Using Randomized Response Techniques for Privacy-Preserving Data Mining *

Wenliang Du and Zhijun Zhan
Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, NY 13244
Tel: 315-443-9180 Fax: 315-443-1122
Email: {wedu,zhzhan}@ecs.syr.edu

ABSTRACT

Privacy is an important issue in data mining and knowledge discovery. In this paper, we propose to use the randomized response techniques to conduct the data mining computation. Specially, we present a method to build decision tree classifiers from the disguised data. We conduct experiments to compare the accuracy of our decision tree with the one built from the original undisguised data. Our results show that although the data are disguised, our method can still achieve fairly high accuracy. We also show how the parameter used in the randomized response techniques affects the accuracy of the results.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

Keywords

Privacy, security, decision tree, data mining

1. INTRODUCTION

The growth of the Internet makes it easy to perform data collection on a large scale. However, accompanying such benefits are concerns about information privacy [1]. Because of these concerns, some people might decide to give false information in fear of privacy problems, or they might simply refuse to divulge any information at all. Data-analysis and knowledge-discovery tools using these data might therefore produce results with low accuracy or, even worse, produce false knowledge.

Some people might be willing to selectively divulge information if they can get benefit in return [11]. Examples of the benefit provided include discount of purchase, useful recommendations, and information filtering. However, a significant number of people are not willing to divulge their information because of privacy concerns. In order to understand Internet users' attitudes towards privacy, a survey was conducted in 1999 [3]. The result shows 17% of respondents are privacy fundamentalists, who are extremely concerned about any use of their data and generally unwilling to provide their data, even when privacy protection measures were in place. However, 56% of respondents are a pragmatic majority, who are also concerned about data use, but are less concerned than the fundamentalists; their concerns are often significantly reduced by the presence of privacy protection measures. The remaining 27% are marginally concerned and are generally willing to provide data under almost any condition, although they often expressed a mild general concern about privacy. According to this survey, providing privacy protection measures is a key to the success of data collection.

After data are collected, they can be used in many data analysis and knowledge discovery computations, the results of which can benefit not only the data collectors but also their customers; those benefits include market trend analysis, better services, and product recommendation. A widely used useful knowledge discovery method is *Data Mining*. Data mining, simply stated, refers to extracting or "mining" knowledge from large amounts of data [6]; its goal is to discover knowledge, trends, patterns, etc. from a large data set. Examples of the knowledge include classification, association rules, and clustering. As we mentioned before, because of privacy concerns, collecting useful data for data mining is a great challenge: on one hand, such data collection needs to preserve customers' privacy; on the other hand, the collected data should allow one to use data mining to "mine" useful knowledge.

In this paper, we particularly focus on a specific data mining computation, the decision-tree based classification, namely we want to find out how to build decision trees when the data in the database are disguised. We formulate our *DTPD Problem* (Building Decision Tree on Private Data) in the following:

PROBLEM 1. *(DTPD: Building Decision Tree on Private Data) Party A wants to collect data from users, and form a central database, then wishes to conduct data mining on this database. A sends out a survey containing N questions; each customer needs to answer those questions and sends back the answers. However, the survey contains some sensitive questions, and not every user feels comfortable to disclose*

*This material is based upon work supported by the National Science Foundation under Grant No. 0219560 and by the Center for Computer Application and Software Engineering (CASE) at Syracuse University.

SIGKDD '03, August 24-27, 2003, Washington, DC, USA.

his/her answers to those questions. How could A collect data without learning too much information about the users, while still being able to build reasonably accurate decision tree classifiers?

We propose to use the *Randomized Response* techniques to solve the DTPD problem. The basic idea of randomized response is to scramble the data in such a way that the central place cannot tell with probabilities better than a pre-defined threshold whether the data from a customer contain truthful information or false information. Although information from each individual user is scrambled, if the number of users is significantly large, the aggregate information of these users can be estimated with decent accuracy. Such property is useful for decision-tree classification since decision-tree classification is based on aggregate values of a data set, rather than individual data items.

The contributions of this paper are as follows: (1) We modify the ID3 classification algorithm [6] based on randomized response techniques and implement the modified algorithm. (2) We then conducted a series of experiments to measure the accuracy of our modified ID3 algorithm on randomized data. Our results show that if we choose the appropriate randomization parameters, the accuracy we have achieved is very close to the accuracy achieved using the original ID3 on the original data.

The rest of the paper is organized as follows: we discuss related work in Section 2. In Section 3, we briefly describe how randomized response technique work. Then in Section 4, we describe how to modify the ID3 algorithm to build decision trees on randomized data. In Section 5, we describe our experimental results. We give our conclusion in Section 6.

2. RELATED WORK

Agrawal and Srikant proposed a privacy-preserving data mining scheme using random perturbation [2]. In their scheme, a random number is added to the value of a sensitive attribute. For example, if x_i is the value of a sensitive attribute, $x_i + r$, rather than x_i, will appear in the database, where r is a random number drawn from some distribution. The paper shows that if the random number is generated with some known distribution (e.g. uniform or Gaussian distribution), it is possible to recover the distribution of the values of that sensitive attribute. Assuming independence of the attributes, the paper then shows that a decision tree classifier can be built with the knowledge of distribution of each attribute.

Evfimievski et al. proposed an approach to conduct privacy preserving association rule mining based on randomized response techniques [5]. Although our work is also based on randomized response techniques, there are two significant differences between our work and their work: first, our work deals with classification, instead of association rule mining. Second, in their solution, each attribute is independently disguised. When the number of attributes becomes large, the data quality will degrade very significantly.

Another approach to achieve privacy-preserving data mining is to use Secure Multi-party Computation (SMC) techniques. Several SMC-based privacy-preserving data mining schemes have been proposed [4, 7, 9]. [7] considers the problem of the decision tree building over horizontally partitioned data, i.e., one party has a set of records (rows) and the other has another set of different records. [9] and [4] consider the problems of association rule mining and decision tree building respectively over vertically partitioned data, i.e., for each record, some of the attributes (columns) are in one source, and the rest are in the other source. These studies mainly focused on two-party distributed computing, and each party usually contributes a set of records. Although some of the solutions can be extended to solve our DTPD problem (n party problem), the performance is not desirable when n becomes big. In our proposed research, we focus on centralized computing, and each participant only has one record to contribute. All records are combined together into a central database before the computations occur. In our work, the bigger the value of n is, the more accurate the results will be.

3. RANDOMIZED RESPONSE

Randomized Response (RR) techniques were developed in the statistics community for the purpose of protecting surveyee's privacy. We briefly describe how RR techniques are used for single-attribute databases. In the next section, we propose a scheme to use RR techniques for multiple-attribute databases.

Randomized Response technique was first introduced by Warner [10] in 1965 as a technique to solve the following survey problem: to estimate the percentage of people in a population that has attribute A, queries are sent to a group of people. Since the attribute A is related to some confidential aspects of human life, respondents may decide not to reply at all or to reply with incorrect answers. Two models (*Related-Question Model and Unrelated-Question Model* have been proposed to solve this survey problem. We only describe *Related-Question Model* in this paper.

In this model, instead of asking each respondent whether he/she has attribute A, the interviewer asks each respondent two related questions, the answers to which are opposite to each other [10]. For example, the questions could be like the following:

1. I have the sensitive attribute A.
2. I do not have the sensitive attribute A.

Respondents use a randomizing device to decide which question to answer, without letting the interviewer know which question is answered. The randomizing device is designed in such a way that the probability of choosing the first question is θ, and the probability of choosing the second question is $1 - \theta$. Although the interviewer learns the responses (e.g. "yes" or "no"), he/she does not know which question was answered by the respondents. Thus the respondents' privacy is preserved. Since the interviewer's interest is to get the answer to the first question, and the answer to the second question is exactly the opposite to the answer to the first one, if the respondent chooses to answer the first question, we say that he/she is telling the truth; if the respondent chooses to answer the second question, we say that he/she is telling a lie.

To estimate the percentage of people who has the attribute A, we can use the following equations:

$$\begin{aligned} P^*(A = yes) &= P(A = yes) \cdot \theta + P(A = no) \cdot (1 - \theta) \\ P^*(A = no) &= P(A = no) \cdot \theta + P(A = yes) \cdot (1 - \theta) \end{aligned}$$

where $P^*(A = yes)$ (resp. $P^*(A = no)$) is the proportion of the "yes" (resp. "no") responses obtained from the survey

data, and $P(A = yes)$ (resp. $P(A = no)$) is the estimated proportion of the "yes" (resp. "no") responses to the sensitive questions. Getting $P(A = yes)$ and $P(A = no)$ is the goal of the survey. By solving the above equations, we can get $P(A = yes)$ and $P(A = no)$ if $\theta \neq \frac{1}{2}$. For the cases where $\theta = \frac{1}{2}$, we can apply *Unrelated-Question Model* where two unrelated questions are asked with one probability for one of the questions is known.

4. BUILDING DECISION TREES ON DISGUISED PRIVATE DATA

4.1 Data Collection

The randomized response techniques discussed above consider only one attribute. However, in data mining, data sets usually consist of multiple attributes; finding the relationship among these attributes is one of the major goals for data mining. Therefore, we need the randomized response techniques that can handle multiple attributes while supporting various data mining computations. Work has been proposed to deal with surveys that contain multiple questions [8]. However, their solutions can only handle very low dimensional situation (e.g. dimension = 2), and cannot be extended to data mining, in which the number of dimensions is usually high. In this paper, we propose a new randomized response technique for multiple-attribute data set, and we call this technique the Multivariate Randomized Response (MRR) technique. We use MRR to solve our privacy preserving decision tree building problem.

Suppose there are N attributes, and data mining is based on these N attributes. Let E represent any logical expression based on those attributes. Let $P^*(E)$ be the proportion of the records in the whole disguised data set that satisfy $E =$ `true`. Let $P(E)$ be the proportion of the records in the whole undisguised data set that satisfy $E =$ `true` (the undisguised data set contains the true data, but it does not exist). $P^*(E)$ can be observed directly from the disguised data, but $P(E)$, the actual proportion that we are interested in, cannot be observed from the disguised data because the undisguised data set is not available to anybody; we have to estimate $P(E)$. The goal of MRR is to find a way to estimate $P(E)$ from $P^*(E)$ and other information.

For the sake of simplicity, we assume the data is binary, and we use the finding of the probability of $(A_1 = 1) \wedge (A_2 = 1) \wedge (A_3 = 0)$ as an example to illustrate our techniques[1], i.e., in this example, we want to know $P(A_1 = 1 \wedge A_2 = 1 \wedge A_3 = 0)$. To simplify our presentation, we use $P(110)$ to represent $P(A_1 = 1 \wedge A_2 = 1 \wedge A_3 = 0)$, $P(001)$ to represent $P(A_1 = 0 \wedge A_2 = 0 \wedge A_3 = 1)$, and so on.

For the related model, users tell the truth about all their answers to the sensitive questions with the probability θ; they tell the lie about all their answers with the probability $1 - \theta$. For example, using the above example, assume an user's truthful values for attributes A_1, A_2, and A_3 are 110. The user generates a random number from 0 to 1; if the number is less than θ, he/she sends 110 to the data collector (i.e., telling the truth); if the number is bigger than θ, he/she sends 001 to the data collector (i.e., telling lies about all the questions). Because the data collector does not know the random number generated by users, the data collector cannot know that is the truth or a lie.

[1] "$\wedge$" is the logical `and` operator.

Because the contributions to $P^*(110)$ and $P^*(001)$ partially come from $P(110)$, and partially come from $P(001)$, we can derive the following equations:

$$\begin{aligned} P^*(110) &= P(110) \cdot \theta + P(001) \cdot (1 - \theta) \\ P^*(001) &= P(001) \cdot \theta + P(110) \cdot (1 - \theta) \end{aligned}$$

By solving the above equations, we can get $P(110)$, the information needed for decision tree building purpose.

4.2 Building Decision Trees

Classification is one of the forms of data analysis that can be used to extract models describing important data classes or to predict future data. It has been studied extensively by the community in machine learning, expert system, and statistics as a possible solution to the knowledge discovery problem. Classification is a two-step process. First, a model is built given the input of training data set which is composed of data tuples described by attributes. Each tuple is assumed to belong to a predefined class described by one of the attributes, called the class label attribute. Second, the predictive accuracy of the model (or classifier) is estimated. A test set of class-labeled samples is usually applied to the model. For each test sample, the known class label is compared with predictive result of the model.

The decision tree is one of the classification methods. A decision tree is a class discriminator that recursively partitions the training set until each partition entirely or dominantly consists of examples from one class. A well known algorithm for decision tree building is ID3 [6]. We describe the algorithm below where S represents the training samples and AL represents the attribute list:

ID3(S, AL)

1. Create a node V.
2. **If** S consists of samples with all the same class C **then** return V as a leaf node labeled with class C.
3. **If** AL is empty, **then** return V as a leaf-node with the majority class in S.
4. Select test attribute (TA) among the AL with the highest information gain.
5. Label node V with TA.
6. **For** each known value a_i of TA
   (a) Grow a branch from node V for the condition $TA = a_i$.
   (b) Let s_i be the set of samples in S for which $TA = a_i$.
   (c) **If** s_i is empty **then** attach a leaf labeled with the majority class in S.
   (d) **Else** attach the node returned by **ID3(s_i, $AL - TA$)**.

According to ID3 algorithm, each non-leaf node of the tree contains a splitting point, and the main task for building a decision tree is to identify an attribute for the splitting point based on the information gain. Information gain can be computed using *entropy*. In the following, we assume there are m classes in the whole training data set. $Entropy(S)$ is defined as follows:

$$Entropy(S) = -\sum_{j=1}^{m} Q_j \log Q_j \qquad (1)$$

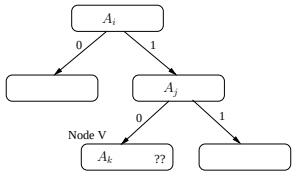


Figure 1: The Current Tree

where Q_j is the relative frequency of class j in S. Based on the entropy, we can compute the information gain for any candidate attribute A if it is used to partition S:

$$Gain(S, A) = Entropy(S) - \sum_{v \in A} (\frac{|S_v|}{|S|} Entropy(S_v)) \quad (2)$$

where v represents any possible values of attribute A; S_v is the subset of S for which attribute A has value v; $|S_v|$ is the number of elements in S_v; $|S|$ is the number of elements in S. To find the best split for a tree node, we compute information gain for each attribute. We then use the attribute with the largest information gain to split the node.

When the data are not disguised, we can easily compute the information gain, but when the data are disguised using the randomized response techniques, computing it becomes non-trivial. Because we do not know whether a record in the whole training data set is true or false information, we cannot know which records in the whole training data set belong to S. Therefore, we cannot directly compute $|S|$, $|S_v|$, Entropy(S), or Entropy(S_v) like what the original ID3 algorithm does. We have to use estimation.

Without loss of generality, we assume the database only contains binary values, and we will show, as an example, how to compute the information gain for a tree node V that satisfies $A_i = 1$ and $A_j = 0$ (suppose A_i and A_j are splitting attributes used by V's ancestors in the tree). Let S be the training data set consisting of the samples that belong to node V, i.e. all data samples in S satisfy $A_i = 1$ and $A_j = 0$. The part of the tree that is already built at this point is depicted in Figure 1.

Let E be a logical expression based on attributes. Let $P(E)$ be the proportion of the records in the undisguised data set (the true but non-existing data set) that satisfy $E =$ `true`. Because of the disguise, $P(E)$ cannot be observed directly from the disguised data, and it has to be estimated. Let $P^*(E)$ be the proportion of the records in the disguised data set that satisfy $E =$ `true`. $P^*(E)$ can be computed directly from the disguised data.

To compute $|S|$, the number of elements in S, let

$$\begin{aligned} E &= (A_i = 1) \wedge (A_j = 0) \\ \overline{E} &= (A_i = 0) \wedge (A_j = 1) \end{aligned}$$

If we use the randomized response technique with the related-question model, we can get the following equations:

$$\begin{aligned} P^*(E) &= P(E) \cdot \theta + P(\overline{E}) \cdot (1 - \theta) \\ P^*(\overline{E}) &= P(\overline{E}) \cdot \theta + P(E) \cdot (1 - \theta) \end{aligned} \quad (3)$$

We can compute $P^*(E)$ and $P^*(\overline{E})$ directly from the (whole) disguised data set. Therefore, by solving the above equations (when $\theta \neq \frac{1}{2}$), we can get $P(E)$. Hence, we get $|S| = P(E) * n$, where n is the number of records in the whole training data set.

To compute $Entropy(S)$, we need to compute Q_0 and Q_1 first (we assume the class label is also binary for this example, and the class label is also disguised). Let

$$\begin{aligned} E &= (A_i = 1) \wedge (A_j = 0) \wedge (Class = 0) \\ \overline{E} &= (A_i = 0) \wedge (A_j = 1) \wedge (Class = 1) \end{aligned}$$

Sometimes, the class label is not sensitive information, and is not disguised. Therefore, the information of the class label is always true information. We can slightly change the above definition of $\overline{E}$ as the following:

$$\begin{aligned} E &= (A_i = 1) \wedge (A_j = 0) \wedge (Class = 0) \\ \overline{E} &= (A_i = 0) \wedge (A_j = 1) \wedge (Class = 0) \end{aligned}$$

We compute $P^*(E)$ and $P^*(\overline{E})$ directly from the (whole) disguised data set. Then we solve Equations 3 and get $P(E)$. Therefore, $Q_0 = \frac{P(E)*n}{|S|}$, $Q_1 = 1 - Q_0$, and $Entropy(S)$ can be computed. Note that the $P(E)$ we get here is different from the $P(E)$ we get while computing $|S|$.

Now suppose attribute A_k is a candidate attribute, and we want to compute $Gain(S, A_k)$. A number of values are needed: $|S_{A_k=1}|$, $|S_{A_k=0}|$, Entropy($S_{A_k=1}$), and Entropy($S_{A_k=0}$). These values can be similarly computed. For example, $|S_{A_k=1}|$ can be computed by letting

$$\begin{aligned} E &= (A_i = 1) \wedge (A_j = 0) \wedge (A_k = 1) \\ \overline{E} &= (A_i = 0) \wedge (A_j = 1) \wedge (A_k = 0) \end{aligned}$$

Then we solve Equations 3 to compute $P(E)$, and thus getting $|S_{A_k=1}| = P(E) * n$. $|S_{A_k=0}|$ can be computed similarly.

The major difference between our algorithm and the original ID3 algorithm is how $P(E)$ is computed. In ID3 algorithm, data are not disguised, $P(E)$ can be computed by simply counting how many records in the database satisfy E. In our algorithm, such counting (on the disguised data) only gives $P^*(E)$, which can be considered as the "disguised" $P(E)$ because $P^*(E)$ counts the records in the disguised database, not in the actual (but non-existing) database. The proposed randomized response techniques allow us to estimate $P(E)$ from $P^*(E)$.

4.3 Testing and Pruning

To avoid over-fitting in decision tree building, we usually use another data set different from the training data set to test the accuracy of the tree. Tree pruning can be performed based on the testing results, namely how accurate the decision tree is. Conducting the testing is straightforward when data are not disguised, but it is a non-trivial task when the testing data set is disguised, just as the training data set. Imagine, when we choose a record from the testing data set, compute a predicted class label using the decision tree, and find out that the predicated label does not match with the record's actual label, can we say this record fails the testing? If the record is a true one, we can make that conclusion, but if the record is a false one (due to the randomization), we cannot. How can we compute the accuracy score of the decision tree?

We also use the randomized response techniques to compute the accuracy score. We use an example to illustrate how we compute the accuracy score. Assume the number of attributes is 5, and the probability $\theta = 0.7$. To test a record 01101 (i.e. the value for the first attribute is 0, for the second attribute is 1, and so on), we feed both 01101 and its complement 10010 to the decision tree. We know one of the class-label prediction result is true. If both prediction results are correct (or incorrect), we can make an accurate conclusion about the testing results of this record. However, if the prediction result for 01101 is correct and for 10010 is incorrect, or vice versa, we can only make a conclusion with 0.7 certainty. Therefore, when the number of testing records is large, we can estimate the accuracy score. Next we will show how to estimate the accuracy score.

Using the (disguised) testing data set S, we construct another data set $\overline{S}$ by reversing the values in S (change 0 to 1 and 1 to 0), i.e. each record in $\overline{S}$ is the complement of the corresponding record in S. We say that $\overline{S}$ is the complement of the data set S. We conduct the testing using both S and $\overline{S}$. Similarly, we define U as the *original undisguised* testing data set, and $\overline{U}$ as the complement of U. Let $P^*(correct)$ be the proportion of correct predictions from testing data set S, and $\overline{P}^*(correct)$ be the proportion of correct predictions from testing data set $\overline{S}$. Let $P(correct)$ be the proportion of correct predictions from the *original undisguised* data set U, $\overline{P}(correct)$ be the proportion of correct predictions from $\overline{U}$. $P(correct)$ is what we want to estimate.

Because both $P^*(correct)$ and $\overline{P}^*(correct)$ are partially contributed by $P(correct)$, and partially contributed by $\overline{P}(correct)$, we have the following equations:

$$\begin{aligned} P^*(correct) &= P(correct) \cdot \theta + \overline{P}(correct) \cdot (1-\theta) \\ \overline{P}^*(correct) &= \overline{P}(correct) \cdot \theta + P(correct) \cdot (1-\theta) \end{aligned}$$

$P^*(correct)$ and $\overline{P}^*(correct)$ can be obtained from testing S and $\overline{S}$. Therefore, by solving the above equations, we can get $P(correct)$, the accuracy score of the testing.

5. EXPERIMENTAL RESULTS

5.1 Methodology

To evaluate the effectiveness of our randomized-response-based technique on building a decision tree classifier, we compare the classification accuracy of our scheme with the original accuracy, which is defined as the accuracy of the classifier induced from the original data.

We used a data set from the UCI Machine Learning Repository[2]. The original owners of data set is US Census Bureau. The data set was donated by Ronny Kohavi and Barry Becker in 1996. It contains 48842 instances with 14 attributes (6 continuous and 8 nominal) and a label describing the salary level. Prediction task is to determine whether a person's income exceeds \$50k/year based on census data. We used first 10,000 instances in our experiment.

We modified the ID3 classificaiton algorithm to handle the randomized data based on our proposed methods. We run this modified algorithm on the randomized data, and built a decision tree. We also applied the original ID3 algorithm to the original data set and built the other decision tree. We then applied the same testing data to both trees. Our goal is to compare the classification accuracy of these two trees. Clearly we want the accuracy of the decision tree built based on our method to be close to the accuracy of the decision tree built upon the original ID3 algorithm.

The following is our experiment steps:

1. Preprocessing: Since our method assumes that the data set contains only binary data, we first transformed the original non-binary data to the binary. We split the value of each attribute from the median point of the range of the attribute. After preprocessing, we divided the data set into a training data set D and a testing data set B. We also call the training data set D the original data set. Note that B will be used for comparing our results with the benchmark results, it is not used for tree pruning during the tree building phase.

2. Benchmark: We use D and the original ID3 algorithm to build a decision tree T_D; we use the data set B to test the decision tree, and get an accuracy score. We call this score the original accuracy.

3. θ Selection: For $\theta = 0.1$, 0.2, 0.3, 0.4, 0.45, 0.51, 0.55, 0.6, 0.7, 0.8, 0.9, 1.0, we conduct the following 4 steps:

   (a) Randomization: In this step, we need to create a disguised data set, G. For each record in the training data set D, we generate a random number r from 0 to 1 using uniform distribution. If $r \leq \theta$, we copy the record to G without any change; if $r > \theta$, we copy the complement of the record to G, namely each attribute value of the record we put into G is exactly the opposite of the value in the original record. We perform this randomization step for all the records in the training data set D and generate the new data set G.

   (b) Tree Building: We use the data set G and our modified ID3 algorithm to build a decision tree T_G.

   (c) Testing: We use the data set B to test T_G, and we get an accuracy score S.

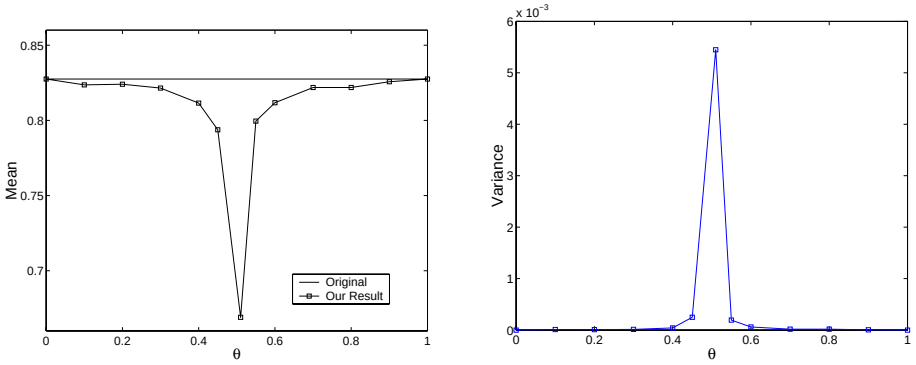
   (d) Repeating: We repeat steps 3a-3c for 50 times, and get $S_1, \ldots, S_{50}$. We then compute the mean and the variance of these 50 accuracy scores.

5.2 The Results

Figure 2(a) shows the mean value of the accuracy scores for each different θ values, and the original accuracy score. Figure 2(b) shows the variance of the accuracy scores for different θ's. We can see from the figures that when $\theta = 1$ and $\theta = 0$, the results are exactly the same as the original ID3 algorithm. This is because when $\theta = 1$, the randomized data set G is exactly the same as the original data set D; when $\theta = 0$, the randomized data set G is exactly the opposite of the original data set D. In both cases, our algorithm produces the accurate results (comparing to the original algorithm), but privacy is not preserved in either cases because an adversary can know the real values of all the records provided that he/she knows the θ value.

When θ moves from 1 and 0 towards 0.5, the degree of randomness in the disguised data is increased, the variance

[2]ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult



(a) Mean (b) Variance

Figure 2: Experiment Results

of the estimation used in our method should become large. Our results have demonstrated this. However, as we observed from the results, when θ is in the range of $[0, 0.4]$ and $[0.6, 1]$, our method can still achieve very high accuracy and low variance when comparing to the original accuracy score.

When θ is around 0.5, the mean deviates a lot from the original accuracy score, and the variance becomes large. The variance is caused by two sources. One is the sample size, the other is the randomization. Since we used the same sample size, the difference among variances when θ is different mainly comes from the difference of the randomization level. When θ is near 0.5, the randomization level is much higher and true information about the original data set is better disguised, in other words, more information is lost;. therefore the variance is much larger than the cases when θ is not around 0.5.

5.3 Privacy Analysis

When $\theta = 1$, we disclose everything about the original data set. When θ is away from 1 and approaches to 0.5, the privacy level of the data set is increasing. Our previous example shows that for a single attribute, when θ is close to 0.5, the data for a single attribute become uniformly distributed. On the other hand, when $\theta = 0$, all the true information about the original data set is revealed. When θ is moving toward 0.5, the privacy level is enhancing.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a method to build decision tree classifiers while preserving data's privacy. Our method consists of two parts: the first part is the multivariate data disguising technique used for data collection; the second part is the modified ID_3 decision tree building algorithm used for building a classifier from the disguised data. We presented experimental results that show the accuracy of the decision tree built using our algorithm. Our results show that when we select the randomization parameter θ from $[0.6, 1]$ and $[0, 0.4]$, we can get fairly accurate decision trees comparing to the trees built from the undisguised data.

In our future work, We will apply our techniques to solve other data mining problems (i.e., association rule mining). We will also extend our solution to deal with the cases where data type is not binary.

7. REFERENCES

[1] Office of the Information and Privacy Commissoner, Ontario, *Data Mining: Staking a Claim on Your Privacy, January 1998*. Available from `http://www.ipc.on.ca/web_site.eng/matters/sum_pap/papers/datamine.htm`.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD on Management of Data*, pages 439–450, Dallas, TX USA, May 15 - 18 2000.

[3] L. F. Cranor, J. Reagle, and M. S. Ackerman. Beyond concern: Understanding net users' attitudes about online privacy. Technical report, AT&T Labs-Research, April 1999. Available from `http://www.research.att.com/library/trs/TRs/99/99.4.3/report.htm`.

[4] W. Du and Z. Zhan. Building decision tree classifier on private data. In *Workshop on Privacy, Security, and Data Mining at The 2002 IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, December 9 2002.

[5] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 2002.

[6] J. Han and M. Kamber. *Data Mining Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[7] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Advances in Cryptology - Crypto2000, Lecture Notes in Computer Science*, volume 1880, 2000.

[8] A. C. Tamhane. Randomized response techniques for multiple sensitive attributes. *The American Statistical Association*, 76(376):916–923, December 1981.

[9] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 23-26 2002.

[10] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *The American Statistical Association*, 60(309):63–69, March 1965.

[11] A. F Westin. Freebies and privacy. Technical report, Opinion Research Corporation, July 1999. Availabe from `http://www.privacyexchange.org/iss/surveys/sr990714.html`.