Privacy Leakage in Multi-relational Learning via Unwanted Classification Models

Hongyu Guo*, Herna L. Viktor+ and Eric Paquet*++

*Institute for Information Technology, National Research Council of Canada
Email: {hongyu.guo,eric.paquet}@nrc-cnrc.gc.ca

+School of Electrical Engineering and Computer Science, University of Ottawa
Email: hlviktor@site.uottawa.ca

Abstract

Multirelational classification algorithms aim to discover patterns across multiple interlinked tables in a relational database. However, when considering a complex database schema, it becomes difficult to identify all possible relationships between attributes. This is because a database often contains a very large number of attributes which come from different interconnected tables with non-determinate (such as one-to-many) relationships. A set of seemingly harmless attributes across multiple tables, therefore, may be used to learn unwanted classification models to accurately determine confidential information, leading to data leaks. Furthermore, eliminating or distorting confidential attributes may be insufficient to prevent such data disclosure, since values may be inferred based on prior insider knowledge. This paper proposes an approach to identify such “dangerous” attribute sets. For data publishing, our method generates a ranked list of subschemas which maintain the predictive performance on the class attribute, while limiting the disclosure risk, and predictive accuracy, of confidential attributes. We demonstrate the effectiveness of our method against several databases.

1 Introduction

The number of commercial relational databases, with vast amounts of data, including financial transactions, marketing surveys, medical records, and healthinformatics observations, is growing exponentially. This fact has invoked a recent surge of interests on multi-relational classification, which aims to discover patterns across multiple interlinked tables in a database [10, 16]. State-of-the-art multi-relational methods such as CrossMine [35], TILDE [4], and MRC [15], however, require full access to the schema and unmasked values of the provided databases, thus preventing the application of popular data perturbation techniques for privacy protection [1, 7, 14].

Another challenge for privacy protection in multi-relational classification is that, for relational databases, it is difficult to identify all attribute interrelationships. Often, a database contains a large number of attributes which come from multiple interconnected tables with non-determinate (such as one-to-many) relationships. A set of seemingly harmless attributes across multiple tables, therefore, may be used to construct unwanted classification models which can accurately predict the values of confidential, or so-called sensitive, attributes, leading to data leaks.

As an example, let’s consider the database provided for the ECML 1998 challenge. This database was extracted from a data warehouse of a Swiss insurance company and the schema is depicted in Figure 1 [23]. Originally, two learn-
Figure 1: ECML98 database schema with eight background tables: the classification target attribute (partner category) and the confidential attribute (household category) are highlighted in red.

ing tasks were presented from this database: task A is to classify the partner category into 1 or 2, and task B categorizes the household category of class positive or negative. Eight so-called background relations, namely Eadr, Hhold, Padr, Parrol, Part, Tfkomp, Tfrol and Vvert, are provided for the two learning tasks. This background information describes partners (customers) of the insurance companies, their households, addresses, and their insurance contacts. Suppose that the multirelational classification task here aims to classify the partner category in Task A. We consider the household category in Task B as being confidential and it follows that it should be protected. That is, we are interested in protecting the confidential information whether, or not, a household category is positive.

In this example, if the entire database schema is used, one may use a multi-relational classification method such as the MRC approach [15] to predict the partner category in the Task A relation with an accuracy of 95.14%. Suppose, we shift our target variable from the partner category in the Task A table to the household category in the Task B table. In this case, we are able to build a MRC model to predict if a household is positive, or not, with an accuracy of 93.66%.

These existing unwanted models against the confidential attribute pose a potential threat for privacy leakage. Firstly, to apply state-of-the-art multi-relational classification methods, one may have to publish the database with unmasked values. In this scenario, an attacker can learn unwanted models to accurately determine the values of confidential attributes. Secondly, even if the sensitive attributes are eliminated or distorted from the database, an attacker may still be able to use a small percentage of known values of the sensitive attributes (i.e., a small number of labeled instances) to accurately predict the confidential information well.

For example, as shown by Nigam and Ghani in [24], for the benchmarking WebKB-Course data set [5], which consists of 1051 web pages collected from computer science departments at four universities, a Naive Bayes classifier obtains an error of 3.3% if 788 training instances are labeled and the rest 25% of instances are held aside as a test set. An EM-based and a co-training algorithms, however, achieve errors of 4.3% and 5.4%, respectively, by using only 12 of the 788 labeled instances. In another classification task based on UseNet newsgroups [24], a co-training algorithm can achieve an error of 3.7% by using only 6 labeled documents and 1000 unlabeled documents. This error is lower than the 3.9% obtained by a Naive Bayes method with all the 1006 instances labeled. From a privacy perspective, a small number of labeled sensitive values in a database could be inferred, for example, by utilizing prior insider knowledge [33]. Consequently, unwanted accurate learning models may be able to leverage the small number of labeled instances to predict the removed/distorted sensitive attributes well, leading to privacy leakage.

A possible solution would be to prevent the prediction of the household category with high confidence, but still maintain the predictive performance against the partner category. For example, say we publish the subschema that only consists of tables Task A, Padr, and Parrol. We are still able to predict a partner’s category with an accuracy of 84.12%, but the predictive accuracy against the sensitive attribute household category drops from 93.66% to 58.54%.

Motivated by the above observations, we propose a method to identify attribute sets which can be used to build accurate classification
models against the confidential attributes. In particular, our method generates a ranked list of database 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 several databases.

This paper is organized as follows. Section 2 presents the background and problem formulation. Next, in Section 3, we introduce our method for privacy protection. This is followed, in Section 4, by the discussion of our experimental studies. Finally, Section 6 concludes the paper.

2 Background

Privacy leakage protection in data mining strives to prevent revealing sensitive data without invalidating the data mining results [1, 7]. Often data anonymization operations are applied [11].

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, 9]. For example, the k-anonymity model [27] and the perturbing method [2] are two techniques for achieving this goal.

Recent research deals with correlation and association between attributes to prevent the inference of sensitive data [28, 29, 30, 33]. For example, Association Rule Hiding (ARH) methods sanitize datasets in order to prevent disclosing sensitive association rules from the modified data [30]. Tao et al. propose a method to distort data in order to hide correlations between non-sensitive attributes [28].

Xiong et al. present a semi-supervised learning approach to prevent privacy attacks which make use of the observation that similar data objects tend to have similar class labels [33]. In addition, data leakage prevention in releasing multiple views from databases has also been intensively studied [34]. Also, privacy leakage in multi-party environment has been investigated [21].

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. This stands in contrast to the above-mentioned anonymization techniques, such as generalization, suppression, anonymization, permutation, and perturbation. Furthermore, our approach is not tied to a specific data mining technique and there is no need to learn from masked data, which will enable the application of current multi-relational classification algorithms. In our earlier work [18], we used a specific multi-relational learning method, namely the CrossMine algorithm, to find a set of rules to identify potentially dangerous attribute sets of a database. In contrast, the research presented in this paper entails the creation of an entropy-based general computational schema that does not rely on particular multi-relational learning strategies, aiming at collecting all potential dangerous subschemas of a database.

In this paper, a relational database \( \mathcal{R} \) is described by a set of tables \( \{ R_1, \ldots, R_n \} \). 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 \( m \) tables \( R_1 \bowtie \cdots \bowtie R_m \) 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 \( \{ J \} \) [10, 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,

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

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. \( C \in \{ R_b \} \), and \( C \) has either to be removed from the database or 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.

3 Privacy Preserving Multirelational Classification

Recall that our Privacy Preserving Multirelational Classification (PPMC) method aims to prevent the prediction of confidential attributes, while maintaining the predictive performance of the target variable. 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 detail, the PPMC method consists of the following two stages.

Firstly, the PPMC method collects two ranked subschema lists from the original database: one for the target attribute (target subschemas) and another for the confidential variable (privacy subschemas). The target subschema list contains subschemas that may be used to build accurate classification models for the target variable. The confidential subschema list has subschemas that may potentially be used to build accurate prediction models against the confidential variable.

In the second phase of the method, the PPMC approach compares each of the target subschemas with the subschemas on the privacy list, aiming to determine how well each target subschema predicts the confidential variable. As a result, each subschema on the target list will also have a value indicating its predictive performance against the confidential variable. After having such a list, the owner of the database may decide to publish one of the generated subschemas which has an acceptable trade-off between sensitive attribute protection and target variable prediction.

3.1 Target Subschema Construction

To collect target subschemas, the PPMC approach initially decomposes the relational do-

\begin{algorithm}
\caption{Target Subschema Construction}
\textbf{Input:} database $\mathcal{R} = (R_t, \{R_b\})$; target variable $Y \in R_t$
\begin{enumerate}
\item convert schema $\mathcal{R}$ into undirected graph $\mathcal{G}(V, E)$, there $R_t$ and $R_b$ as nodes $V$ and joins $J$ as edges $E$
\item using $Y$ as the classification target, construct a set of subgraphs from $\mathcal{G} \Rightarrow \text{subgraphs set } \{G_1, \cdots, G_n\}$
\item for each potential subgraph subset $G_s_i \subset \{G_1, \cdots, G_n\}$ do
\item compute the $I$ of the subgraph subset (i.e., subschema $\mathcal{R}'^{i} \in \mathcal{R}$)
\item end for
\item rank the $\{\mathcal{R}'^{i}\}$ based on their $I$ values
\item return the ranked $\{\mathcal{R}'^{i}\}$
\end{enumerate}
\end{algorithm}

main into subgraphs, each corresponding to a unique join path of the database. From this set, it searches and evaluate the predictive performance, in terms of predicting the target variable, of different subschemas that each consists of a set of subgraphs. The PPMC method collects subgraph subsets where subgraphs are strongly uncorrelated with each other (i.e., containing less redundant information), but correlated with the target class (i.e., benefiting target prediction). Note that, following Tao et. al, [28] we use the term correlation to denote the associations, interrelationships or links between attributes in our database. In the final step, the PPMC method ranks the collected subschemas according to their prediction capability to the target variable. In detail, the target subschema construction process consists of four (4) stages, as follows.

3.1.1 Subgraph Construction

The subgraph construction process aims to build a set of subgraphs given a relational schema. In our environment, for each subschema, we create several subgraphs. Each such subgraph starts at the target relation and 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. The subgraph construction process proceeds 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. Next, correlations between constructed subgraph are computed in the following two steps.

### 3.1.2 Subgraph SubInfo Creation

Each subgraph created above may be used to build a relational classifier using traditional single-shape learning algorithms (such as decision trees [26] or support vector machines (SVMs) [6]). These methods require “flat” data presentations. In order to employ these “flat” data methods, aggregation operators are usually used to squeeze a bag of tuples into one attribute-based entity in order to “link” the relations together [23]. For example, in the sample database, the count function may be used to determine the number of insurance contracts associated with a particular partner, and thus “link” the Task A and Parrol tables. Through applying aggregate functions such as min, max, and count, each subgraph may separately be “flattened” into a set of attribute-based training instances. Next, the trained subgraph classifiers are used to generate Subgraph Information (SubInfo) variables, which denote the information captured by a subgraph.

The choice of SubInfo is motivated by the success of meta-learning algorithms [13, 32]. In a meta-learning setting such as Stacked Generalization [32], knowledge of the base learners is conveyed through their predictions in the meta level. These predictions serve as the confidence measure made by a given individual learner [32]. Following the same line of thought, we use class probabilistic predictions generated by a given subgraph classifier as its corresponding subgraph’s SubInfo.

### 3.1.3 Subschema Informativeness

To calculate the correlation among different subgraph subsets, a heuristic measurement is used to evaluate the informativeness of a set of subgraphs, for building an accurate classification model. The informativeness of a set of subgraphs $\mathcal{I}$ is calculated as follows.

$$\mathcal{I} = \frac{k \mathcal{U}_{cf}}{\sqrt{k + k(k - 1) \mathcal{U}_{ff}}} \quad (1)$$

Here, $k$ is the number of SubInfo variables in the subset (i.e., subschema), $\mathcal{U}_{cf}$ is the average SubInfo variable-to-target variable correlation, and $\mathcal{U}_{ff}$ 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 [12, 20, 36]. Hall has adapted it into the CFS feature selection strategy [19], where this measurement aims to discover a subset of features which are highly correlated to the class. Also, Guo and Viktor [15] utilized this formula to select a subset of useful views for multirelational classification.

The Symmetrical Uncertainty ($\mathcal{U}$) [25] is used to measure the degree of correlation between SubInfo variables and the target class ($\mathcal{U}_{cf}$) as well as the correlations between the SubInfo variables themselves ($\mathcal{U}_{ff}$). This score is a variation of the Information Gain (InfoGain) measure [26]. It compensates for InfoGain’s bias toward attributes with more values, and has been used by Ghiselli [12] and Hall [19]. Symmetrical Uncertainty is defined as follows:

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

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

$$H(V) = -\sum_{v \in V} p(v) \log_2(p(v))$$

And the InfoGain is given by

$$\text{InfoGain} = -\sum_{v \in V} p(v) \log_2(p(v)) \quad + \sum_{w \in W} p(w) \sum_{v \in V} p(v|w) \log_2(p(v|w))$$

### 3.1.4 Subschema Searching and Ranking

In order to identify a set of subgraphs that has a large $\mathcal{I}$ value, the evaluation procedure uses a best-first search strategy [22] to search potential SubInfo variable subsets, compute their $\mathcal{I}$ values, and then construct a ranking of them. A SubInfo variable subset estimated to be closer to the best $\mathcal{I}$ value is extended first.
by a backtrack enabled best-first search strategy. Often, a stopping criteria is imposed to a best first search. The search will be terminated if a number of consecutive nonimprovement expansions occur. Based on our experimental observations, we heuristically set the number to five.

The best-first method starts with an empty set of SubInfo variables, and keeps expanding, one variable at a time. At each expansion iteration, the best variable subset, namely the subset with the highest $I$ value, is chosen. As a result, the method generates a ranked list of subgraph sets with different $I$ values. In this way, a ranked list of subgraph sets with different predictive capability against the target variable is constructed. The above processes are described in Algorithm 1.

After forming a list of target subschemas (where each consists of a set of subgraphs), dangerous subschemas that can accurately determine the values of confidential attributes are identified. This identification process is based on the construction of a list of privacy subschemas, which is discussed next.

### 3.2 Privacy Subschemas Construction

The construction of privacy subschemas is similar to that of the target subschemas. However, two issues have to be addressed while searching for the ranking list of privacy subschemas, namely incompleteness and bias.

#### 3.2.1 Incompleteness Issue

Although the target subschema construction process is able to find a set of subschemas that accurately predict the classification variable, the selected subschemas may not include all good subschemas (incompleteness). Recall from Section 3.1.4 that, a best-first heuristic search strategy is employed when constructing a set of good target subschemas. That is, the target subschema search component aims to find a good set of subschemas only, but do not intend to exhaustively find all such subschemas. This means that a subschema with good prediction for the sensitive attributes may still exist, if the search process is directly adopted to construct the list of dangerous privacy subschemas.

Consider the following case. Suppose a relation is duplicated in the database, and the search component for the target variable uses the first version of this relation. In this case, the searching method will exclude the second version of the relation when selecting good subschemas, because this table is strongly correlated with its duplicated one. Thus, if the second version contains essentially the same information and potentially leaking privacy, then identifying and excluding only the first relation will not lead to privacy protection.

To address this incompleteness problem, the

---

**Algorithm 2 The PPMC Approach**

**Input:** $\mathcal{R} = (R_t, \{R_b\}); \ Y \in R_t$ is the target variable, $C \in \{R_b\}$ is a confidential attribute; correlation threshold $\vartheta$

1: convert schema $\mathcal{R}$ into undirected graph $G(\mathcal{V}, \mathcal{E})$
2: using $C$ as the classification target, construct a set of subgraphs from $G \Rightarrow$ subgraphs set $\{\mathcal{G}_1, \ldots, \mathcal{G}_m\}$
3: for each potential subgraph subset $\mathcal{G}_{s_i} \subset \{\mathcal{G}_1, \ldots, \mathcal{G}_m\}$ do
   4: compute its $I$ (with respect to variable $C$)
   5: end for
6: repeat
7: for each selected subgraph subset $\mathcal{G}_{s_i}$
   8: for each subgraph $\mathcal{G}_m \notin \mathcal{G}_{s_i}$
   9: if $\mathcal{G}_m$ correlates to subgraph $\mathcal{G}_s \in \mathcal{G}_{s_i}$ (> $\vartheta$)
      then
      10: derive a new subgraph set from $\mathcal{G}_{s_i}$ by replacing $\mathcal{G}_s$ with $\mathcal{G}_m$; add the new formed subset $\mathcal{G}_{s_j}$ to the list
      end if
   end for
11: end for
12: end for
13: end for
14: until the subgraph subset $\mathcal{G}_{s_i}$ is stabilized
15: $n$ subschemas ($\mathcal{R}')$ ← rank the subgraph sets using $I$ values
16: for $i = 1$ to $n$ do
   17: for $j = 0$ to $i$ do
   18: if tables of $\mathcal{R}_j \subset$ tables of $\mathcal{R}_i$ then
   19: $\mathcal{R}_j$'s $I$ value ← $\mathcal{R}_i$'s $I$ value
   20: end if
   end for
   end for
21: end for
22: target subschema set ($\mathcal{R}'^t$) ← call Algorithm 1
23: for each target subschema in $\mathcal{R}'^t$ do
   24: for $i = 0$ to $n$ do
      25: if $\mathcal{R}_j \subset \mathcal{R}_i$ then
      26: the sensitive degree of $\mathcal{R}_j^t$ ← $I$ value of $\mathcal{R}_j$
      end if
   end for
   29: end for
30: end for
31: return the ranked $\{\mathcal{R}'^t\}$, each with two $I$ values
PPMC method includes the following so-called sibling subgraph search step, after a ranked list of privacy subschemas (where each consists of a set of subgraphs) has been formed using Equation 1. As described in Algorithm 2, this sibling search proceeds as follows.

First, for each subgraph set $G_s_i$ in the list $\{G_s_i\}_i^1$, the PPMC method searches for subgraphs that do not belong to the list but are strongly correlated (above a pre-set threshold value $\vartheta$) with any of subgraph belonging to it. When identifying such a correlated subgraph, the PPMC method derives a new set of subgraphs $G_s_i$ from $G_s_i$. Finally, the newly derived subgraph set is added to the candidate list $\{G_s_i\}_i^1$. This search process repeats until the list is stabilized. That is, no new subgraph subset is added to the list. In this way, subgraph sets that are strongly correlated to the existing privacy subgraph sets in the list are identified and collected.

### 3.2.2 Bias Issue

The second issue required to be addressed, while searching for a ranking list of privacy subschemas, is the bias.

From Equation 1, research has shown that when adding an additional variable into the constructed variable subset, the SubInfo variable-to-SubInfo variable correlation may well predominate over the SubInfo variable-to-target variable [20]. Also, Ghiselli has noted that it is unlikely to encounter a variable set in which all variables are highly correlated with the class variable but at the same time have low correlations with one another [12]. These observations imply that the subgraph set search component for the target variable discourages subgraph subsets with a large number of subgraphs. This makes sense when identifying a good subset of subgraphs for better predictive performance on the target variable. However, such bias may exclude subgraph sets which may contain a good predictor for the confidential attribute.

Consider, for example, a subgraph subset, where two subgraphs $\{A, B\}$ has been formed, and the correlation between $A$ and $B$ is 0.1. Suppose that there is another subgraph $C$ that is strongly correlated with $B$ with a correlation of 0.98, and its correlation with $A$ is 0.1. Assuming the correlations between $A$, $B$, $C$, and the class variable are all 0.5. In this case, the subset of $\{A, B, C\}$ has a $I$ value of 0.647, but the subset of $\{A, B\}$ has a $I$ value of 0.674. That is, the subset $\{A, B, C\}$ has lower $I$ value than that of subset $\{A, B\}$, according to Equation 1. From a privacy protection perspective, subgraph set $\{A, B, C\}$ is as dangerous as subgraph set $\{A, B\}$, because one can still use a subset of it, e.g., the set of $\{A, B\}$, to build an accurate model to predict the sensitive attribute well.

To address this bias problem, the PPMC method executes a so-called powerset search procedure, after a ranked list of privacy subgraph sets has been formed using Equation 1. The search aims to compensate for the bias of the $I$ metric, as described in Equation 1, toward privacy subgraph sets with a small number of subgraphs. It proceeds as follows.

After the ranked list of privacy subgraph sets, sorted in a descending order based on their $I$ values, has been constructed, for each subgraph set $G_s_i$ in the list, the PPMC method verifies, by tracing down the list from the top, if there is any subgraph set that is included in $G_s_i$. If such a subgraph set is found, the $I$ value associated with the first found subgraph set is assigned as the $I$ value for $G_s_i$. By doing so, adding more subgraphs into an existing subgraph subset will not decrease its $I$ value, thus compensating for Equation 1’s bias toward privacy subgraph sets with less subgraphs.

### 3.3 Algorithm Overview

The detail of the PPMC approach is described in Algorithm 2. It consists of the following four (4) steps.

Firstly, the algorithm collects different subschemas (privacy subschemas) of the database and compute their degrees of sensitivity. Secondly, the PPMC method shifts the classification target from the confidential variable to the target attribute. In this stage, another ranked list of subschemas of the provided database, regarding the target attribute, is constructed (as discussed in Section 3.1). Next, each of the constructed target subschemas is compared with the privacy subschema base, aiming to de-
termine how well each target subschema predicts the confidential variable. Finally, a ranking list of target subschemas is provided. For each subschema, its performance measurement when predicting the target variable, along with its privacy sensitivity level, is associated.

4 Experimental Evaluation

We implemented the PPMC algorithm using Weka [31]. Three benchmarking databases for multi-relational classification were used. The data reside in a MySQL database system in a 64-bit workstation running Windows Vista. We reported the average accuracy obtained by the MRC method using 10-fold cross validation. It follows that there are other relational learning methods that may be used to evaluate the effectiveness of the subschemas selected by the PPMC method. However, we employed the MRC strategy due to its ability to efficiently, directly and transparently search patterns from a relational database, without any data conversion [15]. In these experiments, the pre-set correlation threshold of the PPMC algorithm was set to 0.01, which will allow the PPMC method to search a very large number of attribute sets before the collected subschema list is stabilized.

For each database, the PPMC approach will generate two ranking lists, namely one for the target subschemas and another for the privacy subschemas. The subschemas in each list are in descending order, based on their $I$ values. In the experiment results, we use a subschema’s ranking index value (position) in the list to identify the subschema. Accordingly, the subschema with a lower index value indicates that the subschema has a better $I$. For example, the first (top) subschema in a ranked list has an index value of zero.

4.1 Databases Used

Our first experiment used the example ECML98 database as described in Section 2. Recall that this database was extracted from a Swiss insurance company. In the experiment, we used the star schema prepared in [23]. We aim to prevent the prediction of the class of Task B (household category), and consider the related Task A (partner category) as the target variable. We assume that the household category information will either be eliminated from the database, or distorted, prior to being published.

The second database used was published for the PKDD 1999 discovery challenge [3]. The multirelational classification task aims to predict a new customer’s risk level for a personal loan, i.e. good or bad. Our experiment used the data prepared by Yin et al. in [35]. 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 target attribute is the loan status in the Loan table. In this experiment, we consider the payment type in the Order table as being confidential and it follows that it should be protected. The payment type attribute indicates 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.

Our last experiment used the Thrombosis database from the PKDD 2001 Discovery Challenge [8]. Thrombosis is one of the most important and severe complications arisen from different collagen diseases. It has been found that this complication is closely related to anti-cardiolipin antibodies [8]. This database is organized using five background relations, namely Antibody_exam, Patient info, Diagnosis, Ana pattern, and Thrombosis. In our experiment, we set the Patient gender as the target attribute and the IgG concentration level as the confidential variable. The IgG level records a patient’s anti-cardiolipin antibody IgG concentration. We are interested in protecting whether, or not, a patient’s anti-cardiolipin antibody IgG level is zero.

4.2 Experimental Results

The experimental results against the ECML98 database was presented in Figure 2 and Table 1. There are three (3) subfigures in Figure 2. The
Figure 2: Experimental results against the ECML98 database. The three subfigures show the target subschemas, the privacy subschemas, and the subschema candidates for publishing, respectively.

Figure 3: Experimental results against the PKDD99 database. The three subfigures show the target subschemas, the privacy subschemas, and the subschema candidates for publishing, respectively.

Figure 4: Experimental results against the PKDD01 database. The three subfigures show the target subschemas, the privacy subschemas, and the subschema candidates for publishing, respectively.

Table 1: Predictive performance obtained against both the target and confidential variables for selected subschema candidates of the ECML98 Database

<table>
<thead>
<tr>
<th>ranking index</th>
<th>target attribute</th>
<th>privacy attribute</th>
<th>subschema tables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>target accuracy</td>
<td>privacy accuracy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>precision recall</td>
<td>precision recall</td>
<td></td>
</tr>
<tr>
<td>full schema</td>
<td>95.14</td>
<td>0.951</td>
<td>0.989</td>
</tr>
<tr>
<td>0 (top)</td>
<td>95.26</td>
<td>0.952</td>
<td>0.989</td>
</tr>
<tr>
<td>87</td>
<td>92</td>
<td>0.834</td>
<td>0.998</td>
</tr>
<tr>
<td>113</td>
<td>88</td>
<td>0.795</td>
<td>1.0</td>
</tr>
<tr>
<td>116(bottom)</td>
<td>91</td>
<td>0.795</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Table 2: Predictive performance obtained against both the target and confidential variables for selected subschema candidates of the PKDD99 Database

<table>
<thead>
<tr>
<th>ranking index</th>
<th>target attribute accuracy</th>
<th>privacy attribute accuracy</th>
<th>precision</th>
<th>recall</th>
<th>subschema tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>full schema</td>
<td>92.25</td>
<td>1.0</td>
<td>0.592</td>
<td>68.95</td>
<td>acc,.card,client,disp.,demo.,loan,order,tran.</td>
</tr>
<tr>
<td>0 (top)</td>
<td>5</td>
<td>92.25</td>
<td>1.0</td>
<td>0.592</td>
<td>account,client,disp.,loan,order,tran.</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>86.0</td>
<td>0.954</td>
<td>0.276</td>
<td>account,client,disp.,demo.,loan,order,tran.</td>
</tr>
<tr>
<td>18</td>
<td>34</td>
<td>81.5</td>
<td>0.533</td>
<td>0.210</td>
<td>account,client,demo.,loan,order</td>
</tr>
<tr>
<td>27 (bottom)</td>
<td>20</td>
<td>81.0</td>
<td>0.0</td>
<td>0.0</td>
<td>account,card,disp.,district,loan</td>
</tr>
</tbody>
</table>

Table 3: Predictive performance obtained against both the target and confidential variables for selected subschema candidates of the PKDD01 Database

<table>
<thead>
<tr>
<th>ranking index</th>
<th>target attribute accuracy</th>
<th>privacy attribute accuracy</th>
<th>precision</th>
<th>recall</th>
<th>subschema tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>full schema</td>
<td>91.97</td>
<td>0.889</td>
<td>1.0</td>
<td>71.44</td>
<td>ana_pattern,antibody_exam,antibody_level,diag.,patient,gend.,thrombosis</td>
</tr>
<tr>
<td>0 (top)</td>
<td>10</td>
<td>91.97</td>
<td>0.889</td>
<td>1.0</td>
<td>ana_pat.,antibody_exam,diag.,patient,gend.</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>91.97</td>
<td>0.889</td>
<td>1.0</td>
<td>antibody_level,diagnosis,gend.</td>
</tr>
<tr>
<td>26</td>
<td>42</td>
<td>91.97</td>
<td>0.889</td>
<td>1.0</td>
<td>antibody_level,patient_info,gend.</td>
</tr>
<tr>
<td>47 (bottom)</td>
<td>42</td>
<td>64.34</td>
<td>0.643</td>
<td>1.0</td>
<td>patient,gend.</td>
</tr>
</tbody>
</table>

The left subfigure shows the ranking index values of the subschemas, along with their I values, generated by the PPMC algorithm for the target attribute. This subfigure also provides the accuracy obtained by the MRC method when the best (with largest I value) and the worse (with smallest I value) subschemas were provided for learning.

The middle subfigure describes the subschemas generated by the PPMC algorithm for the confidential variable, along with their I values. Also in this subfigure, we present the precision obtained by the MRC method when the best and the worse privacy subschemas were provided for learning.

In the right subfigure, we provide the subschema candidates for publishing. Each candidate (denoted by a circle point in the subfigure) is associated with a target subschema index value and a privacy subschema index value. This subfigure was created as follows. For each target subschema in the left subfigure, the algorithm searches the privacy subschemas, as depicted in the middle subfigure, and then returns the best I value privacy subschema that is part of the target subschema. Also, in the right subfigure, the best and worse I value target subschemas (depicted in the most left and right dot points of the subfigure), as well as the two other subschemas with good trade-off between the target variable prediction and privacy attribute protection were highlighted in red.

Table 1 reports the predictive performance for the above-mentioned four subschema candidates, including the accuracy, precision, and recall obtained by the MRC method against both the target and privacy attributes. Also, the last column of the table provides the relations within the subschema. In addition, we also provide, in Table 1, the predictive performance obtained by the MRC method against the full schema of the tested database.

Similarly, the experiment results for the PKDD99 and PKDD01 databases were provided in Figure 3 and Table 2, and Figure 4 and Table 3, respectively.

4.2.1 Target and Privacy Subschemas

Results in the left and middle subfigures in Figures 1, 2, and 3 show that the PPMC method was able to generate subschemas with different predictive performance. For example, as shown in Figure 1, the predictive accuracy of different subschemas derived from the ECML98 database ranged between 79.56% and 95.26% against the target variable, and the precision against the privacy variable ranged between 0.562 and 0.908. Similarly, against the PKDD89 database, these two ranges were between 81.02% and 92.25%, and between 0.582 and 0.683, respectively. Against the PKDD01
database, these values were between 64.34% and 91.97%, and between 0.606 and 0.760, respectively.

These experimental results indicate that a wide range of good subschemas, in terms of predictive capability for both the target and confidential variables, has been collected by the PPMC method. Next, we are ready to choose a subschema candidate that can prevent the prediction of the confidential variables with high confidence, but still maintain the predictive performance against the target variables.

4.2.2 Subschema Candidate for Publishing

The graphs depicted in the right subfigures of Figures 1, 2, and 3 indicate that the PPMC method has created a list of subschemas with different predictive capability against the target variable and the confidential attribute for the ECML98, PKDD99, and PKDD01 databases, respectively. The experimental results show that one can select a subschema with a good trade-off between the two predictive capabilities. In particular, through consulting with the privacy subschema list, one is able to identify the dangerous subschemas, that pose a high data leakage risk.

In the right subfigures, one may prefer to choose a subschema with small target ranking index value, but large privacy ranking index value. These subschemas locate in the top-right corner of the subfigure. A subschema with small \( I \) value is weakly correlated with the classification variable, thus tend to provide less information for building an accurate classification model.

As an example, let’s consider the results obtained against the ECML98 database, as shown in Table 1. Consider the subschema with target index value of zero. In this case, the accuracy obtained against the target variable was slightly higher than that of against the full database schema. However, for the confidential attribute, the precision is also high (with a value of 0.908). 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. In order to have more confidence on protecting the sensitive attribute, one may prefer to increase the privacy index value of the subschema candidates. For instance, one can publish the subschema with target index value of 87 and privacy index value of 92. Using this subschema, one is able to predict the confidential attribute, as shown in Table 1, with a precision of 0.562 (a drop from 0.908 for privacy subschema with index value of zero). It follows that this precision value is only slightly better than random guessing. Nevertheless, as shown in the table, the accuracy against the target variable will be compromised. It drops from 95.26% to 84.12%.

Similar results are deduced from Tables 2 for the PKDD99 database. For example, when choosing the target subschema with ranking index value of zero, the target accuracy and privacy precision were 92.25% and 0.683, respectively. These numbers are very close to the performance against the full database schema. If one chooses the target index value of 17, the subschema, as expected, will decrease the target prediction from 92.25% to 86.0%, but the privacy precision will degrade from 0.698 to 0.604. If one wants to reduce the prediction on the sensitive variable further, one may choose subschema candidates with larger privacy index value. For example, one may prefer the subschema candidate with target index value of 18 and privacy index value of 34. In this case, the target prediction accuracy was 81.5%, but the sensitive precision will drop further to 0.582.

Finally, consider the results against the PKDD01 database. The results shown in Figure 3 suggest that, one may select the subschema with target index value of 5 or 26. These two subschemas will give a low precision against the sensitive variable, with a precision of 0.610 and 0.606, respectively (as shown in Table 3). Promisingly, in this case these two subschemas did not reduce the accuracy against the target variable. In other words, for this database, one may release only a subset of the database, and the prediction against the target variable will not degrade. However, the predictive capability, in terms of precision against the sensitive variable, will drop from 0.747 to 0.610 and 0.606, respectively.

In summary, these experimental results show that the PPMC method was able to generate a
ranked list of subschemas with different trade-off between the multirelational classification accuracy and the predictive capability against the confidential attributes.

4.3 Scalability Analysis

The computational cost of the PPMC algorithm heavily depends on the subschema evaluation procedure, as depicted in Sections 3.1 and 3.2. To evaluate the scalability of the PPMC method against complex databases, we generated six synthetic databases with different characteristics. The aim of these experiments was to further explore the applicability of the PPMC algorithm when considering relational repositories with a varying number of relations and tuples.

4.3.1 Synthetic Databases

The database generator was obtained from Yin et al. [35]. In their paper, Yin et al. used this database generator to create synthetic databases to mimic real-world databases in order to evaluate the scalability of the CrossMine multirelational classification algorithm. To create a database, the generator first generates a relational schema with a specified number of relations. Next, synthetic tuples with categorical attributes (integer values) are created and added to the database schema. Using this generator, users can specify the expected number of tuples, attributes, relations, and joins, etc., in a database in order to obtain various kinds of databases. Interested readers are referred to the paper presented by Yin et al. [35] for a detailed discussion of the database generator.

For each of the generated databases, the expected number of tuples and attributes were set to 1000 and 15, respectively. Default values, as found in Yin et al. [35], were used for the other parameters of the data generator. The six databases were generated with 10, 20, 50, 80, 100, and 150 relations (denoted as SynR10, SynR20, SynR50, SynR80, SynR100, and SynR150), respectively. We ran these experiments on a PC with a 2.66Ghz Intel Quad CPU and 4 GByte of RAM.

4.3.2 Scalability Performance

In our experiment, we varied the length of join path, as defined in Section 2, from zero (0) to six (6), against each of the six synthetic databases. That is, we allowed a subgraph, i.e., a join path, to contain up to seven (7) tables. As a result, the number of subgraphs to be evaluated by the PPMC algorithm for each database increases substantially. For example, the number of subgraphs for the SynR50, SynR80 and SynR100 databases are 570, 287 and 435, respectively, when the length of the join path is six. More details can be found in Figure 5(a). The running time required to complete the calculation (in seconds) for each of the databases and the number of subgraphs collected by the PPMC approach are shown in Figures 5(a) and 5(b), respectively.

The experimental results, as presented in Figures 5(a) and 5(b), indicate that, even
though the number of subgraphs that has to be explored by the PPMC approach increases quasi exponentially, with respect to the length of join path in databases with a large number of relations and tuples, the execution time as required for the PPMC algorithm increases much less rapidly. For example, as shown in Figure 5(a), for the SynR50 database, the number of subgraphs that was explored by the PPMC strategy jumped from 44 to 119, and then to 570 when the length of join paths allowed were set to 3, 4, and 6, respectively. However, the running time associated with these processes was 10, 19, and 50 seconds, respectively. Similar results are observed with the remaining five databases. Importantly, for all six databases, although the number of subgraphs explored by the PPMC method is very large, the execution time required by the PPMC approach is relatively small. These experimental results suggest that the PPMC strategy scale relatively well in term of running time when applied to complex databases.

5 Discussions

So far we have discussed the application of the PPMC method against only one target variable. However, our algorithm is applicable to databases with multiple confidential variables. Let us suppose, in order to illustrate, that we want to add one more sensitive attribute; the generalization to multiple sensitive attributes is immediate. There are two cases to be considered. That is, the two attributes are either 1) highly correlated or 2) weakly correlated or uncorrelated. In the first case, one may re-use the subschemas selected for the first attribute, since both attributes are highly correlated. In the second setting, the subschema selection element can be simply repeated for the second attribute, since both attributes are statistically independent.

In terms of computational cost, although Section 4.3.2 tested the scalability of our method against only one variable, our algorithm generalizes to multiple variables with, at most, linear complexity. In the first case mentioned above, namely two sensitive attributes are highly correlated, one re-uses the results obtained from the calculation associated with the first attribute. In the second scenario, i.e., two attributes are weakly or uncorrelated, as mentioned earlier, the subschema search calculations are repeated for the second attribute. It follows that the outcomes of both calculations are independent, since both attributes are statistically independent. The complexity of such an operation is linear because of the independence of the two attributes.

When considering databases where the target and confidential attributes are highly correlated, the direct application of the PPMC method may not easily find a subschema that predicts the class attribute well, while limiting the disclosure risk of confidential attributes. In such databases, one may first break the correlation between the target attribute and the confidential attributes before employing the PPMC algorithm. For example, one may use the correlation hiding approach proposed by Tao et al. [28] to make the target and confidential attributes appear mutually independent, through distorting some values in the database. As a result, the PPMC algorithm may then be applied to the transformed data.

In data privacy practices where data distortion techniques are preferred, one can use the PPMC method as a pre-processing and/or post-processing element. In the former, the method may significantly reduce the size of the database, in terms of the number of tuples and attributes, for data distortion, resulting in reducing the efforts for data masking. For the latter, the PPMC strategy can be used to examine if the masked data is immune to attacks that use unwanted classification models.

6 Conclusions and Future Work

Unwanted classification models that are able to accurately predict the values of confidential attributes, pose potential for privacy disclosure when publishing a database. We propose a method to identify such relevant attribute sets. These attributes may reside in multiple interconnected tables with non-determinate relationships. Our algorithm generates subschemas that maintain the predictive performance on
the target variable, but limit the prediction accuracy against the confidential attribute. We conducted experiments on three databases to show the effectiveness of our strategy.

Our future work will include applying the PPMC method to real-world databases with very large number of tuples and complex schema. Also, it would be interesting to empirically study the application of the PPMC algorithm to databases where the target and confidential variables are strongly correlated, and to databases where multiple confidential attributes exist.

References


