Protection of Sensitive Data based on Reducts in a Distributed Knowledge Discovery System

Seunghyun Im
University of Pittsburgh at Johnstown
Computer Science Department
Johnstown, PA, 15904, USA
sim@pitt.edu

Zbigniew W. Raś
University of North Carolina at Charlotte
Computer Science Department
Charlotte, NC, 28223, USA
ras@uncc.edu

Abstract

This paper discusses data confidentiality in a Distributed Knowledge Discovery System (DKDS) . In particular, we provide a method that protects values of confidential attributes from being reveled by Chase algorithm [8] using reducts [13]. The method presented in this paper is intended for use in DKDS that the set of rules used by Chase algorithm is not completely known in advance. Reduct is used to determine the optimal set of data to be additionally hidden.

1. Introduction

The issue of data confidentiality with respect to Chase was addressed in [9], that the rules extracted from remote information systems can be used to mine confidential data with data reconstruction process of Chase algorithm. More specifically, sensitive data (e.g. medical record) stored in attribute of an information system can be hidden from the users to maintain confidentiality. However, users in distributed knowledge discovery system (DKDS) may acquire knowledge (in terms of rules) from remote sites, and run Chase algorithm with these knowledge to reveal the hidden data. Information systems mined by DKDS are built independently, and they collaborate with each other through knowledge sharing. Therefore, confidentiality of sensitive data should be maintained at each site. The basis of the idea of data protection against Chase is to hide additional data from local information system so that the inference rule extracted from remote sites cannot be applied. If the rules reconstructing confidential data are completely known, we can compare conditional part of the rules with the objects in the information system to identify the set of attribute values applied by the inference rules [3]. However, the assumption may not hold for some DKDSs that allow users to extract rules without restrictions so that the rules not stored in knowledge base (KB) are used for Chase. If that is the case, we need a way to identify the optimal set of attribute values to be additionally hidden without all possible rules. The notion of object reduct [13] has been used to develop a solution for this problem which discovers the essential relations between attribute values in the information system.

2. Background and Related Work

2.1 Incomplete Information System

One of the assumptions made by most rule extraction algorithms is that rules are discovered from an information system that the information about objects is either precisely known or not known at all. This implies that either a single value of an attribute is assigned to an object as its property or no value is assigned. However, it happens quite often that users do not have exact knowledge about objects, which makes it difficult to determine a unique set of values for an object. To overcome this problem, the notion of incomplete information system [1] was introduced which is a generalization of an information system introduced by Pawlak [5][6].

More formally, by an information system, we mean $S = (X, A, V)$, where X is a finite set of objects, A is a finite set of attributes, and V is a set of attribute values. In particular, we say that S is an incomplete information system of type λ [1] if the following three conditions hold:

- $a_S(x) = Null$ or $(\exists m)(\forall i \leq m)(\exists a_i)(\exists p_i)[a_S(x) = \{(a_i, p_i) : 1 \leq i \leq m\}]$, for any $x \in X$, $a \in A$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(a_i, p_i) : 1 \leq i \leq m\}) \rightarrow (\sum_{i=1}^{m} p_i = 1)]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(a_i, p_i) : 1 \leq i \leq m\}) \rightarrow (\forall_i)(p_i \geq \lambda)]$.

Incompleteness is understood by having a set of weighted attribute values as a value of an attribute. The concept of multiple possible values is used for replacing null values in Chase.

2.2 Chase Algorithm and Null Value Imputation

The incomplete value imputation algorithm $Chase$ converts information system S of type λ to a more complete information system $Chase(S)$ of the same type. This algorithm assumes partial incompleteness of data (sets of weighted attribute values can be assigned to an object as its value) in system S. The main phase of the algorithm is the following,

1. Identify all incomplete attribute values in S.
2. Extract rules from S describing these incomplete attribute values.
3. Incomplete attribute values in S are replaced by values suggested by the rules.
4. Repeat steps 1-3 until a fixed point is reached.

More formally, suppose that $KB = \{(t \rightarrow v_c) \in D : c \in In(A)\}$ (called a knowledge base) is a set of all rules extracted from $S = (X, A, V)$ by $ERID(S, \lambda_1, \lambda_2)$, where $In(A)$ is the set of incomplete attributes in S and λ_1, λ_2 are thresholds for minimum support and minimum confidence, correspondingly. *ERID* [10] is the algorithm for discovering rules from incomplete information systems, and used as a part of null value imputation algorithm Chase. Now, let $R_s(x_j) \subseteq KB$ be the set of rules that all of the conditional part of the rules match with the attribute values in $x_j \in S$, and $d(x_j)$ is the null value. Then, there are three cases:

- Case 1 : $R_s(x_j) = \phi$
- Case 2: $R_s(x_j) = \{r_1 = [t_1 \rightarrow d_1], r_2 = [t_2 \rightarrow d_1], ..., r_k = [t_k \rightarrow d_1]\}$
- Case 3 : $R_s(x_j) = \{r_1 = [t_1 \rightarrow d_1], r_2 = [t_2 \rightarrow d_2], ..., r_k = [t_k \rightarrow d_k]\}$

In case 1, d cannot be replaced because there is no rule predicting the decision value d. In case 2, every rule implies a single decision attribute value, and we replace the null value d with d_1 with confidence of 1. In case 3, rules imply multiple decision values, and the replacement is determined by the confidence of the values predicted.

Assume that $\alpha(t_i)$, where $\alpha(t_i) \subseteq A$, is a set of attributes listed in t_i. To calculate the confidence of d', we use the following formula [3]. Assuming now that $t_i = \prod\{a(t_i) : a \in \alpha(t_i)\}$, support and confidence of a rule r_i is $[s_i, c_i]$, the weight in S associated with $a(t_i)$ is $p_{a(t_i)}$, then:

$$conf(d') = \frac{\sum\{[\prod p_{a(t_i)}] \cdot s_i \cdot c_i : [d' = d_i]\}}{\sum\{[\prod p_{a(t_i)}] \cdot s_i \cdot c_i\}}, 1 \leq i \leq k \quad (1)$$

Note that we replace the null value assigned to d with d' when $conf(d')$ is greater than a threshold value λ.

Chase is an iterative process. Execution of the algorithm which creates a new information system is repeated until it does not improve the confidence of attribute values.

We can also use Chase algorithm in a DKDS. It is very possible that an attribute is missing in one of sites while it occurs in many others. Also, in one information system, an attribute might be partially hidden, while in other systems the same attribute is either complete or close to being complete. In such a case, network communication technology is used to get definitions of these unknown attributes from other information systems. All these new definitions form a knowledge base that can be used to chase missing attributes at the client site.

2.3 Related Work

Several methods meeting different requirements have been proposed to improve data confidentiality. For example, to achieve minimum loss of data from an information system, bottom up approach [2] uses Chase closure [2] to find the maximum set of data that do not reconstruct confidential data. Another algorithm presented in [4] is a top down approach that hides a set of data that eliminates largest number of rules involved in hidden data reconstruction. The algorithm in [3] takes advantage of the information systems represented in hierarchical attribute structures. Experimental results show that the algorithm in [3] has the smallest data loss in terms of the number of null value replacements. Another research [11] was conducted in order to minimize knowledge loss. The objective for minimum knowledge loss is to minimize the changes of existing knowledge as much as possible. The interestingness of knowledge for a site or domain was determined based on most commonly used factors of rule significance, such as certainty and support.

3 Sensitive Data Protection based on Reducts

3.1 Problem Statement and Goal

The problem that motivated our work is the data confidentiality problem in distributed knowledge discovery systems where knowledge is generated from local and remote sites, and provided to users in terms of rules. In particular,

one or more attributes in the information systems contain confidential data. We assume that the system allows users to generate rules with their own support and confidence values.

If the DKDS restricts users to generate rules (e.g. users can only access the rules provided by DKDS), the set of attribute values applicable by Chase is fixed. Then, hiding additional attribute values using one of the existing algorithms [3] [2] is sufficient to ensure data confidentiality. That is to hide attribute values appearing in the conditional part of the rules, object by object. However, running Chase algorithm with different set of rules may result in disclosure of part of the confidential data. For example, 28% of confidential attribute values protected by bottom up approach algorithm [2] were revealed when confidence value used in ERID was changed from 80% to 70% for the data set 'congressional voting' obtained from [12]. Running Chase algorithm using rules with higher confidence (e.g. from 80% to 90%) also revealed some of the confidential data if extracted rules are different from the rules used by the protection algorithm.

A naive solution to this problem is to run the algorithm with a large number of rules generated with wide range of confidence and support values. However, as we increase the size of KB, more attribute values will most likely have to be hidden. In addition, malicious users may use even lower values for rule extraction attributes, and we may end up with hiding all data. In fact, ensuring data confidentiality against all possible rules is difficult because Chase does not enforce minimum support and confidence of rules when it reconstructs missing data. Therefore, the security against Chase should aim to reduce the confidence of the reconstructed values, particularly, by meaningful rules, such as rules with high support or high confidence, instead of trying to prevent data reconstruction by all possible rules. In other words, malicious users may obtain some of the confidential data, but the confidence on the reconstructed data should be lower.

3.2 Design Principal

We designed our method according to the following design principles:

- Remove attribute values that will more likely be used to predict confidential attribute values with high confidence.
- The amount of attribute values to be hidden from S is adjustable

As discussed, we assume that Chase applicable rules are unknown or only partially known when we run our method. To find the optimal set of attribute values to be additionally hidden without using all Chase applicable rules, we use the notion of object reduct [13] [7]. Reduct is the set of essential attributes (or attribute values if object based reducts are generated) that identifies the relations between attributes in an information system. The logic behind this approach is that hiding data from these essential attributes (related to confidential attribute) will also reduce or remove data reconstruction from large portion of possible rules because data in these relations contribute to the most frequently occurring patterns. Conventional method for object reducts [13] eliminates almost-certain relations in the information system if there exists a small amount of noise in the data set. Also it often generates an excessively large number of reducts which equally represent the unique relations of data in the information system. We use the notion of *relative reduct* to overcome these problems. The idea of relative reduct is to take a sample from an information system S, say 95%, multiple times. A weight is calculated based on frequency of the object reducts appearing in each run. Clearly, object reducts containing the confidential attribute have to be extracted from remote sites.

Now, assuming that object reducts are acquired, there are two possible ways to determine the amount of attribute values to hide from S.

1. the number of additional attribute values to be hidden from S, denoted as τ_1
2. the degree of confidential data prediction to be decreased, denoted as τ_2.

Suppose that $R_d = \{r = (t \rightarrow d)\}$ is the set of all object reducts for attribute d which is generated with μ percents of sample data, and $\alpha(t)$ is the set of attribute values used in t, where t is their conjunction. To hide additional attribute values based on τ_1 we start hiding $v \in t$ having the highest weight (computation of these weight will be discussed in next section with an example). If the number of data removed from S is less then τ_1 choose another v with the next highest weight and hide them until the total number of additionally hidden data has reached to τ_1.

τ_2 based hiding does not limit the number of data to be hidden. Instead, it measures the difference of overall prediction accuracy of the confidential data [4]. Let the sum of initial prediction accuracy be $\phi(S)$. After hiding each attribute values from $v \in t$ we measure the difference $\phi(S) - \phi(S')$, where S' is the new information system. The process is repeated until $\phi(S) - \phi(S') < \lambda_2$

3.3 Method Description

We assume that an incomplete information system S (see Table 1) has a confidential attribute d. Let S_d be a new information system that d is hidden from the users of S in

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

Table 1. Information System S

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

Table 2. Information System S'_d

order to protect sensitive data in d. Suppose that the protection is based on τ_1, and its value is given as 5. Also, the following objects reducts are extracted from remote sites.

$r_1 : a_1 \cdot b_1 \rightarrow d_1$
$r_2 : b_1 \cdot c_1 \rightarrow d_1$
$r_3 : a_1 \cdot c_1 \rightarrow d_2$
$r_4 : e_2 \rightarrow d_2$

Let c_v be the total amount of data reconstruction made by r_v, where $v \in r$, and f_v be the frequency of $v \in \alpha(t)$. Then, the weight is given by the $w = \frac{c_v}{f_t}$. For example, the a_1 appears in the conditional part of r_1 and r_3. Therefore, $f_{a_1} = 2$. r_1 reconstructs the confidential attribute value d_1 for object x_1, x_3, x_7 with the confidence of $\frac{2}{3} + \frac{1}{2} + 1 = \frac{13}{6}$. r_3 reconstructs d_2 in x_4 with confidence of 1. So, $c_{a1} = \frac{19}{6}$. Therefore, the weight for a_1, $w_{a1} = \frac{19}{12}$. In the same manner, we compute $w_{b1} = \frac{47}{24}$, $w_{c1} = \frac{11}{8}$, and $w_{e2} = \frac{1}{3}$.

After calculating the weights, we start hiding b_1 from x where $d = d_1$ while the total number of hidden values is less than or equal to 5. So, b_1 is removed from $\{x_1, x_3, x_6, x_7\}$, and we still need to hide one more attribute value.

Now, we calculate weights again with object reducts not containing b_1. So, we have $w_{a1} = \frac{1}{1}$, $w_{c1} = \frac{1}{1}$, and $w_{e2} = \frac{1}{3}$ for the following reducts,

$r_3 : a_1 \cdot c_1 \rightarrow d_2$
$r_4 : e_2 \rightarrow d_2$

If the weights for two or more vs are the same, we randomly select one. So, a_1 is removed from x_8. The new information system after hiding these 5 attribute values are shown in Table 2.

4 Conclusion

Confidential data in an information system can be reconstructed by Chase in DKDS. This article discussed a method that mitigate the risk of confidential data disclosure using object reducts. Reduct was used to identify and remove attribute values that are strongly associated with confidential data.

References

[1] A. Dardzińska and Z. Raś. On rules discovery from incomplete information systems. In *the proceedings of ICDM 03 Workshop on Foundations and New Directions of Data Mining*, November 2003.

[2] S. Im. Privacy aware data management and chase. *Fundamenta Informaticae Journal, Special issue on inteligent information systems*, 2007, will appear.

[3] S. Im and Z. Raś. Ensuring data security against knowledge discovery in distributed information system. In *the proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. Springer, 2005.

[4] S. Im, Z. Raś, and A. Dardzińska. Building a security-aware query answering system based on hierarchical data masking. In *the proceedings of the ICDM Workshop on Computational Intelligence in Data Mining*, November 2005.

[5] Z. Pawlak. *Rough sets-theoretical aspects of reasoning about data*. Kluwer, 1991.

[6] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko. Rough sets. *Commun. ACM*, 38(11):88–95, 1995.

[7] Z. Ras. Reducts-driven query answering for distributed autonomous knowledge systems. *International Journal of Intelligent Systems*, 17(2):113–124, 2002.

[8] Z. Raś and A. Dardzińska. Chase-2: Rule based chase algorithm for information systems of type lambda. In *the proceedings of the Second International Workshop on Active Mining*, October 2005.

[9] Z. Raś and A. Dardzińska. Data security and null value imputation in distributed information systems. In *Advances in Soft Computing*, pages 133–146. Springer-Verlag, 2005.

[10] Z. Raś and A. Dardzińska. Extracting rules from incomplete decision systems: System erid. In *the proceedings of Foundations and Novel Approaches in Data Mining*, pages 143–154, 2005.

[11] Z. Ras, S. Im, and O. Gurdal. Data security versus knowledge loss, invited paper. In *the priceedings of Artificial Intelligence Studies*, volume 3, pages 5–11. Publishing House of University of Podlasie, 2006.

[12] C. B. S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.

[13] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. 11:331–362, 1992.