RESEARCH PAPER | Computer Science | Volume : 3 | Issue : 9 | Sept 2013 | ISSN - 2249-555X

An Efficient Technique for Privacy Preserving Decision Tree Learning

KEYWORDS: Data Mining, Classification, Decision tree learning, ID3, C4.5

Dr. S.Vijayarani
Assistant Professor, School of Computer Science and Engineering, Bharathiar University, Coimbatore

M. Sangeetha
Research Scholar, School of Computer Science and Engineering, Bharathiar University, Coimbatore

ABSTRACT *Data mining helps to extract hidden predictive information from large databases. There are several techniques and algorithms used for extracting the hidden patterns from the large data sets and finding the relationships between them. Privacy preservation is an important factor in data mining. The problem of privacy preservation in data mining has become more important in recent years because of increasing need to store vast data about users. In this research work, a new privacy preserving approach is applied to decision tree learning. This approach converts the original sample datasets into a group of unreal datasets. The original sample datasets cannot be reconstructed from it. Meanwhile, an accurate decision tree is built using those unreal datasets. C4.5 algorithm is used to build decision tree. The experimental results show that accurate and efficient decision tree is built in C4.5 algorithm than existing algorithm.*

1. INTRODUCTION

Data mining is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [1]. In data mining and machine learning, preserving of privacy is important process. Privacy-preserving processes have been developed to sanitize private information from the samples while keeping their utility.

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. Many privacy protection approaches preserve private information of sample datasets, but not precision of data mining outcomes. Hence, the utility of the sanitized datasets is downgraded.

This paper provides an approach that preserves privacy and utility of sample datasets for decision-tree data mining. 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.

This approach converts original samples into a group of unreal datasets from which the original samples cannot be reconstructed without the entire group of unreal data sets. Meanwhile, an accurate decision tree can be built directly from those unreal data sets.

This paper is organized as follows: Section 2 explains a brief discussion about the decision tree learning. Section 3 provides discussion on the previous works related to the topic. Section 4 describes the existing approaches of decision tree learning and the proposed algorithm for decision tree learning. Section 5 involves the Conclusion and future works.

2. DECISION TREE LEARNING

Decision tree learning is one of the most widely used and practical methods for inductive inference. Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. Data comes in records of the form:

$(X, Y) = (x_1, x_2, x_3, ..., x_k, Y)$

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector x is composed of the input variables, x_1, x_2, x_3 etc., that are used for that task.

Decision trees used in data mining are of two main types:

- Classification tree analysis is when the predicted outcome is the class to which the data belongs.
- Regression tree analysis is when the predicted outcome can be considered a real number.

DECISION TREE LEARNING ALGORITHM

Most Algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. This approach is exemplified by the ID3 algorithm (Quinlan 1986) and it's Successor C4.5 (Quinlan 1993).

ID3 ALGORITHM

ID3 is a nonincremental algorithm, meaning it derives its classes from a fixed set of training instances. The classes created by ID3 are inductive, that is, given a small set of training instances, the specific classes created by ID3 are expected to work for all future instances. The distribution of the unknowns must be the same as the test cases. Induction classes cannot be proven to work in every case since they may classify an infinite number of instances.

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 $S=s_1, s_2...$ of already classified samples. Each sample s_i consists of a p-dimensional vector, $(x_{1,i}, x_{2,i}, ..., x_{p,i})$ where the x_j represent attributes or features of the sample, as well as the class in which s_i falls.

The general algorithm for building decision tree is:

1. Check for base cases
2. For each attribute a
1. Find the normalized information gain from splitting on a
3. Let a' be the attribute with the highest normalized information gain
4. Create a decision node that splits on a'
5. Recurse on the sublists obtained by splitting on a', and add those nodes as children of node

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain .The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sublists.

Volume : 3 | Issue : 9 | Sept 2013 | ISSN - 2249-555X

This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

3. RELATED WORKS

A wide research has been devoted to the protection of sensitive information when samples are given to third parties for processing or computing [2], [3], [4], [5], [6]. Samples may be leaked or stolen anytime during the storing process or while residing in storage. This paper focuses on preventing such attacks to the samples by third parties.

Contemporary research in privacy preserving data mining mainly falls into one of two categories: 1) perturbation and randomization-based approaches, and 2) secure multiparty computation (SMC)-based approaches [7].SMC approaches employ cryptographic tools for collaborative data mining computation by multiple parties. Samples are distributed among different parties and they take part in the information computation and communication process. SMC research focuses on protocol development [8] for protecting privacy among the involved parties [9] or computation efficiency [10]; however, centralized processing of samples and storage privacy is out of the scope of SMC.

This approach 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. 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.

4. PROBLEM DEFINITION & PROPOSED METHODOLOGY

In this paper we have proposed a privacy preserving approach that can be applied to decision tree learning. This approach converts the original sample datasets into a group of unreal datasets. The original sample datasets cannot be reconstructed from it. Meanwhile, an accurate decision tree is built, using C4.5 algorithm, from those unreal datasets.

- Unrealized dataset conversion
- Decision Tree Generation
- Distribution
- Comparison

UNREALIZED DATASET CONVERSION:

For conversion of unrealized dataset, we use the algorithm of unrealized training set. Data modification techniques maintain privacy by modifying attribute values of the sample data sets. For this process K-anonymity approach is used for the modification purpose. In this process datasets are inserted into the data table. Data unrealization algorithm is used for this process. Inserted dataset are unreal dataset.

First we load the universal set and the sample set. This sample set and universal set is implemented by the unrealized training set algorithm. Finally the output of the unrealized data set is training set and perturbation set. T^U, the universal set of data table T, is a set containing all possible datasets in data table T. Let T associates with attributes <Wind, Play> where Wind= {Strong, Weak}

and Play= {Yes, No} then T^U= {<Strong, Yes>, <Strong, No>, <Weak, Yes>, <Weak, No>}. T_S is constructed by inserting sample data sets into a data table. T^P is a perturbing set that generates unreal datasets which is used for converting T_S into unrealized training set T'.

Algorithm Unrealize-Training-Set (T_S,T^U, T' , T^P)

Input: T_S , a set of input sample data sets

T^U , a universal set

T' , a set of output training data sets

T^P, a perturbing set

Output: T' , T^P

1. if T_S is empty then return(T', T^P)
2. t ← a data set in T_S
3. if t is not an element of T^P or T^P = t then
4. T^P ← T^P + T^U
5. T^P ←T^P – {t}
6. t' ←the most frequent dataset in T^P
7. return Unrealize-TrainingSet
T_s-{t},T^U,T'+{t'},T^P-(t'))

SYSTEM DESIGN

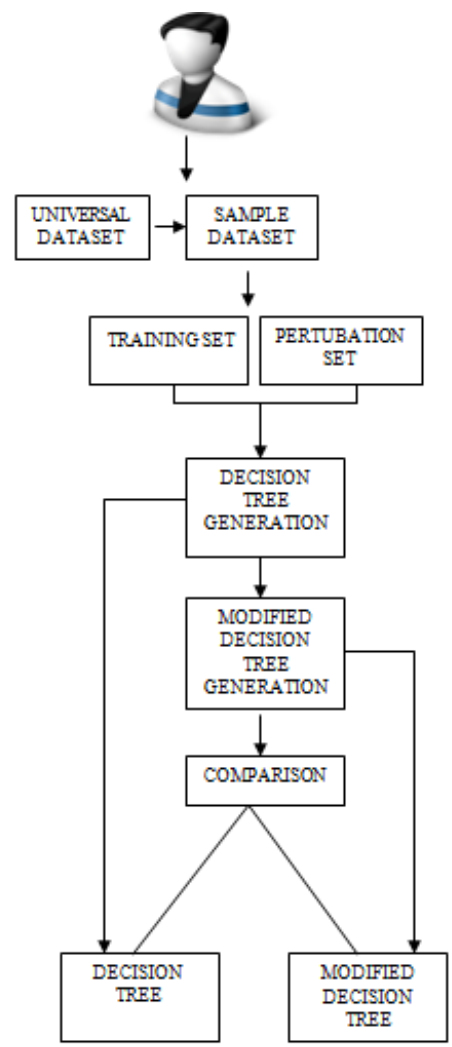


Figure 1: System Architecture of Proposed Methodology

DECISION TREE GENERATION:

The well-known ID3 algorithm builds a decision tree by call-

Volume : 3 | Issue : 9 | Sept 2013 | ISSN - 2249-555X

ing algorithm Choose-Attribute recursively. This algorithm selects a test attribute 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 ←Majority _ Value(T_S)
3. if $H_{ai}(T_S) = 0$ then return default
4. else if attribs is empty then return default
5. else
6. best ←Choose-Attribute(attribs, T_S)
7. tree← a new decision tree with root attribute best
8. for each value v_i of best do
9. T_{Si}← {datasets in T_s as best = k_i}
10. subtree ←Generate-Tree(T_{Si}, attribs-best, default}
11. connect tree and subtree with a branch labelled k_i
12. return tree

C4.5 algorithm is used for making decision tree process. A decision tree is done by calling algorithm Choose-Attribute recursively. There are 2 types of tree is generated in this process, first decision tree using the majority values in the gain calculation and modified decision tree using minority values in the gain calculation.

Algorithm Generate-Tree' (size, T', T^P , attribs, default)

Input: size, size of qT^U

T', the set of unreal training data sets

T^P , the set of perturbing data sets

attribs, set of attributes default,

default value for the goal predicate

Output: tree, a decision tree

1. if (T'; T^P) is empty then return default

2. default ← Minority _ Value(T', T^P)

3. if H_{ai} (q[T'+T^P]^c)=0 then return default

4. else if attribs is empty then return default

5. else

6. best ← Choose-Attribute'(attribs, size, (T', T^P))

7. tree ← a new decision tree with root attribute best

8. size ← size/number of possible values k_i in best

9. for each value v_i of best do

10. T'_i←{data sets in T' as best =k_i}

11. T^P_i←{data sets in T^P as best =k_i}

12. subtree← Generate-Tree(size, T'_i, T^P_i, attribs-best, default)

13. connect tree and subtree with a branch labelled k_i

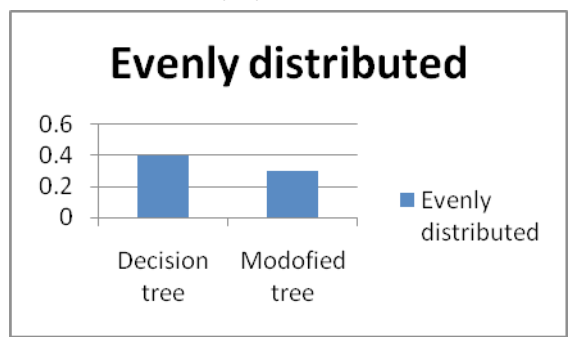
14. return tree

DISTRIBUTION:
Here we performs the calculation using the even distribution, extremely uneven distribution, and normal distribution. And also we perform the process of accuracy and accuracy calculation. Dummy values can be added for any attribute such that the domain of the perturbed sample data sets will be expanded while the addition of dummy values will have no impact on Training Set.

COMPARISON GRAPH:
We generate the accuracy graph and the time complexity graph.

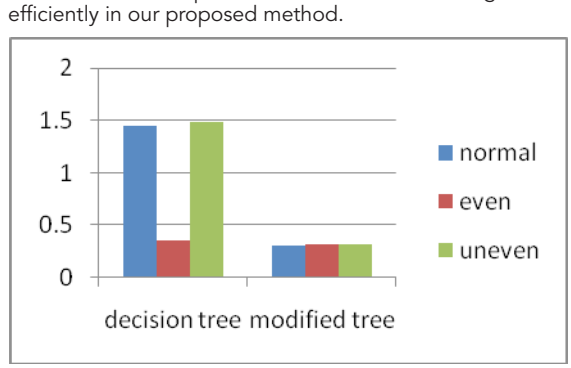
Accuracy Graph:
The storage requirement increases while the required storage may be doubled if dummy attribute values technique is applied to double the sample domain. The best case happen when samples are evenly distributed.

Samples with even distribution are taken. In even distribution, all datasets have the same counts. Decision tree is generated with increased storage required in existing method while it is not with our proposed method.



Time complexity Graph:
The worst case happens when the samples are in uneven distribution. Based on the randomly picked tests, Time complexity of storage for our approach is less than five times (without using dummy values) and eight times (with dummy values) than that of the original samples.

Normally distributed, evenly distributed and extremely uneven distributed samples are taken. Decision tree is generated efficiently in our proposed method.



5. CONCLUSION

We introduced a new privacy preserving approach that converts the sample data sets, training set, into some unreal data sets, such that any original data set is not able to reconstruct, if an unauthorized party where to steal some portion of data set. Privacy preservation via data set complementation fails if all training data sets are leaked because the data set reconstruction algorithm is generic. Therefore, further research is required to overcome this limitation. This paper covers the application of a new privacy preserving approach with the C4.5 decision tree learning algorithm and discrete-valued attributes only.

Volume : 3 | Issue : 9 | Sept 2013 | ISSN - 2249-555X

REFERENCE

[1] Arun K Pujari: Data Mining Techniques, Universities Press (India) Private Limited 2001. | | [2] S. Ajmani, R. Morris, and B. Liskov, "A Trusted Third-Party Computation Service," Technical Report MIT-LCS-TR-847, MIT, 2001. | [3] R. Agrawal and R. Srikant, "Privacy Preserving Data Mining," Proc. ACM SIGMOD Conf. Management of Data (SIGMOD '00), pp. 439-450, May 2000. | [4] Q. Ma and 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] S.L. Wang and A. Jafari, "Hiding Sensitive Predictive Association Rules," Proc. IEEE Int'l Conf. Systems, Man and Cybernetics, pp. 164- 169, 2005. | [6] J. Gitanjali, J. Indumathi, N.C. Iyengar, and 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. | [7] L. Liu, M. Kantarcioglu, and B. Thuraisingham, "Privacy Preserving Decision Tree Mining from Perturbed Data," Proc. 42nd Hawaii Int'l Conf. System Sciences (HICSS '09), 2009. | | [8] Y. Zhu, L. Huang, W. Yang, D. Li, Y. Luo, and F. Dong, "Three New Approaches to Privacy-Preserving Add to Multiply Protocol and Its Application," Proc. Second Int'l Workshop Knowledge Discovery and Data Mining, (WKDD '09), pp. 554-558, 2009. | | [9] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '02), pp. 23- 26, July 2002. | | [10] M. Shaneck and Y. Kim, "Efficient Cryptographic Primitives for Private Data Mining," Proc. |