Multi-Dimensional Inference and Confidential Data Protection with Decision Tree Methods

LiWu Chang & James Tracy
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375 USA
{*lchang,tracy*}*@itd.nrl.navy.mil*

Abstract

We present a novel approach to the challenging issue of database confidential data protection. We adopt the decision tree framework as our baseline and extend it to cope with databases where the class_label attribute is not specified. We are interested in confidential data that are randomly distributed over different attributes (referred to as *multi-dimensional inference*). For confidential data protection, our method (referred to as *adaptive modification*) mitigates inference by evaluating and modifying some, not all, relevant data records. We localize data modification in a decision tree and, instead of exhaustively evaluating all modification possibilities, we select informative data to modify. Our proposed method is effective in protection of confidential data and scalable for handling large databases.

1 INTRODUCTION

Safeguarding confidential data of a database has been a challenging issue in the past and emerges as one of the most critical information technologies today. The pressing demand for such a protection technique is partly due to the trend of information sharing between institutions and among coalition members, and the opening of the government databases to the public. The problem that arises when confidential information can be derived from released data by unauthorized users is commonly called the *database inference* problem.

Many of today's efforts in confidentiality protection have been geared towards modifying to-be-released data in order to mitigate inference. Methods of modification include perturbation ([4]) (i.e., alteration of an attribute value to a new value), blocking ([2][13]) (i.e., replacement of an existing attribute value with a "?" indicating ignorance), and aggregation (i.e., combination of several values into one coarser category) ([14]). These modifications are made on the basis of a probabilistic model ([2][5]), decision tree ([1]), association rules ([6][7]) or the rough set theory ([10]).

Our goal is to lay a sound theoretical foundation for confidential data protection. In this paper, we develop inference prevention methods on the basis of a decision tree framework ([12]). The decision tree method conveniently provides a more localized description of data records. The structure of the tree may easily be traced back to individual instances, and the effect of the modification of particular instances on decision making is more clear. It also delivers excellent performance against many benchmark test datasets ([12]). In [1], we applied the decision tree method as our baseline approach to the inference problem, where confidential data were represented as values of the class_label (attribute) of the test data. However, confidential data may be composed of data from different sources and may not be restricted to one attribute (i.e., the class_label). It is cases in which confidential data are distributed over the entire database (referred to here as *multi-dimensional inference*) that interest us. In this paper, we extend the decision tree method in order to handle distributed confidential data.

Decision theoretical-based approaches often suffer from the inability to scale-up to cope with large databases. What limits these approaches the most is not the intricate decision analysis required, but the exhaustive evaluation of the entire databases in a repeated manner during the modification process. Our approach adopts

an adaptive modification strategy which gives effective performance and desirable results.

2 INFERENCE PROBLEM

We consider a simple two-leveled security protocol ([8]) which has *High* and *Low* users. The High users (e.g., the database manager) view the entire database, and the Low users share the High view with the exception of any confidential data. When data are shared, High releases some of the non-confidential data to Low.

Authors of ([11]) have introduced a conceptual model for database inference and discussed the necessary steps involved in dealing with the inference problem. High generates rules from the available data set, and then determines whether there is inference. If the inference is excessive, then it implements a protection plan to lessen the inference (i.e., decides to modify by deleting certain data from the database as it appears to Low). In fact, many database inference papers have alluded to our inference model. The output of our inference model is the database that can be released to Low. Our goal is to make modifications as parsimoniously as possible and thus avoid imposing unnecessary changes which lessen functionality.

2.1 Decision Tree Method

Our analysis of data protection is based on C4.5 decision tree ([12]). The C4.5 decision tree uses an information theoretic test to evaluate the quality of decision tree generation. It classifies a new data record by assigning it the class label possessed by the majority of data records that are at the same *leaf node* (i.e., the end of a branch where a class label is assigned) of the decision tree as the new data record. By convention, the attribute used as the class_label is deterministic. To deal with multi-dimensional inference, we evaluate the possibility of inference on an arbitrary attribute by designating it as the class_label (thereby the original class_label becomes an ordinary attribute.)

In our method, attributes that contain confidential data are viewed as the class labels of the testing data, and the remainder of the database is considered non-confidential. In [1], the database inference problem was viewed as traditional decision tree learning, and the prevention of database inference dealt exclusively with attribute values of the training data (which are only part of the non-confidential data). Confidential data were associated with only one attribute (i.e., the class_label). Inference prevention that structured in this way is clearly insufficient. In this paper, we take into account the entire body of non-confidential data

Table 1: Relational Table for Evaluation. A_j denotes the jth attribute and the "?" denotes an unknown value, a piece of confidential datum, or a previously modified value.

key	A1	A2	..	A_k	..	A_M	class_label
training data	..	?	..	..	..	..	..
	?	..	..	?	..	?	..
	..	..	..	..	..	..	..
testing	..	..	..	?	..	..	?
	..	?	..	..	..	..	?

2.2 Metric

Table 1 shows an instant view of a relational data table. Modification results in placing perhaps more "?"s in the database. At this instant, one of the M attributes has been selected as the class_label. Data modification is likely to incur degradation of database performance. In our approach, important performance metrics include the effectiveness measure of confidential data protection (E) and the measure of the loss of functionality (F) in a database. In terms of the decision tree method, the effectiveness measure for the attribute currently selected as the class_label is determined by the classification error of the test data (i.e. the confidential data), while the measure of loss of functionality is a function of the classification error of the training data (i.e. the to-be-released data).

Suppose the jth attribute is posted as the class_label. Let the measure of protection effectiveness with respect to the jth attribute be denoted as E_j and the measure of the loss of functionality be denoted as F_j. The overall measure of E and F for the entire database are the function (e.g.,weighted average) of E_is and F_is. The measure F is usually has an upper bound of a given threshold υ (i.e., $F \leq \upsilon$) that represents the maximum level of information loss that users are willing to tolerate. With the definitions of E and F in mind, our optimization goal is to ([3])

$$\text{Minimize } E, \text{ while keeping } F \leq \upsilon,$$

i.e., we optimize E with F as the objective function.

Note that the effect of protection is evaluated from High's perspective, while the database functionality is evaluated from Low's view.

3 ADAPTIVE MODIFICATION

Our adaptive strategy exploits the property of *localization* inherent to the decision tree method and modifies not all attribute values, but rather only selected ones. We examine the leaf nodes of a decision tree and study the statistics of the data records at different leaf nodes. By restricting modification to a small area of the database, our approach preserves the database functionality. During modification, we visit the attribute that contains the largest number of confidential data records and receives the lowest classification error (i.e., the highest inference threat). With this selected attribute in mind, we examine the distribution of associated confidential data and modify the leaf node that has the highest population. Clearly, our search strategy may not yield an overall optimal solution. However, the controlled modification scheme can effectively avoid the high computational complexity incurred by exhaustive search.[1] We will describe our modification procedures at three levels in the following sections.

3.1 Selection of Attributes

Let the total number of confidential attribute values be denoted as S, the number of total attribute values be D, and the classification error of the test data with respect to the jth attribute be $Cerr_j$. At the first level, we select from all the M attributes the one that maximizes the product of the number of associated confidential data records and the inverse of the classification error

$$U_{attr} = \max\ (1 - Cerr_j)(\frac{TE_j}{S})$$

where TE_j is the number test data associated with the jth attribute. Suppose attribute A_j is selected. Thus, A_j is posted as the class_label and denoted as C_j.

3.2 Selection of A Leaf Node

For the given C_j, we decide among all leaf nodes from the corresponding decision tree, DT_j, a leaf node to visit. The selection of the leaf node is determined, at least, by (1) the number of correctly labeled training data records at a leaf, and (2) the number of correctly classified test data records at a leaf. Less correctly labeled training data implies that less effort is required to alter the present class label at a leaf node. On the other hand, the more correctly classified test records are, the higher the effect in protection is from modifying the leaf node. Let the selected leaf node be denoted as L_k and the immediate attribute (i.e., the last attribute on the branch) of L_k be A_i. With L_k and C_j in mind, we record the following statistical information:

- R_j: training records w.r.t. C_j
- E_j: testing records w.r.t. C_j
- Tr: correctly labeled training records at L_k
- Te: correctly classified testing records at L_k
- Fr: incorrectly labeled training records at L_k
- Fe: incorrectly classified testing records at L_k

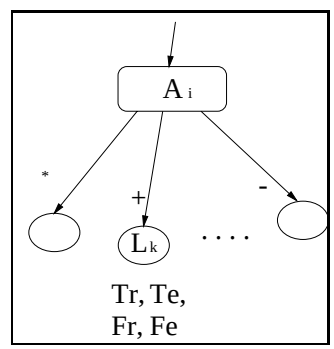

Figure 1: L_k is the leaf node and A_i is the immediate attribute. Te, Tr, Fe and Fr denote the statistical information of the data records associated with L_k. Assume that at L_k class_label is "+".

With these statistical information, the utility function that combines the above two factors is as follows:

$$U_{leaf} \ = \ \frac{Te}{E_j}(1 - \frac{Tr - Fr}{R_j \ + \ E_j}),$$

We need information about the data distribution of the neighbors (i.e., leaf nodes of different values of A_i) of L_k. At A_i, we store the relationship of data records to its fellow leaf nodes as the ratio $a1 : a2 : ... : al$, for l different values of A_i. Furthermore, those leaf nodes with the same class label as L_k will be collectively denoted as a_+ and those with different labels be a_-.

[1] Exhaustive search means the evaluation of every possible batch of attribute values of non-confidential data.

3.3 Selection of Modification Methods

At the leaf node L_k, our strategy to mitigate inference involve two aspects:

(S1) Reduce the correctly classified test records.
(S2) Reduce the correctly labeled training records.

The result of (S1) is expected to produce higher classification error of the test data, while the result of (S2) may cause the change of the value of the class_label at L_k and thus, affect the outcome of decision analysis. To implement our strategy, we envision the following three possible ways:

- (I1) Modify attribute values of correctly classified test records.
- (I2) Remove the value of class_label of correct training records.
- (I3) Modify attribute values correctly labeled training records.

For the purpose of minimizing the impact of modification, in both I1 and I3, we localize changes by replacing only those values of the immediate attribute (i.e., A_i) with "?"s.[2] In I2, the values of the class_label of some training data records are blocked. As a consequence, a training data record with its class label being blocked will be excluded from the training data set. Item (S2) is carried out by implementing I2 and I3, while item (S1) consists only of I1. In both I1 and I3, we increase the uncertainty of classification by moving (or, redistributing) correct testing and training data records to neighboring leaf nodes. In both I2 and I3, the effort is to make the number of incorrectly labeled training records to outnumber incorrectly labeled ones. The difference between I2 and I3 is that the effect of I3 depends upon the distribution of data records in the neighborhood of L_k. Unlike I2, I1 and I3 are intended to 'smear' data records of neighbor leaf nodes.

The choice between the three modification methods will depend on an estimation of the computational *cost* and *gain* in confidentiality, where the cost refers to the total number of modifications executed and the gain refers to the number of data records whose class labels have been successfully altered.

[2] In decision tree analysis, a data record with "?"s at its attribute values is called the uncertain evidence. Suppose the attribute value of hth attribute is "?" and the hth attribute is used in its classification path. Then the impact (or weight) of this data record is split among the group of leaf nodes under the hth attribute according to the population distribution.

I1. The cost of I1 is Te, for all correctly classified test records will be modified. On the other hand, the gain is $\frac{a_-}{a_+ + a_-}Te$, because those $\frac{a_-}{a_+ + a_-}Te$ data records that used to be correct now become incorrect, where $\frac{a_-}{a_+ + a_-}$ is the ratio of the number of re-distributed test records will receive incorrect label. Summing the loss and the gain yields the net loss of I1, which is $\frac{a_-}{a_+ + a_-}Te$.

I2. For I2, the condition of applicability is that Tr and Fr are close. Because the removal of the class label of a training record results in deletion, the cost of I2 is $Tr - Fr + 1$, meaning that the amount of removal is determined by the difference between Tr and Fr. After deletion, the associated class label of L_k will change. This means there are Te test records that will change to a wrong class sign and Fe test records that will change to the correct one. So, the gain is $Te - Fe$. (Note that if $Te \leq Fe$, modification of L_k is avoided.) The net loss for I2 is therefore, $((Tr - Fr + 1) - (Te - Fe)$.

I3. For I3, the applicability is the same as I2. As in the case of I1, modification will be restricted to the value of the nearest attribute (i.e., A_i). The gain that arises from applying I3 is also $Te - Fe$ as that of I2. However, the calculation of the cost is more involved, because a modified training record will become an uncertain evidence whose impact (or, weight) will be distributed among different values (i.e., leaf nodes) of A_i. The number of changes (denoted as c) needed to alter the associated class label can be iteratively determined by [3]

$$T_r - c + \sum_{i=1}^{c} \alpha_i \leq F_r.$$

In this case, the cost of I3 is c. By putting together the cost and the gain, the net loss of I3 can be obtained as $(c - (Te - Fe))$.

[3] Let c be the amount of necessary changes with values at A_i being replaced by "?"s. For each modification, Tr becomes $(Tr - 1 + \alpha_j)$, where α_j is the fraction of the mass of this modified record that gets back from re-distribution. For the sign of the class label to change, we want

$$T_r - c + \sum_{i=1}^{c} \alpha_i \leq F_r,$$

with

$$\alpha_{i+1} = \frac{a_+ - i + \sum_{j=1}^{i} \alpha_i}{a_+ + a_- - 1}.$$

3.4 Control Step

By comparing the net losses of the three approaches, we pick the modification method with the minimum loss. Modification hides one attribute value at a time until either the leaf node is exhausted or the threshold of allowed modifications of the present class_label is reached, where the threshold of modification is determined according to the ratio of number of confidential values that is with this particular attribute (i.e., the class_label) and with other attributes. After modification is carried out, we compute E and F and determine whether or not F exceeds the given threshold. If it does not, our modification procedures will be repeated from the top level.

4 DISCUSSION

Decision theoretical-based approaches to confidential data protection have been widely pursued by researchers from different fields. Exhaustive evaluation incurs extremely high computational complexity and hence, impedes the scalability of existing approaches. We presented an adaptive modification method with a basis in the decision tree framework. The transparency of decision trees make them an excellent tool for analyzing how specific data modifications may affect inference possibilities. Our adaptive strategy selects and modifies the most informative attribute values, with information about statistical distribution obtained from decision tree analysis, to effectively and parsimoniously handle the database inference problem. Furthermore, it localizes modification operations in a manner that preserves database performance.

4.1 Complexity

The gain in computational complexity is obvious. Let M, N, S and G denote, respectively, the number of attributes, data records, confidential attribute values, and modified attribute values that are sufficient for data protection. In the (batch) exhaustive evaluation, the complexity is the combinatorial $G\binom{(MN)-S}{G}$[4], while in our approach, it is the polynomial order of M^2S. In fact, because exhaustive search involves large number of repeated tree generation, it becomes impractically expensive to use for even a small relational table of the dimension of $N = 20$ and $M = 5$ with υ=25%. With our proposed method, we are able to obtain satisfactory results in terms of performance and protection.

We are presently experimenting with some data sets from UCI repositories (e.g., [9]) and will test various KDD databases. We have tested methods of exhaustive search, single-attribute-valued best-first search[5], and our informative modification.

4.2 Evaluation

As mentioned, our evaluation of confidential data protection is based on the average of the classification error of the test data (i.e., the modified non-confidential data) with respect to each class_label. We justify the proposed approach by comparing the results with those obtained from a best-first search. With the well-known voting records dataset ([9]), the proposed approach selects and modifies attribute values of test records with the modification method I1 being chosen. The results of modification is close to those obtained from the best-first search. For instance, with 20 confidential attribute values, in the best-first search the classification error increases from 5.26% to 57.90% with 11 modifications. The proposed adaptive modification selects 9, that are part of the 11, modifications. The modification method I2 is likely to be selected in the case that a leaf node is associated with a small number of training records, but a large number of test data records. The adaptive modification was motivated by our experiments with best-first search and thus, the performance of our proposed selection strategy is expected to be very close to the performance of the best-first strategy on the voting record data. In addition to the reduction of computational complexity, the adaptive modification avoid unnecessary modifications, which can be a large quantity, at the beginning and towards the end of selection that the best-first strategy may face (due to the ineffectiveness of the data selection criterion under certain conditions.)

[4] If the number of attribute values to be modified is not known *a priori*, different values of G will be tested until performance bound υ is met in the average sense.

[5] In this approach, for each attribute value, we estimate its impact on the average classification error. After evaluating all attribute values, we hide the attribute value with the maximum impact and update decision tree. Then, we resume the next round selection.

4.3 Restoration

An attacker may know our inference prevention strategy. As a result, modified attribute values can be restored and hence, confidential data are not correctly protected. We perform experiments in which the remaining unhidden attribute values were used to infer the attribute values that had been hidden in our first experiment. In term of voting data, the result shows that the previously hidden attribute values (e.g., "physician fee freeze") are restored by using some other attributes (e.g., "El Salvador aid"). In the wake of the possibility of modified values being restored, we repeat the process of attribute value hiding, making previously hidden attributes confidential, until restoration risk goes below a specified threshold. The need of repeated hiding (referred to as the *ramification* problem of database inference [2]) presents a challenge to the value suppression (e.g., blocking) modification strategy. We will explore the restoration and other types of attacks in our future work.

5 FUTURE WORK

Our present blocking-based modification may not be the most effective means of modifying data. We will also experiment with perturbation method. We have not yet discussed the value-restoration problem, in which an attacker might restore blocked values in the same manner that he restores confidential data. Also, we did not discuss particular numerical values which might be assigned to the tolerance level. We leave these issues as part of our future work.

Our evaluation of the proposed method is based on empirical study by comparing it with different decision tree approaches. We will evaluate it against other existing method based on different frameworks for microdata suppression.

References

[1] Chang, L. & Moskowitz, I. S. (1998) "Parsimonious Downgrading and Decision Trees Applied to the Inference Problem," Proc. NSP Workshop, pp. 82-89.

[2] Chang, L & Moskowitz, I. S. (2000) "An Integrated Framework for Database Inference and Privacy Protection," *Data And Applications Security*, (eds. Thuraisingham, van de Riet, Dittrich & Tari), Kluwer, pp. 161-172.

[3] Chang, L & Moskowitz, I. S. (2002) "A Study of Inference Problems in Distributed Databases," Proc. of IFIP Data Security and Applications, Cambridge, UK, pp 229-243.

[4] Cox, L. (1987) "A Constructive Procedure for Unbiased Controlled Rounding," J. of the American Statistical Association, 82, pp. 520-524.

[5] Dobra, A. & Fienberg, S.E. (2000). "Bounds for cell entries in contingency tables given marginal totals and decomposable graphs," Proceedings of the National Academy of Sciences, 97, No. 22, 11885-11892.

[6] Evfimievski, A., Srikant, R., Agrawal, R. & Gehrke, J. "Privacy Preserving Mining of Association Rules", Proc. of the 8th ACM SIGKDD, Canada, July 2002.

[7] Johnsten, T. & Raghavan, V. (1999) "Impact of Decision-Region Based Classification Mining Algorithms on Database Security," IFIP WG 11.3, Seattle, pp. 177-191.

[8] Kang, M.H., Froscher, J.N., & I.S. Moskowitz (1997) "An Architecture for Multilevel Secure Interoperability," Proc. 13th ACSAC Conference, San Diego, CA.

[9] http://kdd.ics.uci.edu/ .

[10] Lin, T. Y. (1993) "Rough Patterns in Data-Rough Sets and Intrusion Detection Systems," J. of Foundation of Computer Science and Decision Support, Vol. 18, No. 3-4, pp. 225-241.

[11] Moskowitz, I. S. & Chang, L. (2000) "A Computational Intelligence Approach to the Database Inference Problem," *Advances in Intelligent Systems:Theory and Applications* IOS Press, pp. 377-387.

[12] Quinlan, R. (1992) *C4.5*, Morgan Kaufmann.

[13] Saygin, Y., Verykios, V. & Clifton, C. (2001) "Using Unknowns to Prevent Discovery of Association Rules." SIGMOD Record 30(4): pp. 45-54.

[14] Zayatz, L. & Rowland, S. (1999) "Disclosure Limitation for American Factfinder," Census Bureau report (manuscript).