This fact can be presented as a statement $(\forall x \in X)(\forall a \in A)[\Psi(a_{S_1}(x)) = \Psi(a_{S_2}(x))]$. If containment mapping Ψ converts an information system S_1 to S_2, then we say that S_2 is more complete than S_1. It means, that for a minimum one pair $(a, x) \in A \times X$, either Ψ has to decrease the number of attribute values in $a_S(x)$ or the average difference between confidences assigned to attribute values in $a_S(x)$ has to be increased by Ψ.

Example

Let us take two information systems S_1, S_2 both of type λ, represented as Table 1 and Table 2.

X	a	b	c	d	e
x_1	$(a_1, \frac{1}{3})$, $(a_2, \frac{2}{3})$	$(b_1, \frac{2}{3})$, $(b_2, \frac{1}{3})$	c_1	d_1	$(e_1, \frac{1}{2})$, $(e_2, \frac{1}{2})$
x_2	$(a_2, \frac{1}{4})$, $(a_3, \frac{3}{4})$	$(b_1, \frac{1}{3})$, $(b_2, \frac{2}{3})$		d_2	e_1
x_3		b_2	$(c_1, \frac{1}{2})$, $(c_3, \frac{1}{2})$	d_2	e_3
x_4	a_3		c_2	d_1	$(e_1, \frac{2}{3})$, $(e_2, \frac{1}{3})$
x_5	$(a_1, \frac{2}{3})$, $(a_2, \frac{1}{3})$	b_1	c_2		e_1
x_6	a_2	b_2	c_3	d_2	$(e_2, \frac{1}{3})$, $(e_3, \frac{2}{3})$
x_7	a_2	$(b_1, \frac{1}{4})$, $(b_2, \frac{3}{4})$	$(c_1, \frac{1}{3})$, $(c_2, \frac{2}{3})$	d_2	e_2
x_8		b_2	c_1	d_1	e_3

Table 1: Information System S_1

For explanation the Definition 2, let us look at the values of attribute *a* in both systems.

Assume that $a_{S_1}(x)=\{(a_1,\frac{1}{3}),(a_2,\frac{2}{3})\}$ and $a_{S_2}(x)=\{(a_1,\frac{1}{5}),(a_2,\frac{4}{5})\}$.

X	a	b	c	d	e
x_1	$(a_1,\frac{1}{5})$, $(a_2,\frac{4}{5})$	$(b_1,\frac{2}{3})$, $(b_2,\frac{1}{3})$	c_1	d_1	$(e_1,\frac{1}{3})$, $(e_2,\frac{2}{3})$
x_2	$(a_2,\frac{1}{4})$, $(a_3,\frac{3}{4})$	b_1	$(c_1,\frac{1}{3})$, $(c_2,\frac{2}{3})$	d_2	e_1
x_3	a_1	b_2	$(c_1,\frac{1}{2})$, $(c_3,\frac{1}{2})$	d_2	e_3
x_4	a_3		c_2	d_1	e_2
x_5	$(a_1,\frac{3}{4})$, $(a_2,\frac{1}{4})$	b_1	c_2		e_1
x_6	a_2	b_2	c_3	d_2	$(e_2,\frac{1}{3})$, $(e_3,\frac{2}{3})$
x_7	a_2	$(b_1,\frac{1}{4})$, $(b_2,\frac{3}{4})$	c_1	d_2	e_2
x_8	$(a_1,\frac{2}{3})$, $(a_2,\frac{1}{3})$	b_2	c_1	d_1	e_3

Table2: Information System S_2

Clearly S_2 is closer to a complete system than S_1 with respect to $a(x)$, since uncertainty in the value of attribute a for x is lower in S_2 than in S_1. It means that the containment mapping Ψ converts system S_1 to S_2.

3 Query Processing with Distributed Data and Chase

The knowledge-base *L(D)* , described as: $L(D)=\{(t\to v_c)\in D : c\in In(A)\}$, is a set of all rules extracted at a remote site for *S=(X,A,V)* by ERID, where *In(A)* is the set of incomplete attributes in *S* and there are given two thresholds for minimum support and minimum confidence.

Algorithm ERID is the algorithm for discovering rules from incomplete information systems, presented in [2]. The type of incompleteness in [2] is the same as in this paper.

Assume now that a query $q(B)$ is submitted to system *S=(X,A,V)*, where *B* is the set of all attributes used in $q(B)$ and that $A\cap B\neq\emptyset$. Attributes belonging to the set $B\setminus[A\cap B]$ are called hidden attributes in information system *S*.
Hidden attributes for *S* can be seen as attributes which are entirely incomplete in *S*, which means exact or partially incomplete values of such attributes have to be ascribed to all objects in *S*. Stronger the consensus among sites on a value to be ascribed to *x*, better the result of the ascription process for *x* can be expected. Assuming that systems S_1, S_2 store the same sets of objects and use the same attributes to describe them, system S_1 is *finer* than system S_2, if $\Psi(S_2)=S_1$.

Let us assume that *S=(X,A,V)* is an information system of type λ and *t* is a term constructed in a standard way (for predicate calculus expression) from values of attributes in *V* seen as *constants* and from two functors + and *.

By $N_S(t)$ we mean the standard interpretation of a term *t* in *S* defined as in [6]

- $N_S(v)=\{(x,p):(v,p)\in a(x)\}$, for any $v\in V_a$,
- $N_S(t_1+t_2)=N_S(t_1)\oplus N_S(t_2)$,
- $N_S(t_1 * t_2)=N_S(t_1)\otimes N_S(t_2)$.

where, for any $N_S(t_1)=\{(x_i,p_i)\}_{i\in I}$, $N_S(t_2)=\{(x_j,q_j)\}_{j\in J}$, we have:

- $N_S(t_1)\otimes N_S(t_2)=\{(x_i,p_i\cdot q_i)\}_{i\in I\cap J}$
- $N_S(t_1)\oplus N_S(t_2)=\{(x_j,p_j)\}_{j\in J\setminus I}\cup\{(x_i,p_i)\}_{i\in I\setminus J}\cup\{(x_i,\max(p_i,q_i))\}_{i\in I\cap J}$

The incomplete value imputation algorithm Chase [8], based on the above semantics converts information system S of type λ to a more complete, new information system of the same type. Algorithm ERID can be used to extract rules from the first information system and next can be applied in Chase.

3 Security Problem of Hidden Attributes

Assume that system S is a distributed information system, and the attribute $h \in A$ is hidden. We also assume, that $S_h=(X,A,V)$, where

- $(\forall a \in A-\{h\})(\forall x \in X) a_S(x) = a_{S_h}(x)$
- $(\forall x \in X) h_{S_h}(x)$ is undefined
- $h_S(x) \in V_h$.

The assumption is, that the user can only submit a query to S_h and not to S. We show how to find a minimal number of additional values which should be hidden to be sure that the values of attribute h can not be reconstructed by Chase for any $x \in X$.

Example

Let us take *IS* from Table 1. Let this system be a system of type $\lambda = \frac{1}{4}$. Let us assume that attribute d is hidden in *S*.

X	a	b	c	d	e
x_1	$(a_1,\frac{1}{3})$, $(a_2,\frac{2}{3})$	$(b_1,\frac{2}{3})$, $(b_2,\frac{1}{3})$	c_1		$(e_1,\frac{1}{2})$, $(e_2,\frac{1}{2})$
x_2	$(a_2,\frac{1}{4})$, $(a_3,\frac{3}{4})$	$(b_1,\frac{1}{3})$, $(b_2,\frac{2}{3})$			e_1
x_3		b_2	$(c_1,\frac{1}{2})$, $(c_3,\frac{1}{2})$		e_3
x_4	a_3		c_2		$(e_1,\frac{2}{3})$, $(e_2,\frac{1}{3})$
x_5	$(a_1,\frac{2}{3})$, $(a_2,\frac{1}{3})$	b_1	c_2		e_1
x_6	a_2	b_2	c_3		$(e_2,\frac{1}{3})$, $(e_3,\frac{2}{3})$
x_7	a_2	$(b_1,\frac{1}{4})$, $(b_2,\frac{3}{4})$	$(c_1,\frac{1}{3})$, $(c_2,\frac{2}{3})$		e_2
x_8		b_2	c_1		e_3

Table 3: Information System S_d

Assume, the following rules where extracted at the remote sites for S_d:

$r_1 : a_2 \cdot b_2 \to d_2$ $\quad$ $r_2 : b_2 \cdot c_1 \to d_1$
$r_3 : b_2 \cdot c_3 \to d_2$ $\quad$ $r_4 : a_1 \cdot c_1 \to d_2$
$r_5 : a_1 \cdot b_2 \to d_1$ $\quad$ $r_6 : a_2 \cdot c_1 \to d_2$

All the above rules have confidence equal 1. Additional rules r_2 and r_5 have support 2, rules r_1, r_3, r_4 have support 3, and rule r_6 has support equals 4. Let us consider the first tuple x_1 from S_d (Table 3). It supports rules r_1, r_2, r_4, r_5, r_6.

Rule r_1 supports d_2 with weight $\frac{2}{3} \cdot \frac{1}{3} \cdot 3 \cdot 1 = \frac{2}{3}$.
Rule r_2 supports d_1 with weight $\frac{1}{3} \cdot 1 \cdot 2 \cdot 1 = \frac{2}{3}$.
Rule r_4 supports d_2 with weight $\frac{1}{3} \cdot 1 \cdot 3 \cdot 1 = 1$.
Rule r_5 supports d_1 with weight $\frac{1}{3} \cdot \frac{1}{3} \cdot 2 \cdot 1 = \frac{2}{9}$.
Rule r_6 supports d_2 with weight $\frac{2}{3} \cdot 1 \cdot 4 \cdot 1 = \frac{8}{3}$.
Because $\frac{8}{9} : \frac{13}{3} < \frac{1}{4}$, the value d_1 is rule out and the same can not be predicted by Chase.

Following the similar strategy for all the objects from information system we obtain a new information system S_d (Table 4)

X	a	b	c	d	e
x_1	$(a_1,\frac{1}{3})$, $(a_2,\frac{2}{3})$	$(b_1,\frac{2}{3})$, $(b_2,\frac{1}{3})$	c_1		$(e_1,\frac{1}{2})$, $(e_2,\frac{1}{2})$
x_2	$(a_2,\frac{1}{4})$, $(a_3,\frac{3}{4})$	$(b_1,\frac{1}{3})$, $(b_2,\frac{2}{3})$			e_1

x_3		b_2	$(c_1, \frac{1}{2})$, $(c_3, \frac{1}{2})$		e_3
x_4	a_3		c_2		$(e_1, \frac{2}{3})$, $(e_2, \frac{1}{3})$
x_5	$(a_1, \frac{2}{3})$, $(a_2, \frac{1}{3})$	b_1	c_2		e_1
x_6					$(e_2, \frac{1}{3})$, $(e_3, \frac{2}{3})$
x_7		$(b_1, \frac{1}{4})$, $(b_2, \frac{3}{4})$	$(c_1, \frac{1}{3})$, $(c_2, \frac{2}{3})$		e_2
x_8		b_2	c_1		e_3

Table 4: New Information System S_d

The values of attributes *a,b,c* for tuple x_6 have been removed. The hidden attribute can not be reconstructed by Chase from the available data in S_d, for any object *x*.

Assume, that the knowledge base contains rules extracted in *DIS* at server sites for S_d with a goal to reconstruct hidden attribute *d*. For each object *x*, first of all we discover all rules supported by the tuple. We have to take into consideration all the possibilities, which are as follows:

- there is only one rule supported by object *x* in S_d
- there is a set of rules supported by object *x* in S_d

In the first case, when we have one rule $r : t \to f$ supported by object *x*, and $d = f$, the value *f* is predicted correctly by *r,* which means, that minimum one of the attributes listed in *t* has to be additionally hidden for *x*.

In the second case, there is a situation, where $r_1 : t_1 \to f_1, ..., r_i : t_i \to f_1$, is the set of all rules supported by x and $d = f$.

In this case a minimal set of attributes covering all terms $\{t1, t2, \ldots, t_i\}$ needs to be additionally hidden for x in S.

There exists also the third possibility, where there is a set of rules, such that $r_1 : t_1 \to f_1, ..., r_i : t_i \to f_i$, supported by *x*. In this case we calculate support of the rule s_i, and its confidence c_i. Let λ be also given threshold for minimal confidence in attribute values for objects in *S*.

If Conf $S(f, x) > \lambda$ and $(\exists e \neq f)$[Conf $S(e, x) > \lambda$], we do not have to hide any additional slots for x.

If Conf $S(f, x) > \lambda$ and $(\forall e \neq f)$[Conf $S(e, x) < \lambda$], we have to hide additional slots for x.

If Conf $S(f, x) < \lambda$ and $(\exists e \neq f)$[Conf $S(e, x) > \lambda$], we do not have to hide additional slots for x.

The confidence in attribute value *e* for x in S is as follows:

Conf $S(e, x) =$

$$\Sigma\{s_i \cdot c_i : 1 \le i \le k \wedge e = d_i\} / \Sigma\{s_i \cdot c_i : 1 \le i \le k\}$$

4 Conclusion

Presented method seems to be very interesting and promising in hiding some values of attributes from data security point of view. We showed the possibility and importance of hiding the attributes in information systems. For any tuple *x* we are able to identify all the rules supported by that tuple. On the basis of these rules, we calculate the total support for each value of the hidden attribute. These total supports are used to calculate the confidence in each of these values. If the confidence is below the given threshold λ, then such value is rule out. We need minimum two weighted values remaining if the correct value is one of them. This can be achieved by replacing some values by null ones. The suggested strategy provides a way to identify a minimal number of additional slots in *IS* required to be hidden if one of the attributes in *IS* has to be hidden.

References:

[1] Benjamins, V. R., Fensel, D., Pérez, A. G., Knowledge management through ontologies, *Proceedings of the 2nd International Conference on Practical Aspects of Knowledge Management (PAKM-98)}, Basel, Switzerland*, 1998

[2] Dardzinska, A., Ras, Z.W., On Rules Discovery from Incomplete Information Systems, *Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining*, (Eds: T.Y. Lin, X. Hu, S. Ohsuga, C.Liau), Melbourne, Florida, IEEE Computer Society, 2003, 31-35

[3] Fensel, D., Ontologies: a silver bullet for knowledge management and electronic commerce, *Springer-Verlag*, 1998

[4] Guarino, N., Formal Ontology in Information *System, IOS Press*, Amsterdam, 1998.

[5] Guarino, N., Giaretta, P, Ontologies and knowledge bases towards a terminological

clarification, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, IOS Press*, 1995.

[6] Ras, Z.W., Joshi, S. Query approximate answering system for an incomplete DKBS, *Fundamenta Informaticae Journal, IOS Press*, Vol. 30, No. 3/4, 1997, 313-324

[7] Ras, Z.W, Dardzinska, A, Ontology Based Distributed Autonomous Knowledge Systems, *Information Systems International Journal, Elsevier*, Vol.29, No.1, 2004, 47-58.

[8] Ras, Z.W., Dardzinska, A. Query answering based on collaboration and chase, *Proceedings of FQAS 2004 Conference*, Lyon, France, LNCS/LNAI, No. 3055, Springer-Verlag, 125-136

[9] Sowa, J.F., Ontology, metadata and semiotics, *B.Ganter & G.W.*ineau, *Conceptual Structures: Logical, Linguistic, and Computational Issues*, LNAI, No.1867, Spronger-Verlag, Berlin, 2000, pp.55-81.

[10] Sowa, J.F., Knowledge Representation: Logical, Philosophical and Computational Foundations, *Brooks/ColePublishing Co., Pacific Grove,* CA, 2000.

[11] Sowa, J.F., Ontological categories, L. Albertazzi,ed., *Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics*, Kluwer Academic Publishers, Dordrecht, 1999, pp. 307-340.

[12] Van Heijst, G., Schreiber, A., Wielinga, B. Using explicit ontologies in KBS development, *International Journal of Human and Computer Studies*, Vol. 46, No. 2/3, 183-292.