Protecting Sensitive Rules Based on Classification in Privacy Preserving Data Mining

Ms. Devangi L. Kotak, Mrs. Shweta Shukla

M. Tech (C.S.)
Department of Computer and Science
Rajasthan College of Engineering for Women
Jaipur, Rajasthan, India

***Abstract*- In this paper, we propose a method of hiding sensitive classification rules from data mining algorithms for categorical datasets. Our approach is to reconstruct a dataset according to the classification rules that have been checked and agreed by the data owner for releasing to data sharing. Unlike the other heuristic modification approaches, firstly, our method classifies a given dataset. Subsequently, a set of classification rules is shown to the data owner to identify the sensitive rules that should be hidden. Then we replace known values with unknown values ("?") in those transactions to hide a given sensitive classification rule. Finally the sanitized dataset is generated from which sensitive classification rules are no longer mined. Our experiments show that the sensitive rules can be hidden completely on the reconstructed datasets. While non-sensitive rules are still able to discovered without any side effect.**

Key words- Data Mining, Privacy Preserving, ClassificationRule Hiding.

I. INTRODUCTION

During some period of time, databases have grown exponentially in large stores and companies. In the old days, system analysts faced many difficulties in finding enough data to feed into their models. The picture has changed and now the reverse picture is a daily problem–how to understand the large amount of data we have accumulated over the years. Simultaneously, investors have realized that data is a hidden treasure in their companies. With data, one can analyze the behavior of competitors, understand the system better, and diagnose the faults in strategies and systems. Research into statistics, machine learning, and data analysis has been resurrected. Unfortunately, with the amount of data and the complexity of the underlying models, traditional approaches in statistics, machine learning, and traditional data analysis fail to cope with this level of complexity. The need therefore arises for better approaches that are able to handle complex models in a reasonable amount of time. These approaches have been named data mining (sometimes data farming) to distinguish them from traditional statistics, machine learning, and other data analysis techniques. [12]

With this increasing, new threats to privacy of the individual are also increases. Thus, an interesting new direction of data mining research has been developed, known as privacy preserving data mining (PPDM). The aim of these algorithms is the extraction of relevant knowledge from large collection of data, while protecting private information simultaneously. So, the main objective in PPDM is to develop algorithms for modifying the original data in such a way, so that the private data and knowledge remain private even after the mining process. The most of the existing PPDM approaches can be classified as two categories. First that aims to protect the sensitive data itself in the mining process and the second that aims to protect the sensitive data mining results (i.e. the extracted knowledge produced by the data mining process. Here we focus our attention on privacy preserving approaches that prohibit the disclosure of any sensitive knowledge patterns. These approaches modify the original dataset in such a way that certain sensitive knowledge patterns are hidden while mining the data.

In this paper, we focus on the problem of classification rules hiding or classification rules privacy preservation. Classification rule hiding is studied to a substantially lesser extent than association rule hiding [2]. Classification rule hiding algorithms consider a set of classification rule as a sensitive and aim to protect them. Our goal is the successful hiding of the sensitive classification rules. The remainder of this paper is organized as follows. In section 2 we present overview of privacy preserving techniques and related work. In section 4 we formalize the problem. Section 4 provides outline of rule hiding process and proposed algorithm. Experimental results are given in section 5. Finally we conclude our discussion in section 6.

II. THEORETICAL BACKGROUND WITH TECHNIQUES

The primary goal in privacy preserving is to protect the sensitive data before it is released for analysis. However the data may reside at same place or at different places. In such a scenario appropriate algorithms or techniques should be used which preserves any sensitive information in the knowledge discovery process. To address this issue there are many approaches adopted for privacy preserving data mining.

A.PRICVACY PRESERVING TECHNIQUES

It can be classified based on the following dimensions [3]:

•Data distribution

Based on the distribution of data, the PPDM algorithms can be first divided into two major categories, centralized and distributed data. In a centralized database scenarios, data are all stored in a single database; while, in a distributed database environment, data are stored in different databases. Distributed data environment can be further classified Into horizontal and vertical data distributions.

•Data modification

In general, data modification is used to ensure high privacy protection when it is necessary to modify the original values of a database that needs to be released to the public [2]. Methods of modification include:

• Perturbation, altering the value of an attribute by a new value (i.e., changing a 1-value to a 0-value, or adding noise)

• Blocking, replacement of an existing attribute value with a "?"

•Aggregation or merging, combination of several values into a coarser category

• Swapping, interchanging values of individual records.

•Data mining algorithm

This is actually something that is not known beforehand, but it facilitates the analysis and design of the data hiding algorithm. Various data mining algorithms have been considered in isolation of each other. Among them, the most important ideas have been developed for classification data mining algorithms, like decision tree inducers, association rule mining algorithms, clustering algorithms and Bayesian networks.

•Data or rule hiding

The PPDM algorithms can be further classified into two types, data hiding and rule hiding [3], according to the purposes of hiding. Data hiding refers to the cases where the sensitive data from original database like identity, name, and address that can be linked, directly or indirectly, to an individual person are hided. In contrast, in rule hiding, we remove the sensitive knowledge (rule) derived from original database after applying data mining algorithms.

• Privacy preservation

This refers to the privacy preservation technique used for the selective modification of the data. The techniques used are:

• Heuristic-based techniques modify selected values i.e. changing some data values in a given dataset from an original value to another value.

• Reconstruction-based techniques where the original distribution of the data is reconstructed. These algorithms are implemented by perturbing the data first and then reconstructing the distributions.

• Cryptography-based techniques like secure multiparty computation where a computation is secure if at the end of the computation, no party knows anything except its own input and the results.

III RELATED WORK

The hiding of classification rules is our main focus. Most of the initial work on this field addresses the problem of individual privacy. However, over the past few years interest has increased towards dealing with the problem of hiding sensitive patterns. In [4], a reconstruction algorithm is proposed for classification rules hiding. They proposed an algorithm to preserve the privacy of the classification rules by using reconstruction technique for categorical datasets. In which, only non-sensitive rules of the dataset are used to build a decision tree. Finally, the new dataset which contains only non-sensitive classification rules is reconstructed from the tree.

Another reconstruction based approach was proposed in [5]. They proposed an approach, called the Least Supported Attribute (LSA) modification algorithm, to classification rule hiding in categorical datasets. It uses the nonsensitive rules of original dataset to reconstruct its sanitized counterpart D'. If any transaction that supports nonsensitive rule also supports sensitive classification rule, LSA modifies it. The new value that will be assigned to the selected attribute will be the one that is supported the least by the transactions of the original dataset. By doing this, they minimize the side–effects in the sanitized dataset.

A data reduction approach was adopted in [6]. They addressed the problem of sensitive classification rule hiding by using data reduction approach. i.e. removing the whole selected tuples in the given dataset.

In [7] data perturbation approach (ROD) was proposed for hiding sensitive classification rules in categorical datasets. Their approach modifies the tuples of sensitive rules of a dataset D in such a way that these are distributed to the more important non-sensitive rules. In ROD tuples belonging to the sensitive rules are assigned to the non-sensitive rules based on their rank in the ruleset. First ROD identifies the sensitive and non-sensitive rules, then it selects the tuples of the non-sensitive rules and assigns them to the perturbed dataset D′. It then proceeds to the sensitive rules. For each tuple of a sensitive rule several attribute-value pairs are being altered in order to match a non-sensitive rule.

Our approach for classification rule hiding is motivated by a blocking based approach proposed in [8] and [9] for association rules privacy preserving. They increase or decrease the support of item by placing unknowns ("?") in place of 1's or 0's .So, It is difficult for an adversary to know the value behind unknowns ("?"). More efficient approaches than other approaches as in [8] [9] proposed in [10].

In [11], they propose two heuristic blocking based to preserve privacy for sensitive association rules in database. To hide sensitive rules, proposed algorithms replace the 1's or 0's by unknowns ("?") in fewer selected transactions for decreasing their confidence. To decrease the confidence of specified rules, first algorithm increases the support of rule antecedent, while another decreases the support of rule consequent. They can hide many rules at a time in rule clusters.

IV. PROBLEM MOTIVATION AND DEFINITION

In perturbation methods, values are changes, for example, 1 is replaced by 0 and 0 is replaced by 1. Sometimes it may have bad consequences. Consider a medical institution (an example discussed in [8]) that will publish some of its data, and data is sanitized by replacing actual values by true or false values. Researchers may use this data. But they may obtain wrong results (for example by using data mining tools to learn

rules). In worst case, such wrong rules could be used for critical purposes (like diagnosis). Therefore for many situations it is safer if the sanitization process place unknown values. This obscures the sensitive rules, while protecting the user of the data from learning 'false' rules.

A. Problem Statement

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

V. PROPOSED ALGORITHM

Our proposed technique applies to applications where we can store unknown values for some attributes, when actual values are not available or confidential. Here we propose blocking based algorithm to preserve privacy for sensitive classification rules in database. To hide sensitive rules, proposed algorithm replaces the 1's or 0's by unknowns ("?") in selected transactions. So, that rule will not be generated from the dataset.

TABLE 1: SAMPLE DATABASE OF TRANSACTION

TID	A	B	C	D
T1	1	1	0	1
T2	0	1	0	0
T3	1	0	1	1
T4	1	1	0	0
T5	1	1	0	1

TABLE 2: SAMPLE DATABASE OF TRANSACTION WITH UNKNOWN ATTRIBUTE VALUES

TID	A	B	C	D
T1	?	1	0	1
T2	0	1	0	0
T3	1	0	1	?
T4	1	?	0	0
T5	1	?	0	1

The goal of the algorithm presented here is to obscure a given set of sensitive rule by replacing known values with unknown values. Proposed algorithm, for each sensitive rule, it scans the original database by idexing and find outs the transactions supporting sensitive rules. We can say transaction supports any rule when the left side of the rule (attribute –value) pair is a subset of attribute values pair of the transaction and the right hand side of the rule is same as the class attribute of the transaction. Then for each transaction that supports sensitive rule, algorithm places "?" (unknown) values in place of attribute value which appears in rule. This procedure continues until all the sensitive rules are hidden. Finally the sanitized dataset which contains unknown values is released to public.

Example:
A diabetes dataset as a sample transaction database D is shown in table 3. It is taken from WEKA data set. Here preg, plas, pres, skin, insu, mass, pedi, age and class are the attributes of the transaction. And class is the decision attribute. Fig. 2 shows the set of classification rules generated from this dataset. Rules are generated using WEKA 3.7 tool with JRIP classification algorithm. Now suppose Rule No. 1 is considered as sensitive. We have to find the set of transactions that satisfies rule 1. Here, we can see from table 3 that tuple no. 1, 5, 9, 12 and 14 are the supporting transactions of rule 1. In order to hide rule 1, we will place unknown ("?") in these transaction. The modified dataset is shown in table 4. Fig. 3 shows the classification rules generated from the sanitized dataset.

Input: Initial database D, set of classification rules R.
Output: sanitized database D'
1. Begin
2*D'* ←{}.
3. sort the data in D
4. *S* ←*D*
5. *Rs* ←set of sensitive rules.
6. For each rule *Ri* ϵ *Rs*
7. {
8. For each tuple *t* ϵ *S* do
9. {
10. If *t* supports *Ri* then
11. *Ti* ←*t*. (add *t* to *Ti*)
12. }
13. }
14. For each *Ti*, where *Ri* ϵ *Rs* do
15. {
16. For each transaction *t* ϵ *Ti* do
17. {
18. Replace the value of attributes (which appears in *Rs*) with unknown ("?") in the *t*.
19. Update *S*.
20. }
21. }
22. *D'* ←*S*
23. End

Figure 1. Proposed Algorithm

TABLE 3. EXAMPLE DATASET

preg	**plas**	**pres**	**Skin**	**insu**	**mass**	**pedi**	**age**	**class**
6	148	72	35	0	33.6	0.627	50	tested_positive
1	85	66	29	0	26.6	0.351	31	tested_negative
8	183	64	0	0	23.3	0.672	32	tested_positive
1	89	66	23	94	28.1	0.167	21	tested_negative
0	137	40	35	168	43.1	2.288	33	tested_pos

								itive
5	116	74	0	0	25.6	0.201	30	tested_negative
3	78	50	32	88	31	0.248	26	tested_positive
10	115	0	0	0	35.3	0.134	29	tested_negative
2	197	70	45	543	30.5	0.158	53	tested_positive
8	125	96	0	0	0	0.232	54	tested_positive
4	110	92	0	0	37.6	0.191	30	tested_negative
10	168	74	0	0	38	0.537	34	tested_positive
10	139	80	0	0	27.1	1.441	57	tested_negative
1	189	60	23	846	30.1	0.398	59	tested_positive

```
1.(plas >= 132) and (mass >= 30) =>
  class=tested_positive
2.(age >= 29) and (insu >= 125) and
  (preg <= 3) =>
  class=tested_positive
```

Figure 2. Dataset's classification rules

TABLE 4. SANITIZED DATASET

preg	plas	pres	skin	insu	mass	pedi	age	class
?	?	?	?	?	?	?	?	?
1	85	66	29	0	26.6	0.351	31	tested_negative
8	183	64	0	0	23.3	0.672	32	tested_positive
1	89	66	23	94	28.1	0.167	21	tested_negative
?	?	?	?	?	?	?	?	?
5	116	74	0	0	25.6	0.201	30	tested_negative
3	78	50	32	88	31	0.248	26	tested_positive
10	115	0	0	0	35.3	0.134	29	tested_negative
?	?	?	?	?	?	?	?	?
8	125	96	0	0	0	0.232	54	tested_positive
4	110	92	0	0	37.6	0.191	30	tested_negative
?	?	?	?	?	?	?	?	?
10	139	80	0	0	27.1	1.441	57	tested_negative
?	?	?	?	?	?	?	?	?

VI. EXPERIMENTS AND RESULTS

In our experiment we have used three real life dataset from UCI repository. The detail of each dataset is given in table 3.

TABLE 5. DATASET DETAIL

Dataset	#attributes	#instances	# rules
Iris	5	150	4
Labor	16	57	4
Vote	17	435	4

The experiments have two parts. Firstly from the set of classification rules one rule is randomly selected as the sensitive rule. The second experiment is same as first but it is for multi-rule hiding.

In experiments classification rules are generated using WEKA 3.7 data mining tool. Then a sensitive rule is randomly selected for hiding. Proposed algorithm is applied to hide the sensitive rule.

Another experiment is for hiding more than one rule. The procedure is same as single rule hiding. We can check the algorithm for multi-rule hiding.

A. Evaluation metrics

- Hiding Failure

First evaluation metric is hiding failure, i.e. the percentage of sensitive information that is still discovered, after the data has been sanitized, gives an estimate of the hiding failure parameter. Here, we are placing unknowns in transaction which satisfy the sensitive classification rule such that they no longer satisfy sensitive rule. So, sensitive rule will not discovered from sanitized dataset.

- Side effects

The second metric is the side effects generated due to the rule hiding procedure. We can measure the side effects in terms of number of false dropped rules and number of ghost rule. False dropped rules are those nonsensitive rules which were present in the original dataset but hidden in the sanitized dataset. And ghost rules are those rules which were not there in the original dataset but are present in the sanitized dataset.

B. Experimental result

Table 4 shows experimental results when a sensitive rule is hidden. Here, no sensitive rule is discovered from the sanitized dataset. Table 5 shows the experimental results when more than one rule are hidden. It can be seen from table 4 that our proposed approach can be used to hide sensitive classification rule with minimum side effects (i.e. no. of false dropped rules and no. of ghost rules).

TABLE 6. EXPERIMENTAL RESULTS (SINGLE RULE)

Dataset	Discovered sensitive rule	#False dropped rules	#Ghost rules
Iris	0	0	0
Labor	0	0	0
Vote	0	0	0.3

TABLE 7. EXPERIMENTAL RESULTS (MULTI RULE)

Dataset	Discovered sensitive rule	#False dropped rules	#False dropped rules	#Ghost rules
Iris	2 3	0	0	0

Labor	2 3	0	0	0
Vote	2 3	0	0	0

VII. CONCLUSION AND FUTURE WORK

From the analysis it is concluded that the algorithm places unknown values in place of known values in the transactions that support the sensitive rules. So, from the modified database, the sensitive rules are no longer generated. Here the comparison faster as compared to other algorithm provided in this area.

In this algorithm focus is on centralized database so in future the algorithm can be used for distributed database to hide sensitive rules.

REFERENCES

[1] W. F. Richards, S. E. Davidson and S.A Long, Dual band reactively loaded microstrip patch antenna, IEE Trans. Antenna Propa., 33 (1985) 556-561.
[2] S. A. Division, S. A. long, W. F. Richards, Dual band microstrip antennas with monolithic reactive loading, Electronic Lett., 21 (1985) 936-937.
[3] Wang E, A novel dualband patch antenna for WLAN communication, PIER C, (2009) 93-102.
[4] D. D. Krishna, M. Gopikrishna, C. K. Aanandan, P. Mohanan and K. Vasudevan, Compact dualband slot loaded circular microstrip antenna with a superstrate, PIER, 83 (2008) 245-255.
[5] T .Huynh, and K. F. Lee, Single layer single patch wideband microstrip antenna, Electronics Lett., 31 (1995) 1310-1312.
[6] J. A. Ansari, R. B. Ram, Broadband stacked U-slot microstrip patch antenna,PIER Letters, 4 (2008)17-24.
[7] S. S. Sharma, B. R. Vishvakarma, Analysis of slot loaded rectangular microstrip patch antenna, Indian J. Radio Space Phys., 34(2005) 424-430.
[8] E. A. Wolf, Antenna Analysis, Artech house, Narwood (USA), 1998.
[9] R. Garg, P. Bhartia, I. Bahl, A. Ittipiboon Microstrip Antenna Design, Handbook Artech house, Boston, London, 2003.
[10] L. C. Chen, et al., Resonant frequency of circula disk printed circuit antenna. IEEE Trans. Antenna Propa., 25 (1997) 595-596.
[11] Zeland softwere, Inc., IE3D simulation software, version 14.05, C A, 2008.
[12] Data mining- A Huristic Approach by Abbas, Sarker, Newton.
[13] http://www.dba-oracle.com/t_indexing_power.htm
[14] http://msdn.microsoft.com/en-us/library/ms190197(v=sql.105).aspx