

Perturbation Approach for Protecting Data Server used for Decision Tree Mining

¹T. Nirosh Kumar, ²G. Jaya Raju

^{1,2}Dept. of CSE, Lenora College of Engineering, Rampachodavaram, AP, India

Abstract

Data Mining is the step by step process for extracting interesting rules from large amount of data. The data can be stored at database server, file, data warehouse, and the data servers must be protected from an authenticated person access. Decision tree mining is one of the classification algorithms that construct rules from centralized data set. This paper gives description for how to protect data in the server used for decision tree mining and description of perturbation privacy preserving algorithm. It constructs two data sets unrealized equivalent original data set. Here the main idea is the decision tree derived from original data set same as the decision tree derived from unrealized data set. The experiment results shows that approach out performs the other techniques

Keywords

Data Mining, Decision Tree Mining, Privacy

I. Introduction

Data mining is widely used by researchers for science and business purposes. Data collected (referred to as “sample data sets” or “samples” in this paper) from individuals (referred to in this paper as “information providers”) are important for decision making or pattern recognition. Therefore, privacy-preserving processes have been developed to sanitize private information from the samples while keeping their utility.

A large body of research has been devoted to the protection of sensitive information when samples are given to third parties for processing or computing [1-5]. It is in the interest of research to disseminate samples to a wide audience of researchers, without making strong assumptions about their trustworthiness. Even if information collectors ensure that data are released only to third parties with non malicious intent (or if a privacy preserving approach can be applied before the data are released), there is always the possibility that the information collectors may inadvertently disclose samples to malicious parties or that the samples are actively stolen from the collectors). Samples may be leaked or stolen anytime during the storing process [6-7] or while residing in storage [8-9]. This paper focuses on preventing such attacks on third parties for the whole lifetime of the samples. We introduce a new perturbation and randomization based approach that protects centralized sample data sets utilized for decision tree data mining. Privacy preservation is applied to sanitize the samples prior to their release to third parties in order to mitigate the threat of their inadvertent disclosure or theft. In contrast to other sanitization methods, our approach does not affect the accuracy of data mining results. The decision tree can be built directly from the sanitized data sets, such that the originals do not need to be reconstructed. Moreover, this approach can be applied at any time during the data collection process so that privacy protection can be in effect even while samples are still being collected. The following assumptions are made for the scope of this paper: first, as is the norm in data collection processes, a sufficiently large number of sample data sets have been collected to achieve significant data mining results covering the whole research target. Second, the number of data sets leaked to potential attackers constitutes a small portion of

the entire sample database. Third, identity attributes (e.g., social insurance number) are not considered for the data mining process because such attributes are not meaningful for decision making. Fourth, all data collected are discredited; continuous values can be represented via ranged value attributes for decision tree data mining.

II. Related work

In Privacy Preserving Data Mining: Models and Algorithms [10], Agrawal and Yu classify privacy preserving data mining techniques, including data modification and cryptographic, statistical, query auditing and perturbation-based strategies. Statistical, query auditing and most cryptographic techniques are subjects beyond the focus of this paper. In this section, we explore the privacy preservation techniques for storage privacy attacks. Data modification techniques maintain privacy by modifying attribute values of the sample data sets. Essentially, data sets are modified by eliminating or unifying uncommon elements among all data sets. These similar data sets act as masks for the others within the group because they cannot be distinguished from the others; every data set is loosely linked with a certain number of information providers. k-anonymity [11] is a data modification approach that aims to protect private information of the samples by generalizing attributes. k-anonymity trades privacy for utility. Further, this approach can be applied only after the entire data collection process has been completed. Perturbation-based approaches attempt to achieve privacy protection by distorting information from the original data sets. The perturbed data sets still retain features of the originals so that they can be used to perform data mining directly or indirectly via data reconstruction. Random substitutions [12] is a perturbation approach that randomly substitutes the values of selected attributes to achieve privacy protection for those attributes, and then applies data reconstruction when these data sets are needed for data mining. Even though privacy of the selected attributes can be protected, the utility is not recoverable because the reconstructed data sets are random estimations of the originals. Most cryptographic techniques are derived for secure multiparty computation, but only some of them are applicable to our scenario. To preserve private information, samples are encrypted by a function, f , (or a set of functions) with a key, k , (or a set of keys); meanwhile, original information can be reconstructed by applying a decryption function, f^{-1} , (or a set of functions) with the key, k , which raises the security issues of the decryption function(s) and the key(s). Building meaningful decision trees needs encrypted data to either be decrypted or interpreted in its encrypted form. The (anti)monotone framework [13] is designed to preserve both the privacy and the utility of the sample data sets used for decision tree data mining. This method applies a series of encrypting functions to sanitize the samples and decrypts them correspondingly for building the decision tree. However, this approach raises the security concerns about the encrypting and decrypting functions. In addition to protecting the input data of the data mining process, this approach also protects the output data, i.e., the generated decision tree. Still, this output data can normally be considered sanitized because

it constitutes an aggregated result and does not belong to any individual information provider. In addition, this approach does not work well for discrete-valued attributes.

III. Problem Description

Frame Work and Algorithm

A. Frame Work

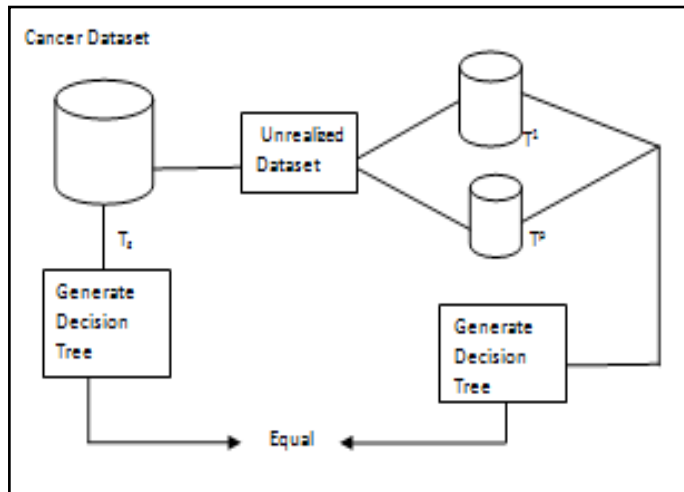


Fig. 1:

- In this data mining the cancer data set is converted into unrealized dataset.
- After that these unrealized dataset is converted into training dataset(T1) and perturbing sets.
- Again both these T1 and Tp are decision tree generated.
- Simultaneously the cancer dataset is also generate decision tree.
- In both cases generate decision trees are compare to be equal.

B. Decision Tree Using Gini Index:

Gini index builds decision trees from a set of training data in the same way as id3, using the concept of information entropy. The training data is a set $s = s_1, s_2$, of already classified samples. Each sample $s_i = x_1, x_2$, is a vector where x_1, x_2 , represent attributes or features of the sample. The

Training data is augmented with a vector $c = c_1, c_2$, where c_1, c_2 , represent the class that each sample belongs to. Gini index uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets.

Gini index examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller sub lists.

C. Decision Tree using C4.5 Algorithm

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = sbs_2$, of already classified samples. Each sample $S_i = X_1, X_2$, is a vector where x_1, x_2 , represent attributes or features of the sample. The training data is augmented with a vector $C = cbc_2$,

Where cbc_2 , represent the class that each sample belongs to. C4.5 uses the fact that each attribute of

The data can be used to make a decision that splits the data into

smaller subsets. C4.5 examines the normalized information gain (difference in entropy)

That results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision.

The algorithm then recurs on the smaller subsists. This algorithm has a few base cases; the most common base case is when all the samples in your list belong to the same class. Once this happens, you simply create a leaf node for your decision tree telling you to choose that class. It might also happen that none of the features give you any information gain; in this case C4.5 creates a decision node higher up the tree using the expected value of the class. It also might

Happen that you've never seen any instances of a class; again, C4.5 creates a decision node higher up the tree using expected value. Improvements from ID3 algorithm C4.5 made a number of improvements to ID3. Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value

Is above the threshold and those that are less than or equal to it. Handling training data with missing attribute values - C4.5 allows attribute values to be marked as? For missing. Missing attribute values are

Simply not used in gain and entropy calculations. Handling attributes with differing costs. Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes. C4.5 is

An extension of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on. C4.5 Algorithm Check for base cases For each attribute 'a' Find the normalized information gain from splitting on 'a' Let 'a' best be the attribute with the highest normalized information gain Create a decision node that splits on a best Recur on the sub lists obtained by splitting on a best and add chose nodes as children of node.

1. Unrealized Training Set

Traditionally, a training set, TS, is constructed by inserting sample data sets into a data table. However, a data set complementation approach, as presented in this paper, requires an extra data table, TP. TP is a perturbing set that generates unreal data sets which are used for converting the sample data into an unrealized training set, T0. The algorithm for un realizing the training set, TS, is shown as follows:

Algorithm

Un realize-Training-Set (T_s, T^u, T^1, T^p)

Input: TS, a set of input sample data sets

T^u , a universal set

T^1 , a set of output training data sets

T^p , a perturbing set

Output: (T^1, T^p)

1. If TS is empty then return (T^1, T^p)
2. $t \leftarrow$ a data set in T_s
3. if t is not an element of T^p or $T^p = \{t\}$ then
4. $T^p \leftarrow T^p + T^u$
5. $T^p \leftarrow T^p - \{t\}$
6. $t^1 \leftarrow$ the most frequent dataset in T^p
7. return Un realize-Training Set ($T_s - \{t\}, T^u, T^1 + \{t^1\}, T^p - \{t^1\}$)

To un realize the samples, T_s , we initialize both T^1 and T^p as empty sets, i.e., we invoke the above algorithm with Un realize-Training-Set ($T^s, T^u, \{\}, \{\}$). Figs. 2(b) and 2(c) show the tables that result from the un realizing process of the samples in fig. 2(a). The resulting unrealized training set contains some dummy data sets excepting the ones in T_s . The elements in the resulting data sets are unreal individually, but meaningful when they are used together to calculate the information required by a modified ID3 algorithm,

2. Decision Tree Generation

The well-known ID3 algorithm shown above builds a decision tree by calling algorithm Choose-Attribute recursively. This algorithm selects a test attribute (with the smallest entropy) according to the information content of the training set T_s .

Algorithm

Generate-Tree (T_s , attribs, default)

Input: T_s , the set of training data sets attribs, set of attributes default, default value for the goal predicate

Output: tree, a decision tree

1. if T_s is empty then return default
2. default $\leftarrow$ Majority- Value(T_s)
3. if $H_{A_i} T_s = 0$ then return default
4. else if attribs is empty then return default
5. else
6. best $\leftarrow$ Choose-Attribute (attribs, T_s)
7. tree $\leftarrow$ a new decision tree with root attribute best
8. for each value v_i of best do
9. $T_{s_i} \leftarrow \{\text{datasets in } T_s \text{ as best} = k_i\}$
10. subtree $\leftarrow$ Generate-Tree(T_{s_i} , attribs-best, default)
11. connect tree and subtree with a branch labelled k_i
12. return tree

Results

LOGIN

USERID

PASSWORD

Fig. 2: Design Preview [Privacypreservation View]

PRIVACY PRESERVING DECISION TREE LEARNING USING UNREALIZED DATA SETS

Fig. 3: Design Preview [Welcome]

DATASET UPLOAD

FILE NAME

Fig. 4: Design Preview [Data Set Upload]

DATASET UPLOAD

FILE NAME

File Uploaded Sucessfully

Fig. 5: Design Preview [Data Set Upload]

DATA SET GENERATION

NO OF ATTRIBUTES

NAME OF TYPE ATTRIBUTES

SNO	ATTRIBUTE	POSSIBLE VALUES
1		
2		
3		
4		
5		

Fig. 6: Design Preview [Data Set Generation]

RANDOM DATA SET GENERATION

NO OF TUPLES

TYPE OF DISTRIBUTION

NORMALLY
NORMALLY
EVENLY
UNEVENLY

Fig. 7: Design Preview [Random Data Set Generation]

SOURCE DATA SET GENERATED(T_s)

SNO	Attrib1	Attrib2	Attrib3

Fig. 8: Design Preview [Source Data Set Generated (T_s)

UNIVERSAL DATA SET

SNO	Attrib1	Attrib2	Attrib3

Fig. 9: Design Preview [Universal Data Set]

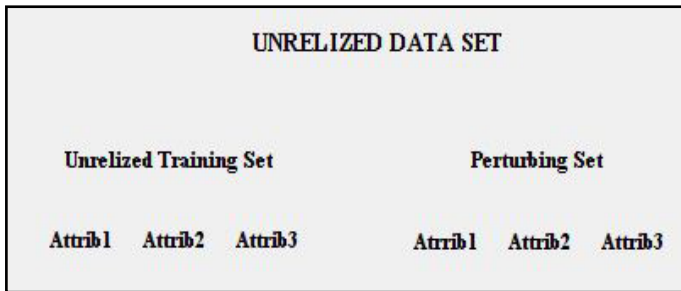


Fig. 10: Design Preview [Unrelized Data Set]

VI. Conclusion

This paper described a new privacy preserving approach via data set complementation which confirms the utility of training data sets for decision tree learning. This approach converts the sample data sets, TS, into some unreal data such that any original data set is not re constructible if an unauthorized party were to steal some portion of. Meanwhile, there remains only a low probability of random matching of any original data set to the stolen data sets, TL.

This paper also covers the application of this new privacy preserving approach with other algorithms, such as C4.5 and C5.0, and data mining methods with mixed discretely—and continuously valued attributes.

References

- [1] S. Ajmani, R. Morris, B. Liskov, "A Trusted Third-Party Computation Service", Technical Report MIT-LCS-TR-847, MIT, 2001.
- [2] S.L. Wang, A. Jafari, "Hiding Sensitive Predictive Association Rules", Proc. IEEE Int'l Conf. Systems, Man and Cybernetics, pp. 164- 169, 2005.
- [3] R. Agrawal, R. Srikant, "Privacy Preserving Data Mining", Proc. ACM SIGMOD Conf. Management of Data (SIGMOD '00), pp. 439-450, May 2000.
- [4] Q. Ma, P. Deng, "Secure Multi-Party Protocols for Privacy Preserving Data Mining", Proc. Third Int'l Conf. Wireless Algorithms, Systems, and Applications (WASA '08), pp. 526-537, 2008.
- [5] J. Gitanjali, J. Indumathi, N.C. Iyengar, N. Sriman, "A Pristine Clean Cabalistic Foruity Strategize Based Approach for Incremental Data Stream Privacy Preserving Data Mining", Proc. IEEE Second Int'l Advance Computing Conf. (IACC), pp. 410-415, 2010.
- [6] N. Lomas, "Data on 84,000 United Kingdom Prisoners is Lost", Retrieved Sept. 12, 2008, [Online] Available: http://news.cnet.com/8301-1009_3-10024550-83.html, Aug. 2008.
- [7] BBC News Brown Apologises for Records Loss. Retrieved Sept. 12, 2008, [Online] Available: http://news.bbc.co.uk/2/hi/uk_news/politics/7104945.stm, Nov. 2007.
- [8] D. Kaplan, Hackers Steal 22,000 Social Security Numbers from Univ. of Missouri Database, Retrieved Sept. 2008, [Online] Available: <http://www.scmagazineus.com/Hackers-steal-22000-Social-Security-numbers-from-Univ.-of-Missouri-database/article/34964/>, May 2007.
- [9] D. Goodin, "Hackers Infiltrate TD Ameritrade client Database", Retrieved Sept. 2008, [Online] Available: http://www.channelregister.co.uk/2007/09/15/ameritrade-database_burgled/, Sept. 2007.
- [10] C. Aggarwal, P. Yu, "Privacy-Preserving Data Mining", Models and Algorithms. Springer, 2008.

- [11] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy", Int'l J. Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 10, pp. 557-570, May 2002.
- [12] J. Dowd, S. Xu, W. Zhang, "Privacy-Preserving Decision Tree Mining Based on Random Substitutions", Proc. Int'l Conf. Emerging Trends in Information and Comm. Security (ETRICS '06), pp. 145-159, 2006.
- [13] S. Bu, L. Lakshmanan, R. Ng, G. Ramesh, "Preservation of Patterns and Input-Output Privacy", Proc. IEEE 23rd Int'l Conf. Data Eng., pp. 696-705, Apr. 2007.



T. Nirosh Kumar received the Bachelor's Degree in Computer Science from the Aditya Engineering College, Affiliated to Jawahar Lal Nehru Technological University, Hyderabad, in 2008. He is pursuing M.Tech Degree in the Department of Computer Science, Lenora College of Engineering at Rampachodavaram, East Godavari Dt, AP. His interests include Data Mining and Cloud Computing.



G. Jaya Raju received the Master's Degree in Computer Science from the Aditya Engineering College, Affiliated to Jawahar Lal Nehru Technological University, Kakinada, in 2011. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Lenora College of Engineering at Rampachodavaram, East Godavari Dt, AP. His interests include Data Mining, Networking, Mobile Computing and

Cloud Computing.