12

# IMPACT OF DECISION-REGION BASED CLASSIFICATION MINING ALGORITHMS ON DATABASE SECURITY

Tom Johnsten and Vijay V. Raghavan

**Abstract** In this paper, we investigate issues pertaining to the assessment of the impact of classification mining on database security. Specifically, the security threat presented by a category of classification mining algorithms that we refer to as decision-region based is analyzed. Providing safeguards against this threat requires, in part, the development of new security policies. Our specific contributions are the proposal of a set of security policies for use in the context of decision-region based classification mining algorithms along with the specification and implementation of a security risk measure that allows for the realization of a subset of the proposed policies.

**Keywords:** Classifier, data mining

## Introduction

Chris Clifton and Don Marks are among a small number of researchers who have examined the potential impact of KDD technology on database security. In their paper, *Security and Privacy Implications of Data Mining,* Clifton and Marks outline several general strategies designed to eliminate or reduce the security risk presented by this new technology [2]. Their strategies include allowing users access to only a subset of data, altering existing data or introducing additional (spurious) data. They contend that the application of such policies is most effective in the context of specific learning tasks. These tasks include classification, estimation, clustering, characterization and association [1, 4, 3]. Of special interest to the current work are the classification mining algorithms, which have the potential to disclose sensitive information whenever a database contains both "sensitive" and "non-sensitive" data [2]. Specifically,

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35508-5_22

V. Atluri et al. (eds.), *Research Advances in Database and Information Systems Security*

our current work has focused upon the need to develop security policies designed to minimize or eliminate the threat presented by classification mining in the context of the relational data model.

A valid security policy with respect to classification mining is the protection of all the attribute values of a tuple whenever a tuple includes at least one sensitive, or *protected*, data element. That is, the tuple is entirely eliminated from the user's view. In general, such a policy unnecessarily restricts a user's access to the data. An alternative policy, which is the one we propose, is to protect only data elements that need to be concealed in order to prevent the disclosure of sensitive information through classification mining. This type of policy has the obvious advantage of allowing maximum use of the data and at the same time protecting sensitive information. However, the implementation of such a policy requires an accurate assessment of the data in order to determine a protected data element's risk of disclosure and the need to conceal additional data elements.

A possible assessment strategy is to assess a protected data element's risk of disclosure in the context of a specific classification algorithm. Then, based on the results for several selected methods, a decision can be made with regards to a protected data element's risk of disclosure. An alternative strategy is to make a generic assessment of a protected data element's risk of disclosure that is independent of a specific classification method. This strategy has a number of potential advantages over the former. These include:

- Producing security policies that are applicable to a general set of classification methods.
- Providing insight on how to modify the protection level of a protected data element.
- Reducing the time complexity of the process of assessing the risk of disclosure.

Unfortunately, a completely generic assessment that is independent of a specific classification method is in all likelihood an impossibility as a result of variations among classification mining algorithms. However, such an assessment becomes feasible when the scope of the evaluation is limited to a specific group of classification algorithms and/or certain restrictions are placed on the domain of the given attributes. The realization of this condition requires, of course, the partitioning of classification algorithms into groups that have uniform assessment properties of a protected data element. We have currently identified one such group of algorithms, which we will refer to as decision-region based.

The primary focus of this paper is on the assessment of a protected data element's risk of disclosure with respect to the decision-region based classification algorithms. To that end, the rest of this paper is organized as follows. Section one presents a general overview of classification mining along with an example to illustrate the security threat presented by classification mining algorithms. Section two proposes a set of security policies that when implemented have the potential to provide a high level of protection against classification mining and at the same time maximize access to the given data. Section three characterizes the decision-region based classification algorithms and describes the uniform assessment properties possessed by this group of algorithms. Section four proposes a security measure designed specifically for assessing a protected data element's risk of disclosure with respect to the decision-region based algorithms. Section five presents an outline of an evaluation algorithm, called Orthogonal Boundary (OB), which when executed results in the application of the proposed security measure against a given relation instance. The application of the measure allows for the implementation of the security policies presented in section two. Section six presents the results of experiments that were conducted in order to assess the validity of both the proposed security measure and a subset of the proposed security policies. Section seven presents an outline of future research projects.

# 1. CLASSIFICATION MINING AND SECURITY

The goal of classification mining is to discover patterns that classify objects, or tuples in the context of the relational data model, into predefined classes [1, 4, 3]. This goal is achieved, in part, through the successful completion of three specific tasks. The first task is the selection of an attribute from the given relation. The selected attribute is typically referred to as the decision variable since its purpose is to partition tuples into disjoint sets or classes. The next task is to generalize, if needed, the current values of the selected decision variable to form a set of named classes. Table 1.1 shows a generalized instance of a relation in which the values of the decision variable Mileage have been replaced by the class labels, *low*, *med* and *high*. The final task is to partition the available data into two disjoint sets, a training set and a validation set [7]. The training set is analyzed by a classification mining algorithm to discover patterns that are relevant to the classification of objects into the predefined classes, while the validation set is used to judge the validity of the discovered patterns. Obviously, the generalizability (or predicatability) of the results are only as good as the extent of agreement of patterns or relationships between the training and validation sets as judged by the validation process.

We now illustrate how the disclosure of sensitive information may occur through the execution of a classification mining algorithm. Our example is

*Table 12.1* Relation instance.

| *Id* | *Fuel* | *Cyl* | *Power* | *Prod* | *Tran* | *Mileage* |
|---|---|---|---|---|---|---|
| T1 | efi | 4 | high | n | manu | med |
| T2 | efi | 6 | high | n | manu | med |
| T3 | 2-bbl | 6 | high | n | auto | low |
| T4 | efi | 6 | med | n | manu | med |
| T5 | efi | 4 | high | n | manu | high |
| T6 | 2-bbl | 4 | med | n | manu | high |
| T7 | efi | 6 | high | n | auto | low |
| T8 | efi | 6 | med | n | manu | low |
| T9 | efi | 4 | med | n | auto | med |
| T10 | 2-bbl | 4 | high | n | manu | high |
| T11 | efi | 4 | med | n | manu | med |
| T12 | efi | 4 | high | n | auto | high |
| T13 | 2-bbl | 4 | low | n | manu | high |
| T14 | efi | 6 | high | n | auto | med |
| T15 | 2-bbl | 4 | high | y | auto | high |
| T16 | efi | 6 | med | y | auto | low |
| T17 | 2-bbl | 4 | low | y | auto | med |

based on the data shown in Table 1.1. In particular, suppose that the car company that owns the data has implemented the following security policy: "junior engineers may not access the mileage class of pre-production cars". This policy might be the result of company officials attempting to reduce the chance that someone outside the company will learn the mileage class of a newly designed car. Therefore, company officials have concealed from junior engineers the mileage value associated with the tuples T15, T16 and T17. In the resulting instance, the Mileage attribute is referred to as the *protected attribute* since it contains the protected data elements; and, the attributes Id, Fuel, Cyl, Power, Prod, and Tran are referred to as *non-protected attributes* since they contain no protected data elements. Similarly, we refer to the tuples that contain a protected data element as *protected tuples*. In this case the protected tuples are T15, T16 and T17.

The security risk presented in this example is the extent to which the voluntarily released data facilitates the disclosure of a protected mileage value. The disclosure of that information can be achieved through the process of solving a classification problem. In other words, a junior engineer may be able to correctly infer a protected mileage value through the application of a classification mining algorithm to instances with a known mileage value. For example, ac-

cess to the rule set in Table 1.2 would allow a junior engineer to infer with a relatively high degree of confidence the protected data element of the protected tuple T15 since this tuple satisfies the antecedent of Rule-2 and the predicted accuracy of Rule-2 is higher than that of a simple naïve prediction that always predicts a *med* mileage value (since Pr(Mileage = *med*) = $\frac{6}{14}$). In contrast, the risk of disclosure of the protected data element in tuple T17 is relatively low

*Table 12.2* Production rules.

| |
|---|
| Rule-1: IF (Cyl = 6) $\wedge$ (Tran = auto) THEN |
| (Mileage = low) [31.4%] |
| Rule-2: IF (Fuel = 2-bbl) $\wedge$ (Cyl = 4) THEN |
| (Mileage = high) [63.0%] |
| Rule-3: IF (Cyl = 4) $\wedge$ (Power = high) THEN |
| (Mileage = high) [45.3%] |
| Rule-4: IF (Fuel = efi) THEN |
| (Mileage = med) [44.1%] |

with respect to the given rule set since it is assigned an incorrect class label by Rule-2.

This example motivates the need for security policies to minimize security violation through classification mining.

## 2. INFERENCE BASED SECURITY POLICIES

It is possible to view the security threat presented by a classification mining algorithm in terms of the expected occurrence of an unauthorized inference [2]. The inputs into a classification inference system are a set of tuples having a defined security classification at or below some level $L$ and a protected tuple that contains a protected data element with a defined security classification level at some level $\hat{L}$, where $\hat{L} > L$. The output of the system, referred to as a class-accuracy set, is a set of ordered pairs $(c_i, a_i)$, where $c_i$ is the $i^{th}$ attribute value (class label) in the domain of the protected attribute, and $a_i$ is the predicted accuracy, according to the classification mining algorithm, of assigning to the protected tuple the class label $c_i$. Suppose, for example, that $c_1$, $c_2$, $c_3$, and $c_4$ are the attribute values of a protected attribute. In this case, if a classification inference system produces the class-accuracy set, $\{(c_1, .5), (c_2, .2), (c_3, .8), (c_4, .1)\}$, then the protected tuple is assigned the class label $c_1$, $c_2$, $c_3$ and $c_4$ with predicted accuracy .5, .2, .8 and .1, respectively. The behavior of a set of classification inference systems can be simulated through the specifica-

tion of an accuracy measure based on which the accuracy values, $a_i$'s, can be predicted.

We have identified two general criteria for assessing the security risk associated with the output of a classification inference system. The first criterion is based on a chosen threshold value. This criterion involves security policies that require predicted accuracy values to be below a specified threshold. The other criterion is to assess the output of the system based on a ranking of predicted class-accuracy values. This particular criterion involves security policies that require the ranked position of the protected data element to lie within a specified range. These two criteria are referred to as threshold and rank criteria, respectively.

The threshold and rank criteria have led to the development of four inference-based security policies. Two of the four policies, *maximum threshold* and *maximum range*, are defined independently of the predicted accuracy value of the protected data element. Specifically, an instance of a maximum threshold policy is *satisfied* for some threshold value, $\varepsilon$, and for some class-accuracy set, $\{(c_1, a_1), (c_2, a_2), ..., (c_n, a_n)\}$, if all $a_i$ $(1 \leq i \leq n)$ are less than $\varepsilon$ . In this paper, "$\varepsilon$" is assumed to represent either an organization defined constant, expression or function whose value depends, in part, on the desired level of protection. The other type of security policy, which is also defined independently of a protected data element, called maximum range, is *satisfied* for some threshold value, $\varepsilon$, and for some class-accuracy set, $\{(c_1, a_1), (c_2, a_2), ..., (c_n, a_n)\}$, if [MAX $(a_1, a_2, ..., a_n)$ - MIN $(a_1, a_2, ..., a_n)$] $< \varepsilon$. To illustrate the maximum threshold and maximum range policies consider again the class-accuracy set, $\{(c_1, .5), (c_2, .2), (c_3, .8), (c_4, .1)\}$. In this particular case the maximum threshold and maximum range policies are violated if the specified $\varepsilon$-value is less than or equal to .8 and .7, respectively.

The remaining two identified policies, *protected threshold* and *protected rank*, are both defined in terms of the predicted accuracy value of the protected data element. In particular, the protected threshold policy is *satisfied* for some threshold value, $\varepsilon$, and for some class-accuracy set, $\{(c_1, a_1), (c_2, a_2), ..., (c_n, a_n)\}$, if $a_i < \varepsilon$, where $a_i$ is the predicted accuracy value associated with the protected data element. The protected rank policy is *satisfied* for some class-accuracy set, $\{(c_1, a_1), (c_2, a_2), ..., (c_n, a_n)\}$, if the ranked position of the protected data element is not within the range [L, U], where L and U are positive integers such that $1 \leq L \leq U$ and $L \leq U \leq |\{a_1, a_2, ..., a_n\}|$. We refer to the interval [L, U] as the *non-secure rank range*. We illustrate the protected threshold and protected rank policies using the previously given class-accuracy set, $\{(c_1, .5), (c_2, .2), (c_3, .8), (c_4, .1)\}$. In this case, assuming $c_1$ is the actual value of the protected data element, the protected threshold policy is violated

if the specified $\varepsilon$-value is less than or equal to .5 and the protected rank policy is violated if the specified non-secure rank range is greater than [L= 1, U=1].

# 3. DECISION-REGION BASED CLASSIFICATION ALGORITHMS

Decision-region based classification algorithms share a common set of properties that give rise to a uniform assessment of the predicted accuracy values of a class accuracy set. Before stating these properties, we first characterize the decision-region based class of algorithms. A classification algorithm, A, is a decision-region based algorithm if and only if the following two conditions are satisfied:

- *Condition-1*: It is possible to identify *a priori* a finite set of descriptions, D, in terms of the properties present in an object O such that the particular description d used by A to classify O is an element of D.

- *Condition-2*: The predicted accuracy of assigning an object O satisfying a description $d \in D$ to a class C is dependent on the distribution of class label C relative to all other class labels among the objects that satisfy d in the training set.

The first condition leads to the property that the effective assessment of the security risk for decision-region based classification algorithms requires explicit or implicit determination of the predicted accuracy values of the class-accuracy set associated with each description $d \in D$. The second condition enables us to select a particular method of computing the predicted accuracy values of a class-accuracy set. In general, inference-based security policies may be applied at two levels. If it is known *a priori* that a particular description d will be selected relative to the protected tuple in a protected relation, then we can apply a policy just to that description. This case is referred to as the *description level* security policy. This form of evaluation is possible only if we wish to do the assessment for a particular classification algorithm. An alternative to this approach is referred to as *description space level* security policy. In this case, we must ensure that a chosen security policy is satisfied no matter which d is chosen by a class of classification algorithms.

Given the above definitions and *Condition-1*, we concluded that the specification of inference-based security for decision-region based classification algorithms should be carried out at the description space level. In the following section, we propose a measure for computing the predicted accuracy values of a class-accuracy set with respect to an arbitrary decision-region based classification algorithm.

# 4. COMPUTING PREDICTED ACCURACY VALUES

The proposed measure for computing the predicted accuracy values of a class-accuracy set in the context of a decision-region based algorithm is as follows. Let C be a class label in the domain of a protected attribute. Given a description d $\in$ D, the *predicated accuracy* $a_i$ of assigning the protected tuple T the label C is the ratio of the number of tuples that are assigned label C and satisfy d to the number of tuples that satisfy d. The proposed measure is equivalent to the classification accuracy measure defined in [5].

We now illustrate the application of the measure using the description, (Fuel =*efi*) $\wedge$ (Cyl = *4*), applied against Table 1.1. In this instance there are zero tuples with a *low* gas mileage label that satisfy the description, three tuples with a *med* gas mileage label that satisfy the description, and two tuples with a *high* gas mileage label that satisfy the description. Thus, the predicted accuracy value for *low*, *med* and *high* is 0.0, 0.6 and 0.4, respectively. The Orthogonal Boundary (OB) algorithm, presented in the next section, is designed, in part, to compute the class accuracy values using the proposed measure with respect to an arbitrary description d $\in$ D that might be chosen by a specific subset (see next section) of the decision-region based algorithms.

# 5. ORTHOGONAL-BOUNDARY (OB) ALGORITHM

The Orthogonal-Boundary (OB) algorithm has been designed for use with decision-region based classification algorithms that produce a specific type of class description. In particular, we require that each description, d $\in$ D, represent a logical conjunction of attribute name value pairs that are sufficient, but may or may not be necessary. For example, this type of description is produced by decision tree classifiers.

The set of such descriptions D, corresponding to a protected tuple T, is the set of all logical conjunctions formed from one or more non-protected attribute name value pairs that appear in T. We refer to this set of descriptions as the description space, D*, of the protected tuple T. It follows from *Condition-1* that the assignment of a class label to tuple T, by a decision-region based algorithm that produces a description space equivalent to D*, is necessarily a label that it associates with one of the descriptions d $\in$ D*. Obviously, there is no way to identify *a priori* the description d $\in$ D* chosen by a classifier without making explicit assumptions about the operation of such an algorithm. Unfortunately, the number of descriptions belonging to a protected tuple's description space, D*, is exponential in terms of the number of non-protected attributes. There are, however, several conditions that can be exploited in order to reduce the number of inspected descriptions (e.g. reduce the size of the search space).

One such condition is the recognition of a special set of descriptions that we refer to as *zero* descriptions. The classes constructed from these descriptions contain no tuples with a class label corresponding to the protected data element. The recognition of a zero description implies that there is no need to inspect any description that is a specialization of the zero description since the resulting class will also contain zero instances of the protected data element.

Another condition that can also reduce the number of inspected descriptions is the transformation of a *non-secure description* into a *secure description.* A description is considered secure if its computed class-accuracy set satisfies the chosen security policy. Based on our measure of predicted accuracy and either a protected threshold or protected rank security policy, a transformation of a non-secure description into a secure description requires a percentage reduction in the number of tuples satisfying the description with a class label equal to the protected data element. Obviously, such a reduction occurs when either the number of tuples satisfying the description with a class label equal to the protected data element is decreased, or the number of tuples satisfying the description with a class label that is not equal to the protected data element is increased. A possible transformation scheme, especially when the objective is to maximize the amount of accessible data without altering non-protected data values, is to "protect" additional values of the protected tuple so as to prevent the assignment of the tuple to the class defined by the non-secure description. This particular solution has the added benefit of reducing the required number of inspected descriptions.

Unfortunately, the protection of additional attribute values of a protected tuple T, in general, causes a decision-region based algorithm to violate *Condition-1* in the assignment of a class label to T. Such a case occurs in the application of C4.5's "consult" interpreter which is designed to classify previously unseen tuples based on a constructed decision tree and to output a ranking of the possible class labels that correspond to the tuple [8]. The interpreter is able, through the use of conditional probabilities, to assign a class label to a tuple that contains unknown or concealed attribute values. It is this latter feature that potentially results in a violation of *Condition-1* since the assignment of a class label to a protected tuple T may not be based on a specific description $d \in D^*$. An alternative scheme to transforming a non-secure description into a secure description is to protect a subset of attribute values not belonging to the protected tuple. This solution requires the protection of attribute values such that a decrease occurs in the number of tuples satisfying the non-secure description with a class label equal to that of the protected data element. The advantage of the alternative scheme is that it ensures that the assignment of a class label to a protected tuple satisfies *Condition-1*; however, this scheme does not support maximum access to the data. The current implementation of

the OB algorithm adheres to the first transformation scheme, the protection of additional attribute values of the protected tuple.

A third condition that can reduce the number of inspected descriptions is the establishment of an upper bound on the number of descriptions. By statically or dynamically protecting a subset of a protected tuple's non-protected attribute values, we can reduce the size of the tuple's description space. Of course, the disadvantage of such a strategy is that it does not guarantee maximum access to the data.

A high-level description of the OB algorithm is shown below. A more detailed description can be found in [6].

**OB algorithm**

```
k = 1
while (∃ descriptions to inspect)
  D = k-level descriptions requiring inspection
  for each description d ∈ D
    if (d == zero_description)
      append all specializations of d to zero_description list
    else if (d == non_secure description)
      append d to non_secure description list
  end_for
  transform non_secure descriptions to secure descriptions
  k = k + 1
end_while
```

In the above algorithm, a k-level description represents a description defined in terms of k attributes. The result of executing the OB algorithm is the implicit or explicit inspection of a protected tuple's description space. The inspection process ensures that all descriptions belonging to the tuple's description space satisfy the user's specified description level security policy.

## 6. EXPERIMENTAL INVESTIGATION

In this section, experiments are conducted to validate a proposed approach to establish security policies based on the proposed class-accuracy measure and their results are reported. The objective of the experiments is to test the following hypothesis: there exist a protected threshold policy applied at the description level that produces a protected rank policy at the description space level with a non-secure rank range of [L = 1, U = 1]. In other words, we wish to identify a description level protected threshold policy that, when applied to

the individual descriptions of a protected tuple's description space, results in an appropriate description space level protected rank policy. In general, the determination as to whether a specific description space level policy has been successfully implemented must be based on an evaluation of the output from one or more decision-region based algorithms that have been applied to the protected tuple. In conducting the experiments, we restricted the evaluation of the description space level protected rank policy, [L = 1, U = 1], to the C4.5 decision tree classifier [8]. The application of the classifier is described in the next section.

We anticipate that the implementation of the description space level protected rank policy, [L = 1, U = 1], will provide a high level of protection. This statement is based on the assumption that a user will assign the protected tuple, or more specifically the protected data element, the class label that is assigned the top rank by the chosen decision-region based algorithm. In addition, the non-secure rank range, [L = 1, U = 1], introduces a relatively high degree of uncertainty in a user's assignment of a class label to a protected tuple. This is because, even a user who has knowledge of the fact that the implemented description space level protected rank policy is [L = 1, U = 1] can only logically eliminate from consideration the class label that has been assigned the top rank by the decision-region based algorithm. Hence, a user is at best forced to make a random guess from $n$ - 1 class labels; where $n$ is the number of possible labels.

## 6.1 EXPERIMENTAL PARAMETERS

The execution of the experiments required the construction of several *protected relation instances*. We define a protected relation instance as a relation consisting of at least one non-protected attribute and exactly one protected data element. A relation instance that contains n-protected data elements is viewed as n-instances of a protected relation. The protected relations used in this investigation were constructed through the insertion of a protected tuple into non-protected relation instances constructed through the execution of the Synthetic Classification Data Set (SCDS) program [9]. A total of four non-protected relation instances were constructed through the use of the program and each instance was produced with the parameter values shown in Table 1.3.

The protected tuples, unlike the non-protected relation instances, were manually generated from randomly selected attribute values. Specifically, six protected tuples were constructed with respect to Relation-1, six with respect to Relation-2, four with respect to Relation-3, and five with respect to Relation-4. Each protected tuple was evaluated against a set of protected threshold policies applied at the description level. The implemented policies included those defined at an $\varepsilon$-value of 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1. These

description level policies were applied to each protected tuple, T, through the execution of the OB algorithm. The result in each case was the *transformation* of a protected tuple, T, into a protected tuple $T'$ such that each description, d, belonging to $T'$'s description space D* satisfied the stated policy as defined in section three. In general, the implementation of the individual description level policies resulted in a protected tuple, $T'$, that contained multiple protected attribute values.

In order to assess the generality of the implemented description level policies, two distinct decision tree models were generated for each of the four non-protected relation instances. One set of models corresponded to the *gain attribute selection* criterion, while the other set of models corresponded to the *gain ratio attribute selection* criterion [8]. These two criteria are both supported by C4.5 and provide a decision rule for selecting the interior nodes of a decision tree. Each protected tuple $T'$ was evaluated against instances of both models using C4.5's "consult" interpreter. In conducting the experiments, an attribute value was specified as unknown if the interpreter requested a concealed attribute value.

*Table 12.3* Construction of non-protected relation instances.

| | *Relation-1* | *Relation-2* | *Relation-3* | *Relation-4* |
|---|---|---|---|---|
| Tuples | 5000 | 5000 | 5000 | 5000 |
| Classes | 5 | 5 | 5 | 5 |
| Relevant Attrs. | 15 | 10 | 15 | 15 |
| Irrelevant Attrs. | 3 | 0 | 2 | 0 |
| Masked Relevant Attrs. | 1 | 2 | 0 | 3 |

## 6.2 EXPERIMENTAL RESULTS

The results of the experiments, with respect to nineteen of the twenty-one protected relation instances, are summarized in Tables 1.4 and 1.5. The two tables display the average, highest and lowest rank positions of the protected data elements across all nineteen protected relation instances. Specifically, the tables display statistics about the rank positions produced by the "consult" interpreter when applied to the individual protected tuples, $T'$, that were generated by the application description level protected threshold policies using the OB algorithm.

If the assumption is made that a valid description space level protected rank policy is one defined with respect to a non-secure rank range, [L = 1, U = 1], then a valid protected rank policy is achievable based on the above results

through the application of a description level protected threshold policy defined at a threshold value ($\varepsilon$) of approximately 0.45. We suspect that when a description level protected threshold policy is defined in terms of a relatively high $\varepsilon$-value, (0.9, 0.8, 0.7, or 0.6), the percentage limit on the number of tuples with a class label equal to the protected data element is insufficient to ensure an adequate level of protection at the description space level. On the other hand, description level protected threshold policies defined in terms of a low $\varepsilon$-value (0.3, 0.2, or 0.1) over protect the protected data element. As a result, the assignment of a class label to a protected tuple is based entirely upon the dominant relationships, or patterns, that exist within the data, independent of the accessible attribute values of the protected tuple.

The two protected relation instances not represented in Tables 1.4 and 1.5 are exceptions to the notion of a valid description space level protected rank policy defined in terms of a description level protected threshold policy. The rank position of the two tuples' protected data element as specified by the "consult" interpreter consistently occupied the top position across all implemented description level protected threshold policies. We refer to such protected relation instances as *inherently non-secure*. In the case of such a relation instance the only logical course of action is to entirely eliminate the protected tuple from the user's view. Our preliminary work (not reported in this paper) in this area indicates that such relation instances are avoidable if the transformation of a non-secure description to a secure description is accomplished by protecting additional attribute values not belonging to the protected tuple (no violation of *Condition-1*); or, such relation instances are identifiable through the application of an alternative predicated class-accuracy measure.

*Table 12.4* Rank positions (gain ratio criterion).

| *Threshold($\varepsilon$)* | *Avg. Rank* | *Highest Rank* | *Lowest Rank* |
|---|---|---|---|
| 1.0 | 2.00 | 1 | 5 |
| 0.9 | 2.76 | 1 | 5 |
| 0.8 | 3.05 | 1 | 5 |
| 0.7 | 2.86 | 1 | 5 |
| 0.6 | 3.19 | 1 | 5 |
| 0.5 | 3.33 | 2 | 5 |
| 0.4 | 3.52 | 2 | 5 |
| 0.3 | 3.14 | 1 | 5 |
| 0.2 | 4.10 | 1 | 5 |
| 0.1 | 3.48 | 1 | 5 |

*Table 12.5* Rank positions (gain criterion).

| *Threshold(ε)* | *Avg. Rank* | *Highest Rank* | *Lowest Rank* |
|---|---|---|---|
| 1.0 | 2.67 | 1 | 5 |
| 0.9 | 2.86 | 1 | 5 |
| 0.8 | 3.05 | 1 | 5 |
| 0.7 | 3.19 | 1 | 5 |
| 0.6 | 3.38 | 1 | 5 |
| 0.5 | 3.48 | 2 | 5 |
| 0.4 | 3.52 | 2 | 5 |
| 0.3 | 3.33 | 1 | 5 |
| 0.2 | 4.00 | 2 | 5 |
| 0.1 | 3.67 | 1 | 5 |

## 7. FUTURE WORK

We have several research projects planned with respect to this new and challenging research area. Our immediate plans include, addressing the issue of inherently non-secure relation instances, improvement of the efficiency of the OB algorithm, mapping of continuously valued attributes on to the description space, D*, and development of additional security measures for other groups of classification mining algorithms.

## Acknowledgments

This work is supported in part by a grant from the U.S. Department of Energy (under grant no. DE-FG02-97ER1220).

## References

[1] Berry, M. and Linoff, G. (1997). *Data Mining Techniques: For Marketing, Sales, and Customer Support*, Wiley & Sons.

[2] Clifton, C. and Marks, D. (1996). Security and privacy implications of data mining. *1996 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pp. 15–19.

[3] Deogun, J., Raghavan, V., Sarkar, A. and Sever, H. (1996). Data mining: Trends in research and development. *Rough Sets and Data Mining: Analysis for Imprecise Data*, pp. 9–45.

[4] Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). From data mining to knowledge discovery: An overview. *Advances in Knowledge Discovery and Data Mining* (eds. U. Fayyad, G. Piatetsky-Shapiro, P.Smyth and R. Uthurusamy), pp. 1–34.

[5] Holsheimer, M. and Siebes, A. (1994). *Data Mining: The Search for Knowledge in Databases*, Report CS-R9406, Computer Science, CWI, Amsterdam, The Netherlands.

[6] Johnsten, T. (1998). *Impact of Data Mining on Database Security*, Dissertation. University of Southwestern Louisiana.

[7] Mitchell, T. (1997). *Machine Learning*, McGraw-Hill.

[8] Quinlan, J. (1993). *C4.5: Programs For Machine Learning*, Morgan Kaufmann.

[9] *Synthetic Classification Data Sets*, http://fas.sfu.ca/cs/people/Grad Students/melli/SCDS/intro.html.