International Journal of Research in Advent Technology, Vol.5, No.2,February 2017
E-ISSN: 2321-9637
Available online at www.ijrat.org

A Robust Privacy Preserving of Multiple and Binary Attribute by Using Supermodularity with Perturbation

Megha Dhiran[1], Prof. Raj kumar Paul[2]
Computer Science and Engineering, Vedica Institute of Technology
Email: megha.dhiranbe.it@gmail.com[1]

Abstract- The age of large database is now an big issue. So researchers try to develop a high performance *platform* to efficiently secured these kind of data before publishing. Here proposed work has resolve this issue of digital data security by finding the relation between the columns of the dataset which is based on the highly relative association patterns. Here use of supermodularity is also done which balance the risk and utilization of the data. Experiment is done on large dataset which have all kind of attribute for implementing proposed work features. Results are compare with previous existing techniques and it was obtained that proposed work was better on different evaluation parameters.

Index Terms- Privacy Preserving Mining, Association Rule Mining, Data Perturbation, Aggregation, Data Swapping

.

1 INTRODUCTION

Data mining methodology can help associating knowledge gaps in human understanding. Such as analysis of any student dataset gives a better student model yields better instruction, which leads to improved learning. More accurate skill diagnosis leads to better prediction of what a student knows which provides better assessment. Better assessment leads to more efficient learning overall. The main objectives of data mining in practice tend to be prediction and description [4, 5]. Predicting performance involves variables, IAT marks and

assignment grades etc. in the student database to predict the unknown values. Data mining is the core process of knowledge discovery in databases. It is the process of extracting of useful patterns from the large database. In order to analyze large amount of information, the area of Knowledge Discovery in Databases (KDD) provides techniques by which the interesting patterns are extracted. Therefore, KDD utilizes methods at the cross point of machine learning, statistics and database systems.

Different approach of mining is done for different type of data such as textual, image, video, etc. Information extraction is done in digital for resolving many issues. But some time this data contain information that is not fruitful for an organization, country, raise, etc. So before extraction such kind of information is remove. By doing this privacy for such unfair information is done. This is very useful for the security of data which contain some kind of medical information about the individual, financial information of family or any class. As this make some changes on the dataset, so present information in the dataset get modify and make it general for all class or rearrange so that miner not reach to concern person.

So privacy preserving mining consist of many approaches for preserving the information at various level form the individual to the class of items [3, 4]. But vision is to find the information from the dataset by observing repeated pattern present in the fields or data which can provide information of the individual, then perturb it by different methods such as suppression, association rules, swapping, etc.

2. RELATED WORK

In [14] present a hybrid discovery algorithm called HyFD, which combines fast approximation techniques with efficient validation techniques in order to find all minimal functional dependencies in a given dataset. While operating on compact data structures, HyFD not only outperforms all existing approaches, it also scales to much larger datasets.

Li et al (2013), problem of finding the minimal set of constants for conditional functional dependency present in used dataset. Here minimal set of conditional functional dependency is obtained by minimal generator as well as by clousers of those sets. Here proposed work has find the pruning criteria so overall work get reduce and unwanted generator, closures get shorten. So based on the proposed work a dataset modal is generate where each node act as a data row. Pruning of node is depending on two condition first is node have no conditional functional dependency rules. Second is descendent node of the node have no conditional functional dependency rules.

In [15] The discovery of functional dependencies from relations is an important analysis technique. We present TANE, a proficient algorithm for finding functional dependencies from larger databases. TANE is based on partitioning the sets of rows with respect to their attribute values which makes testing the validity of functional dependency fast even for big databases.

The results have shown that the algorithm is faster in use. It is observed that for benchmark databases the running times have improved.

In [16] original data is distributed among multiple parties. Here data is horizontally and vertically distribute by utilizing the random tree distribution with homomorpic schema distribution. So all party agree with schema of distributed tree. Here problem of building time is high with increase in number of attributes of the entity. Then data loss is next issue in this paper as schema construction is random so classification accuracy is less.

Yka Huhtala et. al. in [5], has proposed a work that generate conditional functional dependency and approximation rules by utilization of partitions. So by dividing the large dataset in to some partitions generation or searching of functional rules get easy and accurate.

Hong Yao [7] has developed an algorithm named as FDMine (Functional Dependency Mining). Here FD-Mine develops rules by utilizing the functional dependency properties in theory which reduce dataset size for searching as well as filter some of the unwanted or unfruitful rules. It has also proved in the work that pruning of rules not lead to loss of information in the work. Here whole work experiment is done on IS UCI datasets. Here pruning of rules are more as compare to previous works while evaluating results get improved.

3. PROPOSED WORK

3.1 Pre-Processing

As the dataset obtained from the above steps contain many unnecessary information which one need to be removed for making proper operation on those sets. This can be understood as let the name be the same as it is in the original set so to put this column in the original dataset is not necessary and it can be removed move from the above set of vectors, while if to hide information of the salary of the individual then one has to make changes from the original, therefore this kind of numeric data which need to be hide is perturbed by our method.

3.2 Multi-attribute Supermodularity

In this step whole multi attributes are replace by its hierarchy value in the supermodularity tree, while

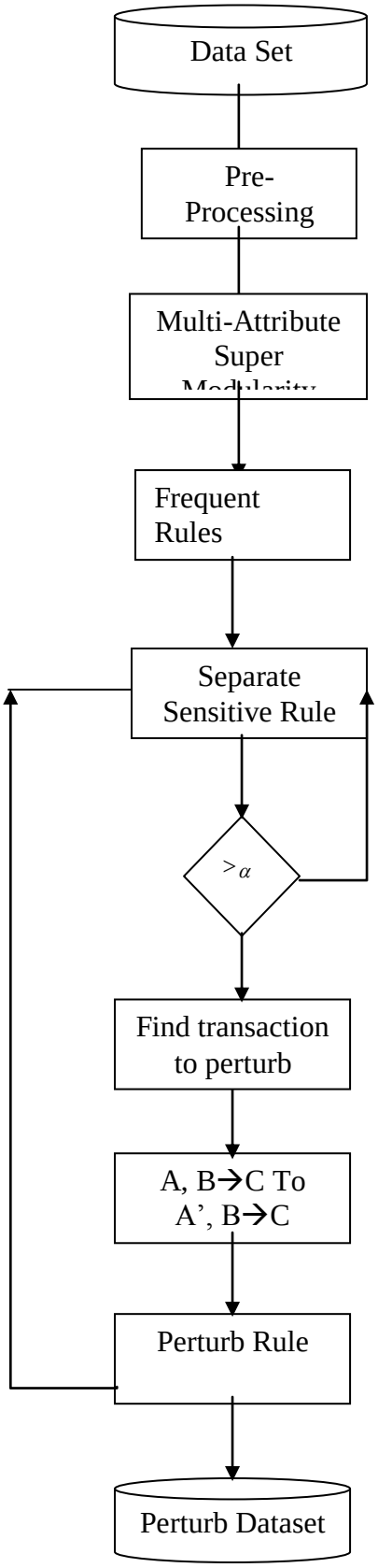

Fig. 1 Block diagram of proposed work.

replacing it is required to balance the dataset utility and risk by making required changes. This was done in [Base Paper]. This replacement is so designed that utility of the data get increase while risk remain below under some threshold value.

3.3 Generate Rules

In order to hide the information from the dataset one approach is to reduce the support and confidence of the desired item. For finding the item set which is most desired one has to find that the frequent pattern in the dataset. There are many approaches of pattern finding in the dataset which are most frequent one of the most popular is aprior algorithm.

3.4 Separate Sensitive Rule

Now from the generated rule one can get bunch of rules then it is required to separate those rules from the collection into sensitive and non- sensitive rule set. Those rules which contain sensitive items are identified as the sensitive rules while those not containing are indirect rules. This can be understood as the Let A, B →C where A is set of sensitive item then this rule is sensitive rule, where B, C are non sensitive items. If D, B→ C is a rule and D is the non sensitive item set the this rule is not sensitive rule.

3.5 Hide Sensitive Pattern:

So in order to hide an pattern, {X, Y), it can decrease its support to be smaller than user-specified minimum support transaction (MST). To decrease the support of a rule, there is a approach: Decrease the support of the item set {X ,Y}. For this case, by only decrease the support of Y, the right hand side of the rule, it would reduce the support faster than simply reducing the support of {X , Y}. To decrease the confidence of a rule, there is two approach:

(1) Increase the support of X, the left hand side of the rule, but not support of X → Y.

(2) Decrease the support of the item set X →Y .For the second case, if we only decrease the support of Y, the right hand side of the rule, it would reduce the confidence faster than simply reducing the support of X → Y.

Here it only reduce the RHS item Y of the rule correspondingly. So for the rule Bread→ Milk can generate reduce the support of Y only. Now it need to find that for how many transaction this need to be done. So calculation of that number is done by

Here it only reduce the RHS item Y of the pattern correspondingly. So for the pattern {Bread, Milk} can generate reduce the support of Y only. Now it need to find that for how many transaction this need to be done. So calculation of that number is done by

((Rule_support – Minimum_ support) * Total_transaction)/100

Above formula specify the number of transaction where one can modify and overall support of that hiding pattern is lower then the minimum support.
Table 4.3: Number of session to hide sensitive dataset.

3.6 Proposed Algorithm

For this algorithm t is a transaction, T is a set of transactions, P is used for pattern, RHS (R) is Right Hand Side of rule R, LHS (R) is the left hand side of the pattern P, support (S) is the rule R, a set of items H to be hidden.

Hiding Rules Algorith

Input: A source database D, A minimum support in_support (MST).
Output: The sanitized database D, where rules containing X on Left Hand Side (LHS) or Right Hand Side (RHS) will be hidden.

Steps of algorithm:

1. P[c] ← Apriorr(D) // s = support

2. Loop I = For each P

3. If Intersect(P[I], H) and P[I] > MST

4. New_transaction ← Find_transaction(P[I], MST)

5. While (T is not empty OR count = New_treansaction)

6. If t←T have XUY rule then

7. Remove Y from this transaction

8. End While

9. EndIf

10. End Loop

In proposed algorithm input is original dataset (DS), MST threshold and output contain perturbed dataset (PDS). In whole algorithm frequent rules (FR) are generated then rules are filter by sensitive rule. Then in-order to suppress those discriminating rules (DR) find number of sessions to perturb and perturb those session where those item set is present.

4 EXPERIMENT AND RESULT

This section presents the experimental evaluation of the proposed perturbation and de-perturbation technique for privacy prevention. All algorithms and utility measures were implemented using the MATLAB tool. The tests were performed on an 2.27 GHz Intel Core i3 machine, equipped with 4 GB of RAM, and running under Windows 7 Professional.

4.1 Dataset

In order to analyze proposed algorithm, it is in need of the dataset. One grocery shop dataset is use that has following attribute {items, date_of_birth, gender, salary}. Here personal information are from date_of_birth, gender, salary. While sensitive items are important for the Shop owner. So for the privacy preservation both things need to be hide. So in ordr to provide protection against the private data of the customer one concept of K-Anonymity has been include which make multiple copy of the same customer with different values. Then for hiding the useful or sensitive data transaction, in other words the most frequent item set association rules are find and hide them. This work can provide privacy to those datasets only which have the pattern generation values in the transactions. In this dataset it contain different item set such as jeans, T-shirt, shoes, etc. This data set consists of 20,000 records. The data set has 14 attributes (without class attribute).

4.2 Evaluation Parameters

Lost Patterns: Representing the number of non-sensitive patterns (i.e., classification patterns) which are hidden as side-effect of the hiding process

False Patterns: Representing the number of art factual patterns created by the adopted privacy preserving technique.

Missed Pattern: Representing the number of Sensitive patterns still present in dataset even after applying adopted privacy preserving technique.

Privacy Percentage: This specify the percentage of the privacy provide by the adopting technique.

4.3 Results

Table. 1. Represent comparison of proposed and previous work on the basis of Lost Patterns.

Support	**Lost Patterns Percentage**	
	Previous work	**Proposed Work**
14	0	0
2	0	0
3	0	0
4	0	0
5	0	0

From table 1 it is obtained that proposed work has not affect non sensitive patterns in the dataset. While previous work do not apply any approach for pattern preservation so no affect on those patterns are present after previous approach.

Table. 2. Represent comparison of proposed and previous work on the basis of False Patterns.

Support	**False Patterns Percentage**	
	Previous work	**Proposed Work**
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0

From table 2 it is obtained that proposed work has not generate any sensitive as well non sensitive patterns in the dataset. While previous work do not apply any approach for pattern preservation so no affect on those patterns are present after previous approach.

Table. 3. Represent comparison of proposed and previous work on the basis of Missed Patterns.

Support	**Missed Patterns Percentage**	
	Previous work	**Proposed Work**
1	100	0
2	100	0
3	100	0
4	100	0
5	100	0

From table 3 it is obtained that proposed work has not preserve all sensitive patterns in the dataset. While previous work do not apply any approach for pattern preservation so no affect on those patterns are present after previous approach. Here all sensitive information is hide in proposed work.

5 CONCLUSION

In this work, a set of algorithms and techniques were proposed to solve privacy-preserving data mining problems. The experiments showed that the proposed algorithms perform well on large databases. It work better as the Maximum lost pattern percentage is zero a certain value of support. Then this work shows that false patterns value is zero. Comparison with the other algorithm it is obtained that including the differential privacy and then directly hide the sensitive information. It is shown in the results that accuracy of the perturbed dataset is preserved for low support values as well. Here Proposed work has resolve the multi party data distribution problem as well as different level trust party get different level of perturbed dataset copy.

REFERENCES

[1] Abedjan, Z., Grütze, T., Jentzsch, A., Naumann, F.: Mining And Profiling RDF Data With Prolod++. In: Proceedings Of The International Conference On Data Engineering (ICDE), Pp. 1198–1201(2014).

[2] Rostin, A., Albrecht, O., Bauckmann, J., Naumann, F., Leser,U.: A Machine Learning Approach To Foreign Key Discovery. In: Proceedings Of The ACM SIGMOD Workshop On The Web And Databases (Webdb) (2009)

[3] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schonberg, Jakob Zwiener And Felix Naumann, "Functional Dependency Discovery: An Experimental Evaluation Of Seven Algorithms", Proceedings Of VLDB 2015.

[4] Mohamed R. Fouad, Khaled Elbassioni, Member, IEEE, And Elisa Bertino . "A Supermodularity-Based Differential Privacy Preserving Algorithm For Data Anonymization". IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 7, JULY 2014.

[5] Huhtala, Y., Karkkainen, J., Porkka, P., And Toivonen, Dependencies Using Partitions, IEEE ICDE 1998.

[6] Shyue-Liang Wang, Jenn-Shing Tsai And Been-Chian Chien, "Mining Approximate Dependencies Using Partitions On Similarity-Relation-Based Fuzzy Databases", IEEE International Conference On Systems, Man And Cybernetics(SMC) 1999.

[7] Yao, H., Hamilton, H., And Butz, C., FD_Mine: Discovering Functional Dependencies In A Database Using Equivalences, Canada, IEEE ICDM 2002.

[8] Wyss. C., Giannella, C., And Robertson, E. (2001), Fastfds: A Heuristic-Driven, Depth-First Algorithm For Mining Functional Dependencies From Relation Instances, Springer Berlin Heidelberg 2001.

[9] Russell, Stuart J. And Norvig, Peter. Arti Cial Intelligence: A Modernapproach. Prentice Hall, 1995.

[10] Mannila, H. (2000), Theoretical Frameworks For Data Mining, ACM SIGKDD Explorations, V.1, No.2, Pp.30-32.

[11] Stephane Lopes, Jean-Marc Petit, And Lotfi Lakhal, "Efficient Discovery Of Functional Dependencies And Armstrong Relations", Springer 2000.

[12] Heikki Mannila And Kari-Jouko R¨Aih¨A. Design By Example: An Application Of Armstrong Relations. Journal Of Computer And System Sciences, 33(2):126{141, 1986.

[13] Wenfei Fan, Jianzhong Li, Nan Tang, And Wenyuan Y. "Incremental Detection Of Inconsistencies In Distributed Data". Ieee Transactions On Knowledge And Data Engineering, Vol. 26, No. 6, June 2014 1367

14. Thorsten Papenbrock, Felix Naumann ." A Hybrid Approach To Functional Dependency Discovery". SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA C 2016 ACM. ISBN 978-1-4503-3531-7/16/06. .

15. Akshay Kulkarni, Sachin Batule, Manoj Kumar Lanke, Adityakumar Gupta. "Functional Dependencies Discovery In RDBMS". International Journal Of Advanced Research In Computer Science And Software Engineering Volume 6, Issue 4, April 2016 ISSN: 2277 128X.

[16] Jaideep Vaidya, Senior Member, IEEE, Basit Shafiq, Member, IEEE, Wei Fan, Member, IEEE, Danish Mehmood, And David Lorenzi. "A Random Decision Tree Framework For Privacy-Preserving Data Mining" . IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 11, NO. 5, SEPTEMBER/OCTOBER 2014