PRIVACY-PRESERVING MULTI-PARTY DECISION TREE INDUCTION

Justin Z. Zhan, LiWu Chang, and Stan Matwin

Abstract We propose a new scheme for multiple parties to conduct data mining computations without disclosing their actual data sets to each other. We then apply the new scheme to let multiple parties build a decision tree classifier on their joint data set. We evaluate our scheme through a set of experiments. The empirical results show the tradeoffs between privacy and accuracy can be obtained.

1. INTRODUCTION

Collaboration is an important trend in the current world. It is valuable because of the mutual benefit it brings. This paper studies a very specific collaboration that becomes more and more prevalent. The problem is the collaborative data mining. The goal of our research is to develop technologies to enable multiple parties to conduct data mining collaboratively without disclosing their private data to each other.

Data mining has emerged as a means for identifying patterns and trends from a large amount of data. It includes various algorithms such as classification, association rule mining, sequential pattern mining and clustering, etc. This paper studies the decision tree classification which is one of the most successful data mining algorithms. The research on decision tree classification is extensive. However, the problem of how to jointly build decision tree classifier among multiple parties while preserving data privacy is a challenge to the information security and privacy community.

In this paper, we provide a solution to tackle this challenge. The basic idea of our solution is that we select one among all the involved parties and treat it as a data collector. Other parties use multi-variant randomized response technique, which will be discussed in Section 4, to disguise their data sets. They then send their disguised data sets to the data collector. The data collector combines the disguised data and conduct mining based on our proposed estimation model. Our contributions are (1) to propose a flexible scheme for multiple parties to build a decision tree on their joint data sets without compromising their data privacy, and (2) to conduct a set of experiments to evaluate the scheme.

The paper is organized as follows: Section 2 discusses the related work. We then formally define the problem in Section 3. In Section 4, we describe our proposed scheme. Section 5 shows how to build a decision tree classifier using the proposed scheme. We conduct experiments to evaluate our scheme in Section 6. Further discussion is conducted in Section 7. We give our conclusion in Section 8.

2. RELATED WORK

2.1 Secure Multi-party Computation

Briefly, a Secure Multi-party Computation (SMC) problem deals with computing any function on any input, in a distributed network where each participant holds one of the inputs, while ensuring that no more information is revealed to a participant in the computation than can be inferred from that participant's input and output [6]. The SMC problem literature is extensive. It has been proved that for any function, there is a secure multi-party computation solution [5]. The approach used is as follows: the function F to be computed is first represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. Every participant gets corresponding shares of the input wires and the output wires for every gate. This approach, though appealing in its generality and simplicity, means that the size of the protocol depends on the size of the circuit, which depends on the size of the input, hence it is highly impractical.

2.2 Privacy-Preserving Multi-party Data Mining

In early work on such a privacy preserving data mining problem, Lindell and Pinkas [8] propose a solution to the privacy preserving classification problem using the oblivious transfer protocol, a powerful tool developed by the secure multi-party computation research. The solution, however, only deals with the horizontally partitioned data[1]. Vaidya and Clifton [3] proposed to use the scalar product as the basic component to tackle the problem of association rule mining in vertically partitioned data. Later, they proposed a permutation scheme to solve the K-means clustering [2] over vertically partitioned data. In [15], a secure protocol is developed to deal with two-party decision tree induction problem. In [12], a secure procedure is further provided to conduct multi-party association rule mining over vertically partitioned data sets. In this paper, we will propose a new scheme to tackle the problem of privacy-preserving decision tree induction over the vertically partitioned data.

[1] This terminology refers to data represented in a table, in which columns correspond to attributes and rows to data records. Partition of this table by grouping together rows is the horizontal partition. In contrast, partitioning by grouping together different attributes is the vertical partition.

3. PRIVACY-PRESERVING MULTI-PARTY DECISION TREE INDUCTION PROBLEM

We consider the scenario where multiple parties, each having a private data set denoted by D_1, D_2, $\cdots$, and D_n respectively, want to collaboratively construct a decision tree classifier on the union of their data sets. Because they are concerned about data privacy, neither party is willing to disclose its raw data set to others. Without loss of generality, we make the following assumptions on the data sets: First, the identities of each row (record) in D_1, D_2, $\cdots$ and D_n are the same. Second, all parties share the class labels of all the records and also the names of all the attributes.

Privacy-Preserving Multi-party Decision Tree Induction Problem: Party 1 has a private data set D_1, party 2 has a private data set D_2, $\cdots$ and party n has a private data set D_n. Data set $[D_1 \cup D_2 \cup \cdots \cup D_n]$ is the combination of $D_1, D_2, \cdots$ and D_n where the ith record in $D_1, D_2, \cdots, D_n$ becomes the ith record of $[D_1 \cup D_2 \cup \cdots \cup D_n]$. Let *N* be the total number of records with N_i representing the ith record. The problem is that the parties want to build a decision tree classifier on $[D_1 \cup D_2 \cup D_3 \cdots \cup D_n]$, but they do not want to disclose their private data sets to each other.

4. MULTI-VARIANT RANDOMIZED RESPONSE (MRR) TECHNIQUES

Randomized Response techniques were 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.

To enhance the level of cooperation, instead of asking each respondent whether he/she has attribute *A*, the data collector 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: First, you have the sensitive attribute *A*, is it true? Second, you do not have the sensitive attribute *A*, is it true?

Respondents use a randomizing device to decide which question to answer, without letting the data collector 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 data collector 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.

The randomized response technique discussed above considers only one attribute. However, in data mining, data sets usually consist of multiple attributes; multi-variant randomized response technique (MRR) [16] was proposed to deal with multiple attributes. In this paper, we will develop a scheme to let multi-parties to conduct decision tree induction based on the MRR technique.

4.1 Multi-variant Randomized Response Technique

In the multiple attribute case, all the attributes are either all reversed together or all keep the same values. In other words, when sending the private data to the data collector, respondents either tell the truth (select the first question to answer) about all their answers to the sensitive questions or tell the lie (select the second question to answer) about all their answers. The probability of the first event is θ, and the probability for the second event is $1-\theta$. For example, assume a respondent's truthful values for attributes A_1, A_2, and A_3 are 110. The respondent 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 greater 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 respondents, the data collector cannot know whether a respondent tells the truth or a lie. To simplify our presentation, $P(110)$ is utilized to represent $P(A_1 = 1 \wedge A_2 = 1 \wedge A_3 = 0)$, $P(001)$ to represent $P(A_1 = 0 \wedge A_2 = 0 \wedge A_3 = 1)$, where $\wedge$ 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 general model is described as follows:

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

where E represents any logical expression of attributes, e.g., $E = (A_1 = 1 \wedge A_2 = 0)$, $\overline{E}$ denotes the opposite of E. For example, for the E in the previous example, $\overline{E} = (A_1 = 0 \wedge A_2 = 1)$. $P^*(E)$ be the proportion of the records in the whole *randomized* data set that satisfy E = `true`. *P(E)* be the proportion of the records in the whole *non-randomized* data set that satisfy E = `true` (the non-randomized data set contains the true data, but it does not exist). *P*(E)* can be observed directly from the randomized data, but *P(E),* the actual proportion that we are interested in, cannot be observed because the non-randomized data set is not available to the data collector. Through the above equations, we can estimate *P(E)*.

4.2 Multi-party Scheme

4.2.1 Two-party Case. In the two-party case, we assume there are only two parties in the collaboration. Each party has one data set (D_1 and D_2) which contains a set of attributes. For each record, they apply the multi-variant randomized response techniques on it and finally obtain the randomized data sets G_1 and G_2 respectively. To build a decision tree on the joint data set, they need to combine the disguised data sets G_1 and G_2 into a single data set *G*. One party needs to hold the disguised data set and conduct the mining. Therefore, we randomly select a party (i.g., party 1), who will play the role of data collector, to hold the disguised data set. The other party (i.g., party 2) then sends its disguised data sets to the data collector (i.g., party 1) who then combines the disguised data sets into a single data set *G*. The challenge is how the data collector compute the entropy for each splitting nodes. In other words, how to estimate the probabilities terms needed to compute the entropy, e.g., how to estimate $P(E_1E_2)$, where E_1 comes from party 1 and E_2 is from party 2, based on the combined disguised data set *G*.

To show how to estimate $P(E_1E_2)$, we look at all the contributions to $P^*(E_1E_2)$. There are four parts that contribute to $P^*(E_1E_2)$: (1)$P(E_1E_2)$: both parties tell the truth about all the attributes for their data set; the probability for this event is θ^2. (2) $P(E_1\overline{E_2})$: party 1 tells the truth about all the attributes for its data set D_1 and party 2 tells the lie about all the attributes for D_2; the probability for this event is $\theta(1-\theta)$. (3)$P(\overline{E_1}E_2)$: party 1 tells the lie about all the attributes for D_1 and tells the truth about all the attributes for D_2; the probability for this event is $(1-\theta)\theta$. (4)$P(\overline{E_1}\overline{E_2})$: both parties tell the lie about all the attributes for their data sets; the probability of this event is $(1-\theta)^2$. We then have the following equation:

$$\begin{aligned} P^*(E_1E_2) = P(E_1E_2)\cdot\theta^2 + P(E_1\overline{E_2})\cdot\theta(1-\theta) + \\ P(\overline{E_1}E_2)\cdot\theta(1-\theta) + P(\overline{E_1}\overline{E_2})\cdot(1-\theta)^2 \end{aligned} \quad (1)$$

There are four unknown variables in the above equation ($P(E_1E_2)$, $P(E_1\overline{E_2})$, $P(\overline{E_1}E_2)$, $P(\overline{E_1}\overline{E_2})$). To solve the above equation, we need three more equations. We can derive them using the similar method. We can then obtain the following equation by combining $P^*(\overline{E_1}E_2)$ with $P^*(E_1\overline{E_2})$ and $P(\overline{E_1}E_2)$ with $P(E_1\overline{E_2})$.

$$\begin{pmatrix} P^*(0Bar) \\ P^*(1Bar) \\ P^*(2Bar) \end{pmatrix} = M_2 \cdot \begin{pmatrix} P(0Bar) \\ P(1Bar) \\ P(2Bar) \end{pmatrix}, \tag{2}$$

where $P^*(0Bar) = P^*(E_1E_2), P^*(1Bar) = P^*(E_1\overline{E_2}) + P^*(\overline{E_1}E_2)$, $P^*(2Bar) = P^*(\overline{E_1}\overline{E_2})$; $P(0Bar) = P(E_1E_2)$, $P(1Bar) = P(E_1\overline{E_2}) + P(\overline{E_1}E_2)$ and $P(2Bar) = P(\overline{E_1}\overline{E_2})$. M_2 is the coefficiency matrix,

$$M_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3}$$

a_{ij} can be derived as follows: $a_{11} = \theta^2$, $a_{12} = \theta(1-\theta)$, $a_{13} = (1-\theta)^2$, $a_{21} = 2\theta(1-\theta)$, $a_{22} = \theta^2 + (1-\theta)^2$, $a_{23} = 2\theta(1-\theta)$, $a_{31} = (1-\theta)^2$, $a_{32} = \theta(1-\theta)$,and $a_{33} = \theta^2$. $P^*(0Bar)$, $P^*(1Bar)$ and $P^*(2Bar)$ can be obtained from the randomized data, θ is determined before the parties randomize their data. By solving the above equations, we can get $P(E_1E_2) = P(0Bar)$.

4.2.2 Multi-party Case. In the real applications, the collaboration may involve multiple parties. In this case, we also let each party apply the multivariant randomized response techniques for their data set *independently*. A data collector is again randomly selected. All other parties send their randomized data sets to the data collector who then combine the randomized data sets into one. The data collector then conducts data mining computation on the combined randomized data set. To estimate $P(E_1E_2E_3\cdots E_n)$ that we are interested in, we provide the following estimation model.

$$\begin{pmatrix} P^*(0Bar) \\ P^*(1Bar) \\ P^*(2Bar) \\ \cdots \\ P^*(nBar) \end{pmatrix} = M_n \cdot \begin{pmatrix} P(0Bar) \\ P(1Bar) \\ P(2Bar) \\ \cdots \\ P(nBar) \end{pmatrix},$$

where $P^*(0Bar) = P^*(E_1E_2E_3\cdots E_n)$, $P^*(1Bar) = P^*(\overline{E_1}E_2\cdots E_n) + P^*(E_1\overline{E_2}\cdots E_n) + \cdots + P^*(E_1E_2\cdots\overline{E_n})$, $\cdots$, $P^*(nBar) = P^*(\overline{E_1}\overline{E_2}\cdots\overline{E_n})$; $P(0Bar) = P(E_1E_2E_3\cdots E_n)$, $P(1Bar) = P(\overline{E_1}E_2\cdots E_n) +$
$P(E_1\overline{E_2}\cdots E_n) + \cdots + P(E_1E_2\cdots\overline{E_n})$, $\cdots$, $P(nBar) =$
$P(\overline{E_1}\overline{E_2}\cdots\overline{E_n})$ and M_n is the coefficiency matrix.

$$M_n = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1(n+1)} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2(n+1)} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3(n+1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{(n+1)1} & a_{(n+1)2} & a_{(n+1)3} & \cdots & a_{(n+1)(n+1)} \end{bmatrix} \tag{4}$$

For different party scheme, the coefficiency matrix is different. the values of a_{ij} can be similarly derived as we did for two-party scheme. For instance, we can derive the values for three-party scheme as follows: $a_{11} = \theta^3$, $a_{12} = \theta^2(1-\theta)$, $a_{13} = \theta(1-\theta)^2$, $a_{14} = (1-\theta)^3$, $a_{21} = 3\theta^2(1-\theta)$, $a_{22} = \theta^3 + 2\theta(1-\theta)^2$, $a_{23} = (1-\theta)^3 + 2\theta^2(1-\theta)$, $a_{24} = 3\theta(1-\theta)^2$, $a_{31} = 3\theta(1-\theta)^2$, $a_{32} = 2\theta^2(1-\theta) + (1-\theta)^3$, $a_{33} = 2\theta(1-\theta)^2 + \theta^3$, $a_{34} = 3\theta^2(1-\theta)$, $a_{41} = (1-\theta)^3$, $a_{42} = \theta(1-\theta)^2$, $a_{43} = \theta^2(1-\theta)$, and $a_{44} = \theta^3$. $P^*(0Bar)$, $P^*(1Bar)$, $\cdots$ and $P^*(nBar)$ can be obtained from the randomized data. After we derive the coefficiency matrix, we can solve the above equation and obtain $P(E_1E_2E_3 \cdots E_n) = P(0Bar)$. We need to point out that when $\theta = 0.5$, the related model cannot be applied, and other techniques such as randomized response techniques using the *unrelated-question* model [10] may be employed.

5. HOW TO BUILD DECISION TREES USING MULTI-PARTY SCHEME

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 records from one class. A well known algorithm for building decision tree classifiers is ID3 [1]. 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 this paper, we assume the data are binary and there are 2 classes in the whole training data set (but our scheme can be extended to deal with categorical data). *Entropy(S)* is then defined as follows:

$$Entropy(S) = -\sum_{j=1}^{2} Q_j \log Q_j, \tag{5}$$

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*:

$$Gain(S, A) = Entropy(S) - \sum_{v \in A} \left(\frac{|S_v|}{|S|} Entropy(S_v)\right), \tag{6}$$

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 randomized, we can easily compute the information gain, but when the data are disguised using the multi-party scheme, computing it becomes non-trivial. We do not know whether a record in the whole training data set is true or false information, and we cannot know which records in the whole training data set belong to *S* (referring to the ID3 procedure [1]). For instance, in Fig. 1, let's assume some record in the original non-randomized data set fall into the node A_j and the value of this record for attribute A_i is 1. Because of the data randomization, the value for A_i can be 0 or 1. We don't exactly know the original value of A_i by knowing the randomized data. Thus, we are not sure that this record should fall in the left sub-tree of node A_i or the right sub-tree. Therefore, we cannot directly compute $|S|$, $|S_v|$, Entropy(S), or Entropy(S_v) on the randomized data. We have to use estimation.

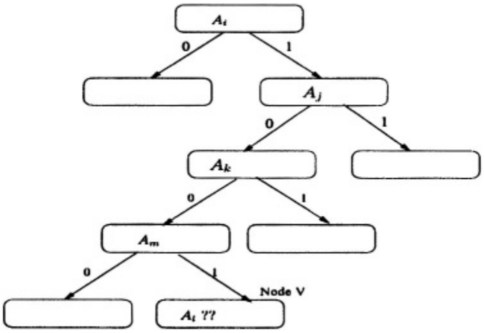


Figure 1. The Current Tree

Let's use an example to show how to compute the information gain for a tree node *V* that satisfies $(A_i = 1) \wedge (A_j = 0) \wedge (A_k = 0) \wedge (A_m = 1)$. For simplicity, we only show how to conduct these computations using the three-party scheme. Without loss of generality, we assume A_i and A_j belong to party 1, A_k belongs to party 2, and A_m belongs to party 3. Let *S* b e the training data set consisting of the records that belong to node *V*, i.e., all data records in *S* satisfy $(A_i = 1) \wedge (A_j = 0) \wedge (A_k = 0) \wedge (A_m = 1)$. The part of the tree that has already been built at this point is depicted in Fig. 1.

To compute $|S|$, the number of elements in *S*, let $E_1 = (A_i = 1) \wedge (A_j = 0)$, $\overline{E_1} = (A_i = 0) \wedge (A_j = 1)$, $E_2 = (A_k = 0)$, $\overline{E_2} = (A_k = 1)$, $E_3 = (A_m = 1)$ and $\overline{E_3} = (A_m = 0)$. $P^*(E_1E_2E_3)$, $P^*(\overline{E_1}E_2E_3)$, $P^*(E_1\overline{E_2}E_3)$, $P^*(E_1E_2\overline{E_3})$, $P^*(\overline{E_1}\overline{E_2}E_3)$, $P^*(E_1\overline{E_2}\overline{E_3})$, $P^*(\overline{E_1}E_2\overline{E_3})$ and $P^*(\overline{E_1}\overline{E_2}\overline{E_3})$ can be directly obtained from the (whole) disguised data set. Feeding the above terms into three-party model (when $\theta \neq \frac{1}{2}$), we can obtain $P(E_1E_2E_3)$. Hence, we can get $|S| = P(E_1E_2E_3) * n$, 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. Let $E_1 = (A_i = 1) \wedge (A_j = 0) \wedge (Class = 0(or\ 1))$, $\overline{E_1} = (A_i = 0) \wedge (A_j = 1) \wedge (Class = 0(or\ 1))$, $E_2 = (A_k = 0) \wedge (Class = 0)(or\ 1)$, $\overline{E_2} = (A_k = 1) \wedge (Class = 0)(or\ 1)$, $E_3 = (A_m = 1) \wedge (Class = 0)(or\ 1)$, and $\overline{E_3} = (A_m = 0) \wedge (Class = 0)(or\ 1)$. We can compute $P^*(E_1E_2E_3)$, $P^*(\overline{E_1}E_2E_3)$, $P^*(E_1\overline{E_2}E_3)$, $P^*(E_1E_2\overline{E_3})$, $P^*(\overline{E_1}\,\overline{E_2}E_3)$, $P^*(E_1\overline{E_2}\,\overline{E_3})$, $P^*(\overline{E_1}E_2\overline{E_3})$ and $P^*(\overline{E_1}\,\overline{E_2}\,\overline{E_3})$ directly from the combined randomized data set. Then we apply three-party estimation model and get $P(E_1E_2E_3)$. Therefore, $Q_0 = \frac{P(E_1E_2E_3)*n}{|S|}$, $Q_1 = 1 - Q_0$, and *Entropy*(S) can be computed.

Now suppose attribute A_l, which belongs to party 3, is a candidate attribute, and we want to compute $Gain(S, A_l)$. A number of values are needed: $|S_{A_l=1}|$, $|S_{A_l=0}|$, $\textsf{Entropy}(S_{A_l=1})$, and $\textsf{Entropy}(S_{A_l=0})$. These values can be similarly computed. For example, $|S_{A_l=1}|$ can be computed by letting $E_1 = (A_i = 1) \wedge (A_j = 0)$, $\overline{E_1} = (A_i = 0) \wedge (A_j = 1)$, $E_2 = (A_k = 0)$, $\overline{E_2} = (A_k = 1)$, $E_3 = (A_m = 1) \wedge (A_l = 1)$, and $\overline{E_3} = (A_m = 0) \wedge (A_l = 0)$. We then apply three-party estimation model to compute $P(E_1E_2E_3)$, and thus obtain $|S_{A_k=1}| = P(E_1E_2E_3) * n$. $|S_{A_k=0}|$ can be similarly computed.

The major difference between our algorithm and the original ID3 algorithm is how $P(E_1E_2E_3)$ is computed. In the ID3 algorithm, data are not disguised, $P(E_1E_2E_3)$ can be computed by simply counting how many records in the joint data sets that satisfy E_1, E_2, and E_3. In our algorithm, such counting (on the randomized data) only gives $P^*(E_1E_2E_3)$, which can be considered as the "randomized" $P(E_1E_2E_3)$ because $P^*(E_1E_2E_3)$ counts the records in the randomized data set, not in the original (but non-existing) data set. The proposed multi-party scheme allows us to estimate $P(E_1E_2E_3)$ from $P^*(E_1E_2E_3)$.

5.1 Testing

Conduct the testing is straightforward when data are not randomized, but it is a non-trivial task when the testing data set is randomized. 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 original 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 apply the multi-party scheme once again to compute the accuracy score. For simplicity, we only describe how to conduct testing for the three party case. We use an example to illustrate how we compute the score. Assume the number of attributes is 5. To test a record $(A_1 = 1, A_2 = 0, A_3 = 1, A_4 = 0, A_5 = 1)$ denoted by 10101, with A_1 and A_2 belonging to party 1, A_3 belonging to party 2, and A_4 and A_5 belonging to party 3, we feed 10101, 10110, 10001, 10010, 01101, 01110, 01001, and 01010 to the decision tree. We know one of the class-label prediction result is true, but don't exactly know which one. However, with enough testing data, we can estimate the total accuracy score, even though we do not know which test case produces the correct prediction result.

Using the (randomized) testing data set $S = S_1S_2S_3$, we construct other data sets $\overline{S_1}S_2S_3$, $S_1S_2\overline{S_3}$, $S_1\overline{S_2}S_3$, $S_1\overline{S_2}\overline{S_3}$, $\overline{S_1}S_2S_3$, $\overline{S_1}S_2\overline{S_3}$, $\overline{S_1}\overline{S_2}S_3$, and $\overline{S_1}\overline{S_2}\overline{S_3}$ by reversing the corresponding values in S_1, S_2 and S_3 (change 0 to 1 and 1 to 0). Note that each record in $\overline{S_i}$ (for $i \in [1, 2, 3]$) is the opposite of the corresponding record in S_i. We say that $\overline{S_i}$ is the opposite of the data set S_i. Similarly, we define U_i (for $i \in [1, 2, 3]$) as the *original non-randomized* testing data set, and $\overline{U_i}$ as the opposite of U_i.

Let $P^*(c_1c_2c_3)$ be the proportion of correct predictions from testing data set $S_1S_2S_3$, $P^*(\overline{c_1}c_2c_3)$ be the proportion of correct predictions from testing data set $\overline{S_1}S_2S_3$, $\cdots$, $P^*(\overline{c_1}\overline{c_2}\overline{c_3})$ be the proportion of correct predictions from testing data set $\overline{S_1}\overline{S_2}\overline{S_3}$. Similarly, let $P(c_1c_2c_3)$ be the proportion of correct predictions from the *original non-randomized* data set $U_1U_2U_3$, $P(\overline{c_1}c_2c_3)$ be the proportion of correct predictions from $\overline{U_1}U_2U_3$, $\cdots$, $P(\overline{c_1}\overline{c_2}\overline{c_3})$ be the proportion of correct predictions from $\overline{U_1}\overline{U_2}\overline{U_3}$. $P(c_1c_2c_3)$ is what we want to estimate.

Because $P^*(c_1c_2c_3)$, $P^*(\overline{c_1}c_2c_3)$, $\cdots$ and $P^*(\overline{c_1}\overline{c_2}\overline{c_3})$ consist of contributions from $P(c_1c_2c_3)$, $P(\overline{c_1}c_2c_3)$, $\cdots$ and $P(\overline{c_1}\overline{c_2}\overline{c_3})$, we have the following equation:

$$\begin{pmatrix} P^*(0Bar) \\ P^*(1Bar) \\ P^*(2Bar) \\ P^*(3Bar) \end{pmatrix} = M_3 \cdot \begin{pmatrix} P(0Bar) \\ P(1Bar) \\ P(2Bar) \\ P(3Bar) \end{pmatrix},$$

where $P^*(0Bar) = P^*(c_1c_2c_3)$, $P(0Bar) = P(c_1c_2c_3)$, $P^*(1Bar) = P^*(\overline{c_1}c_2c_3) + P^*(c_1\overline{c_2}c_3) + P^*(c_1c_2\overline{c_3})$, $P(1Bar) = P(\overline{c_1}c_2c_3) + P(c_1\overline{c_2}c_3) + P(c_1c_2\overline{c_3})$, $\cdots$, $P^*(3Bar) = P^*(\overline{c_1}\overline{c_2}\overline{c_3})$ and $P(3Bar) = P(\overline{c_1}\overline{c_2}\overline{c_3})$. M_3 is the coefficiency matrix defined in three-party scheme.

$P^*(0Bar)$, $P^*(1Bar)$, $P^*(2Bar)$, and $P^*(3Bar)$ can be obtained from testing data sets. By solving the above equation, we can get $P(0Bar) = P(c_1c_2c_3)$, the accuracy score of testing.

6. EXPERIMENTAL RESULTS

To evaluate the effectiveness of our proposed scheme, we conducted experiments on two real life data sets *Adult* and *Breast Cancer* which were obtained from the UCI Machine Learning Repository (ftp://ftp.ics.uci.edu/pub/machine-learning-databases).

6.1 Experimental Steps

We modified the ID3 classification 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 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. Obviously we want the accuracy of the decision tree built based on our method to be close to the accuracy of the decision tree built from the ID3 algorithm. Our experiments consist of the following steps:

Preprocessing:. Since we assume 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 sets into a training data set *D* and a testing data set *B*.

For simplicity, we only conduct the experiments for three-party scheme. In the experiments, we randomly split the whole data set into three parts in a vertical partition way such that each data set contains different attributes.

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 benchmark score.

θ Selection:. For $\theta = 0$, 0.1, 0.2, 0.3, 0.4, 0.45, 0.51, 0.55 0.6, 0.7, 0.8, 0.9, and 1.0, we conduct the following 4 steps:

1 Randomization: We randomly split *D* into three sub-data sets D_1, D_2, and D_3. Each party generates a random number r from 0 to 1 using uniform distribution. If $r \leq \theta$, we copy the record to G_1 without any change; if $r > \theta$, we copy the opposite of the record to G_1, namely each attribute value of the record we put into G_1 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_1 and conduct the above randomization for each party, finally obtain randomized data set G_1, G_2, and G_3. We then combine G_1, G_2 and G_3 together and form a data set *G*. Note that the random numbers used for different sub-data sets should be independent to each other.

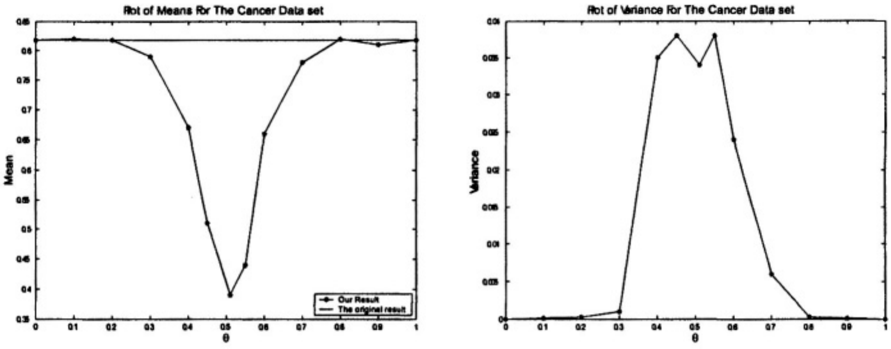
2 Tree Building: We use the data set *G* and our modified ID3 algorithm to build a decision tree T_G.

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

4 Repeating: We repeat steps 1-3 for 100 times, and get $S_1, \ldots, S_{100}$. We then compute the mean and the variance of these 100 accuracy scores.

6.2 The Result Analysis

6.2.1 The Analysis of Accuracy. Fig. 2(a) [Fig. 2(b)] and 3(a) [Fig. 3(b)][2] show the mean [variance] values of the accuracy scores for *Breast-Cancer* and *Adult* data sets respectively. We can see from the figures that when $\theta = 1$ and $\theta = 0$, the results are exactly the same as the results when the original ID3 algorithm is applied. This is because when $\theta = 1$, the randomized data sets are exactly the same as the original data set *D;* when $\theta = 0$, the randomized data sets are 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 case because the data collector can know the real values of all the records. When θ moves from 1 and 0 towards 0.5, the mean of accuracy has the trend of decreasing. When θ is around 0.5, the mean deviates a lot from the original accuracy score.



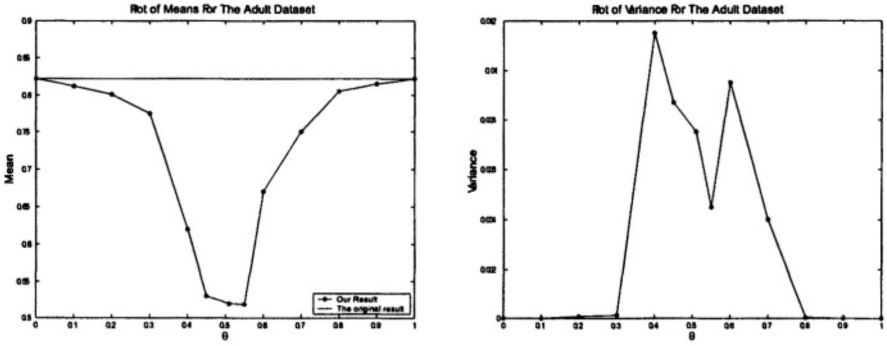
(a) Mean

(b) Variance

Figure 2. The Results On The Breast-Cancer Data Set

6.2.2 The Analysis of Privacy. Privacy comes from the randomization which is under the control of parameter θ. When $\theta = 1$, we disclose everything about the original data set. When θ is away from 1 and approaches to 0.5, the probability of disclosing the original data is decreasing and the privacy level of

[2] The case that $\theta = 0.5$ is excluded.



(a) Mean

(b) Variance

Figure 3. The Results On The Adult Data Set

the data set increases. 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 probability of disclosing the original data is decreasing and the privacy level is enhancing.

7. DISCUSSION

There are several issues needed to be addressed. Firstly, how do we select data collector among the collaborative parties. In our scheme, we randomly select one party as the data collector who collect the randomized data from other parties. However, there are alternative ways. For example, we could introduce a semi-trusted party who doesn't belong to the group of collaboration and will collect data from the parties and conduct the mining tasks for the collaborative parties. The shortcoming of this scheme is that we introduce an extra party, which may not be desirable from security point of view. The other alternative is that the parties, instead of sending the randomized data to each other, just send the probability terms (e.g., Q_j in Eq. 5) needed for the data mining computations. Thus, all the parties can be involved in the data mining computations, such as naive Bayesian classification. Secondly, does the party who is chosen as the data collector need to randomize its data? Based on the current estimation model, it is necessary. However, the estimation model of our proposed scheme can be modified to deal with this case. Thirdly, the proposed scheme can also deal with the cases where different parties use different randomization level (θ) to randomize their data. As a result, the values in the coefficiency matrix will be modified, e.g., a_{11} in Eq. 3, instead of being θ^2, will be $\theta_1\theta_2$. Finally, in the paper, we only consider the fully-dependent case in which the attributes for each party are randomized together, but our scheme can be generalized to deal with the case where the attributes for each party may not be fully-dependent. For example, suppose party 1 has 2 attributes A_1 and A_2. Based on the current scheme, A_1 and A_2 are randomized together, i.e., either all keep the same values or all flip the values to the opposite. It is considered as the fully-dependent case. In the non-fully-dependent case, A_1 and A_2 can be randomized separately. In other words, A_1 can keep the original value while A_2 can be flipped to the opposite and vice versa.

We also need to point out that we deal with heterogeneous collaboration in this paper. However, homogeneous collaboration can also be solved by the proposed scheme. We will describe the details in our future work.

8. CONCLUSION

In this paper, we have presented a multi-party scheme to build decision tree classifiers based on multi-variant randomized response techniques. Our method consists of two parts: the first part is the multi-party scheme used for data disguising; the second part is the modified ID_3 algorithm for decision tree induction, which is used to build a classifier from the randomized data, based on our proposed multi-party scheme. We conducted a set of experiments to evaluate our scheme. The empirical results show that, although the original data are not revealed, the tradeoffs between privacy and accuracy are possible. As future work, we will apply our scheme to other data mining algorithms.

Acknowledgments

The authors acknowledge the support of the Natural Sciences and Engineering Research Council of Canada, of the Communications and Information Technology Ontario and of the Office of Naval Research, U.S.A. for this research.

References

[1] R. Quinlan. Introduction of Decision Trees. In Journal of Machine Learning, Vol. 1, Pages: 81-106, 1986.

[2] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In Proc. of the 8th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, 2003, Washington, D.C, USA.

[3] J. Vaidya and C. Clifton. Privacy-Preserving Association Rule Mining in Vertically Partitioned Data. In Proc. of the 8th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, 2002, Edmonton, Canada.

[4] W. B. Barksdale. New Randomized Response Techniques for Control of Nonsampling Errors in Surveys. University of North Carolina, Chapel Hill, 1971.

[5] O. Goldreich. Secure Multi-party Computation (working draft), 1998.

[6] S. Goldwasser. Multi-Party Computations: Past and Present. Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, 1997, Santa Barbara, CA USA,August 21-24.

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

[8] Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. Advances in Cryptology - CRYPTO '00,2000,1880 of Lecture Notes in Computer Science. Spinger-Verlag, 36-54.

[9] A. C. Tamhane. Randomized Response Techniques for Multiple Sensitive Attributes. The American Statistical Association, 1981, volume 76, pages 916-923, December.

[10] S. L. Warner. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. The American Statistical Association, 1965, March, volume 60, pages 63-69.

[11] A. C. Yao. Protocols for secure computations. Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science, 1982.

[12] Z. Zhan and L. Chang. Privacy-Preserving Collaborative Data Mining, Workshop on Foundation and New Direction of Data Mining at The 2003 IEEE International Conference on Data Mining (ICDM'03), 2003 November 19, Melbourne, Florida, USA.

[13] O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game. Proceedings of the 19th Annual ACM Symposium on Theory of Computing, pages: 218-229, 1987.

[14] M. Franklin, Z. Galil and M. Yung. An Overview of Secure Distributed Computing, Department of Computer Science, Columbia University, 1992.

[15] W. Du and Z. Zhan. Building Decision Tree Classifier on Private Data, Workshop on Privacy, Security, and Data Mining at The 2002 IEEE International Conference on Data Mining, December 9, Maebashi City, Japan, 2002.

[16] W. Du and Z. Zhan. Using Randomized Response Techniques For Privacy-Preserving Data Mining. Proceedings of The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , August 24-27, 2003, Washington, DC, USA.