Privacy-Preserving Data Mining Using Multi-Group Randomized Response Techniques

Zhijun Zhan and Wenliang Du
Systems Assurance Institute
Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, NY 13244
Email: {zhzhan, wedu}@ecs.syr.edu

Abstract

Privacy is an important issue in data mining and knowledge discovery. In this paper, we describe a specific privacy preserving data mining problem: Company C wants to collect data from its customers to form a data set for data mining purpose. For the data collection, C sends out a survey containing a set of questions; each customer needs to answer those questions and sends back the answers. However, because the survey contains sensitive questions, not every user feels comfortable to disclose his/her answers to those questions; how could we develop a method such that C cannot find out any customer's actual answers, while still being able to derive reasonably accurate data mining results?

We propose to use the randomized response techniques to conduct the data collection. Such a method adds certain degree of randomness to the answers to prevent the data collector from learning the true information. In order to enhance the privacy level, we propose a multi-group scheme in which the customers partition all their answers into multiple groups, and for different groups, they randomize data separately. We then present a method to build decision tree classifiers from the disguised data based on the multi-group scheme. Finally we 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 parameters used in our randomized response techniques affect the accuracy of the results.

Keywords: privacy, security, decision tree, data mining, randomized response

1 INTRODUCTION

Data mining has emerged as a means for identifying patterns and trends from a large amount of data [10]. To conduct data mining computations, we need to collect data first. Without privacy concerns, data can be directly collected. However, because of privacy concerns, some people might decide to selectively divulge information, or give false information, or simply refuse to disclose any information at all. A survey was conducted in 1999 [5] to understand Internet users' attitudes towards privacy. 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.

1.1 Privacy-Preserving Data Mining Problem

There are many ways to collect data. One way is to collect data using transaction records of users. This can be done without users' knowledge, and they have no control over what information can be collected. Another way to collect data is to solicit users' responses via surveys, for example, users might be asked to rate certain products, or they might be asked whether they have alcohol addiction problem, and so on. The collected data will be put into a database. Although the second approach gives users the control over whether they want to disclose their information or not, privacy concerns might hinder the users from telling the truth or responding at all. *How can we improve the chance to collect more truthful data that are useful for data mining while preserving users' privacy*? *How can users contribute their personal information without compromising their privacy*?

One way to achieve privacy is to use anonymous techniques [1, 12, 13], which allow users to disclose their personal information without disclosing their identities. The biggest problem of using anonymous techniques is that there is no guarantee on the quality of the data set. A malicious user (e.g., a competing company) could send a great deal of random information to the database and render the database useless, or a company could send a lot of made-up information to the database with the goal of making their products the most favorable ones. These potential attacks could all render the database useless. If the communication is really anonymous, it is difficult for the database owner to control the quality of the data. To guarantee the quality, it is important for the database owner to verify the identities of the data contributors.

Another way to achieve privacy is to let each user disguise or randomize their data, such that the data collector cannot derive the truthful information about an user's private information. The challenge is how to conduct data mining from the disguised data? To address this challenge, we first propose the following computing model: The model consists of a data collection step and a computation step. In the data collection step, each user utilizes certain techniques to disguise his/her data, then sends the disguised data to the central place; the central place should not be able to find out any user's actual data with probabilities better than a pre-defined threshold. In the computation step, the central place constructs a database using the disguised data, and conducts data mining computations on this database. The goal of the central place is to derive useful information (or knowledge) out of this disguised database. 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 tree classifiers when the data in the database are disguised.

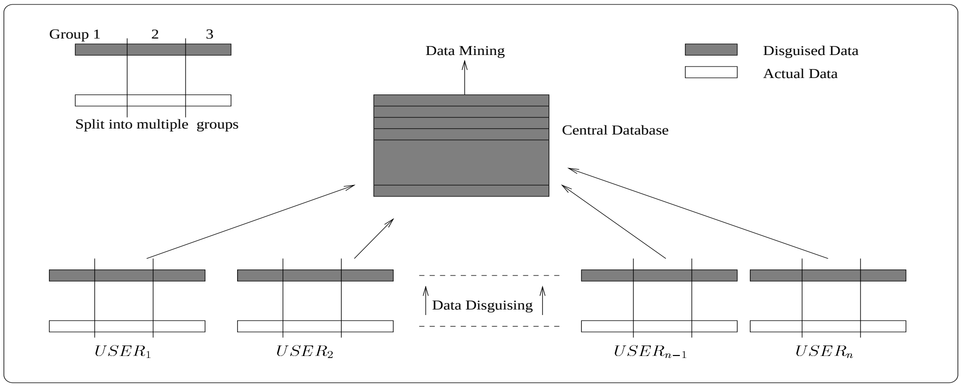


Figure 1: Privacy Preserving Data Mining

1.2 Outline of Our Solution

We propose to use the *Randomized Response* techniques to solve the privacy-preserving data mining 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 based classification since it is based on aggregate values of a data set, rather than individual data items.

One solution was proposed to solve the privacy-preserving data mining problem using the *Randomized Response* techniques [7]. We call this solution the one-group scheme, in which, users either send the truthful answers to all those questions or send the answers that are exactly opposite to their truthful answers for all the questions. Namely, if the value for one attribute in a record is true, then the values for the other attributes in the same record are also true. Thus, once an adversary somehow knows whether data provider tells a truth or not for only one attribute for some record, he/she can obtain all the true values about this record. This is not a desirable property because in practice, due to the hints from other sources, the probability of obtaining the values of some attributes is very high. For example, some public universities publish the statistical information about their faculty's annual salary. From the published data, the adversary knows that the salary of all the professors in these universities is greater than some value (e.g., $40,000). We assume that some surveyees come from these universities and one of the sensitive questions is to ask whether the surveyee's annual salary is more than $40,000. If some professor's response is *no*, the adversary immediately knows that he/she is telling a lie, so the true answers for other attributes can be easily derived.

To further preserve data's privacy, we propose a multi-group scheme, in which we partition the set of the attributes into a number of groups (the model is depicted in Fig. 1.); we then use the randomized response techniques to randomize each group *independently*. For example, we can partition N attributes into N groups (we call this scheme N-group scheme), with each group containing only one attribute. For each group (or each attribute in this situation), users randomly decide whether to disclose its true value (with probability θ) or to disclose the false value (with probability $1 - \theta$). The users repeat this process for all groups; the random decisions are independent for each group.

This N-group scheme is very secure because the value of each attribute is independently disguised. An adversary cannot guess an user's correct answer to any specific question with a probability better than θ. Even if one value is disclosed, other values are still hidden. To correctly guess all the values for each record, the probability is θ^N (where N is the number of attributes). When N is big, the probability becomes almost zero. However when N is large, the accuracy of the data mining computations might become to low.

A compromise between the one-group scheme and the N-group scheme is to partition the attributes to M ($1 < M < N$) groups. For each group, users select a random number θ, then based on θ, they decide whether to disclose the true values or false values for all the attributes in this group (the decision is the same for the attributes in the same group, but decisions for different groups are independent). By selecting an appropriate value of M, we can achieve a balance between privacy and accuracy.

The rest of the paper is organized as the following: we discuss related work in Section 2. In Section 3, we briefly describe how randomized response technique works. 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 scheme for privacy-preserving data mining 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 value 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.

Our work is actually motivated by this paper. The difference between our work and [2] is that the scheme in [2] works only when the attributes are independent; if some attributes are dependent, distribution is not sufficient for building a decision tree classifier because the distribution of each attribute does not capture the correlation among attributes. Our proposed approach takes this into consideration; we use a different approach, the randomized response techniques [17], to disguise data.

Evfimievski et al. proposed an approach to conduct privacy preserving association rule mining based on randomized response techniques [8]. 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. To address this problem, we propose a group-oriented randomization scheme, in which we divide all the attributes to a number of groups, and attributes in a group are disguised in the same way, but attributes from different groups are disguised independently. Our goal is to achieve a better balance between data quality and privacy.

Another approach to achieve privacy-preserving data mining is to use Secure Multi-party Computation (SMC) techniques [9, 18, 19]. Several SMC-based privacy-preserving data mining schemes have been proposed [6, 11, 15, 16]. 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 problem (n party problem), the performance is not desirable when n becomes large. 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 larger the value of n is, the more accurate the results will be.

Interestingly, the solutions from our work can be easily extended to solve the two-party (or even n-party) privacy preserving data mining problems using the following scheme: one party A uses the randomized response techniques to disguise its data, then sends the disguised data set to the other party B. B combines A's disguised data sets and his own data set into a hybrid one, part of which consists of disguised data and the other part consists of true data. B then conducts data mining using this data set. Our proposed two-group scheme can be easily applied to this situation.

SMC-based solutions generate exactly the same results as their corresponding non-secure solutions (those without considering privacy concerns), but with a degradation in performance whereas our work achieves an acceptable balance between accuracy and performance.

3 Randomized Response

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

Randomized Response techniques were first introduced by Warner [17] 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. In this paper, we only describe the first one.

In *Related-Question 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 [17]. For example, the questions could be like the following. If the statement is correct, the respondent answers "yes"; otherwise he/she answers "no".

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 for 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:

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

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 the probability for one of the questions is known.

4 Building Decision Tree classifiers Using Multi-Group Randomized Response Techniques

The randomized response techniques discussed in Section 3 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 [3, 14]. However, their solutions can only handle very low dimensional situation (e.g., dimension = 2), and cannot be extended to solve data mining problems, in which the number of dimensions is usually high. We have developed a multi-group randomized response technique (MRR) to deal with multiple attributes.

4.1 Notations

In this work, we assume data are binary, but the techniques can be extended to categorical data. Suppose there are N attributes (A_1, ..., A_N) in a data set. Let E represent any logical expression based on those attributes (e.g., $E = (A_1 = 1) \wedge (A_2 = 0)$); let $\overline{E}$ denote the logical expression that reverses the 1's in E to 0's and 0's to 1's; we call $\overline{E}$ the opposite of E. For example, for the E in the previous example, $\overline{E} = (A_1 = 0) \wedge (A_2 = 1)$.

Let $P^*(E)$ be the proportion of the records in the whole *disguised* data set that satisfy $E = \texttt{true}$. Let $P(E)$ be the proportion of the records in the whole *undisguised* data set that satisfy $E = \texttt{true}$ (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 $P^*(E)$.

In our multi-group scheme, we also divide each expression E to multiple sub-expressions. For example, in a three-group scheme, we write $E = E_1E_2E_3$, where E_i contains only the attributes in the group i.

4.2 One-Group Scheme

In the one-group scheme, all the attributes are put in the same group, and all the attributes are either reversed together or keeping the same values. In other words, when sending the private data to the central database, users either tell the truth about all their answers to the sensitive questions or tell the lie about all their answers. The probability for the first event is θ, and the probability for the second event is $1 - \theta$. For 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 whether data provider tells the truth or a lie. 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 $P(A_1 = 0 \wedge A_2 = 0 \wedge A_3 = 1)$ [1].

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:

$$\begin{aligned} P^*(110) &= P(110)\cdot\theta + P(001)\cdot(1-\theta) \\ P^*(001) &= P(001)\cdot\theta + P(110)\cdot(1-\theta) \end{aligned} \tag{2}$$

By solving the above equations, we can get $P(110)$, the information needed to build a decision tree. The general model for the one-group scheme is described in the following:

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

Using the matrix form, let M_1 denote the coefficiency matrix of the above equations, then

$$\left(\begin{array}{c} P^*(E) \\ P^*(\overline{E}) \end{array}\right) = M_1 \left(\begin{array}{c} P(E) \\ P(\overline{E}) \end{array}\right), \quad where \quad M_1 = \left[\begin{array}{rr} \theta & (1-\theta) \\ (1-\theta) & \theta \end{array}\right] \tag{4}$$

4.3 Two-Group Scheme

In the one-group scheme, if the interviewer somehow knows whether the respondents tell a truth or a lie for one attribute, he/she can immediately obtain all the true values of a respondent's response for all other attributes. To improve data's privacy level, data providers divide all the attributes into two groups [2]. They then apply the randomized response techniques for each group *independently*. For example, the users can tell the truth for one group while telling the lie for the other group. With this scheme, even if the interviewers know information about one group, they will not be able to derive the information for the other group because they are disguised independently.

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

[2] All the data providers should group the attributes in the same ways (e.g., one user lets attribute A_1 and A_2 to be in the group 1, then other users also let attribute A_1 and A_2 to be in the group 1).

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. $P(E_1E_2)$: users tell the truth about all the answers for both groups; the probability for this event is θ^2.

2. $P(E_1\overline{E_2})$: users tell the truth about all the answers for group 1 and tell the lie about all the answers for group 2; the probability for this event is $\theta(1-\theta)$.

3. $P(\overline{E_1}E_2)$: users tell the lie about all the answers for group 1 and tell the truth about all the answers for group 2; the probability for this event is $(1-\theta)\theta$.

4. $P(\overline{E_1E_2})$: users tell the lie about all the answers for both groups; the probability of this event is $(1-\theta)^2$.

We then have the following equation:

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

There are four unknown variables in the above equation ($P(E_1E_2)$, $P(E_1\overline{E_2})$, $P(\overline{E_1}E_2)$, $P(\overline{E_1E_2})$). To solve the above equation, we need three more equations. We can derive them using the similar method. The final equations are described in the following:

$$\begin{pmatrix} P^*(E_1E_2) \\ P^*(E_1\overline{E_2}) \\ P^*(\overline{E_1}E_2) \\ P^*(\overline{E_1E_2}) \end{pmatrix} = M_2 \cdot \begin{pmatrix} P(E_1E_2) \\ P(E_1\overline{E_2}) \\ P(\overline{E_1}E_2) \\ P(\overline{E_1E_2}) \end{pmatrix}, \tag{5}$$

where M_2 is the coefficiency matrix, and

$$M_2 \quad = \begin{bmatrix} \theta^2 & \theta(1-\theta) & \theta(1-\theta) & (1-\theta)^2 \\ \theta(1-\theta) & \theta^2 & (1-\theta)^2 & \theta(1-\theta) \\ \theta(1-\theta) & (1-\theta)^2 & \theta^2 & \theta(1-\theta) \\ (1-\theta)^2 & \theta(1-\theta) & \theta(1-\theta) & \theta^2 \end{bmatrix} \tag{6}$$

By solving the above equations, we can get $P(E_1E_2)$.

4.4 Three-Group Scheme

To further preserve the data's privacy, we can partition the attributes into three groups, and disguised each group independently. The model can be derived using the similar way as we did for the two-group model. The model for the three-group scheme is as follows:

$$\begin{pmatrix} P^*(E_1E_2E_3) \\ P^*(E_1E_2\overline{E_3}) \\ P^*(E_1\overline{E_2}E_3) \\ P^*(E_1\overline{E_2E_3}) \\ P^*(\overline{E_1}E_2E_3) \\ P^*(\overline{E_1}E_2\overline{E_3}) \\ P^*(\overline{E_1E_2}E_3) \\ P^*(\overline{E_1E_2E_3}) \end{pmatrix} = M_3 \cdot \begin{pmatrix} P(E_1E_2E_3) \\ P(E_1E_2\overline{E_3}) \\ P(E_1\overline{E_2}E_3 \\ P(E_1\overline{E_2E_3}) \\ P(\overline{E_1}E_2E_3) \\ P(\overline{E_1}E_2\overline{E_3}) \\ P(\overline{E_1E_2}E_3) \\ P(\overline{E_1E_2E_3}) \end{pmatrix}, \tag{7}$$

where M_3 is the coefficiency matrix and

$$M_3 = \begin{bmatrix} \theta^3 & \theta^2(1-\theta) & \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta(1-\theta)^2 & (1-\theta)^3 \\ \theta^2(1-\theta) & \theta^3 & \theta(1-\theta)^2 & \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta(1-\theta)^2 & (1-\theta)^3 & \theta(1-\theta)^2 \\ \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta^3 & \theta^2(1-\theta) & \theta(1-\theta)^2 & (1-\theta)^3 & \theta^2(1-\theta) & \theta(1-\theta)^2 \\ \theta(1-\theta)^2 & \theta^2(1-\theta) & \theta^2(1-\theta) & \theta^3 & (1-\theta)^3 & \theta(1-\theta)^2 & \theta(1-\theta)^2 & \theta^2(1-\theta) \\ \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta(1-\theta)^2 & (1-\theta)^3 & \theta^3 & \theta^2(1-\theta) & \theta^2(1-\theta) & \theta(1-\theta)^2 \\ \theta(1-\theta)^2 & \theta^2(1-\theta) & (1-\theta)^3 & \theta(1-\theta)^2 & \theta^2(1-\theta) & \theta^3 & \theta(1-\theta)^2 & \theta^2(1-\theta) \\ \theta(1-\theta)^2 & (1-\theta)^3 & \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta^3 & \theta^2(1-\theta) \\ (1-\theta)^3 & \theta(1-\theta)^2 & \theta(1-\theta)^2 & \theta^2(1-\theta) & \theta(1-\theta)^2 & \theta^2(1-\theta) & \theta^2(1-\theta) & \theta^3 \end{bmatrix} \tag{8}$$

Similar techniques can be employed to extend the above schemes to four-group scheme, five-group scheme, and so on. In this paper, we only describe our results for up to three-group scheme because the undesirable performance for schemes beyond the three-group scheme makes them not very useful. We will compare the performance of various schemes in Section 5.

4.5 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 knowledge discovery problems. 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 building decision tree classifiers is ID3 [10]. 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:

$$Entropy(S) = -\sum_{j=1}^{m} Q_j \log Q_j, \tag{9}$$

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:

$$Gain(S, A) = Entropy(S) - \sum_{v \in A} (\frac{|S_v|}{|S|} Entropy(S_v)), \tag{10}$$

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 multi-group 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, and 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) as what the original ID3 algorithm does. We have to use estimation.

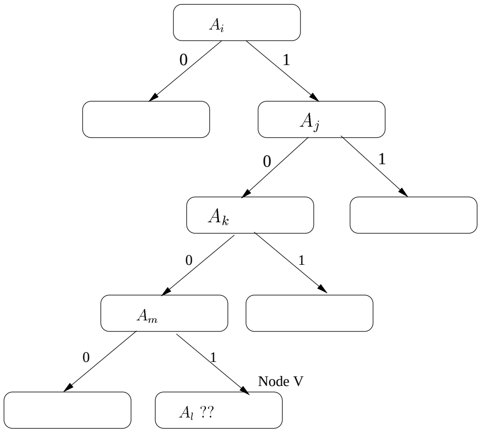


Figure 2: The Current Tree

We will show, as an example, 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-group scheme. Without loss of generality, we assume A_i and A_j belong to group 1, A_k belongs to group 2, and A_m belongs to group 3. Let S be the training data set consisting of the samples that belong to node V (i.e., all data samples in S satisfy $(A_i = 1) \wedge (A_j = 0) \wedge (A_k = 0) \wedge (A_m = 1)$). The part of the tree that is already built at this point is depicted in Figure 2.

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

$$\begin{aligned} E_1 &= (A_i = 1) \wedge (A_j = 0) \\ \overline{E_1} &= (A_i = 0) \wedge (A_j = 1) \\ E_2 &= (A_k = 0) \\ \overline{E_2} &= (A_k = 1) \\ E_3 &= (A_m = 1) \\ \overline{E_3} &= (A_m = 0) \end{aligned} \tag{11}$$

$P^*(E_1E_2E_3)$, $P^*(E_1E_2\overline{E_3})$, $P^*(E_1\overline{E_2}E_3)$, $P^*(E_1\overline{E_2E_3})$, $P^*(\overline{E_1}E_2E_3)$, $P^*(\overline{E_1}E_2\overline{E_3})$, $P^*(\overline{E_1E_2}E_3)$, and $P^*(\overline{E_1E_2E_3})$ can be directly obtained from the (whole) disguised data set. Feeding the above terms into Eq.(7) (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 first (we assume the class label is also binary for this example, and the class label is also disguised). Let

$$\begin{aligned} E_1 &= (A_i = 1) \wedge (A_j = 0) \wedge (Class = 0) \\ \overline{E_1} &= (A_i = 0) \wedge (A_j = 1) \wedge (Class = 1) \\ E_2 &= (A_k = 0) \wedge (Class = 0) \\ \overline{E_2} &= (A_k = 1) \wedge (Class = 1) \\ E_3 &= (A_m = 1) \wedge (Class = 0) \\ \overline{E_3} &= (A_m = 0) \wedge (Class = 1) \end{aligned} \tag{12}$$

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:

$$\begin{aligned} E_1 &= (A_i = 1) \wedge (A_j = 0) \wedge (Class = 0)(or\ Class = 1) \\ \overline{E_1} &= (A_i = 0) \wedge (A_j = 1) \wedge (Class = 0)(or\ Class = 1) \\ E_2 &= (A_k = 0) \wedge (Class = 0)(or\ Class = 1) \\ \overline{E_2} &= (A_k = 1) \wedge (Class = 0)(or\ Class = 1) \\ E_3 &= (A_m = 1) \wedge (Class = 0)(or\ Class = 1) \\ \overline{E_3} &= (A_m = 0) \wedge (Class = 0)(or\ Class = 1) \end{aligned} \tag{13}$$

We can compute $P^*(E_1E_2E_3)$, $P^*(E_1E_2\overline{E_3})$, $P^*(E_1\overline{E_2}E_3)$, $P^*(E_1\overline{E_2E_3})$, $P^*(\overline{E_1}E_2E_3)$, $P^*(\overline{E_1}E_2\overline{E_3})$, $P^*(\overline{E_1E_2}E_3)$, and $P^*(\overline{E_1E_2E_3})$ directly from the (whole) disguised data set. Then we solve Eq.(7) 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. Note that $P(E_1E_2E_3)$ we get here is different from $P(E_1E_2E_3)$ we get while computing $|S|$.

Now suppose attribute A_l, which belongs to group 3, is a candidate attribute, and we want to compute $Gain(S, A_l)$. A number of values are needed: $|S_{A_l=1}|$, $|S_{A_l=0}|$, Entropy($S_{A_l=1}$), and Entropy($S_{A_l=0}$). These values can be similarly computed. For example, $|S_{A_l=1}|$ can be computed by letting

$$\begin{aligned} E_1 &= (A_i = 1) \wedge (A_j = 0) \\ \overline{E_1} &= (A_i = 0) \wedge (A_j = 1) \\ E_2 &= (A_k = 0) \\ \overline{E_2} &= (A_k = 1) \\ E_3 &= (A_m = 1) \wedge (A_l = 1) \\ \overline{E_3} &= (A_m = 0) \wedge (A_l = 0) \end{aligned} \tag{14}$$

We then apply Eq.(7) 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 computed similarly.

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 database satisfy E_1, E_2, and E_3. In our algorithm, such counting (on the disguised data) only gives $P^*(E_1E_2E_3)$, which can be considered as the "disguised" $P(E_1E_2E_3)$ because $P^*(E_1E_2E_3)$ counts the records in the disguised database, not in the actual (but non-existing) database. The proposed multi-group scheme allow us to estimate $P(E_1E_2E_3)$ from $P^*(E_1E_2E_3)$.

4.6 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. An important step in testing is to use the decision tree to classify a data item (a record) from the testing data set, then the classification result is compared to the actual class label of the record. After testing all the records in the testing data set, an accuracy score will be calculated.

Conducting the testing is straightforward when data are not disguised, but it is a non-trivial task when the testing data set is disguised. 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. For simplicity, we only describe how to conduct testing using the three-group scheme (since one-group and two-group schemes are special cases of three-group scheme). We use an example to illustrate how we compute the accuracy score. Assume the number of attributes is 5, and the probability $\theta = 0.8$. 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 group 1, A_3 belonging to group 2, and A_4 and A_5 belonging to group 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 (disguised) testing data set $S = S_1S_2S_3$, we construct other data sets $\overline{S_1}S_2S_3$, $S_1S_2\overline{S_3}$, $S_1\overline{S_2}S_3$, $S_1\overline{S_2S_3}$, $\overline{S_1}S_2S_3$, $\overline{S_1}S_2\overline{S_3}$, $\overline{S_1S_2}S_3$, and $\overline{S_1S_2S_3}$ 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 as the *original undisguised* testing data set, and $\overline{U_i}$ as the opposite of U_i.

Let $P^*(ccc)$ be the proportion of correct predictions from testing data set $S_1S_2S_3$, $P^*(\overline{c}cc)$ be the proportion of correct predictions from testing data set $\overline{S_1}S_2S_3$, $\cdots$, $P^*(\overline{ccc})$ be the proportion of correct predictions from testing data set $\overline{S_1S_2S_3}$. Similarly, let $P(ccc)$ be the proportion of correct predictions from the *original undisguised* data set $U_1U_2U_3$, $P(\overline{c}cc)$ be the proportion of correct predictions from $\overline{U_1}U_2U_3$, $\cdots$, $P(\overline{ccc})$ be the proportion of correct predictions from $\overline{U_1U_2U_3}$. $P(ccc)$ is what we want to estimate.

Because $P^*(ccc)$, $P^*(\overline{c}cc)$, $\cdots$ and $P^*(\overline{ccc})$ consist of contributions from $P(ccc)$, $P(\overline{c}cc)$, $\cdots$ and $P(\overline{ccc})$, we have the following formula:

$$\begin{pmatrix} P^*(ccc) \\ P^*(cc\overline{c}) \\ P^*(c\overline{c}c) \\ P^*(c\overline{cc}) \\ P^*(\overline{c}cc) \\ P^*(\overline{c}c\overline{c}) \\ P^*(\overline{cc}c) \\ P^*(\overline{ccc}) \end{pmatrix} = M_3 \cdot \begin{pmatrix} P(ccc) \\ P(cc\overline{c}) \\ P(c\overline{c}c) \\ P(c\overline{cc}) \\ P(\overline{c}cc) \\ P(\overline{c}c\overline{c}) \\ P(\overline{cc}c) \\ P(\overline{ccc}) \end{pmatrix} \tag{15}$$

where M_3 is defined in Eq.(8). $P^*(ccc)$, $P^*(\overline{c}cc)$, $\cdots$ and $P^*(\overline{ccc})$ can be obtained from testing data set $S_1S_2S_3$, $\overline{S_1}S_2S_3$, $\cdots$ and $\overline{S_1S_2S_3}$. By solving the above formula, we can get $P(ccc)$, the accuracy score of testing.

4.7 Partial Disguise

It is possible in some cases that only a subset of the attributes are sensitive, or maybe just one attribute is sensitive. In this case, disguising all of the attributes in each group together can be dangerous. For example, if it is fairly easy to find out whether a subject is male or female, we should not disguise the gender attribute together with the sensitive attributes, because anyone can derive the values of the sensitive attributes based on the values of an insensitive attribute if they are disguised in the same way. To solve this problem, we should just disguise the sensitive attributes. Our described method can be easily adjusted to deal with this situation.

4.8 Extension

In this paper, we consider the cases where each individual provides a single record to the database, but there are cases when an individual itself is a database owner, and it can contribute multiple records to the new database. For example, this could happen when several companies want to combine their databases together to form a much bigger database, which is then used for data mining computations. When the number of collaborators is not big, there are solutions that can produce accurate data mining results [6, 15]. Although this problem is not our focus, the solution to the problem described in this paper can be easily extended to solve the above problem.

The techniques described in this paper can be also extended to other data mining computations, where the computations are based on aggregate values of data. For example, we can use the same techniques to conduct the association rule mining on the disguised data. Further studies on this extension will be conducted in the future.

5 Experimental Results

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

5.1 Data Setup

We conduct experiments on three real life data sets. We obtain the data sets from the UCI Machine Learning Repository[3]. The first dataset is called *Adult* which was used in [4]. The original owners of the data set

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

Table 1: The Summary of Data Sets

Data Set Name	Adult	Breast-Cancer	Congressional-Voting
Number of Attributes	14	10	16
Number of Records	48,842	786	435

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. The second data set is called *Breast-Cancer*. The original owners are National Institute of Diabetes and Digestive and Kidney Diseases. It came from University of Wisconsin Hospitals. The donor of this data set is Dr. William H. Wolberg. It has 699 instances with 10 attributes. Prediction task is to decide whether a person is benign or malignant. The third data set, which has 435 instances with 16 attributes, is 1984 United States *Congressional-Voting* Records. It was donated by Jeff Schlimmer. We summarize data sets in table 1.

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 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.

5.2 Experimental Steps

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. 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.

We conduct the experiments for three schemes. In the experiments for the one-group scheme, we treat all the data as one data set; for the two-group shceme, we randomly split the whole data set into two groups; and for the three-group scheme, we randomly split the whole data set into three groups.

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 (or the benchmark score).

θ **Selection:** For $\theta = 0.0, 0.1$, 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:

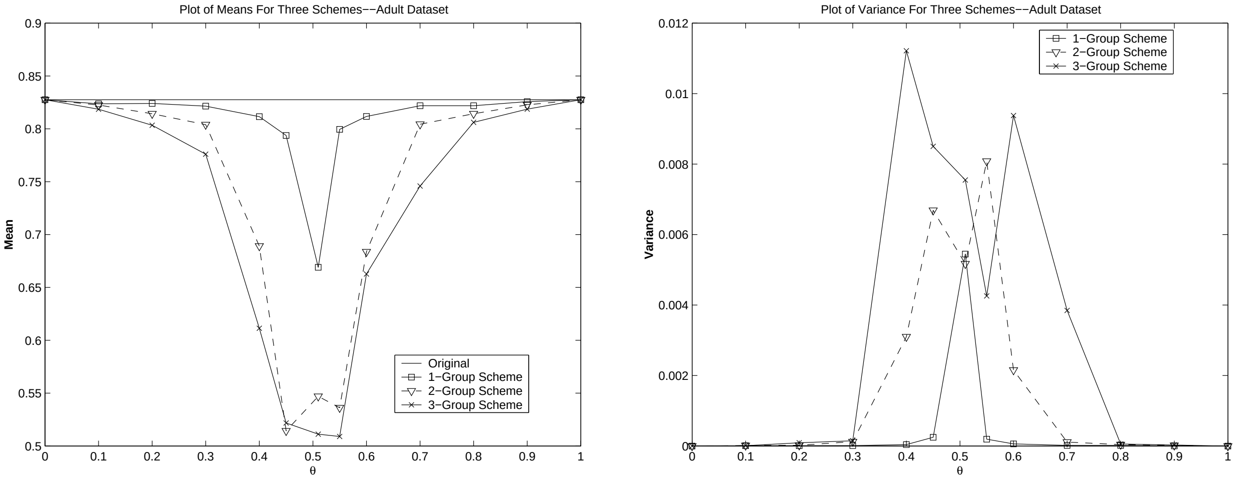
1. Randomization: For the one-group scheme, we create a disguised data set G. For each record in the training data set D, we generate a random number r from 0 to 1 using uniform distribution. If $r \leq \theta$, we copy the record to G without any change; if $r > \theta$, we copy the opposite of the 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. For the two-group and three-group scheme, we randomly split D into two or three groups, and conduct the above randomization for each group, finally obtain disguised data set G. Note that the random numbers used for different groups 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 50 times, and get $S_1, \ldots, S_{50}$. We then compute the mean and the variance of these 50 accuracy scores.



(a) Mean

(b) Variance

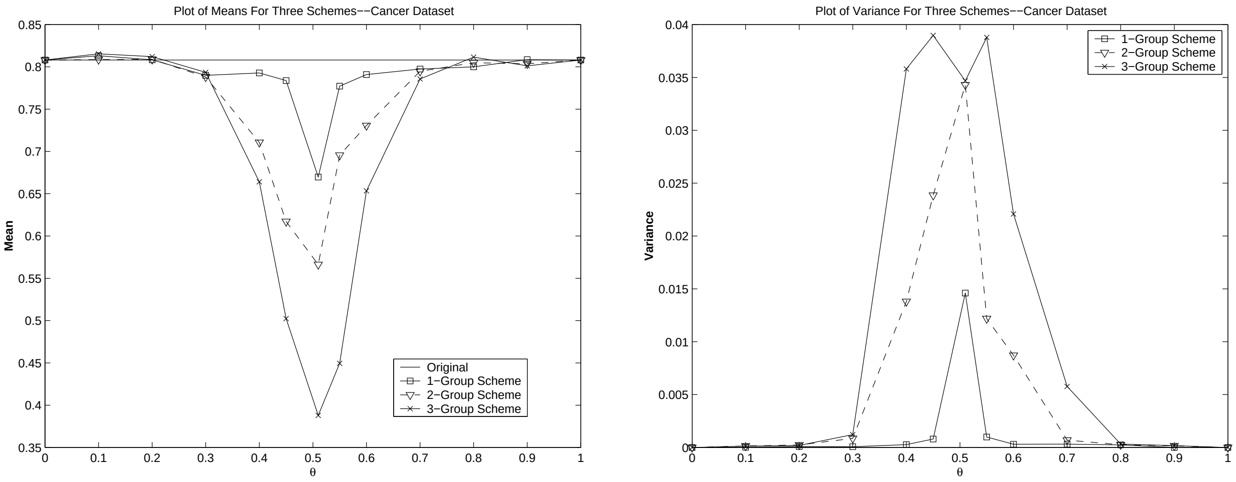
Figure 3: The Results On The Adult Data Set

5.3 The Result Analysis

5.3.1 The Analysis of Mean

Fig. 3(a), 4(a), and 5(a) shows the mean values of the accuracy scores for *Adult*, *Breast-Cancer*, and *Congression-Voting* data sets respectively. In each figure, we combine three schemes together. We can see from the figures that when $\theta = 1$ and $\theta = 0$, the results are exactly the same as the rasults 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 an adversary can know the real values of all the records provided that he/she knows the θ value. Within the same group scheme, when θ moves from 1 and 0 towards 0.5, the mean of accuracy has the trend of decreasing although there are some outliers[4]. When θ is around 0.5, the mean deviates a lot from the original accuracy score. For each randomization

[4] We will further analyse these point in the following.



(a) Mean

(b) Variance

Figure 4: The Results On The Breast-Cancer Data Set

level θ, the mean of the accuracy decrease with the number of group increasing. Fig. 6 provide a snapshot of the relationship between the mean of the accuracy and the number of groups and θ based on *adult* data set. We can see from this figure that with the number of group increasing and θ approaching to 0.5, the mean of accuracy decreases; and with the number of groups decreasing and θ deviating from 0.5, the mean of accuracy increases.

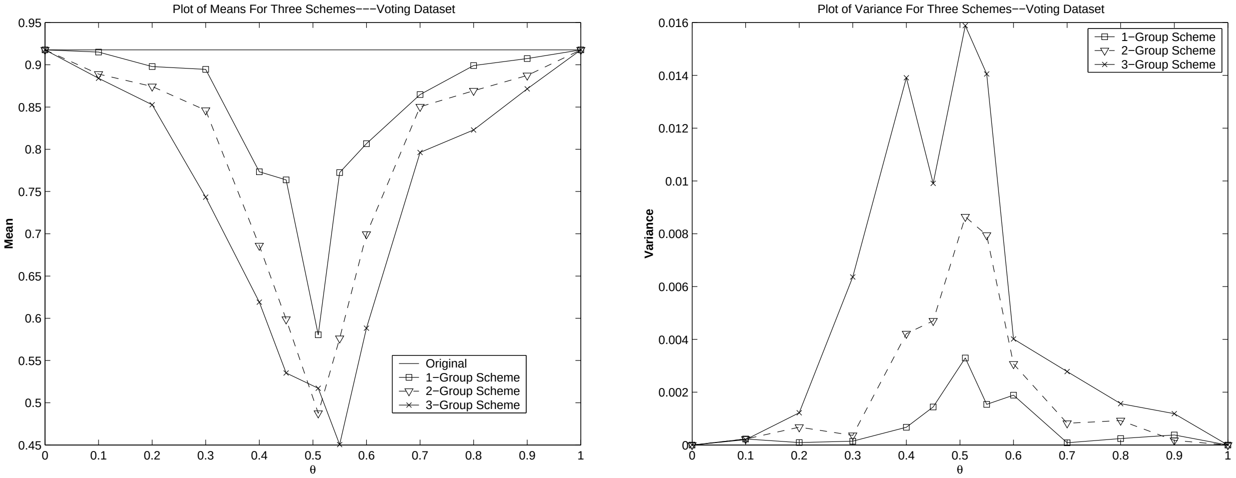
We need also point out that the results on the *Congressional-Voting* data set is not as good as the results on the other two data sets. The possible reason is that the number of records in the *Congressional-Voting* data set is only 435, which is not large enough for achieving accurate estimations.

5.3.2 The Analysis of Variance

Fig. 3(b), 4(b), 5(b) shows the variances of the accuracy scores for *Adult*, *Breast-Cancer*, and *Congression-Voting* data sets respectively. We also combine the results for three schemes together. Within the same scheme, when θ moves from 1 and 0 towards 0.5, the degree of randomness in the disguised data is increased, the variance of the estimation used in our method becomes large. There are two sources which may contribute to the variance.

1. The *within-group* variance which is contributed by the variance within each group. Especially when the randomization level is different, the *within-group* variance will be different. 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 case when θ is not around 0.5. This is actually what we have predicted. We use a simple example to illustrate why this happens. Assume we have just one attribute, with 90% of 1's and 10% of 0's. If we choose $\theta = 0.5$, according to our randomization scheme, the disguised data set will contain $90\% * 0.5 + 10\% * 0.5 = 50\%$ of 1's and another 50% of 0's. If we change the distribution to 10% of 1's and 90% of 0's, we get the same results. This means when $\theta = 0.5$, information about the data distribution is lost. That is why when θ closes to 0.5 the accuracy becomes very low[5], and the

[5] Note, 0.5 is a very low accuracy, because if one just randomly guesses the class label, 1 out 2 guesses will be correct if we have



(a) Mean

(b) Variance

Figure 5: The Results On The Congressional-Voting Data Set

variance becomes very large.

2. The *between-groups* variance which is caused by the variance between groups. As the results show, with the number of groups increasing, the variance increases. The reason is that randomization process is conducted independently for different groups.

5.3.3 Privacy Analysis

We conduct privacy analysis from two aspects: one is the case where we fix the scheme, the other is the case where we fix θ value. For the first case, 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 increases. 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. For the second case, with the number of groups increasing, the privacy level increases since randomization process is conducted independently for different groups.

5.3.4 Summary

Our results on the three real life data sets indicate that the multi-group randomized response techniques can be utilized for privacy-preserving decision tree classification. When θ is 0 or 1, which provides all the true information, the accuracy of the tree is the highest and the privacy level of the data is the lowest. When θ is away from 0(or 1) and approaches to 0.5, the accuracy of the tree decreases and the privacy level of the data increases. On the other hand, when the number of groups increases, the privacy level of data increases but the accuracy of tree decreases. In practice, we can adjust the privacy level and the tree prediction accuracy level by modifying θ value and the number of groups. We need to point out that when

just two class labels. Therefore even the random guess can achieve accuracy of 0.5.

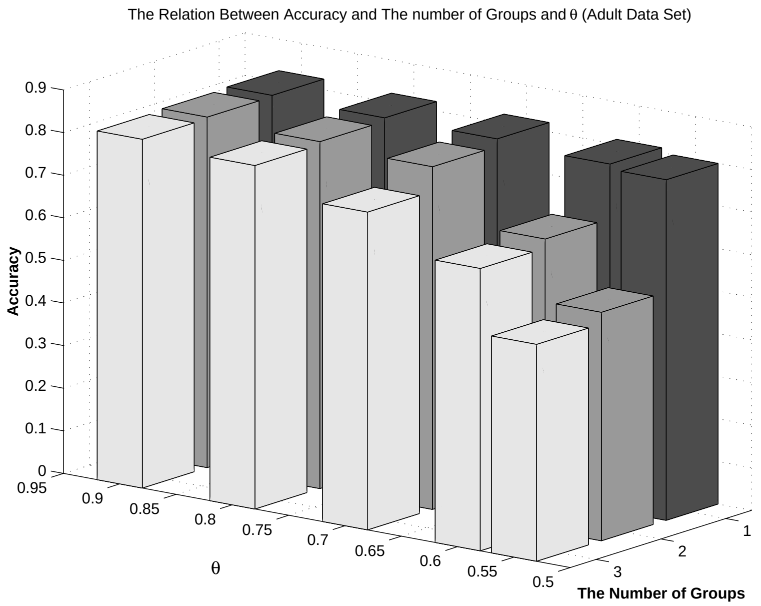


Figure 6: The Relation Between Accuracy and The Number of Groups and θ

$\theta = 0.5$, the related model cannot be applied, and other techniques such as randomized response techniques using the *Unrelated-Question* model may be employed.

6 Conclusion and Future Work

In this paper, we have presented a multi-group scheme for building decision tree classifiers based on randomized response techniques. Our method consists of two parts: the first part is the multi-group data disguising technique used for data collection; the second part is the modified ID_3 algorithm for decision tree building, which is used for building a classifier from the disguised data, based on our proposed multi-group scheme. 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 $[0.55, 1]$ and $[0, 0.45]$ for one-group scheme, $[0.7, 1]$ and $[0, 0.3]$ for two and three- group schemes, we can obtain fairly accurate decision trees comparing to the trees built from the undisguised data. Ideally, we can split the data set to N groups where N is the total number of attributes in the whole data set. As our experiments shows, with the number of groups increasing, the accuracy level decreases and the privacy level increases. In our future work, We will apply our technique to other types of data mining. We will also extend our solution to data sets consisting of non-binary data types.

References

[1] Anonymizer.com: http://www.anonymizer.com.

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

[4] J. Cheng and R. Greiner. Learning Bayesian belief network classifiers: Algorithms and system. *Lecture Notes in Computer Science*, 2056:141–151, 2001.

[5] 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`.

[6] 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.

[7] W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proceedings of The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, August 24-27 2003.

[8] 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.

[9] M. Franklin and M. Yung. Varieties of secure distributed computing. In *Proc. Sequences II, Methods in Communications, Security and Computer Science, Positano, Italy*, pages 392–417, June 1991.

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

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

[12] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transaction. *The ACM Transactions on Information and System Security*, 1(1):Pages 66–92, 1998.

[13] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *Proceedings of 1997 IEEE Symposium on Security and Privacy*, Oakland, California, USA, May 5-7 1997.

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

[15] 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.

[16] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Ddata Mining*, 2003.

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

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

[19] A. C. Yao. How to generate and exchange secrets. In *Proceedings 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.