A FRAMEWORK FOR HIGH PERFORMANCE AND STRICT PRIVACY PRESERVING DATA MINING ALGORITHMS

M. Prakash[1], G. Singaravel[2]

[1]Department of Computer Science and Engineering, K.S.R. College of Engineering,
Affiliated to Anna University, Chennai, Tiruchengode, Tamilnadu, (India)
[2] Department of Information Technology, K.S.R. College of Engineering,
Affiliated to Anna University, Chennai, Tiruchengode, Tamilnadu, (India)

ABSTRACT

The data collection and analysis has seen an unprecedented growth in the last couple of decades due to the advancements in use of technology. Many organizations and individuals generate huge amount of data through everyday activities. The generated data is either centralized for pattern identification or mined in a distributed fashion for efficient knowledge discovery and collaborative computation. This raises serious concerns about privacy issues. The data mining community has responded to this challenge by developing a new variety of algorithms that are privacy preserving. Most of the existing work in privacy preserving data mining fails to serve the purpose when applied to large real-world data mining applications. In this paper we develop a framework for privacy preserving data mining that allows personalization of privacy requirements for individuals in a large database and removes certain assumptions regarding participant behavior, thereby making the framework efficient and real-world adaptable.

Keywords: Data Mining, Data Publishing, Privacy Preserving, Privacy Preserving Techniques, Sensitive Data.

I. INTRODUCTION

Data mining is a technique that deals with the extraction of hidden knowledge from large database. It uses sophisticated algorithms for the process of sorting through large amounts of data sets and picking out relevant information [1]. With the amount of data doubling each year, more data is gathered and data mining is becoming an increasingly important tool to transform this data into information [2]. The applications of data mining [4] includes wide range of areas as, credit card fraud detection, financial forecasting, automatic abstracting, medical diagnosis, analysis of organic compounds etc [6].

Data mining deals with large database which can contain sensitive information. An individual’s private information is one of the example for sensitive information. It requires data preparation which can uncover information or patterns which may compromise confidentiality and privacy obligations [11]. Advancement of efficient data mining technique has increased the disclosure risks of sensitive data [5].

The threat to an individual's privacy comes into play when the data, once compiled, cause the data miner, or anyone who has access to the newly compiled data set, to be able to identify specific individuals, especially when originally the data were anonymous [9][10]. Providing security to sensitive data against unauthorized access has been a long term goal for the database security research community and for the government statistical agencies. Hence, the security issue has become, recently, a much more important area of research in data mining [6]. Therefore, in recent years, privacy preserving data mining has been studied extensively [7][8].

The paper is organized as follows: The framework for privacy preserving data mining is presented in the section 2. In section 3 strict privacy preserving data mining algorithms are described. The experimental analysis is discussed in section 4 and conclusions in section 6.

II. FRAMEWORK

In this section we introduce the notations which are to be used in this paper. The two languages which are the sets of formulas are considered, namely LA and LP where

- LA is used to state the assertions stored in the knowledge database
- LP is used to specify the private information that should be preserved (should not be revealed to the public)

Therefore

- A knowledge database KD is a finite set of formulas of LA
- A privacy P is a formula of LP
- The finite sets of privacy is denoted as FP
- The consequence relation between knowledge database and privacy are to be describes by ConRe which is

ConRe $\subseteq \beta_{fin}$ (LA) x LP

Where β_{fin} is finite power set operator

If a privacy P is a consequence of knowledge database KD, formally ConRe (KD, P), then this means that the private information can be inferred from the knowledge database. In other words, if any one or any agency has access to the knowledge database, then using the knowledge or information from the knowledge database one may derive the private information P, which means the privacy is revealed.

Elucidation – 1

The knowledge database KD preserves privacy with respect to a set of private information FP if for each P Є PF, then ¬ ConRe (KD, P). A user may have a priori knowledge, without accessing the knowledge database. This knowledge is known as background knowledge. The a priori knowledge is integrated in the consequence relation. The example given below shows how a priori knowledge is integrated in the consequence relation.

Example – 1

This example is concerned with medical information which is highly sensitive. The data set is taken from online which is freely available for research purpose. The real world data is not used here, because of the privacy issues of the real world data [1]. The collected data includes administrative data and sociodemographic information.

Administrative data

Insurance details, the length of stay in hospital, diagnosis details

Sociodemo graphic information

Age, gender, region of residence

Let’s assume that the knowledge database KD includes the following information or facts

- If a patient lives in region X, then the patient's diagnosis is flu, cancer or heart disease

RegionX → Flu V Cancer V Heart Disease

- A patient 1 does not receive a high cost treatment

¬ (Patient 1→ High Costs)

Also let's assume the following are a priori knowledge

- A cancer diagnosis does a high cost treatment

Cancer → High Costs

- Heart disease diagnosis entails a high cost treatment

Heart Disease → High Costs

- For a knowledge database T and a private information P, the consequence relation ConRel can be defined by

ConRel (T, P) if and only if T U { (E1), (E2) } ⊨ P

Where, ⊨ is the classical propositional logic for the entailment relation.

- For the knowledge database KD can find that

ConRel (KD, Patient 1→ Flu)

From the above consequence relation, we can conclude that if the agent or user has access to the knowledge database, then one may infer that the patient 1 has the diagnosis 'flu'. Therefore the privacy is violated because the diagnosis information about the patient is sensitive information. The point to be noted here is the knowledge database alone not allows for the conclusion Patient 1→ Flu. The knowledge database and a priori knowledge combine together lead the information leakage.

Elucidation – 2

A knowledge database KD with respect to a set of private information FP is to find a subset KD′ of KD is the distortion problem. To find a subset KD′ of KD such that KD′ preserves privacy with respect to FP and For all KD″ KD with card (KD′) card (KD″); Where that KD″ does not preserve privacy with respect to FP.

Hence such a knowledge database KD′ is a solution for the distortion problem.

A solution KD′ of the distortion problem for KD with respect to a set of private information FP is a maximal subset of KD that preserves the privacy for FP. Consider the example 2, if three facts from KD is removed, then obtained knowledge database does not preserve privacy with respect to

FP = {Patient 1 → Flu}

The solutions are maximal with respect to cardinality. Similarly another possible solution is to be maximal with respect to the subset relation.

Now For all KD″ KD with card (KD′) card (KD″); where that KD″ does not preserve privacy with respect to FP can be read as

For all KD″ KD with card (KD′) card (KD″); where that KD″ does not preserve privacy with respect to FP

However, this also leads to different solutions than the elucidation 2, as shown in the following example.

Example 2

Let KD consists of (E1), (E2), (E3) and the following

Patient lives in region X:

Patient →Region X (E6)

Patient 2's diagnosis is not the flu case

(Patient 2 → Flu) (E7)

Here two private information need to hide

FP = {Patient1 →Flu, Pateint2 → Highcosts}

Simple logical reasoning shows that

KD′ := KD\{(E1)} preserves privacy with respect to FP

However this in not the solution for distortion problems because the knowledge database with greater cardinality KD′ preserves privacy. Similarly KD″ is the solution with respect to subsets because with respect to FP. The point to be noted is, easy to find the solution with respect to subsets; simply can remove the element by elements from the knowledge database until the privacy is preserved. On the other hand finding solution with respect to cardinality is tougher.

III. STRICT PRIVACY PRESERVING DATA MINING ALGORITHMS

This distortion problem can be solved with Depth First Search (DFS) algorithm for a knowledge database. The basic logic or idea of a DFS is s follows

- Check whether KD is privacy preserving
- If yes, answer KD; otherwise remove an element from KD which produces a knowledge database KD′
- Check whether KD′ is privacy preserving
- If yes, answer KD′; otherwise remove an element from KD′ which produces a knowledge database KD″
- Check whether KD″ is privacy preserving
- If yes, answer KD″ ; otherwise remove an element from KD″ which produces a knowledge database KD‴
- Continues so on

This algorithm finds only a solution that is maximized with respect to the subset relation. In order to obtain the solution according to the elucidation discussed earlier that means maximized with respect to cardinality, the algorithm cannot return a privacy preserving knowledge database immediately, but it has to back trace and check whether it finds a larger one.

The following section provides the DFS procedure, which consists of three functions where the argument is a knowledge database.

- depthFirstSearch (KD_A)

It is a main function which calls doBoundedSearch (KD_A, -1, 0)

- doBoundedSearch (KD_A, bound, closed)

It is a recursive function, performs a DFS to obtain solution for the distortion problem. This only looks for solutions that contain closed as a subset

- isPrivacyPreserving (KD_A)

It tests whether KD_A is privacy preserving

It is to assume that the set of private information FP is globally available and hence we do not pass it as an argument to isPrivacyPreserving. Let us now give some comments on the implementation of doBoundedSearch. First, it is checked, whether KD_A is privacy preserving. If so, KD_A is returned; otherwise subsets of KD_A are searched as follows where the variable result stores the best solution found so far and bound stores its size. An element a of KD_A which does not belong to closed is removed from KD_A which gives a knowledge base C. If the size of C is greater than bound, then doBoundedSearch is called recursively with the arguments C, bound, and closed.

There it is important that closed is passed as value and not as reference. That is to guarantee that the call to doBoundedSearch does not change the value of closed. When this recursive call returns, all subsets of $KD_A \setminus \{a\}$ have been searched and if a privacy preserving subset has been found, the variables result and bound are updated. The assertion a is then added to closed, which means that in the following only subsets of KD_A that contain a will be searched. This process is iterated for all a in KD_A that do not belong to closed. We saw in Example 2 that the solution to the distortion problem need not be unique. Our algorithm, as we presented it, will only compute one solution but not produce the complete list of all solutions.

However, it is easy to adapt it such that it will answer all solutions. Instead of looking for privacy preserving subsets with size greater than the current bound the algorithm should look for possible solutions with size greater or equal than the current bound. If such a knowledge base has been found then

i) It is added to the list of solutions if its size equals the bound,

ii) The list of solutions is cleared (all its elements are dropped) and the newly found solution is added (this is then the only element of the list) if its size is greater than the current bound.

Algorithm 1: depthFisrstSearch (KD_A)

Requirement: KD_A is a knowledge database

Assure: Returns a solution to the distortion problem for KD_A with respect to a given set of private information FP

Assure: Returns null if no subset of KD_A preserves privacy with respect to FP

```
return doBoundSearch ($KD_A$, -1, 0)
```

Algorithm 2: doBoundedSearch(KD_A, bound, closed)

Requirement: DA is a knowledge base with card(KD_A) > bound then Bound is an integer Closed is a subset of KD_A

Assure: This returns a solution to the distortion problem for KD_A with respect to the given set of private information P contains more than bound elements and superset of closed.

Ensure: It returns null if no solution exists

```
if isPrivacyPreserving($KD_A$) then
    return $KD_A$
end if`
result ← null
B ← $KD_A$ \ closed
for all a ε B do
    c ← $KD_A$ \ { a }
    if card (C) > bound then
        subResult ← doBoundSearch (C, bound, closed)
    if subResult ≠ null then
        bound ← card (subResult)
        result ← subResult
    end if
    closed ← close U{ a }
end for
return result
```

Algorithm 3: isPrivacyPreserving(KD_A)

Requirement: KD_A is a knowledge database

Assure: it returns true if KD_A preserves privacy with respect to FP. Otherwise it returns false

```
privacyPreserving ← true
i ← 1
while privacyPreserving and I ≤ card (FP) do
    privacyPreserving ← not(ConRe($KD_A$, FPi))
    i ← i + 1
end while
return privacyPreserving
```

Algorithm 4: isPrivacyPreserving2(KD_A)

Requirement: KD_A is a knowledge database

Assure: returns (true, 0) if KD_A preserves privacy with respect to PF

Ensure: it returns (false, queue) otherwise where queue contains elements of KD_A that have been used in deviations of privacy

```
privacyPreserving ← true
queue ← 0
for all a ε  $KD_A$ do
    numberOfUses[a] ← 0
end for
for i := 1 . . . card (PF) do
    (result, list) ← conRe($KD_A$, FPi)
    if result then
        privacyPreserving ← false
        for all a ε list do
            numberOfUse[a] ← numberOfUse[a]+1
        end for
    end if
end for
for all a ε $KD_A$ do
    if numberOfUse[a] > 0 then
        add a to queue with priority numberOfUses[a]
    end if
end for
return (privacyPreserving, queue)
```

IV. EXPERIMENTAL ANALYSIS

The algorithm for doBoundedSearch performs well if it finds the 'right' elements to remove from the knowledge base as early as possible. That is it has to make a clever choice with which element a ε B it will start in the for all loop. So far there is no heuristic built into the algorithm that supports this choice: the loop may iterate in any order through B. Actually, there is no information available to make this an informed choice that selects the ‘right’ elements early. However, depending on the implementation of the decision procedure for ConRe, there exists a simple heuristic that can be added to our algorithm. We only need that if ConRe(KD; P) holds, then a call to ConRe(KD; P) returns a small subset KD′ of KD such that already ConRe(KD′; P) holds.

This is the case, for example, if the implementation of ConRe performs a proof search. That means a call to ConRe(KD; P) tries to construct a derivation of P from KD. If ConRe(KD; P) holds, it can provide a list of elements a_1, a_2, . . . a_m of KD that were actually used in the derivation of P. Since we are looking for a subset KD′′ of KD such that ConRe(KD′′; P) does not hold anymore, it seems to be a good choice to remove one of the a_i above from KD in order to construct a candidate for KD′′. If we removed an element b different from a_1, a_2, . . . a_m then the derivation of P would still be possible, that is ConRe(KD \ {b}; P) holds. Of course, removing an element a_i does not guarantee that ConRe(KD \ {a_i}; P) does not hold since there may be other derivations of P from KD that do not make use of a_i.

We are not only interested in preserving one secret but a whole set of secrets {P_1, P_2, P_3, . . . P_n}. Thus in Algorithm 3, we may not only check ConRe(KD_A; Pi) until privacy is violated; instead we can check ConRe(KD_A; P_i) for all 1 < i < n and keep track of which elements of KD_A are used most often to construct derivations of secrets or private information. If we build C by removing an element a_i of KD_A that has been used in several derivations, then chances are good that C will preserve several secrets.

Hence, Algorithm 3 should not return a boolean value but additionally also a priority queue of elements of KD_A that were used to build the derivations of the secrets. For that queue, the more derivations an element as been used in, the higher priority it receives. The for all loop in Algorithm 2 can make use of this queue to start with an element of maximal priority.

V. CONCLUSIONS

Huge amount of data is collected everyday by many organization and individuals. The collected data are mined for knowledge discovery using numerous data mining algorithms. This raises serious concerns about privacy issues. A framework is developed for privacy preserving data mining which features high performance and strict privacy preserving algorithms.

REFERENCES

[1] L. Sweeney, Privacy Preserving Bio-Terrorism Surveillance, AAAI Spring Symposium, AI Technologies for Homeland Security, 2005.

[2] E. Bertino, I. Fovino and L. Provenza, A Framework for Evaluating Privacy Preserving Data Mining Algorithms, Journal of Data Mining and Knowledge Discovery, 11(2), 2005, pp. 121–154.

[3] M. Prakash, and G. Singaravel, A New Model for Privacy Preserving Sensitive Data Mining, Proceedings of Third International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, 2012.

[4] S. Verykios, E. Bertino, I. Fovino, L. Provenza, Y. Saygin and Y. Theodoridis, State-of-the-art in Privacy Preserving Data Mining, ACM SIGMOD Record, 33(1), 2004, pp. 50–57.

[5] B. Pinkas, Cryptographic Techniques for Privacy Preserving Data Mining, ACM SIGKDD Explorations, 2002.

[6] M. Prakash, and G. Singaravel, A Review on Approaches, Techniques and Research Challenges in Privacy Preserving Data Mining, Australian Journal of Basic and Applied Sciences, 8(10), 2014, pp. 251-259.

[7] E. G. Komishani, and M. Abadi, A Generalization-Based Approach for Personalized Privacy Preservation in Trajectory Data Publishing, Proceedings of Sixth International Symposium on Telecommunications (IST'2012), IEEE, 2012.

[8] R. Agrawal, and R. Srikant, Privacy Preserving Data Mining, ACM SIGMOD Conference, 2000.

[9] Tiancheng Li and Ninghui Li, Slicing - A New Approach for Privacy Preserving Data Publishing, IEEE Transactions on Knowledge and Data Engineering, 24(3), 2012, pp. 561-574.

[10] L. Sweeney, k-Anonymity: A Model for Protecting Privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), 2002, pp. 557-570.

[11] M. Prakash, and G. Singaravel, An approach for prevention of privacy breach and information leakage in sensitive data mining, Journal of Computers and Electrical Engineering, In Press, DOI-http://dx.doi.org/10.1016/j.compeleceng.2015.01.016, 2015