Data Security and Chase

Zbigniew W. Ras
University of North Carolina at Charlotte, USA

Seunghyun Im
University of Pittsburgh at Johnstown, USA

INTRODUCTION

This article describes requirements and approaches necessary for ensuring data confidentiality in knowledge discovery systems. Data mining systems should provide knowledge extracted from their data which can be used to identify underlying trends and patterns, but the knowledge should not be used to compromise data confidentiality. Confidentiality for sensitive data is achieved, in general, by hiding them from unauthorized users in conventional database systems (e.g., data encryption and/or access control methods can be considered as data hiding). However, it is not sufficient to hide the confidential data in knowledge discovery systems (KDSs) due to Chase (Dardzinska & Ras, 2003a, 2003c). Chase is a missing value prediction tool enhanced by data mining technologies. For example, if an attribute is incomplete in an information system, we can use Chase to approximate the missing values to make the attribute more complete. It is also used to answer user queries containing non-local attributes (Ras & Joshi, 1997). If attributes in queries are locally unknown, we search for their definitions from KDSs and use the results to replace the non-local part of the query.

The problem of Chase with respect to data confidentiality is that it has the ability to reveal hidden data. Sensitive data may be hidden from an information system, for example, by replacing them with null values. However, any user in the KDS who has access to the knowledge base is able to reconstruct hidden data with Chase by treating them as incomplete or missing elements. For example, in a standalone information system with a partially confidential attribute, knowledge extracted from a non-confidential part can be used to reconstruct hidden (confidential) values (Im & Ras, 2005a). When a system is distributed with autonomous sites, knowledge extracted from local or remote information systems (Im, Ras, & Dardzinska, 2005b) may reveal sensitive data to be protected. Clearly, mechanisms that would protect against these vulnerabilities have to be implemented in order to build a security-aware KDS.

BACKGROUND

Security in KDS has been studied in various disciplines such as cryptography, statistics, and data mining. A well-known security problem in the cryptography area is how to acquire global knowledge in a distributed system while exchanging data securely. In other words, the objective is to extract global knowledge without disclosing any data stored in each local site. Proposed solutions are based primarily on the idea of secure multiparty protocol (Yao, 1996) that ensures each participant cannot learn more than its own input and outcome of a public function. Various authors expanded the idea to build secure data mining systems. Clifton and Kantarcioglou (2002) employed the concept to association rule mining for vertically and horizontally partitioned data. Du and Zhan (2002) and Lindell and Pinkas (2000) used the protocol to build a decision tree. They focused on improving the generic secure multiparty protocol for ID3 algorithm (Quinlan, 1993). All these works have a common drawback in that they require expensive encryption and decryption mechanisms. Considering that real-world systems often contain an extremely large amount of data, performance has to be improved before we apply these algorithms. Another research area of data security in data mining is called perturbation. A dataset is perturbed (e.g., noise addition or data swapping) before its release to the public, to minimize disclosure risk of confidential data while maintaining statistical characteristics (e.g., mean and variable). Muralidhar and Sarathy (2003) provided a theoretical basis for data perturbation in terms of data utilization and disclosure risks. In the

KDD area, protection of sensitive rules with minimum side effects has been discussed by several researchers. Oliveira and Zaiane (2002) suggested a solution to protecting sensitive association rules in the form of a "sanitization process" where protection is achieved by hiding selective patterns from the frequent itemsets. There has been another interesting proposal for hiding sensitive association rules. Saygin, Verykios, and Elmagarmid (2002) introduced an interval of minimum support and confidence value to measure the degree of sensitive rules. The interval is specified by the user, and only the rules within the interval are to be removed. In this article, we focus data security algorithms for distributed knowledge sharing systems. Previous and related works concentrated only on a standalone information system or did not consider knowledge sharing techniques to acquire global knowledge.

CHASE ALGORITHM

Suppose that we have an incomplete information system $S = (X,A,V)$ where X is a finite set of objects, A is a finite set of attributes (functions from X into V or relations over X×V), and V is a finite set of attribute values. An incomplete information system is a generalization of an information system introduced by Pawlak (1991). It is understood by having a set of weighted attribute values as a value of an attribute. In other words, multiple values can be assigned as an attribute value for an object if their weights ($\underline{\omega}$) are larger than a minimum threshold value (λ).

Chase requires two input parameters to replaces null or missing values: (1) knowledge and (2) existing data in an information system. The main phase of the Chase algorithm for S 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 (with a weight) suggested by the rules
4. Steps 1-3 are repeated until a fixed point is reached

More specifically, suppose that a knowledge base $KB = \{(t \rightarrow v_c) \in D : c \in In(A)\}$ is a set of all rules extracted from S 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* (Dardzinska & Ras, 2003b) is the algorithm for discovering rules from incomplete information systems and used as a part of Chase. Assuming that $Rs(x_i) \subseteq KB$ is the set of rules in which all of the conditional parts of the rules match with the attribute values in $x_i \in S$, and $d(x_i)$ is a null value, there are three cases for value replacements (Dardzinska & Ras, 2003a, 2003c):

1. $Rs(x_i) = \Phi$. $d(x_i)$ cannot be replaced.
2. $Rs(x_i) = \{ r_1 = [t_1 \rightarrow d_1], r2 = [t_1 \rightarrow d_1], ... r_k = [t_k \rightarrow d_1]\}$. $d(x_i) = d_1$ because every rule implies a single decision attribute value.
3. $Rs(x_i) = \{ r_1 = [t_1 \rightarrow d_1], r2 = [t_1 \rightarrow d_2], ... r_k = [t_k \rightarrow d_k]\}$. Multiple values can be assigned as d.

Clearly, the weight of predicted value is 1 for case 2. For case 3, the weight is calculated based on the confidence and support of rules used for the prediction (Ras & Dardzinska, 2005b). As defined, any predicted value with $\omega > \lambda$ is considered to be valid value.

Chase is an iterative process. An execution of the algorithm generates a new information system, and the execution is repeated until it reaches a state where no additional null value imputation is possible. Figure 1 shows the number of null value imputations at each execution for a sample data set describing the U.S. congressional voting (Hettich & Merz, 1998). In its first run, 163 null values are replaced. In the second run, 10 more attribute values are filled. The execution stops after the third iteration.

SECURITY PROBLEMS

Suppose that an information system S is part of a distributed knowledge discovery system (DKDS). Then, there are two cases in terms of the source of knowledge.

1. Knowledge is extracted from local site S.
2. Knowledge is extracted from remote site $S_i \in DKDS$, *for* $S_i \neq S$.

We examine confidential data disclosure for these two cases.