Data Confidentiality and Chase-Based Knowledge Discovery

Seunghyun Im
University of Pittsburgh at Johnstown, USA

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

INTRODUCTION

This article discusses data security in Knowledge Discovery Systems (KDS). In particular, we presents the problem of confidential data reconstruction by Chase (Dardzinska and Ras, 2003c) in KDS, and discuss protection methods. In conventional database systems, data confidentiality is achieved by hiding sensitive data from unauthorized users (e.g. Data encryption or Access Control). However, hiding is not sufficient in KDS due to Chase. Chase is a generalized null value imputation algorithm that is designed to predict null or missing values, and has many application areas. For example, we can use Chase in a medical decision support system to handle difficult medical situations (e.g. dangerous invasive medical test for the patients who cannot take it). The results derived from the decision support system can help doctors diagnose and treat patients. The data approximated by Chase is particularly reliable because they reflect the actual characteristics of the data set in the information system.

Chase, however, can create data security problems if an information system contains confidential data (Im and Ras, 2005) (Im, 2006). Suppose that an attribute in an information system S contains medical information about patients; some portions of the data are not confidential while others have to be confidential. In this case, part or all of the confidential data in the attribute can be revealed by Chase using knowledge extracted at S. In other words, self-generated rules extracted from non-confidential portions of data can be used to find secret data.

Knowledge is often extracted from remote sites in a Distributed Knowledge Discovery System (DKDS) (Ras, 1994). The key concept of DKDS is to generate global knowledge through knowledge sharing. Each site in DKDS develops knowledge independently, and they are used jointly to produce global knowledge without complex data integrations. Assume that two sites S_1 and S_2 in a DKDS accept the same ontology of their attributes, and they share their knowledge in order to obtain global knowledge, and an attribute of a site S_1 in a DKDS is confidential. The confidential data in S_1 can be hidden by replacing them with null values. However, users at S_1 may treat them as missing data and reconstruct them with Chase using the knowledge extracted from S_2. A distributed medical information system is an example that an attribute is confidential for one information system while the same attribute may not be considered as secret information in another site. These examples show that hiding confidential data from an information system does not guarantee data confidentiality due to Chase, and methods that would protect against these problems are essential to build a security-aware KDS.

BACKGROUND

Data Security and Knowledge Discovery System

Security in KDS has been studied in various disciplines such as cryptography, statistics, and data mining. A well known security problem in 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 a secure data mining systems. Clifton and Kantarcioglou employed the concept to association rule mining for vertically and horizontally partitioned data (Kantarcioglou and Clifton, 2002). Du et al, (Du and Zhan, 2002) and Lindell et al, (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 that they require expensive encryption and decryption mechanisms. Considering that real world system often contain 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. 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 (Muralidhar and Sarathy, 2003) provided a theoretical basis for data perturbation in terms of data utilization and disclosure risks. In KDD area, protection of sensitive rules with minimum side effect has been discussed by several researchers. In (Oliveira & Zaiane, 2002), authors suggested a solution to protecting sensitive association rules in the form of "sanitization process" where protection is achieved by hiding selective patterns from the frequent itemsets. There has been another interesting proposal (Saygin & Verykios & Elmagarmid, 2002) for hiding sensitive association rules. They 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 on data security problems in distributed knowledge sharing systems. Related works concentrated only on a standalone information system, or did not consider knowledge sharing techniques to acquire global knowledge.

Chase Algorithm

The overall steps for Chase algorithm is the following.

1. Identify all incomplete attribute values in S.
2. Extract rules from S describing these incomplete attribute values.
3. Null values in S are replaced by values (with their weights) suggested by the rules.
4. Steps 1-3 are repeated until a fixed point is reached.

More specifically, suppose that we have an incomplete information system $S = (X,A,V)$ where X is a finite set of object, A is a finite set of attributes, and V is a finite set of their values. 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 with their weights (w). Assuming that a knowledge base $KB = \{(t \rightarrow v_c) \in D : c \in In(A)\}$ is a set of all classification rules extracted from S by $ERID(S, \lambda_1 \lambda_2)$, where $In(A)$ is the set of incomplete attributes in S, v_c is a value of attribute c, and λ_1, λ_2 are thresholds for minimum support and minimum confidence, correspondingly. *ERID* (Dardzinska and Ras, 2003b) is the algorithm for discovering rules from incomplete information systems, which can handle weighted attribute values. Assuming further that $Rs(x_i) \subseteq KB$ is the set of rules that all of the conditional part of the rules match with the attribute values in $x_i \in S$, and $d(x_i)$ is a null value, then, there are three cases for null value imputations (Dardzinska and Ras, 2003a, 2003c):

1. $Rs(x_i) = \Phi$. $d(x_i)$ cannot be replaced.
2. $Rs(x_i) = \{ r_1 = [t_1 \rightarrow d_1], r2 = [t_1 \rightarrow d_1], \ldots r_k = [t_k \rightarrow d_i]\}$. $d(x_i) = d_1$ because every rule predicts a single decision attribute value.
3. $Rs(x_i) = \{ r_1 = [t1 \rightarrow d_1], r2 = [t_1 \rightarrow d_2], \ldots r_k = [t_k \rightarrow d_k]\}$. Multiple values can replace $d(x_i)$.

Clearly, the weights of predicted values, which represent the strength of prediction, are 1 for case 2. For case 3, weight is calculated based on the confidence and support of rules used by Chase (Ras and Dardzinska, 2005b). Chase is an iterative process. An execution of the algorithm for all attributes in S typically generates a new information system, and the execution is repeated until it reaches a state where no improvement is achieved.