Distortion Based Algorithms For Privacy Preserving Frequent Item Set Mining

[1] K.Srinivasa Rao [2] V.Chiranjeevi

[1]Department of Computer Science and Engineering
Swarna Bharathi College of Engineering, Khammam, Andhra Pradesh, India
ksrao517@gmail.com
[2]Department of Computer Science and Engineering
Swarna Bharathi College of Engineering, Khammam, Andhra Pradesh, India
chiru508@gmail.com

Abstract

Data mining services require accurate input data for their results to be meaningful, but privacy concerns may influence users to provide spurious information. In order to preserve the privacy of the client in data mining process, a variety of techniques based on random perturbation of data records have been proposed recently. We focus on an improved distortion process that tries to enhance the accuracy by selectively modifying the list of items. The normal distortion procedure does not provide the flexibility of tuning the probability parameters for balancing privacy and accuracy parameters, and each item's presence/absence is modified with an equal probability. In improved distortion technique, frequent one item-sets, and non-frequent one item-sets are modified with a different probabilities controlled by two probability parameters fp, nfp respectively. The owner of the data has a flexibility to tune these two probability parameters (fp and nfp) based on his/her requirement for privacy and accuracy. The experiments conducted on real time datasets confirmed that there is a significant increase in the accuracy at a very marginal cost in privacy.

Keywords

Frequent patterns, sensitive patterns, non-sensitive patterns, legitimate patterns, Randomization , privacy preserving.

1. Introduction

The problem of privacy-preserving data mining has become more important in recent years because of the increasing ability to store personal data about users, and the increasing sophistication of data mining algorithms to leverage this information. A number of techniques such as *randomization* and *k-anonymity* have been suggested in recent years in order to perform privacy-preserving data mining. Furthermore, the problem has been discussed in multiple communities such as the database community, the statistical disclosure control community and the cryptography community.

Data products are released mainly to inform public or business policy, and research findings or public information. Securing these products against unauthorized access has been a long term goal of the database security research community and the government statistical agencies. Solutions to such a problem require combining several techniques and mechanisms. It is well known that simply restricting access to sensitive data does not ensure full data protection. It may be the case that sensitive data items can be inferred from non-sensitive data through some

inference process based on some knowledge the user has. Such a problem is known as *inference problem*. This has been widely investigated and possible solutions are identified. All of these approaches address the problem of how to prevent disclosure of sensitive data though the combination of known-inference rules with non-sensitive data. Examples of inference rules include deductive rules and functional dependencies. How ever these approaches do not deal with the problem of how to prevent the discovery of inference rules themselves. In other words rules are not considered sensitive knowledge.

Recent advances in data mining and machine learning increased this risks one may incur when releasing data to outside parties for mining. The main aim of these techniques is to enable the rapid and efficient discovery of hidden knowledge from very large datasets. Therefore the use of data mining techniques enables us to acquire not only knowledge that could be used to infer sensitive data but also sensitive knowledge. The knowledge extracted through data mining techniques is only probabilistic. However, even such probabilistic knowledge may provide sensitive information to users.

1.1. Motivational Examples

consider the following sample of a medical database

Rules Like *"Adult females with malarial infections are also prone to contract tuberculosis"* can be found from the above database using Frequent Itemset Mining algorithms. User may fear that by providing this sensitive details, their employment opportunities may be affected.

PID	Gender	Age	Malaria	Diabetes	TB	Cancer
P100	F	25	1	0	1	0
P200	M	27	1	1	0	0
P300	F	47	1	0	1	1
P400	F	29	0	0	1	1
P500	M	32	1	1	0	1

2. Literature Survey

2.1 The Basics of Mining Frequent Patterns and Association Rules

The discovery of the recurrent patterns in large transactional databases has become one of the main topics in data mining. In its simplest form, the task of finding frequent patterns can be viewed as the process of discovering all item sets, i.e., all combinations of items that are found in a sufficient number of examples, given a frequency threshold. If the frequency threshold is low, then there might be many frequent patterns in the answer set. The items in a frequent pattern are Boolean, i.e., items are either present or absent. For this reason, a transactional database may be represented by a sparse matrix in which the rows correspond to transactions and the columns

correspond to the items available in one store. If the element (i; j) is 1, this indicates that customer i purchased item j, while 0 indicates that the item j was not purchased.

When the frequent patterns are known, finding association rules is simple. Association rules provide a very simple but useful form of rule patterns for data mining. Association rule mining algorithms rely on support and confidence and mainly have two major phases: (1) based on a support _ set by the user, frequent item sets are determined through consecutive scans of the database; (2) strong association rules are derived from the frequent item sets and constrained by a minimum confidence ' also set by the user. Since the main challenge is the discovery of the frequent item sets, we consider only this second phase in our analysis.

2.2 Privacy Preservation: Problem Definition

In this work, our goal is to hide a group of frequent patterns which contains highly sensitive knowledge. We refer to these frequent patterns as restrictive patterns, and we define them as follows:

Definition 1: Let D be a transactional database, P be a set of all frequent patterns that can be mined from D, and Rules H be a set of decision support rules that need to be hidden according to some security policies. A set of patterns, denoted by RP , is said to be restrictive if RP Ć P and if and only if RP would derive the set Rules H. ~RP is the set of non-restrictive patterns such that ~RP U RP = P.

Definition 2 : Let T be a set of all transactions in a transactional database D and RP be a set of restrictive patterns mined from D. A set of transactions is said to be sensitive, as denoted by ST , if ST Ć T and if and only if all restrictive patterns can be mined from ST and only from ST .

The specific problem addressed in this paper can be stated as follows: If D is the source database of transactions and P is a set of relevant patterns that could be mined from D, the goal is to transform D into a database D0 so that the most frequent patterns in P can still be mined from D0 while others will be hidden. In this case, D0 becomes the released database.

2.3 Randomization Process

Randomization process modifies each transaction by replacing some of the existing items with non-existing items, and adding some fake items, thereby preserving the privacy. Randomization is implemented by a concept called Distortion

3 Solution Framework

Let 'I' be a set of 'n' items $\{a_1, a_2 \ldots a_n\}$ and 'T' be a set of transactions $\{t_1, t_2 \ldots t_n\}$ where each transaction t_i is a subset of 'I'.

Each transaction can be considered to be a random Boolean vector $X = \{X_i\}$, such that X_i is either 0 or 1. $X_i = 1$ (or 0) indicates that the transaction represented by X (does not) include(s) the item a_i. We generate the distorted vector from this transaction by computing $Y = distort(X)$ where $Y_i = X_i \oplus R_i'$ and R_j' is the complement of R_i, a random variable with density function

$f(R) = Bernouli(p)$, $(0 \le p \le 1)$ ---------- (12) i.e., R_i takes a value 1 with probability p and 0 with probability $(1 - p)$. Each bit in the vector X is flipped with a probability of p.

In normal distortion scheme [2], each bit is distorted with equal probability. But in optimal distortion technique frequent items are distorted with one probability, and non-frequent items are distorted with a different probability. This is to ensure that good accuracy is achieved even after

distortion. These two probability parameters can be tuned as per the user's requirements for privacy and accuracy.

3.1 Distortion Algorithms

Let *Bitmap[1...n]* contains a bitmap representation of a transaction t_i and *p* be the distortion probability. *Convert_to_bitmap( )* converts an item-list of a transaction to a bitmap. *get_random_double( )* generates a random real number between 0 and 1 with uniform probability.

3.2 Normal Distortion Algorithm

The normal distortion algorithm changes every item with an equal probability say *p.* This algorithm scans the database only once.

```
For i ← 1 to m
  Bitmap ← Convert_to_bitmap(t_i)
For j ← 1 to n
  Rand_num ← get_random_double()
  If Rand_num > p
    Bitmap[i] ← (Bitmap[i]+1)%2
```

Algorithm 1: Normal Distortion

3.3 Improved Distortion Algorithm

The improved distortion algorithm changes frequent items with a less probability (*fp*) and non-frequent items with a greater probability (*nfp*). The values of *fp* and *nfp* can be changed by the user.

This algorithm makes two scans over the entire database. In first scan the supports are calculated for each item, and stored in an array. Let *freqs[1…n]* stores the frequencies of all the items and supp be the minimum support. In the second scan, the actual distortion process takes place as per the following algorithm.

The Algorithm:

```
For i ← 1 to m
  Bitmap ← Convert_to_bitmap(t_i)
For j ← 1 to n
  Rand_num ← get_random_double()
  If freqs[j]< supp & itmap[j]=0
    If Rand_num > nfp
      Bitmap[j] ← Bitmap[j]+1)%2
```

Else Rand_num ← get_random_double()

If Rand_num > fp

Bitmap[j] ← 0

Algorithm 2: Improved Distortion

3.4 Measures of efficiency

Privacy

The framework suggested in [2] is used to quantify the privacy obtained. The probability of reconstructing an item *i* from a random transaction can be calculated by the following formula.

$$R(p, s_i) = \frac{s_i.p^2}{s_i.p + (1-s_i).(1-p)} + \frac{s_i.(1-p)^2}{s_i.(1-p) + (1-s_i).p} \quad \text{---------- (13)}$$

The derivation for the above formula is as follows, Let X_i is original bit and Y_i be the distorted bit

$$R(p, s_i) = P(Y_i = 1 \mid X_i = 1).P(X_i = 1 \mid Y_i = 1) + P(Y_i = 0 \mid X_i = 1).P(X_i = 1 \mid Y_i = 0)$$

$$R(p, s_i) = p.P(X_i = 1 \mid Y_i = 1) + (1-p).P(X_i = 1 \mid Y_i = 0)$$

$$P(X_i = 1 \mid Y_i = 1) = \frac{s_i.p}{P(X_i = 1).P(Y_i = 1 \mid X_i = 1) + P(X_i = 0).P(Y_i = 1 \mid X_i = 0)}$$

$$= \frac{s_i.p}{s_i p + (1-s_i).(1-p)} \quad \text{---------- (14)}$$

Similarly

$$P(X_i = 1 \mid Y_i = 0) = \frac{s_i(1-p)}{s_i(1-p) + (1-s_i)p} \quad \text{---------- (15)}$$

To find the Reconstruction probability of all the items R(p) we apply summation over all the individual item's reconstruction probabilities ($R(p, s_i)$). Hence

$$R(p) = \frac{\sum_{\forall i} s_i.R(p, s_i)}{\sum_{\forall i} s_i} \quad \text{For normal distortion} \quad (16)$$

$$R(p) = \frac{\sum_{i \in F} s_i.R(p, s_i)}{\sum_{i \in F} s_i} + \frac{\sum_{i \in F'} s_i.R(p, s_i)}{\sum_{i \in F'} s_i} \quad \text{For improved distortion (17),}$$

Where F is the set of frequent items and F′ is the set of non-frequent items

Accuracy

To quantify the accuracy, we first find out set of frequent item sets using any off-the-shelf frequent item set mining algorithm from the original database. Let it be *F*. Similarly we find out a set of frequent item sets *F′* from distorted database. Then, the following simple procedure calculates the accuracy.

count = 1

total_patterns = $|F|$

For each pattern $p \in F$

If $p \in F'$

count = count +1

Accuracy = (count/total_patterns)*100

Algorithm 7: Accuracy

3.5 Experimental Results

To assess the effectiveness of our algorithms, the experiments are conducted on three popular real time datasets *retail, BMS-Webview-1, BMS-Webview-2* [7].

For each of the three databases privacy (P), accuracy (A) metrics are calculated for various distortion probabilities (*nfp*) with an interval of 0.5 (Tables 1 to 3) and *fp*=0.95. The bench mark supports used for retail, BMS-Webview-1, BMS-Webview-2 are 0.003, 0.002, and 0.003 respectively. These two metrics are calculated and compared for normal distortion, and improved distortion. As the distortion probability decreases privacy increases, and accuracy decreases. The experimental results show that with a minor reduction in privacy, accuracy can be improved significantly with the improved distortion technique. Moreover, an off-the-shelf frequent item set mining algorithm is used for finding the frequent item sets from the distorted database without any modifications to the original mining algorithm that is used to find frequent itemsets from the original database. The accuracy can be further improved if a frequent itemset mining algorithm can be tailored to mine distorted databases.

Table 4 and Table 5 show the Privacy and Accuracy values for the two databases *BMS-WebView-2* and *Retail* by applying improved distortion procedure. From our experiments we observed that, for most of the databases, if *fp* is increased, privacy decreases and accuracy increases, and if *fp* is decreased, privacy increases and accuracy decreases. The effectiveness of the improved distortion procedure also depends on the percentage of frequent items among all the available items. If only a small fraction of all the available items are frequent then, the improved distortion process performs well. Otherwise we may get inferior values for privacy and accuracy values than those of normal distortion procedure.

Table 6 reports the execution times of a distortion algorithm implemented using item-list file representation (T_I) and bitmap file (T_B) representation of the transactional database. Experiments are conducted in the three datasets used in this paper. For all three databases, distortion algorithm implemented using item-list representation performed better than its bitmap counterpart. In

general the former consumes less space and performs significantly better for the databases in which the available items are more, and the average transaction length is less.

Distortion Probability	Normal Distortion		Improved Distortion (fp=0.95)	
	P	A	P	A
0.95	79.14	85.51	79.14	86.80
0.90	88.52	83.51	79.21	85.18
0.85	92.53	67.57	79.23	82.67
0.80	94.74	62.64	79.74	80.94
0.75	96.10	54.70	80.12	81.31
0.70	96.99	50.9	81.76	81.80

Table 1. BMSWebView1

Distortion Probability	Normal Distortion		Improved Distortion (fp=0.95)	
	P	A	P	A
0.95	81.09	85.04	81.09	86.43
0.90	85.63	71.61	81.68	85.66
0.85	88.07	60.11	81.88	84.88
0.80	89.72	50.60	81.97	84.74
0.75	90.93	43.07	82.03	85.23
0.70	91.83	36.33	82.07	84.74

Table 2. Retail

Distortion Probability	Normal Distortion		Improved Distortion (fp=0.95)	
	P	A	P	A
0.95	90.44	73.37	90.44	75.47
0.90	95.12	56.58	91.07	68.71
0.85	96.93	45.48	91.28	69.82
0.80	97.88	38.53	91.30	70.85
0.75	98.45	32.10	91.45	70.56
0.70	98.82	27.58	91.48	69.52

Table 3. BMSWebView2

Distortion Probability	Improved Distortion			
	fp = 0.98		fp = 0.9	
	P	A	P	A
0.95	83.69	84.71	79.78	86.71
0.90	88.52	85.18	83.93	88.32
0.85	89.97	82.67	86.42	90.42
0.80	91.16	80.94	90.02	91.07
0.75	92.53	81.31	91.77	92.64
0.70	93.44	81.80	92.85	93.82

Table 4. BMSWebView-2 results for various fp values

Distortion Probability	Improved Distortion			
	fp = 0.98		fp = 0.9	
	P	A	P	A
0.95	82.75	85.07	80.03	87.43
0.90	82.90	84.33	81.17	88.76
0.85	83.46	82.95	81.88	90.88
0.80	84.91	80.72	81.97	92.01
0.75	85.13	79.82	82.03	95.23
0.70	87.28	78.86	82.05	97.74

Table 5. Retail results for different values of fp

Dataset	**T_B (sec)**	**T_I (sec)**
Retail	300	43
BMS-Webview-1	134	20
BMS-Webview-2	184	37

Table 6 : Execution times of distortion algorithms

4 Related Work

Some effort has been made to address the problem of privacy preserving in data mining. Such investigation considers how much information can be inferred or calculated from large data repositories made available through data mining algorithms and looks for ways to minimize the leakage of information. This effort has been restricted basically to classification and association rules. In this work, we focus on the latter category.

Atallah et al. (Atallah et al. 1999) considered the problem of limiting disclosure of sensitive rules, aiming at selectively hiding some frequent itemsets from large databases with as little impact on other, non sensitive frequent item sets as possible. Specifically, the authors dealt with the problem of modifying a given database so that the support of a given set of sensitive rules, mined from the database, decreases below the minimum support value. This work was extended in (Dasseni et al. 2001), in which the authors investigated confidentiality issues of a broad category of association

rules. This solution requires CPU-intensive algorithms and, in some way, modifies true data values and relationships.

In the same direction, Saygin et al. (Saygin et al. 2001) introduced a method for selectively removing individual values from a database to prevent the discovery of a set of rules, while preserving the data for other applications. They proposed some algorithms to obscure a given set of sensitive rules by replacing known values with unknowns, while minimizing the side e ects on non-sensitive rules.

Related to privacy preserving in data mining, but in another direction, Evmievski et al. Evmievski et al. 2002) proposed a framework for mining association rules from transactions consisting of categorical items in which the data has been randomized to preserve privacy of individual transactions. Although this strategy is feasible to recover association rules and preserve privacy using a straightforward uniform randomization, it introduces some false drops and may lead a miner to find associations rules that are not supposed to be discovered.

In the context of distributed data mining, Kantarcioglu and Clifton (Kantarcioglu & Clifton 2002) addressed secure mining of association rules over horizontally partitioned data. This approach considers the discovery of associations in transactions that are split across sites, without revealing the contents of individual transactions.

This method is based on secure multi-party computation (Du & Atallah 2001) and incorporates cryptographic techniques to minimize the information shared, while adding little overhead to the mining task. In (Vaidya & Clifton 2002), Vaidya and Clifton addressed the problem of association rule mining in which transactions are distributed across sources. In this approach, each site holds some attributes of each transaction, and the sites wish to collaborate to identify globally valid associations rules. This technique is also based on secure multi-party computation. Although the papers mentioned above deal with privacy preserving in association rules, in this paper, our work is directly related to (Atallah et al. 1999, Dasseni et al. 2001, Saygin et al. 2001). Our work differs from the related work in some aspects, as follows:

First, the hiding strategies behind our algorithms deal with the problem 1 and 2 in Figure 1, and most importantly, they do not introduce the problem 3 since we do not add noise to the original data. Second, we study the impact of our hiding strategies in the original database by quantifying how much information is preserved after sanitizing a database. So, our focus is not only on hiding restrictive patterns but also on maximizing the discovery of patterns after sanitizing a database. More importantly, our sanitizing algorithms select sensitive transactions with the lowest degree of conflict and remove from them the victim item with specific criteria, while the algorithms in related work remove and/or add items from/to transactions without taking into account the impact on the sanitized database.

Third, our framework can achieve a reasonable performance since it is built on indexes. Another difference of our framework from the related work is that we \plug" a transaction retrieval search engine for searching transaction IDs through the transactional database efficiently.

5 Conclusions and Further Enhancement

In this paper, we have introduced a new framework for enforcing privacy in mining frequent patterns, which combines three advances for efficiently hiding restrictive rules: inverted les, one for indexing the transactions per item and a second for indexing the sensitive transactions per restrictive pattern; a transaction retrieval engine relying on Boolean queries for retrieving transaction IDs from the inverted file and combining the resulted lists; and a set of sanitizing algorithms.

This framework aims at meeting a balance between privacy and disclosure of information. In the context of our framework, the integration of the inverted le and the transaction retrieval engine are essential to speed up the sanitization process. This is due to the fact that these two modules feed the sanitizing algorithms with a set of sensitive transactions to be sanitized. It should be noticed that this index schema and the transaction retrieval engine are simple to be implemented and can deal with large databases without penalizing the performance since these two techniques are scalable.

The experimental results revealed that our algorithms for sanitizing a transactional database can achieve reasonable results. Such algorithms slightly alter the data while enabling flexibility for someone to tune them. In particular, the IGA algorithm reached the best performance, in terms of dissimilarity and preservation of legitimate frequent patterns. In addition, the IGA algorithm also yielded the best response time to sanitize the experimental dataset. Another contribution of this work includes three performance measures that quantify the fraction of mining patterns which are preserved in the sanitized database.

The Hiding Failure measures the amount of restrictive patterns that are disclosed after sanitization. Misses Cost measures the amount of legitimate patterns that are hidden by accident after sanitization, and Art factual Patterns measure the artificial patterns created by the addition of noise in the data. We evaluated such metrics by testing different values of the disclosure threshold for our algorithms. The work presented herein addresses the issue of hiding some frequent patterns from transactional databases.

All association rules derivable from these frequent patterns are thus also hidden. This could make the approach sometimes restrictive. For instance, if the pattern ABC is restricted, the pattern ABCD would also be restricted since it includes the previous one, and the association rule ABC→D would be hidden even though initially there was no Restrictions on D. There is no means to specify the constraints on the association rules rather than the frequent patterns. One may want to express that AB→ C is restricted but not C→AB. However, this is not feasible at the frequent patterns level since both rules are derived from the same frequent pattern ABC.

We are investigating new optimal sanitization algorithms that minimize the impact in the sanitized database. We are also investigating, in the context of privacy in data mining, association rules or other patterns, the integration of role-based access control in relational databases with rule-based constraints specifying privacy policies.

Acknowledgments

The authors would like to thank the anonymous reviewers for their careful reading and insightful comments that have helped in improving of this paper.

References

[1]. E.T. Wang, G. Lee, Y.T. Lin, "A novel method for protecting sensitive knowledge in association rules mining," 2005.

[2] . E.T. Wang, G. Lee, "An efficient sanitization algorithm for balancing information privacy and knowledge discovery in association patterns mining," 2008.

[3] . S.R.M. Oliveira, O.R. Zaıane, Y. Saygin, "Secure association rule sharing, advances in knowledge discovery and data mining, in: PAKDD2004.

[4] . Brian, C.S. Loh and Patrick, H.H. Then,2010 Second International Symposium on Data, Privacy, and ECommerce, IEEE.

[5]. GENG Bo,ZHONG Hong,PENG Jun,WANG Da-gang Temporal Rule Distribution Mining of Privacy-reserving,2008.

[6] . Jun Lin Lin, Yung Wei Cheng, "Privacy preserving ite mset mining 2009.

[7] . Agrawal, Shipra and Krishnan, Vijay and Haritsa, Jayant R (2004) On Addressing Efficiency Concerns in Privacy- Preserving Mining. Proceedings 4th International Conference

[8] . Chen, T. 2006. A Novel Method for Protecting Sensitive Knowledge in Association Rules Mining. In Proceedings of the Sixth international Conference on intelligent Systems Design and Applications (ISDA'06) - Volume 01 (October 16 - 18, 2006). ISDA. IEEE Computer Society, Washington, DC, 694-699.

[9] . Kantarcioglu, M. & Clifton, C. (2002), Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data, in `ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery', Madison, Wisconsin, USA.

[10]. Oliveira, S. R. M. & Za□□ane, O. R. (2002), A Framework for Enforcing Privacy in Mining Frequent Patterns, TR02-13, Department of Computing Science, University of Alberta, Canada.

[11] . Rizvi, S. J. & Haritsa, J. R. (2002), Maintaining Data Privacy in Association Rule Mining, in 28th International Conference on Very Large Data Bases', Hong Kong, China.

[12] . Saygin, Y., Verykios, V. S. & Clifton, C. (2001),`Using Unknowns to Prevent Discovery of Association Rules', SIGMOD Record 30(4), 45-54.

[13] . Vaidya, J. & Clifton, C. (2002), Privacy Preserving Association Rules Mining in Vertically Partitioned Data, in `8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', Edmonton, AB, Canada, pp. 639-644.

Authors

K.Srinivasa Rao Received **M.Tech** in Computer Science and Engineering from University College of Engineering ,**JNTU,Kakinada**.B.Tech in Computer Science & Engineering from JNTU,Hyderabad. And now presently working as Head of the Department of CSE, **Swarna Bharathi College of Engineering, Khammam**.His research interests includes DataWarehousing and DataMining,Mobile Computing and Image Processing,Computer Networks.

V.Chiranjeevi Received **M.Tech** in Software Engineering Kakatiya Institute of Technology & Science,Warangal .B.Tech in Computer Science and Engineering from JNTU, Hyderabad. And now presently working as Assistant Professor **Swarna Bharathi College of Engineering, Khammam** .His research interests includes Mobile Computing,Image Processing,DataMining ,Computer Networks and Embedded Systems.