Reconstruction-based Classification Rule Hiding through Controlled Data Modification

Aliki Katsarou, Aris Gkoulalas-Divanis, and Vassilios S. Verykios

Abstract In this paper, we propose a reconstruction–based approach to classification rule hiding in categorical datasets. The proposed methodology modifies transactions supporting both sensitive and nonsensitive classification rules in the original dataset and then uses the supporting transactions of the nonsensitive rules to produce its sanitized counterpart. To further investigate some interesting properties of this methodology, we explore three variations of the main technique which differ in the way they select and sanitize transactions supporting sensitive rules. Finally, through extensive experimental evaluation, we demonstrate the effectiveness of the proposed algorithms towards effectively shielding the sensitive knowledge.

1 Introduction

Recent advances in information technology has provided the means to cost–efficient data collection and analysis. Nowadays, organizations collect vast amounts of data on a daily basis to effectively conduct their business. The collected data is usually organized and stored in a data warehouse to allow for its efficient retrieval and manipulation, when necessary. Apart from accommodating the everyday needs of the organizations, the stored data is also a valuable source of knowledge. Modern organizations are usually willing to integrate and mine their data collectively with other organizations in order to derive global knowledge patterns. However, the collective mining of data poses a threat to privacy, as sensitive knowledge patterns can be inferred from the data. The disclosure of sensitive patterns (e.g., trade secrets) to

Aliki Katsarou
Department of Management, London School of Economics and Political Science, Houghton Str, London WC2A 2AE, U.K., e-mail: A.Katsarou1@lse.ac.uk

Aris Gkoulalas-Divanis · Vassilios S. Verykios
Department of Computer & Communication Engineering, University of Thessaly, 37 Glavani - 28th October Str, Volos GR–38221, Greece, e-mail: {arisgd,verykios}@inf.uth.gr

untrusted entities, such as business competitors, can be deemed catastrophical for the data owner. Therefore, privacy preserving data mining approaches are essential to hide the sensitive knowledge prior to the sharing of the data.

In this paper, we propose an efficient approach for the hiding of sensitive classification rules in categorical datasets. Our proposed methodology uses a rule–based classifier to derive the classification rules for the original dataset, among which there exist rules that are considered as sensitive from the owner's perspective. Given the whole set of classification rules, the algorithm identifies the transactions of the dataset that support both sensitive and nonsensitive rules and modifies them in such a way that they no longer support the sensitive rules. The modification is focused on the attribute–value pair of the transaction that causes the least side–effects to the dataset. Finally, the transactions that support the nonsensitive rules are used to generate the sanitized version of the dataset that can be safely shared. The proposed methodology is simple, time efficient and with very satisfying results.

The remainder of this paper is organized as follows. In Sect. 2, we formalize the problem by providing some basic definitions that allow us to introduce the problem statement. Sect. 3 demonstrates the solution methodology that we followed and presents our basic classification rule hiding algorithm, along with three complementary implementations. In Sect. 4, we present the experimental evaluation of the proposed algorithms. Finally, Sect. 5 presents the related work, and Sect. 6 concludes this paper.

2 Problem Formulation

In this section we first provide some basic definitions that are necessary for the understanding of the proposed methodology, and then we introduce the problem statement.

2.1 Basic Definitions

Definition 1. (**Dataset**) Let a dataset D be a 4–tuple $\{T, A, V, f\}$, where

- T is a nonempty finite set consisting of N transactions.
- A is a nonempty finite set consisting of M attributes, such that any attribute $A_m \in A$ has a domain of supported values V_{A_m}. Among the M attributes, one attribute C is designated as the class for dataset D and consists of a nonempty finite set of class labels.
- V is a nonempty finite set of values for all attributes, s.t. $V_{A_m} \subseteq V$ and $\bigcup_{V_{A_m}} = V$.
- f is a function such that $f : V \times A \longrightarrow V_{A_m}$, i.e. it assigns a value to an attribute of a given record.

Definition 2. (**Classification Rule**) A classification rule $R_i \in R$, extracted from a dataset D, is a sentence of the form $(A_1 = f(A_1)) \wedge (A_2 = f(A_2)) \wedge (A_3 = f(A_3)) \wedge \ldots \wedge (A_m = f(A_m)) \longrightarrow (C = c)$, where $c \in C$ and $A_1, A_2, A_3, \ldots A_m \neq C$.

Having defined the notion of a classification rule, we state that a transaction *supports* a classification rule if all the attribute–value pairs $(A_m = f(A_m))$ of the classification rule (including the class label) appear in the transaction. Furthermore, we define the (supporting) *size* of a classification rule R_i, and denote it as $|R_i|$, to be the number of transactions from D that support the rule. Pertinent to the definition of a classification rule is the definition of a classification problem.

Corollary 1. ***(Classification Problem)*** *A classification problem over dataset D is the task of learning a target function $F : D \longrightarrow C$ that maps each transaction in D to one of the predefined class labels (a.k.a. categories) $c \in C$.*

2.2 Problem Statement

Given a dataset D, a class attribute C, a set of classification rules R over D, as well as a set of sensitive rules $R_S \subset R$, we want to find a dataset D' such that when mining D' for classification rules using the same parameters as those used in the mining of D, only the (nonsensitive) rules in $R - R_S$ can be derived.

3 Solution Methodology

In this section we elaborate on the proposed methodology for classification rule hiding in categorical datasets. First, we present the main reconstruction–based approach[1] that we developed to solve this problem. Then, we introduce three complementary implementations that differ in the way they select and sanitize the transactions supporting sensitive rules.

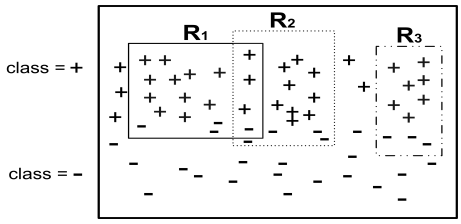

Fig. 1 Rule generation by using a sequential covering algorithm like RIPPER [3]. $\mathbf{R_1}$ and $\mathbf{R_3}$ represent two regions covered by nonsensitive rules, while $\mathbf{R_2}$ is a region covered by a sensitive classification rule.

[1] Reconstruction–based approaches for knowledge hiding, generate the sanitized dataset D' from scratch instead of directly modifying the transactions of the original dataset D.

3.1 The Least Supported Attribute Algorithm

Our proposed methodology, called the Least Supported Attribute (LSA) modification algorithm, uses the nonsensitive rules that are mined from the original database D to reconstruct its sanitized counterpart D'. As a first step, LSA identifies the supporting transactions for each nonsensitive rule $R_i \in R$ in the original dataset. Then, among the supporting transactions for this rule, it selects the ones that are also supporting at least one sensitive rule. Fig. 1 provides an example of such a scenario, generated by a sequential covering algorithm like RIPPER [3] to discriminate between the positive and the negative examples of a two–class classification problem. As one can notice, the rule generation process allows the transactions of the database to support more than one rule, as is the case for rules R_1 and R_2[2].

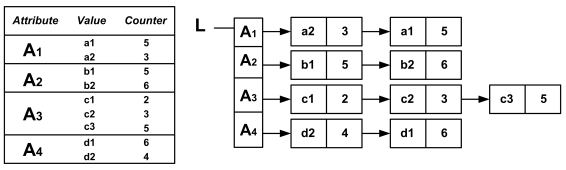

Fig. 2 An efficient data structure **L** used by LSA for the selection of the attribute–value pair that will be modified in a transaction to facilitate knowledge hiding.

For each transaction that supports both a nonsensitive and a sensitive classification rule, LSA modifies it appropriately so that it no longer supports the sensitive rule. The proposed modification affects only one attribute–value pair of the transaction, which is selected to be the one having the least support in D, among the attribute–value pairs of the supported sensitive rule. Furthermore, to minimize the side–effects in the sanitized outcome, the new value that will be assigned to the selected attribute will be the one that is supported the least by the transactions of the original dataset (different from the current value in the sensitive rule). By altering the value of the attribute to equal the one that is least supported in D, we manage to moderate the increment of the support of some attribute–value pairs and thus to minimize the probability of producing rules in D' that were nonexistent in D. To make this possible, we employ a data structure that keeps track of the number of times that each attribute–value pair is met in the transactions of D, as shown in Fig. 2. As one can notice, the proposed data structure **L** is a list of lists, the later of which holds the attribute–value pairs sorted in a descending order of support in dataset D.

Example An example will allow us to better demonstrate how this operation works. Consider the database of Table 1 that consists of four attributes A_1, A_2, A_3, A_4 and a class attribute C with labels 0 and 1. We assume that in the given dataset, the counters for the various attribute–value pairs (as updated based on the support of the attributes–values in the dataset) are provided in Table 1. Let $T_1 = (a1, b1, c1, d1, 1), T_2 = (a1, b2, c2, d1, 1), T_3 = (a1, b1, c2, d1, 1)$ and $T_4 = (a1, b2, c3, d1, 1)$ be four transactions that support the rule $(A_1 = a1) \wedge (A_4 = d1) \longrightarrow (C = 1)$. Among the four transactions, T_4 also supports the sensitive rule $(A_2 = b2) \wedge (A_3 = c3) \longrightarrow (C = 1)$. In order to modify T_4 in such a way that it no longer supports the sensitive rule, LSA will choose to replace

[2] However, due to the rule ordering scheme that is enforced by the rule generation algorithm, only the first rule in the rule set that is supported by a new transaction is used for its classification.

Table 1 An example of using the data structure L on decision making.

Attribute	Value	Counter
A_1	a1	5
	a2	3
A_2	b1	5
	b2	6
A_3	c1	2
	c2	3
	c3	5
A_4	d1	6
	d2	4

Algorithm 1 The Least Supported Attribute (LSA) algorithm.

Input: Original dataset D, Sensitive rules R_S, Nonsensitive rules $R - R_S$.
Output: Sanitized dataset D'.

$D' \leftarrow \varnothing$, $\mathbf{L} \leftarrow$ all counters initialized to zero.
foreach nonsensitive rule $R_i \in R - R_S$ **do**
  $S \leftarrow$ all transactions in D that support R_i.
  foreach transaction $T_n \in S$ **do**
    foreach attribute $A_m \in T_n$ **do**
      update_list (**L**, A_m) ▷ Increase the counter of the appropriate value of attribute A_m.
  foreach transaction $T_n \in S$ **do**
    if T_n supports a sensitive rule $R_j \in R_S$ **then**
      select the attribute–value from R_j that has the minimum counter in **L** and does not appear in R_i.
      replace the value of this attribute in T_n with the one with the minimum counter in **L**.
      update_list (**L**, selected attribute)
  S $\leftarrow$ **S** $\cup (S, |S|)$
foreach pair $(S, |S|) \in$ **S** **do**
  Add $|S| \times N/|\mathbf{S}|$ transactions from S to D' in a round–robin fashion.

the value of attribute A_3 in T_4, since $(A_3 = c3)$ is less supported in D than $(A_2 = b2)$. The new value of A_3 in T_4 will be the one from **L** that is minimally supported in the dataset; that is $c1$.

The rationale behind the modification of the transactions that support sensitive rules in D is as follows. In LSA (as in most of the currently proposed methodologies for classification rule hiding), we consider that the sanitized dataset D' will consist of the same number of transactions N as the ones of dataset D. However, since the sensitive rules cover a set of transactions that are not covered by the nonsensitive rules, and since D' is formulated only from the transactions supporting the nonsensitive rules, it is reasonable to expect that the transactions that support all the nonsensitive rules in D are less than N. Thus, LSA uses the transactions supporting the nonsensitive rules in a round–robin fashion in order to construct the sanitized outcome. However, we need to mention that LSA ensures that the representation of the nonsensitive rules in D' is proportional to their representation in D. Algorithm 1 provides the details of our implementation.

3.2 Three Complementary Implementations of LSA

To experimentally investigate some of the properties of LSA, we implemented three variations of this algorithm. The first variation, called Naïve LSA (NLSA), differs

from LSA in the way it selects the attribute–value pair of a transaction that supports a sensitive rule to facilitate knowledge hiding. Specifically, when a transaction is found to support both a nonsensitive and a sensitive rule, NLSA randomly selects an attribute–value pair of the supported sensitive rule and modifies the value of this attribute based on the counters in **L**.

The second variation, called TR-A (Transaction Removal for All transactions supporting sensitive rules) discards, instead of modifying, all the transactions that support both a nonsensitive and a sensitive rule, while the third variation, called TR-S (Transaction Removal for Selected transactions) is a combination of LSA and TR-A. For every nonsensitive rule, TR-S retrieves its supporting transactions in D. If some of these transactions also support a sensitive rule, then (i) if the number of transactions supporting the nonsensitive rule is greater than the number of instances that have to be generated for this rule in the sanitized dataset D', then any additional transactions from the ones supporting the sensitive rule are removed, and (ii) the remaining transactions that also support the sensitive rule are modified as LSA dictates. Otherwise, the algorithm operates the same way as LSA.

Table 2 The characteristics of the two datasets used for experimentation.

Dataset	# records	# attributes	# rules (prune)	# rules (no prune)
Mushroom	8,124	22	9	8
Vote	435	16	4	10

4 Experimental Evaluation

To experimentally evaluate our proposed algorithms, we implemented the algorithms H(half), H(all), GR and LC, proposed in [5, 6], and then evaluated all the eight methodologies (TR-A, TR-S, NLSA, LSA, H(half), H(all), GR and LC) along two principal dimensions: (i) the number of side–effects (in terms of lost rules, ghost rules and disclosed sensitive rules) caused to the original dataset due to the hiding of the sensitive knowledge, and (ii) the scalability of the approaches under different hiding scenarios. A lost rule is any nonsensitive rule from D that does not appear in D', while a ghost rule is any rule that did not exist in D but is produced for D'. The properties of the categorical datasets that we used to conduct our experiments are shown in Table 2. Both datasets were taken from the UCI machine learning repository (available at http://archive.ics.uci.edu/ml).

Figs. 3–5 present the observed results for the two datasets when a certain number of sensitive rules are hidden. In all cases we tested, both the initial and the final classification rule sets were produced with RIPPER [3], while experiments were conducted with and without rule pruning. As one can notice, LSA and its variations typically achieve better results than their competitors in terms of side–effects caused

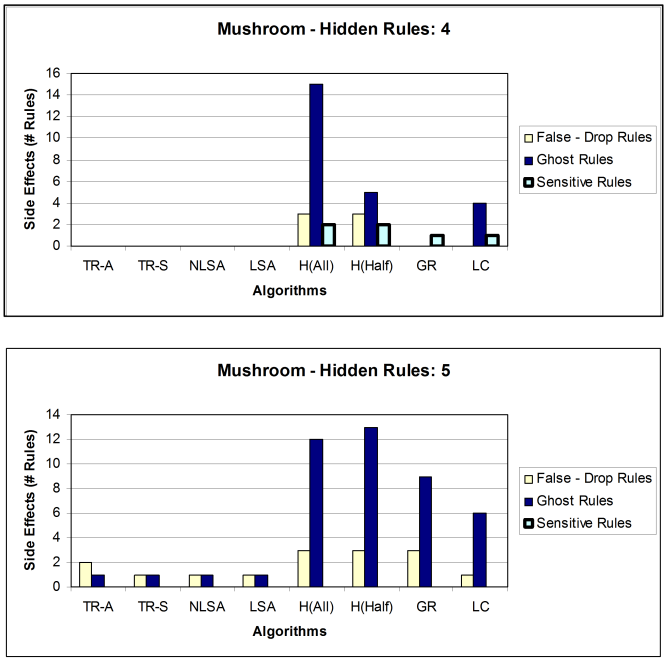

Fig. 3 Experimental results for Mushroom — rules extracted with pruning.

to D' by the sanitization process. Furthermore, the proposed algorithms achieved to appropriately cover–up the sensitive rules in D', in all tested settings.

In terms of scalability, Fig. 6 indicates that the time complexity of the proposed algorithms is kept low, even when the number of rules to be hidden increases. This outcome suggests that the proposed approaches are suitable to facilitate knowledge hiding in very large datasets.

5 Related Work

In the last decade, there has been a lot of active research in the field of privacy preserving data sharing. Vaidya et al. [8] tackle the problem of multiparty data sharing by proposing a distributed privacy preserving version of ID3. The proposed strategy assumes a vertical partitioning of the data where every attribute (including the class) has to be known only by one party. A distributed version of ID3 that is suitable in the case of a horizontal data partitioning scheme, can be found in [9].

Chang and Moskowitz [1] were the first to address the inference problem caused by the downgrading of the data in the context of classification rules. Through a blocking technique, called parsimonious downgrading, the authors block the infer-

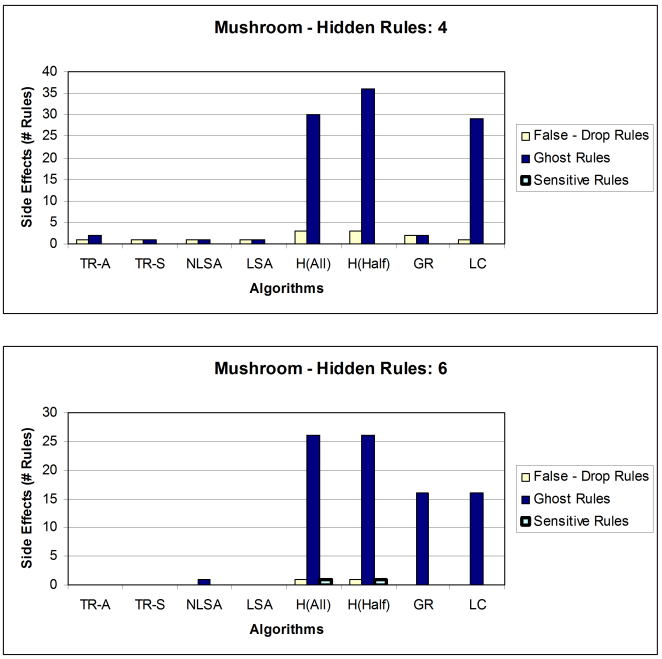

Fig. 4 Experimental results for Mushroom — rules extracted without pruning.

ence channels that lead to the identification of the sensitive rules by selectively sanitizing transactions so that missing values appear in the released dataset. This has as an immediate consequence the lowering of the confidence of an attacker regarding the holding of the sensitive rules.

Chen and Liu [2] present a random rotation perturbation technique to privacy preserving data classification. The proposed methodology preserves the multi–dimensional geometric characteristics of the dataset with respect to task–specific information. As an effect, in the sanitized dataset the sensitive knowledge is protected against disclosure, while the utility of the data is preserved to a large extend.

Natwichai et al. [6] propose a reconstruction algorithm for classification rules hiding. The proposed algorithm uses the nonsensitive rules to build a decision tree from which the sanitized dataset will be generated. To produce the sanitized dataset, the algorithm traverses the paths of the decision tree that correspond to the same rule and repeatedly generates transactions that support this rule in the sanitized outcome. In [6] the decision tree is build based on the gain ratio of the various attributes. An alternative approach that builds the decision tree by using the least common attribute measure is presented in [5]. Finally, in [7] a data reduction approach is proposed, which is suitable for the hiding of a specific type of classification rules, known as canonical associative classification rules.

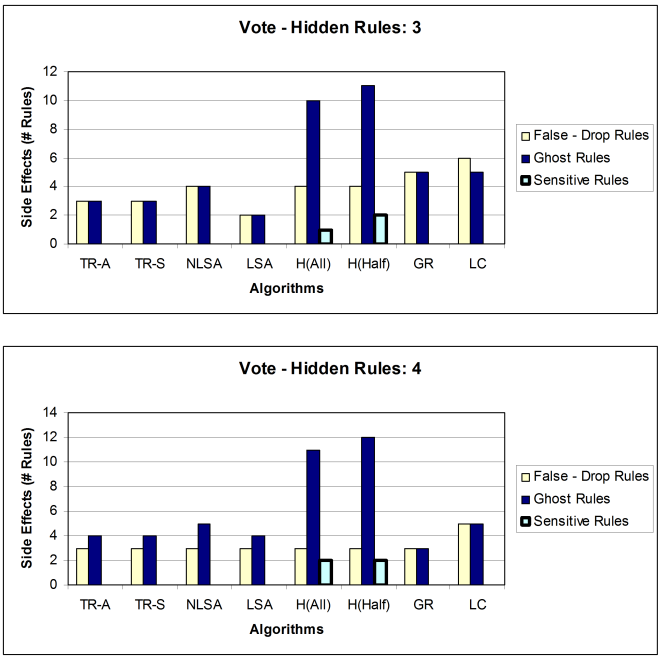

Fig. 5 Experimental results for Vote — rules extracted without pruning.

Finally, Islam and Brankovic [4] present a noise addition framework for the hiding of sensitive classification rules. The suggested framework achieves to protect the sensitive patterns from disclosure, while preserving all the nonsensitive statistical information of the dataset.

6 Conclusions

In this paper, we presented a novel approach to classification rules hiding that guarantees the privacy of the sensitive knowledge, while minimizing the side–effects introduced by the sanitization process. Through a series of experiments, we demonstrated that our approach yields good results in terms of side–effects, while it keeps the computational complexity within reasonable bounds.

Acknowledgements We would like to thank Prof. William W. Cohen from the Carnegie Mellon University for providing us the implementation of the RIPPER algorithm.

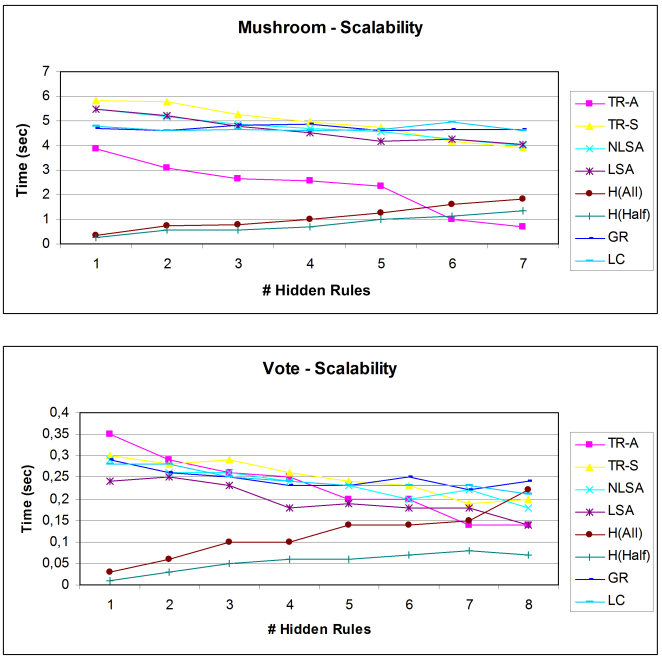

Fig. 6 The scalability of the tested algorithms in the two datasets.

References

1. Chang, L., Moskowitz, I.S.: Parsimonious downgrading and decision trees applied to the inference problem. In: Proceedings 1998 Workshop on New Security Paradigms, pp. 82–89 (1998)
2. Chen, K., Liu, L.: Privacy preserving data classification with rotation perturbation. In: Proceedings 5th IEEE International Conference on Data Mining, pp. 589–592 (2005)
3. Cohen, W.W.: Fast effective rule induction. In: Proceedings 12th International Conference on Machine Learning, pp. 115–123 (1995)
4. Islam, M.Z., Brankovic, L.: A framework for privacy preserving classification in data mining. In: Proceedings 22nd Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation, pp. 163–168 (2004)
5. Natwichai, J., Li, X., Orlowska, M.: Hiding classification rules for data sharing with privacy preservation. In: Proceedings 7th International Conference on Data Warehousing and Knowledge Discovery, pp. 468–467 (2005)
6. Natwichai, J., Li, X., Orlowska, M.E.: A reconstruction-based algorithm for classification rules hiding. In: Proceedings 17th Australasian Database Conference, pp. 49–58 (2006)
7. Natwichai, J., Sun, X., Li, X.: Data reduction approach for sensitive associative classification rule hiding. In: Proceedings 19th Australian Conference on Databases (2007)
8. Vaidya, J., Clifton, C., Kantarcioglu, M., Patterson, A.S.: Privacy-preserving decision trees over vertically partitioned data. ACM Trans. Knowl. Discov. Data **2**(3) (2008)
9. Xiao, M.J., Huang, L.S., Luo, Y.L., Shen, H.: Privacy preserving ID3 algorithm over horizontally partitioned data. In: Proceedings 6th International Conference on Parallel and Distributed Computing Applications and Technologies, pp. 239–243 (2005)