Data Confidentiality versus Chase

Zbigniew W. Raś[1,2], Osman Gürdal[3], Seunghyun Im[4], and Angelina Tzacheva[5]

[1] Univ. of North Carolina, Dept. of Comp. Science, Charlotte, N.C. 28223
[2] Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland
[3] Johnson C. Smith Univ., Dept. of Comp. Sci. and Eng., Charlotte, NC 28216
[4] Univ. of Pittsburgh at Johnstown, Dept. of Comp. Science, Johnstown, PA 15904
[5] Univ. of South Carolina Upstate, Dept. of Informatics, Spartanburg, SC 29303

Abstract. We present a generalization of a strategy, called $SCIKD$, proposed in [7] that allows to reduce a disclosure risk of confidential data in an information system S [10] using methods based on knowledge discovery. The method proposed in [7] protects confidential data against Rule-based Chase, the null value imputation algorithm driven by certain rules [2], [4]. This method identifies a minimal subset of additional data in S which needs to be hidden to guarantee that the confidential data are not revealed by Chase. In this paper we propose a bottom-up strategy which identifies, for each object x in S, a maximal set of values of attributes which do not have to be hidden and still the information associated with secure attribute values of x is protected. It is achieved without examining all possible combinations of values of attributes. Our method is driven by classification rules extracted from S and takes into consideration their confidence and support.

1 Introduction

This article discusses an important issue in data mining: how to provide meaningful knowledge without compromising data confidentiality. In conventional database systems, data confidentiality is achieved by hiding sensitive data from unauthorized users. However, hiding is not sufficient in knowledge discovery systems (KDS) due to null imputation method like rule-based Chase ([2], [4]) which are designed to predict null or missing values. Suppose that attributes in a database contain medical information about patients; some portions are not confidential while others are confidential (they are hidden from users). In this case, part or all of the confidential data in the attribute may be revealed. In other words, self-generated rules extracted from non-confidential portions of data can be used to find secret data.

Security in KDS is studied in many research areas, such as cryptography, statistics, and data mining. A well known security problem in cryptography area is how to acquire global knowledge without revealing the data stored in each local site in a distributed autonomous information system ($DAIS$). Proposed solutions are based primarily on secure multiparty protocol ([12], [5]) which ensures that each participant cannot learn more than its own input data and outcome of a public function. Various authors expanded the idea. Clifton and Kantarcioglou employed the protocol for association rule mining for vertically and horizontally partitioned data [8]. Authors Du and Zhan pursued a similar idea to build a decision tree system [6]. Protection of sensitive rules has been discussed by Oliveira and Zaiane [9]. Authors suggested a solution to protecting sensitive association rules in the form of "sanitization process" that hides selective patterns from frequent itemsets. The data security problem discussed in this article is different from other researches in the following ways. First, we focus on the accuracy of existing data or knowledge instead of statistical characteristics of data. Second, we aim to protect sensitive data in a database instead of sensitive rules.

Our paper takes the definition of an information system proposed by Pawlak [10] as a simplified model of a database. However, the notion of its incompleteness differs from the classical rough set approach by allowing a set of weighted attribute values as a value of an attribute. We also assume that the sum of these weights has to be equal 1. If weights assigned to attribute values have to be greater than a user specified threshold value λ, then we get information system of type λ as introduced in [4].

Additionally we assume that one or more attributes in an information system S of type λ contain confidential data that have to be protected and S is a part of a distributed autonomous information system ($DAIS$) which provides a set of rules applicable at S as a KB [11]. We have to be certain that values of any confidential attribute can not be revealed from the available data in S and KB by *Chase* [2] or any other null value imputation method while minimizing the changes in the original information system. Also, we assume that we can hide the precise information about objects from the user but we can not replace existing data by false data. For instance, if someone is 18 years old, we can say that she is young or her age is unknown but we can not say that she is 24 years old. In pursue of such requirements, we propose a protection method named as *SCIKD* for information systems of type λ. The method identifies weighted transitive closure of attribute values involved in confidential data reconstruction, and uses the result to identify the maximum number of attribute values that can remain unchanged.

2 Chase as Tool for Revealing Hidden Values

We briefly provide some background on a null value imputation algorithm *Chase* based on rule-discovery strategy called *ERID* [2]. Assume that $S = (X, A, V)$, where $V = \bigcup\{V_a : a \in A\}$ and each $a \in A$ is a partial function from X into $2^{V_a} - \{\emptyset\}$. In the first step, Chase algorithm identifies all incomplete attributes in S. An attribute is incomplete if there is an object in S with incomplete information on this attribute. The values of all incomplete attributes in S are treated as concepts to be learned (in a form of rules) either directly from S or from S and its remote sites (if S is a part of $DAIS$). The second step of Chase algorithm is to extract all these rules and store them in a knowledge base D

for S [11]. The next step is to replace incomplete information in S by values provided by rules in D. This process is recursively repeated till no new hidden values in S can be revealed.

Definition:
We say that $S = (X, A, V)$ is a partially incomplete information system of type λ, if the following four conditions hold:

- X is the set of objects, A is the set of attributes, and $V = \bigcup\{V_a : a \in A\}$ is the set of values of attributes,
- $(\forall x \in X)(\forall a \in A)[a_S(x) \in V_a$ or $a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}) \rightarrow \sum_{i=1}^{m} p_i = 1]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}) \rightarrow (\forall i)(p_i \geq \lambda)]$.

An example of an information system of type $\lambda = \frac{1}{5}$ is given in Table 1.

X	a	b	c	d	e	f	g
x_1	$(a_1,\frac{2}{3})(a_2,\frac{1}{3})$	b_1	c_1	d_1	e_1	f_1	g_1
x_2	$(a_2,\frac{2}{5})(a_3,\frac{3}{5})$	$(b_1,\frac{1}{3})(b_2,\frac{2}{3})$		d_2	e_1	f_2	
x_3	a_1	b_2	$(c_1,\frac{1}{2})(c_3,\frac{1}{2})$	d_1	e_3	f_2	
x_4	a_3		c_2	d_1	$(e_1,\frac{2}{3})(e2,\frac{1}{3})$	f_2	
x_5	$(a_1,\frac{2}{3})(a_3,\frac{1}{3})$	$(b_1,\frac{1}{2})(b_2,\frac{1}{2})$	c_2	d_1	e_1	f_2	g_1
x_6	a_2	b_2	c_3	d_1	$(e_2,\frac{1}{3})(e_3,\frac{2}{3})$	f_3	
x_7	a_2	b_1	$(c_1,\frac{1}{3})(c_2,\frac{2}{3})$		e_2	f_3	
			.				
			.				
x_i	$(a_3,\frac{1}{2})(a_4,\frac{1}{2})$	b_1	c_2		e_3	f_2	

Table 1. Information System S

Let us assume that another information system S_2 has the same values as S except $a(x_1)$=$\{(a_1, \frac{3}{4}), (a_2, \frac{1}{4})\}$ and $b(x_5)$=$\{(b_1, \frac{3}{4}), (b_2, \frac{1}{4})\}$. In both cases, an attribute value assigned to an object in S_2 is less general than in S_1.

Now, let us assume that S, S_2 are partially incomplete information systems, both of type λ. They provide descriptions of the same set of objects X using the same set of attributes A. The meaning and granularity of values of attributes in A for both systems S, S_2 is also the same. Additionally, we assume that $a_S(x) = \{(a_i, p_i) : i \leq m\}$ and $a_{S_2}(x) = \{(a2_i, p2_i) : i \leq m_2\}$.

Now, we introduce the relation Ψ, called containment relation. We say that $(S, S_2) \in \Psi$, if the following two conditions hold:

– $(\forall x \in X)(\forall a \in A)[card(a_{S(x)}) \geq card(a_{S_2(x)})]$,

– $(\forall x \in X)(\forall a \in A)[[card(a_S(x)) = card(a_{S_2}(x))] \rightarrow$
$[\sum_{i \neq j} |p2_i - p2_j| > \sum_{i \neq j} |p_i - p_j|]]$.

Instead of saying that containment relation holds between S and S_2, we can equivalently say that S was transformed into S_2 by containment mapping Ψ. Algorithm $Chase_2$, described by Dardzińska and Raś in [2], converts an information system S of type λ to a new more complete information system $Chase_2(S)$ of the same type. The algorithm differs from other known strategies for chasing incomplete data in relational tables because of the assumption concerning partial incompleteness of data (sets of weighted attribute values can be assigned by $Chase_2$ to an object as its new value). This assumption forced authors in [3] to develop a new discovery algorithm, called *ERID*, for extracting rules from incomplete information systems of type λ. The syntax of classification rules discovered by *ERID* is the same as syntax of similar rules discovered by classical methods, like *LERS* or *RSES*. However, the method of computing their confidence and support is different.

X	a	b	c	d	e	f	g
x_1	$(a_1,\frac{2}{3})(a_2,\frac{1}{3})$	b_1	c_1		e_1	f_1	g_1
x_2	$(a_2,\frac{2}{5})(a_3,\frac{3}{5})$	$(b_1,\frac{1}{3})(b_2,\frac{2}{3})$			e_1	f_2	
x_3	a_1	b_2	$(c_1,\frac{1}{2})(c_3,\frac{1}{2})$		e_3	f_2	
x_4	a_3		c_2		$(e_1,\frac{2}{3})(e2,\frac{1}{3})$	f_2	
x_5	$(a_1,\frac{2}{3})(a_3,\frac{1}{3})$	$(b_1,\frac{1}{2})(b_2,\frac{1}{2})$	c_2		e_1	f_2	g_1
x_6	a_2	b_2	c_3		$(e_2,\frac{1}{3})(e_3,\frac{2}{3})$	f_3	
x_7	a_2	b_1	$(c_1,\frac{1}{3})(c_2,\frac{2}{3})$		e_2	f_3	
			.				
			.				
x_i	$(a_3,\frac{1}{2})(a_4,\frac{1}{2})$	b_1	c_2		e_3	f_2	

Table 2. Information System S_d

Algorithm $Chase_2$ based on *ERID* can be used as a null value imputation tool to reveal hidden symbolic data. The method proposed in [7] protects confidential data against $Chase_2$ assuming that it is driven by certain rules. It identifies a minimal subset of additional data in S which needs to be hidden to guarantee that the confidential data can not be revealed by Chase. In this paper we generalize this strategy by proposing an algorithm which protects confidential data against $Chase_2$ driven by *ERID*. It is a bottom-up strategy which identifies, for each object x in S, a maximal set of values of attributes which do not have to be entirely hidden and still the information associated with secure attribute values of x is protected.

3 Algorithm Protecting Confidential Data against Rule-based Chase

In this section we present an algorithm which protects values of a hidden attribute over null value imputation $Chase_2$ based on *ERID*. Suppose we have an information system S as shown in Table 1 of type $\lambda = \frac{1}{5}$. S is transformed to S_d by hiding the confidential attribute d as shown in Table 2. The rules in the knowledge base KB are summarized in Table 3. For instance $r_1 = [b_1 \cdot c_1 \rightarrow a_1]$ is an example of a rule belonging to KB and its confidence is 1.

Rule	*Conf*	a	b	c	d	e	f	g
r_1	1	(a_1)	b_1	c_1				
r_2	1	(a_1)		c_1			f_1	
r_3	$\frac{2}{3}$		(b_1)	c_1				
r_4	1		(b_1)			e_1		
r_5	1	a_1		(c_1)			f_1	
r_6	1	a_1		c_1		(e_1)		
r_7	$\frac{2}{3}$			(c_1)		e_1		g_1
r_8	1	a_1		c_1	(d_1)			
r_9	1		b_1	c_1	(d_1)			
r_{10}	1				(d_1)		f_1	

Table 3. Rules contained in *KB*. Values in parenthesis are decision values

To describe the algorithm, first we define the following sets,

- $\alpha(x) = \{a \in A : a(x) \neq Null\}$, the set of attribute values in S_d used to describe x
- $\alpha(t)$, the set of attribute values used in t, where t is their conjunction
- $R(x) = \{(t \rightarrow c) \in KB : \alpha(t) \subseteq \alpha(x)\}$, the set of rules in KB where the attribute values used in t are contained in $\alpha(x)$
- $\beta(x) = \cup\{\alpha(t) \cup \{c\} : [t \rightarrow c] \in R(x)\}$.

In our example $R(x_1) = \{r_1,r_2,r_3,r_4,r_5,r_6,r_7,r_8,r_9,r_{10}\}$, and $\beta(x_1) = \{a_1, b_1, c_1, d_1, e_1, f_1, g_1\}$. By using $Chase_2$ based on *ERID*, d_1 replaces the hidden slot $d(x_1)$ by rules from $\{r_8, r_9, r_{10}\}$. Rules r_9, r_{10} guarantee the confidence 1 assigned to d_1, whereas the rule r_8 only guarantees the confidence $\frac{2}{3}$ which is above the threshold value $\lambda = \frac{1}{5}$. In addition, other rules from $R(x_1)$ also predict attribute values listed in $\{t_8, t_9, t_{10}\}$. These interconnections often build up a complex chain of inferences. The task of blocking such inference chains and identifying the minimal set of concealing values is not straightforward [7], especially that the confidence assigned to rules in *KB* and the confidence assigned to attribute values in S_d have to be taken into consideration.

To reduce the complexity and minimize the size of the set of hidden values, a bottom up approach has been adapted. We check the values that remain unchanged starting from a singleton set containing attribute value a by using weighted transitive closure [4] (if $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$, which gives us the set $\{a, b, c\}$). What about computing the weights assigned to a, b, c? Let us assume that $a \rightarrow b$ has a confidence λ_1 and $b \rightarrow c$ has a confidence λ_2. Then, weight 1 is assigned to a, weight λ_1 is assigned to b, and weight $(\lambda_1 \cdot \lambda_2)$ is assigned to c. If λ_3 is a weight associated with a, then weight $(\lambda_3 \cdot \lambda_1)$ is assigned to b, and weight $(\lambda_3 \cdot \lambda_1 \cdot \lambda_2)$ is assigned to c. If the weight assigned to any of the elements in $\{a, b, c\}$ is below the threshold value λ, then this element is removed from $\{a, b, c\}$. Our goal is to increase the initial set size as much as possible. Let us notice that any element of the resulting set can be generated by following two different paths. Each path assigns a different weight to that element. In all such cases, the highest weight is chosen by our algorithm. This approach automatically rules out any superset of must-be-hidden values, and minimizes the computational cost. The justification of this is quite simple. Weighted transitive closure has the property that the superset of a set s also contains s. Clearly, if a set of attribute values predicts d_1, then the set must be hidden regardless of the presence/abscence of other attribute values.

To outline the procedure, we start with a set $\beta(x) = \{(a_1, \frac{2}{3}), b_1, c_1, e_1, f_1, g_1\}$ for the object x_1 which construction is supported by 10 rules from KB, and check the transitive closure of each singleton subset $\delta(x)$ of that set. If the transitive closure of $\delta(x)$ contains classified attribute value d_1 and the weight associated with d_1 is greater than λ, then $\delta(x)$ does not sustain, it is marked, and it is not considered in later steps. Otherwise, the set remains unmarked. In the second iteration of the algorithm, all two-element subsets of $\beta(x)$ built only from unmarked sets are considered. If the transitive closure of any of these sets does not contain d_1 with weight associated to it greater than λ, then such a set remains unmarked and it is used in the later steps of the algorithm. Otherwise, the set is getting marked. If either all sets in a currently executed iteration step are marked or we have reached the set $\beta(x)$, then the algorithm stops. Since only subsets of $\beta(x)$ are considered, the number of iterations will be usually not large.

So, in our example the following singleton sets are considered:
$\{(a_1, \frac{2}{3})\}^+ = \{(a_1, \frac{2}{3})\}$ is unmarked
$\{b_1\}^+ = \{b_1, \}$ is unmarked
$\{c_1\}^+ = \{(a_1, \frac{2}{3}), (b_1, \frac{2}{3}), c_1, (e_1, \frac{4}{9}), (d_1, \frac{4}{9})\}$ contains d_1 and $\frac{4}{9} \geq \lambda$ so it is marked
$\{e_1\}^+ = \{b_1, e_1\}$ is unmarked
$\{f_1\}^+ = \{d_1, f_1\}$ contains d_1 so it is marked
$\{g_1\}^+ = \{g_1\}$ is unmarked

Clearly, c_1 and f_1 have to be concealed. The next step is to build sets of length 2 and determine which of them can sustain. We take the union of two sets only if they are both unmarked and one of them is a singleton set.

$\{(a_1, \frac{2}{3}), b_1\}^+ = \{(a_1, \frac{2}{3}), b_1\}$ is unmarked
$\{(a_1, \frac{2}{3}), e_1\}^+ = \{(a_1, \frac{2}{3}), b_1, e_1\}$ is unmarked
$\{(a_1, \frac{2}{3}), g_1\}^+ = \{(a_1, \frac{2}{3}), g_1\}$ is unmarked

$\{b_1, e_1\}^+ = \{b_1, e_1\}$ is unmarked
$\{b_1, g_1\}^+ = \{b_1, g_1\}$ is unmarked
$\{e_1, g_1\}^+ = \{(a_1, \frac{2}{3}), (b_1, \frac{2}{3}), (c_1, \frac{2}{3}), (d_1, \frac{2}{3}), e_1, g_1\}$ contains d_1 and $\frac{2}{3} \geq \lambda$ so it is marked

Now we build 3-element sets from previous sets that have not been marked.

$\{(a_1, \frac{2}{3}), b_1, e_1\}^+ = \{(a_1, \frac{2}{3}), b_1, e_1\}$ is unmarked
$\{(a_1, \frac{2}{3}), b_1, g_1\}^+ = \{(a_1, \frac{2}{3}), b_1, g_1\}$ is unmarked
$\{b_1, e_1, g_1\}^+$ is not considered as a superset of $\{e_1, g_1\}$ which was marked.

We have $\{a_1, b_1, e_1\}$ and $\{a_1, b_1, g_1\}$ as unmarked sets that contain the maximum number of elements and do not have the transitive closure containing d with associated weight greater than λ. In a similar way, we compute the maximal sets for any object x_i.

The corresponding algorithm, called *G-SCIKD*, is a generalization of *SCIKD* strategy presented in [7]. If an attribute values revealed by *G-SCIKD* has a confidence below λ, then this attribute value is removed from consideration. This constraint is semantically similar to the constraint λ used in *ERID* [2].

4 Experiment

We implemented *G-SCIKD* on a PC running Windows XP and Oracle database version 10g. The code was written in PL/SQL language with PL/SQL Developer version 6.

The sampling data table containing 4,000 objects with 10 attributes was extracted randomly from a complete database describing personal income reported in the Census data [1]. The data table was randomly partitioned into 4 tables that each have 1,000 tuples. One of these tables is called *client* and the remaining 3 are called *servers*. Now, we hide all values of one attribute that includes income data at the client site. From the servers, 13 rules are extracted by *ERID* and stored in KB of the client. Additionally, 75 rules describing incomplete or partially hidden attributes at the client site are extracted by *ERID*. All these rules are used to reveal values of incomplete attributes by *Chase* algorithm [2]. It appears that 739 attribute values (7.39% of the total number of attribute values in client table) have to be additionally hidden. The presented method can easily be used to protect two or more confidential attributes in an information system. In this case, a set of attribute values in x_i should be hidden if the closure of this set contains any of the classified data.

5 Acknowledgment

This research was supported by NIH Grant No. 5G11HD38486-06.

References

1. UCI Machine Learning Rep., http://www.ics.uci.edu/ mlearn/MLRepository.html
2. Dardzińska, A., Raś, Z.W. (2003) Rule-Based Chase Algorithm for Partially Incomplete Information Systems, in **Proceedings of the Second International Workshop on Active Mining (AM'2003)**, Maebashi City, Japan, October, 42-51
3. Dardzińska, A., Raś, Z.W. (2003) On Rules Discovery from Incomplete Information Systems, in **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, 31-35
4. Dardzińska, A., Raś, Z.W. (2003) Chasing Unknown Values in Incomplete Information Systems, in **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, 24-30
5. Du, W. and Atallah, M. J. (2001) Secure Multi-party Computation Problems and their Applications: a review and open problems, in **New Security Paradigms Workshop**
6. Du, W. and Zhan, Z. (2002), Building decision tree classifier on private data, in **Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining**
7. Im, S., Raś, Z.W., Dardzińska, A. (2005) SCIKD: Safeguarding Classified Information from Knowledge Discovery, in **Foundations of Semantic Oriented Data and Web Mining**, Proceedings of 2005 IEEE ICDM Workshop in Houston, Texas, Published by Math. Dept., Saint Mary's Univ., Nova Scotia, Canada, 34-39
8. Kantarcioglou, M. and Clifton, C. (2002), Privacy-preserving distributed mining of association rules on horizontally partitioned data, in **Proceedings of the ACM SIGMOD Workshop on Research Isuues in Data Mining and Knowledge Discovery**, 24-31
9. Oliveira, S. R. M. and Zaiane, O. R. (2002), Privacy preserving frequent itemset mining, in **Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining**, 43-54
10. Pawlak, Z. (1991) Information Systems - theoretical foundations, in **Information Systems Journal**, Vol. 6, 205-218
11. Raś, Z.W., Dardzińska, A. (2006) Solving Failing Queries through Cooperation and Collaboration, Special Issue on Web Resources Access, (Editor: M.-S. Hacid), in **World Wide Web Journal**, Springer, Vol. 9, No. 2, 173-186
12. Yao, A. C. (1996) How to generate and exchange secrets, in **Proceedings of the 27th IEEE Symposium on Foundations of Computer Science**, 162-167