



NRC Publications Archive Archives des publications du CNRC

Identifying and Preventing Data Leakage in Multi-relational Classification

Guo, Hongyu; Viktor, Herna L.; Paquet, Eric

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version
acceptée du manuscrit ou la version de l'éditeur.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien
DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1109/ICDMW.2010.33>

*2010 IEEE International Conference on Data Mining Workshops, pp. 458-465,
2010-12-17*

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=886ff751-891b-46e4-aead-bd81dbf5df2e>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=886ff751-891b-46e4-aead-bd81dbf5df2e>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the
first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la
première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez
pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



Identifying and Preventing Data Leakage in Multi-relational Classification

Hongyu Guo*, Herna L. Viktor[†], and Eric Paquet*[†]

**Institute for Information Technology, National Research Council Canada*

Email: {hongyu.guo,eric.paquet}@nrc-cnrc.gc.ca

[†]School of Information Technology and Engineering, University of Ottawa, Canada

Email: hlviktor@site.uottawa.ca

Abstract—Relational database mining, where data are mined across multiple relations, is increasingly commonplace. When considering a complex database schema, it becomes difficult to identify all possible relationships between attributes from the different relations. That is, seemingly harmless attributes may be linked to confidential information, leading to data leaks when building a model. In this way, we are at risk of disclosing unwanted knowledge when publishing the results of a data mining exercise. For instance, consider a financial database classification task to determine whether a loan is considered to be high risk. Suppose that we are aware that the database contains another confidential attribute, such as income level, which should not be divulged. In order to prevent potential privacy leakage, one may thus choose to eliminate, or distort, the income level from the database. However, even after distortion, a learning model against the modified database may accurately determine the income level values. It follows that the database is still unsafe and may be compromised. This paper demonstrates this potential for privacy leakage in multi-relational classification and illustrates how such potential leaks may be detected. We propose a method to generate a ranked list of subschemas which maintains the predictive performance on the class attribute, while limiting the disclosure risk, and predictive accuracy, of confidential attributes. We illustrate our method against a financial database.

I. INTRODUCTION

The number of commercial relational databases, which store vast amounts of real-world data, is growing exponentially. Increasingly, concerns regarding potential data privacy breaches are emerging. One of the main issues organizations face is identifying, avoiding or limiting the inference of attribute values. For relational databases, it is difficult to be able to identify all attribute interrelationships, due to the complexities of the database schemas that contain multiple relations.

Would it, then, not be enough to eliminate, distort, or limit access to confidential data? Our analysis shows that this is not the case. We show that, when following such an approach, there may still be a disclosure of data during multi-relational classification. We demonstrate that, through using publicly available information and insider knowledge, one may still be able to inject an attack which accurately predicts the values of confidential, or so-called sensitive, attributes.

To address the above-mentioned potential privacy breaches, we propose a method which limits the predictive

accuracy against previously identified confidential attributes. To this end, the paper introduces a method which generates a ranked list of subschemas of a database. Each subschema has a different balance between the two prediction accuracies, namely the target variables and the confidential attributes. The objective here is to create subschemas which maintain the predictive performance on the target class label, but limit the prediction accuracy on confidential attributes. We show the effectiveness of our strategy against a financial database.

This paper is organized as follows. Section II presents the background and problem formulation. Next, in Section III, we introduce our method for privacy protection. This is followed, in Section IV, by the discussion of our experimental studies. Finally, Section V concludes the paper and outlines our future work.

II. BACKGROUND AND PROBLEM FORMULATION

Privacy leakage protection in data mining strives to prevent revealing sensitive data without invalidating the data mining results [1], [2], [3]. Often data anonymization operations are applied [4].

Current approaches for privacy preserving data mining aim at distorting individual data values, but enabling reconstruction of the original distributions of the values of the confidential attributes [1], [5], [6], [7], [8]. For example, the k-anonymity model [7] and the perturbing method [8] are two techniques for achieving this goal.

Recent research deals with correlation and association between attributes to prevent the inference of sensitive data [9], [10], [11], [12], [13], [14]. For example, Association Rule Hiding (ARH) methods sanitize datasets in order to prevent disclosing sensitive association rules from the modified data [9], [12], [14]. Zhu and Du [13] incorporate k-anonymity into the association rule hiding process. Tao et al. propose a method to distort data in order to hide correlations between non-sensitive attributes [10]. Data leakage prevention in releasing multiple views from databases has also been intensively studied [15]. In addition, privacy leakage in multi-party environment has been investigated [16].

Our method does not distort the original data in order to protect sensitive information. Rather, we select a subset of data from the original database. The selected attributes are able to maintain high accuracies against the target variables,

while lowering the predictive capability against confidential attributes, thus alleviating the risk of probabilistic (belief) attacks of sensitive attributes [4]. This stands in contrast to the above-mentioned anonymization techniques, such as generalization, suppression, anatomization, permutation, and perturbation.

Furthermore, it follows that our approach is not tied to a specific data mining technique, since there is no need to learn from masked data.

A. Multirelational Database Classification

In this paper, a relational database \mathcal{R} is described by a set of tables $\{R_1, \dots, R_n\}$. Each table R_i consists of a set of tuples T_{R_i} , a primary key, and a set of foreign keys. Foreign key attributes link to primary keys of other tables. This type of linkage defines a *join* between the two tables involved. A set of joins with n tables $R_1 \bowtie \dots \bowtie R_n$ describes a join path, where the *length* of it is defined as the *number of joins* it contains.

A multirelational classification task involves a relational database \mathcal{R} which consists of a target relation R_t , a set of background relations $\{R_b\}$, and a set of joins $\{J\}$ [17]. Each tuple in this target relation, i.e. $x \in T_{R_t}$, is associated with a class label which belongs to Y (target classes). Typically, the task here is to find a function $F(x)$ which maps each tuple x from the target table R_t to the category Y . That is,

$$Y = F(x, R_t, \{R_b\}, \{J\}), x \in T_{R_t}$$

B. Privacy Leakage in Multirelational Classification

We formalize the problem of privacy leakage in multirelational classification as follows.

Given is a relational database $\mathcal{R} = (R_t, \{R_b\})$ with target attribute Y in R_t . Together with this information, we have an attribute C that is to be protected. $C \in \{R_b\}^1$, and C has either to be removed from the database or the values have to be distorted. However, C may potentially be predicted using \mathcal{R} with high accuracy.

Our objective is to find a subschema that accurately predict the target variable, but yields a poor prediction for the confidential attribute. To this end, we generate a ranked list of subschemas of \mathcal{R} . Each subschema $\mathcal{R}' (\mathcal{R}' \subset \mathcal{R})$ predict the target variable Y with high accuracy, but has limited predictive capability against the confidential attribute C . To this end, we construct a number of different subschemas of \mathcal{R} . For each subschema \mathcal{R}' , we determine how well it predicts the target variable Y and we calculate its degree of sensitivity in terms of predicting the confidential attribute C . Finally, we rank the subschemas based on this information. In the next sections, we discuss our approach.

¹in cases where both Y and C reside in the R_t table, one may create two views from R_t which separates the two variables into two relations

III. TARGET SHIFTING MULTIRELATIONAL CLASSIFICATION

Our Target Shifting Multirelational Classification (TSMC) approach aims to prevent the prediction of confidential attributes, while maintain the predictive performance of the target variable. To this end, as described in Algorithm 1, the TSMC method consists of the following four (4) steps.

Firstly, the attributes that are correlated with a confidential attribute, are identified. Note that, following Tao et. al, [10] we here use the term correlation to denote the associations, interrelationships or links between attributes in our database. It follows that such correlated attributes may reside in relations other than the confidential attribute. Secondly, based on the correlation computed from the first step, the degrees of sensitivity for different subschemas of the database are calculated. Next, subschemas consisting of different tables of the database are constructed. Finally, for each subschema, its performance when predicting the target variable, along with its privacy sensitivity level, is computed. As a result, a ranked list of subschemas is provided. These four steps are discussed next.

A. Identify Correlated Attributes Across Interlinked Tables

In the first phase of the TSMC method, the aim is to identify the attributes that are correlated with the confidential attribute C . That is, this step aims to find attributes which may be used to predict C . To find correlated attributes, one needs to compute the correlation between attribute sets across the multiple tables of the database. To address the above issue, the TSMC method learns a set of high quality rules against the confidential attribute C . That is, it searches attributes (attribute sets) across multiple tables to find a set of rules which predict the values of C well. To this end, we employ the CrossMine algorithm, which is able to accurately and efficiently construct a set of conjunctive rules using features across multiple relations in a database [18]. For example, a rule may have the following form:

```
Loan.status = good  $\leftarrow$  (Loan.account-id  $\bowtie$ 
Account.account-id)
(Account.frequency = monthly) (Account.client-id
 $\bowtie$  Client.client-id)
(Client.birth date < 01/01/1970)
```

This rule says a monthly loan where the borrower was born before 1970 is classified as being of low risk. In this rule, the attributes frequency in the Account table and birth date from the Client table work together to predict the loan status in the Loan table. In other words, such a rule is able to capture the interplay between attributes across multiple tables.

In summary, we use CrossMine to learn which other attributes are correlated, or have a relationship with, the confidential attributes.

That is, our approach uses a set of rules as created by this classifier, to identify the most relevant attributes. It

follows that an implicit assumption is that an informative classification model is constructed by CrossMine.

The TSMC method ranks the constructed rules based on their tuple coverages and then selects the first n rules which cover more than 50% of the training tuples. That is, the algorithm considers the set of rules which can predict the confidential attributes better than random guessing.

Algorithm 1 The TSMC Approach

Input: a relational database $\mathcal{R} = (R_t, \{R_b\})$; $Y \in R_t$ is the target variable and $C \in \{R_b\}$ is a confidential attribute

Output: a ranked list of subschemas of \mathcal{R} . Each subschema \mathcal{R}' can predict Y with high accuracy, but has limited predictive capability against C

- 1: using C (instead of Y) as the classification target, construct a set of high quality rules using \mathcal{R}
 - 2: derive the subschema privacy sensitivity \mathcal{P} from the set of rules learned
 - 3: convert schema \mathcal{R} into undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, there R_t and R_b as nodes \mathcal{V} and joins J as edges \mathcal{E}
 - 4: construct a set of subgraphs from $\mathcal{G} \Rightarrow$ subgraphs set $\{\mathcal{G}_{s_1}, \dots, \mathcal{G}_{s_n}\}$
 - 5: **for** each subset $\in \{\mathcal{G}_{s_1}, \dots, \mathcal{G}_{s_n}\}$ **do**
 - 6: compute the \mathcal{PI} (with respect to the target variable Y) of the subgraph subset (namely, subschema $\mathcal{R}' \in \mathcal{R}$)
 - 7: **end for**
 - 8: rank the $\{\mathcal{R}'\}$ based on their \mathcal{PI} values
 - 9: return the ranked $\{\mathcal{R}'\}$
-

B. Assign Privacy Sensitivity to Subschemas

After obtaining a set of rules which find the other attributes that are relevant when learning the confidential attributes, the TSMC method estimates the predictive capability of the subschemas.

Consider the following rule which predicts an order's payment type in the Order table with 70% accuracy.

```
Order.payment type = house payment ←
(Order.amount >= 1833)
(Order.account_id ⋈ Disposition.account_id)
(Disposition.client_id ⋈ Client.client_id)
(Client.birth_date <= 31/10/1937)
```

Thus, if our published subschema includes tables Order, Disposition, and Client, one may use this subschema to build a classification model and then to determine an order's payment type with 70% accuracy.

The previous observation suggests that, by using the set of high quality rules learned, we can estimate the degree of sensitivity (denoted as \mathcal{P}) of a subschema, in terms of its predictive capability on the confidential attribute. \mathcal{P} for a subschema is calculated as follows. Firstly, we identify the set of m ($m \subseteq n$) rules whose conjunctive features

are covered by some *or* all of the tables in the subschema. Next, we sum the number of tuples covered by each one of these rules (denoted as NC_i). Finally, we divide the sum by the total number of tuples (noted as NS) containing the confidential attributes. Formally, the value \mathcal{P} for a subschema is calculated as follows.

$$\mathcal{P} = \frac{\sum_{i=1}^m (NC_i)}{NS} \quad (1)$$

For example, as the above-mentioned rule covers 70% of the total number of tuples, we will assign 0.7 as the degree of sensitivity for the subschema {Order, Disposition, Client}. Note that, there may be another rule against this subschema, such as

```
Order.payment type = house payment ←
(Order.amount >= 1833)
(Order.account_id ⋈ Disposition.account_id)
(Disposition.type = owner)
```

In this case, the sensitivity of subschema {Loan, Account, Client} should be calculated using all tuples covered by the two rules. This is due to the fact that CrossMine is a sequential covering method. The next step of the TSMC method, as described in Algorithm 1, is to construct a set of subschemas and evaluate their contained information, in terms of predicting both the target variable Y and the confidential attribute C .

C. Subschema Evaluation

The TSMC method adopts the subschema construction approach presented by Guo and Viktor in [17]. That is, in the TSMC method, each subschema consists of a set of subgraphs, each corresponds to a unique join path in the relational database. The subgraph construction procedure is discussed, next.

1) *Subgraph Construction:* The subgraph construction process aims to build a set of subgraphs given a relational database schema where each subgraph corresponds to a unique join path. The construction process initially converts the relational database schema into an undirected graph, using the relations as the nodes and the joins as edges.

Two heuristic constraints are imposed on each constructed subgraph. The first is that each subgraph must start at the target relation. This constraint ensures that each subgraph contains the target relation and, therefore, is able to construct a classification model. The second constraint is for relations to be unique for each candidate subgraph. Typically in a relational domain, the number of possible join paths given a large number of relations is very large, making it too costly to exhaustively search all join paths [18]. Also, join paths with many relations may decrease the number of entities related to the target tuples. Therefore, this restriction was introduced for a trade off between accuracy and efficiency.

Using these constraints, the subgraph construction process proceeds initially by finding unique join paths with two

relations, i.e. join paths with a length of one. These join paths are progressively lengthened, one relation at a time. The length of the join path is introduced as the stopping criterion. The construction process prefers subgraphs with shorter length. The reason for preferring shorter subgraphs is that semantic links with too many joins are usually very weak in a relational database [18]. Thus the algorithm specifies a maximum length for the join paths. When this number is reached, the join path extraction process terminates.

After constructing a set of subgraphs, the TSMC algorithm is then able to form different subschemas and evaluate their predictive capabilities on both the target and confidential attributes.

2) *SubInfo of Subgraph*: Recall that each subschema consists of a set of subgraphs. To have better predictive capability for the target variables, we prefer to have a set of subgraphs which are (a) strongly correlated to the target variables, but (b) uncorrelated with one another. The first condition ensures that the subgraphs can be useful for predicting the target variables. The second condition guarantees that information in each subgraph does not overlapped, when predicting the class. That is, we conduct a form of pruning, in order to identify diverse subgraphs. It follows that all new subschemas which are subsumed by, or highly correlated to, a high risk subschema also poses a risk. To enhance privacy, all subschemas should thus be tested before releasing them to the users.

In order to estimate the correlation between subgraphs, we adopted the *SubInfo* calculation as presented by Guo and Viktor in [17]. In their approach, *SubInfo* is used to describe the knowledge held by a subgraph with respect to the target classes in the target relation. Following the same line of thought, the class probabilistic predictions generated by a given subgraph classifier is used as its corresponding subgraph's *SubInfo*. Through generating relational (aggregated) features, each subgraph may separately be "flattened" into a set of attribute-based training instances. Learning algorithms such as decision trees [19] or support vector machines [20] may subsequently be applied to learn the relational target concept, forming a number of subgraph classifiers. Accordingly, the subgraph classifiers are able to generate corresponding *SubInfo* variables.

After generating the *SubInfo* variable for each subgraph, we are ready to compute the correlation among different subgraph subsets, which is discussed next.

3) *Subschema Informativeness*: Following the idea presented in [21], a heuristic measurement has been used to evaluate the "goodness" of a subschema (i.e., a set of subgraphs), for building an accurate classification model. The "goodness" of a subschema \mathcal{I} is calculated as follows.

$$\mathcal{I} = \frac{k\overline{R_{cf}}}{\sqrt{k + k(k-1)\overline{R_{ff}}}} \quad (2)$$

Here, K is the number of *SubInfo* variables in the subset (i.e., subschema), $\overline{R_{cf}}$ is the average *SubInfo* variable-to-target variable correlation, and $\overline{R_{ff}}$ represents the average *SubInfo* variable-to-*SubInfo* variable dependence. This formula has previously been applied in test theory to estimate an external variable of interest [22], [23], [24]. Hall has adapted it into the CFS feature selection strategy [25], where this measurement aims to discover a subset of features which are highly correlated to the class. Also, Guo and Viktor [21] utilized this formula to select a subset of useful views for multirelational classification.

The *Symmetrical Uncertainty* (\mathcal{U}) [26] is used to measure the degree of correlation between *SubInfo* variables and the target class ($\overline{R_{cf}}$) as well as the correlations between the *SubInfo* variables themselves ($\overline{R_{ff}}$). This score is a variation of the *Information Gain* (*InfoGain*) measure [19]. It compensates for *InfoGain*'s bias toward attributes with more values, and has been used by Ghiselli [22] and Hall [25]. *Symmetrical Uncertainty* is defined as follows:

Given variables W and Z ,

$$\mathcal{U} = 2.0 \times \left[\frac{\text{InfoGain}}{H(Z) + H(W)} \right]$$

where $H(W)$ and $H(Z)$ are the entropies of the random variables W and Z , respectively. The entropy of a random variable Z is defined as

$$H(Z) = - \sum_{z \in Z} p(z) \log_2(p(z))$$

And the *InfoGain* is given by

$$\begin{aligned} \text{InfoGain} = & - \sum_{z \in Z} p(z) \log_2(p(z)) \\ & + \sum_{w \in W} p(w) \sum_{z \in Z} p(z|w) \log_2(p(z|w)) \end{aligned}$$

Note that, these measures need all of the variables to be nominal, so *SubInfo* values are first discretized.

4) *Subschema Privacy-Informativeness*: To protect privacy leakage, we need to consider the predictive capabilities against both the target variable (represented by \mathcal{I}) and the confidential variable (represented by \mathcal{P}) when a database subschema is published. Based on this observation, the TSMC method uses a subschema's \mathcal{PI} value to reflect its performance when predicting the target variables as well as its degree of sensitivity in terms of predicting the sensitive attributes.

The \mathcal{PI} value of a subschema is computed using Equations 1 and 2, as follows.

$$\begin{aligned} \mathcal{PI} &= \mathcal{I} * (1 - \mathcal{P}) \\ &= \frac{k\overline{R_{cf}}(1 - \mathcal{P}_k)}{\sqrt{k + k(k-1)\overline{R_{ff}}}} \end{aligned} \quad (3)$$

This formulation suggests that a subschema with more information for predicting the target variable, but with very limited predictive capability on the confidential variable, is preferred. That is, for privacy protection, a subschema should have a larger \mathcal{I} value and a small \mathcal{P} value.

5) *Subschema Searching and Ranking*: In order to identify a subschema, i.e., a set of uncorrelated subgraphs, which has a large \mathcal{I} value but a small \mathcal{P} value, the evaluation procedure searches all of the possible *SubInfo* variable subsets, compute their \mathcal{PI} values, and then constructs a ranking of them.

To search the *SubInfo* variable space, the STMC method uses a best-first search strategy [27]. The method starts with an empty set of *SubInfo* variables, and keeps expanding, one variable at a time. In each round of the expansion, the best variable subset, namely the subset with the highest \mathcal{PI} value is chosen. In addition, the algorithm terminates the search if a preset number of consecutive non-improvement expansions occurs.

As a result, the method generates a ranked list of subschemas with different \mathcal{PI} values. As described in Algorithm 1, the TSMC method calculates such a list. Accordingly, one may select a subschema based on the requirements for the predictive capabilities on both the target variable and the confidential attributes.

IV. EXPERIMENTAL EVALUATION

In this section, we demonstrate the information leakage in multirelational classification with experiments against a financial database. Also, we discuss the outputs resulting from the TSMC method to show its effectiveness for privacy leakage prevention in multirelational classification.

A. Data Set Used

In our experiment, we used the financial database published for the PKDD 1999 discovery challenge [28]. The database was offered by a Czech bank and contains typical business data. Figure 1 shows the database. The multirelational classification task aims to predict a new customer's risk level for a personal loan. The database consists of eight tables. Tables *Account*, *Demographic*, *Disposition*, *Credit Card*, *Transaction*, *Client*, and *Order* are the background relations and *Loan* is the target relation. The class attribute (target attribute) is the loan status in the *Loan* table (highlighted in red in Figure 1), which indicates the status of the loan, namely A (finished and good), B (finished but bad), C (good but not finished), or D (bad and not finished). Our experiment used the data prepared by Yin et al. in [18].

B. Experimental Setup

In this experiment, we consider the *payment type* in the *Order* table as being confidential and it follows that it should be protected. We assume that the *payment type* information will either be eliminated from the database,

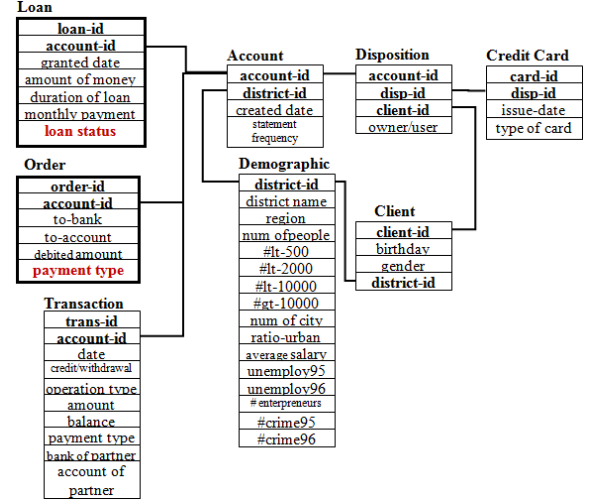


Figure 1. PKDD1999 Financial Database

or distorted, prior to being published ². The *Order* table contains the details of an order to pay a loan. It includes the account information, bank of the recipient, account of the recipient, debited amount, and the previously introduced *payment type*. The *payment type* attribute indicates one of four types of payments, namely for insurance, home loans, leases or personal loans. In this scenario, more than half of the payments are home loan repayment, i.e. there are 3502 home loan payment and 2969 other payment orders. We are interested in protecting the confidential information whether, or not, a client is paying a home loan.

C. Experimental Results

1) *Potential Privacy Leakage*: As a first step, we shifted our target variable from the loan status in the *Loan* table to the *payment type* in the *Order* table (highlighted in red in Figure 1). We used CrossMine to build a classification model [18]. Our experimental results show that we are able to build a set of rules to predict if an order is a household payment, or not, with an accuracy of 72.3%. In other words, even if we eliminate the *payment type* from the *Order* table, we can still use the remainder of the database to predict the values of *payment type* well. That is, there is still a potential for privacy leakage in such database, even if the sensitive attributes are eliminated or distorted in multirelational classification.

A possible solution here would be to prevent the prediction of the type of payment from the *Order* table with high confidence, but still maintain the predictive performance against the loan status in the *Loan* table. The TSMC method is designed to achieve this goal. Next, the execution of the TSMC method against this database is discussed.

²It follows that confidential attributes are removed or distorted after the application of the TSMC method, since the algorithm first needs to build rules against these attributes.

Table I
SAMPLE RULES LEARNED

Order.payment type = house payment ← (Order.amount >= 1833) (Order.account_id ⋈ Disposition.account_id) (Disposition.client_id ⋈ Client.client_id) (Client.birth_date <= 31/10/1937)
Order.payment type = house payment ← (Order.amount >= 1947) (Order.account_id ⋈ Transaction.account_id)(Transaction.type == house) (Order.account_id ⋈ Account.account_id)(Account.district_id ⋈ Demographic.district_id) (Demographic.unemploy95 >= 339) (Demographic.num_lt_10000 >= 3)
Order.payment type = non house payment ← (Order.account_id ⋈ Disposition.account_id) (Disposition.client_id ⋈ Client.client_id) (Client.birth_date >= 27/11/1936)(Client.birth_date <= 04/07/1951) (Order.amount <= 3849)(Order.account_id ⋈ Transaction.account_id)(Transaction.amount >= 8155) (Client.district_id ⋈ Demographic.district_id)(Demographic.unemploy96 <= 539)

Table II
THE NUMBER OF TUPLES COVERED BY THE SET OF SELECTED RULES

Subschemas	# tuples covered
{Order,Disposition,Client}	896
{Order,Disposition,Client,Demographic}	1154
{Order,Disposition,Client,Demographic,Transaction}	1394
{Order,Disposition,Client,Demographic,Transaction,Account}	3562
{Order,Disposition,Client,Demographic,Account}	1800
{Order,Disposition,Client,Account}	1103
{Order,Demographic,Account}	404
{Order,Demographic,Transaction,Account}	1130
{Order,Transaction,Account}	497
{Order,Transaction}	160

Table III
PRIVACY SENSITIVITY OF SUBSCHEMAS

Subschemas	Privacy sensitivity
{Order,Disposition,Client}	0.25
{Order,Disposition,Client,Demographic}	0.32
{Order,Disposition,Client,Demographic,Transaction}	0.39
{Order,Disposition,Client,Demographic,Transaction,Account}	1.0
{Order,Disposition,Client,Demographic,Account}	0.51
{Order,Disposition,Client,Account}	0.31
{Order,Demographic,Account}	0.11
{Order,Demographic,Transaction,Account}	0.32
{Order,Transaction,Account}	0.14
{Order,Transaction}	0.04

2) *Subschema Privacy Sensitivity*: The first step of the TSMC method aims to identify attributes that predict the sensitive attributes, through searching features across multiple tables in the database. From the rules built for predicting the *payment type* in the Order table, 12 high coverage rules were selected. These rules cover 3341 instances in the Order table. That is, over 50% of the examples have been covered by the set of rules selected. The aim for the rule selection

is to identify attributes (across tables) which are useful for predicting the sensitive attributes *payment type* in the Order table.

For example, Table I lists three (3) of the 12 rules learned. The first rule, as described in Table I, indicates that if a payment with an amount larger than 1833 in the Order table, and the client, linked through the Disposition table, was born no later than Oct 31, 1937, then it was a home loan payment. This rule involves two attributes which come from different tables. Similarly, the second rule shows that the amount attribute in the Order table works together with the type attribute in the Transaction table. The rule also indicates that the level of unemployment in 1995 and the number of municipalities with between 2000 and 9999 inhabitants in the Demographic table are of importance to categorize the values for the *payment type* in the Order table. The same idea was demonstrated by the third rule, which includes attributes birth date in the Client table, amount in the Order table, amount in the Transaction table, and the level of unemployment in 1996 from the Demographic table. Importantly, these rules indicate that publicly known statistical data, such as unemployment rates and the number of households in a municipality, may be used to inject attacks when aiming to target individuals. That is, through the combination of public and insider knowledge, an attacker

Table IV
CONSTRUCTED SUBGRAPHS

Loan
Loan ⋈ Account
Loan ⋈ Order
Loan ⋈ Transaction
Loan ⋈ Account ⋈ Disposition
Loan ⋈ Account ⋈ Demographic
Loan ⋈ Account ⋈ Disposition ⋈ Credit Card
Loan ⋈ Account ⋈ Disposition ⋈ Client
Loan ⋈ Account ⋈ Demographic ⋈ Client
Loan ⋈ Account ⋈ Demographic ⋈ Client ⋈ Disposition
Loan ⋈ Account ⋈ Demographic ⋈ Client ⋈ Disposition ⋈ Credit Card

Table V
THE TOP 10 RANKED SUBSCHEMAS AND THEIR ACCURACIES OBTAINED AGAINST THE TARGET AND SENSITIVE VARIABLES

Subschemas Selected For Release	Acc. Target	Acc. Sensi.
{Loan, Order, Transaction, Account, Disposition, Credit Card}	85.0%	66.9%
{Loan, Transaction, Account, Disposition}	82.5%	54.9%
{Loan, Account, Transaction}	87.5%	61.4%
{Loan, Account, Transaction, Order}	87.5%	72.0%
{Loan, Order, Transaction, Account, Disposition}	82.5%	64.7%
{Loan, Transaction, Account, Disposition, Client}	82.5%	59.3%
{Loan, Transaction, Account, Disposition, Client, Order}	85.0%	68.9%
{Loan, Transaction, Account, Demographic}	87.5%	62.2%
{Loan, Transaction, Account, Demographic, Client}	87.5%	60.8%
{Loan, Transaction, Account, Demographic, Client, Order}	87.5%	72.1%
ALL TABLES IN THE DATABASE	87.5%	72.3%

may be able to infer confidential information from the data mining results.

These rules show how attributes across tables work together to predict the confidential attributes. In other words, these rules were able to capture the correlation and their predictive capability among multiple attributes across multiple tables, regardless the attribute types.

As described in Algorithm 1, the second step of the TSMC method is to identify the privacy sensitivity of different subschemas. Let us reconsider the first conjunction rule as shown in Table I. If we evaluate this rule at the table level, we may conclude that the subschema which consists of the tables {Order, Disposition, Client} has attributes for constructing this rule. Thus, we may want to avoid using this subschema or, at least, restrict access to it.

The privacy sensitivity of a subschema is computed using Equation 1, as described in Section III-B. Accordingly, from the tuple coverage of Table II, we calculate the degree of sensitivity for each subschema using Equation 1, and present the results in Table III. As shown in Table III, different subschemas have various privacy sensitivities, in terms of predicting the confidential attribute *payment type* in the Order table. For example, the subschema which consists of tables {Order, Disposition, Client, Demographic, Transaction, Account} has the highest privacy sensitivity. That is, this subschema may be used to build an accurate classification model for determining the value of an order's payment type.

Having the degrees of privacy sensitivity of different subschemas of the provided database, we are able to construct and select different subschemas with various privacy sensitivities against the confidential attributes and predictive capability for the target variables. We will discuss these two elements in detail, next.

3) *Subschema Evaluation*: Following the strategy as described in Section III-C, we construct the set of subgraphs from the provided database. Each subgraph corresponds to a join path starting with the target table. Eleven (11) subgraphs were constructed by the TSMC method. The subgraphs are

presented in Table IV.

After constructing the subgraphs, the search algorithm computes different combinations of subgraphs, resulting in different subschemas. Consequently, each subschema has a \mathcal{PI} value which reflects information about the target variable classification as well as the predictive capability against the confidential attributes. In other words, a ranked list of subschemas, each with a measurement describing the trade-off in between the predictive capability against the target variable and confidential attribute, is created. Table V presents the top ten (10) subschemas generated from the financial database. In this table, we show the tested results against the target label (i.e., the loan status) as well as confidential attribute (namely, the *payment type*). For comparison purpose, we provide the accuracy as obtained against the full database schema at the bottom of the table.

From Table V, one can see that the TSMC method has created a list of subschemas with different predictive capability against the target variable and the confidential attribute. The experimental results, as shown in Table V, suggest that one can select a subschema with a good trade-off between the two predictive capabilities.

Specifically, one is able to identify the dangerous subschemas, that pose a high data leakage risk. For example, in Table V, consider the subschema containing tables {Loan, Transaction, Account, Demographic, Client}. In this case, the accuracy against the target variable remains the same as against the full database schema. However, for the confidential attribute, the accuracy drops from 72.3% to 60.8%. It follows that it is up to the owner of the database, to decide if this potentially high level of leakage is acceptable, or not. On the other hand, in order to have more confidence on protecting the sensitive attribute, one may prefer to publish the subschema with tables {Loan, Transaction, Account, Disposition}. Using this subschema, one is able to predict the confidential *payment type* in the Order table with an accuracy of 54.9%. It follows that this value is only slightly better than random guessing. However, this subschema still predicts the target variable, namely the loan status from the

Loan table with an accuracy of 82.5%, slightly lower than the 87.5% against the original, full database. In summary, the experimental results show that the TSMC method generates a ranked list of subschemas with different trade-off between the multirelational classification accuracy and the predictive capability against the confidential attributes.

V. CONCLUSIONS AND DISCUSSIONS

Multirelational classification discovers patterns across interlinked tables in a relational database [29]. For complex databases, it is becoming more difficult to detect, avoid and limit the inference capabilities between attributes, especially during data mining. This is due to the complexity of the database schema, the involvement of multiple interconnected tables and various foreign key joins, thus resulting in potential privacy leakage. For example, as shown in this paper, one may use the published database to target confidential attributes through shifting the classification target.

To address the above-mentioned challenge, the paper proposes a method to generate a ranked list of subschemas for publishing. These subschemas aim to maintain the predictive performance on the target variable, but limit the prediction accuracy against the confidential attributes. In this way, the owner of the database may decide to rather publish one of the generated subschemas which has an acceptable trade-off between sensitive attribute protection and target variable prediction, instead of the entire database. We conducted experiments on a financial database to show the effectiveness of the strategy. Our experimental results show that our approach generates subschemas which maintain high accuracies against the target variables, while lowering the predictive capability against confidential attributes.

Our future work will include experiments on more databases with complex schemas and a very large number of tuples. As stated earlier, our approach uses a set of rules built by a classifier to detect those attributes that are correlated with a sensitive attribute. We aim to investigate other ways to detect such correlations. Furthermore, it follows that a database may contain many confidential attributes, and we will investigate new methods to address this issue.

REFERENCES

- [1] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *PODS '01*. NY, USA: ACM, 2001, pp. 247–255.
- [2] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala, "Privacy-preserving data publishing," *Found. Trends databases*, vol. 2, no. 1–2, pp. 1–167, 2009.
- [3] A. Gkoulalas-Divanis and V. S. Verykios, "An overview of privacy preserving data mining," *ACM Crossroads*, vol. 15, no. 4, 2009.
- [4] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 1–53, 2010.
- [5] J. Domingo-Ferrer and Y. Saygin, "Recent progress in database privacy," *Data Knowl. Eng.*, vol. 68, no. 11, pp. 1157–1159, 2009.
- [6] L. Guo and X. Wu, "Privacy preserving categorical data analysis with unknown distortion parameters," *Transactions on Data Privacy*, vol. 2, no. 3, pp. 185–205, 2009.
- [7] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Trans. on Knowl. and Data Eng.*, vol. 13, no. 6, pp. 1010–1027, 2001.
- [8] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, 2000.
- [9] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," in *IHW '01*. Springer-Verlag, 2001, pp. 369–383.
- [10] Y. Tao, J. Pei, J. Li, X. Xiao, K. Yi, and Z. Xing, "Correlation hiding by independence masking," in *ICDE*, 2010, pp. 964–967.
- [11] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *KDD*, 2002, pp. 639–644.
- [12] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association rule hiding," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, no. 4, pp. 434–447, 2004.
- [13] Z. Zhu and W. Du, "K-anonymous association rule hiding," in *ASIACCS '10*. ACM, 2010, pp. 305–309.
- [14] A. Gkoulalas-Divanis and V. S. Verykios, "An integer programming approach for frequent itemset hiding," in *CIKM '06*. New York, NY, USA: ACM, 2006, pp. 748–757.
- [15] C. Yao, X. S. Wang, and S. Jajodia, "Checking for k-anonymity violation by views," in *VLDB '05*. VLDB Endowment, 2005, pp. 910–921.
- [16] H. Kargupta, K. Das, and K. Liu, "Multi-party, privacy-preserving distributed data mining using a game theoretic framework," in *PKDD*, 2007, pp. 523–531.
- [17] H. Guo and H. L. Viktor, "Mining relational data through correlation-based multiple view validation," in *KDD '06*, New York, NY, USA, 2006, pp. 567–573.
- [18] X. Yin, J. Han, J. Yang, and P. S. Yu, "Crossmine: Efficient classification across multiple database relations," in *ICDE '04*, Boston, 2004.
- [19] J. R. Quinlan, *C4.5: programs for machine learning*. USA: Morgan Kaufmann Publishers Inc., 1993.
- [20] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [21] H. Guo and H. L. Viktor, "Multirelational classification: a multiple view approach," *Knowl. Inf. Syst.*, vol. 17, no. 3, pp. 287–312, 2008.
- [22] E. E. Ghiselli, *Theory of Psychological Measurement*. McGrawHill Book Company, 1964.
- [23] R. Hogarth, "Methods for aggregating opinions," in *H. Jungermann and G. de Zeeuw, editors, Decision Making and Change in Human Affairs*. Dordrecht-Holland, 1977.
- [24] R. Zajonc, "A note on group judgements and group size," *Human Relations*, vol. 15, pp. 177–180, 1962.
- [25] M. Hall, "Correlation-based feature selection for machine learning, Ph.D diss., Waikato Uni." 1998.
- [26] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 1988.
- [27] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997.
- [28] P. Berka, "Guide to the financial data set." in *A. Siebes and P. Berka, editors, PKDD2000 Discovery Challenge*, 2000.
- [29] J. Han and M. Kamber, *Data mining: concepts and techniques, 2nd Edition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2006.