A DIAMOND method of inducing classification rules for biological data

Han-Lin Li *, Yao-Huei Huang

Institute of Information Management, National Chiao Tung University, Management Building 2, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan

1. Introduction

Classification, the separation of data into distinct classes, is one of the most common tasks in data mining. Recent studies demonstrate that classification can be applied to analyze the effects of clinical, environmental, and demographic factors on diseases. Classification can also be utilized to respond to treatment, and the risk of side effects [1].

Classifying objects and recognizing patterns in biological datasets, such as identifying species or predicting the survival for a cancer patient, are generally difficult tasks. Most supervised learning and classification methods are inductive, i.e., they extract general patterns from data. There are two restrictions for some well-known classification methods (such as decision tree methods and hyper-plane methods):

(i) A restriction on the linear relationship between the input data and the classes to be predicted [2]. Decision tree methods, hyper-plane methods, and many statistical methods assume that classes can be separated by a linear function. These methods will suffer if the boundaries between the classes are non-linear. This linearity is normally represented by a linear discriminant function calculated from \[ \sum \omega_i x_i \] , where \( x_i \) are the attributes and \( \omega_i \) are the weights of each attribute. In fact, the linearity assumption prohibits the practical applications of these classification methods, since many biological datasets have complicated non-linear interactions between attributes and predicted classes.

(ii) A restriction on finding only the rules with high accuracy [3]. Many classification methods regard the accuracy of the induced rules as the single objective to achieve. As a result, many current methods generate rules which either cover only a narrow part of the object or require numerous attributes to explain a classification. In fact, as Einstein stated: “The best explanation should be kept as simple as possible, but not simpler.” As Altman and Royston [4] suggested the usefulness of a rule is determined by how well a model works in practice, and not by how many as there are in associated \( p \) values.

This study proposes another method of inducing classification rules. The proposed method is applicable to current classification problems in biology and medicine, which typically have the following features:

(i) Our method can treat the classification problems where the relationship between the attributes and the class being predicted can be non-linear. Consider the two attribute classification problem in Fig. 1. Where \( \bigcirc \) represents an object of the first class and \( \times \) represents an object of the second class. Fig. 1(a) clearly shows that there is a linear boundary between the objects of these two classes, while Fig. 1(b) depicts a situation in which there is no clear linear relationship between the objects of two classes. Decision tree methods and hyper-plane methods focus on inducing classification rules for the cases in Fig. 1(a). Our proposed method can treat the cases in Fig. 1(a) and (b).

(ii) Our method can fit the classification problems which are not only to find the rules with high accuracy, but to induce the rules which are more general and simpler. A more general
rule means it can cover more objects. A simpler rule means it can use less number of required attributes to explain a class.

Given a biological dataset with several objects, where each object has some attributes and belongs to a specific class, the rules for classifying these objects are the combinations of attributes that best describe the features of a specific class. Li and Chen [5] described three criteria for evaluating the quality of a rule:

(i) *Accuracy rate*: The rule fitting a class should not cover the objects of other classes.

(ii) *Support rate*: The rule fitting a class should be supported by a large number of objects of the same class.

(iii) *Compact rate*: The rule should include as small number of attributes as possible.

Decision tree methods, support vector hyper-plane methods, and integer programming hyper-plane methods are three well-known classification methods; reviewed as follows:

(i) *Decision tree methods*: Decision tree methods [6–8] are heuristics in nature and are similar to the techniques of statistical inference approaches. These methods recursively split the data into hyper-rectangular regions using a single variable. Backward propagation is preformed to prevent over-fitting of the datasets. Attributes leading to substantial entropy reduction are included as condition attributes to partition the data. The main shortcoming of these methods is a fundamentally greedy approach, which may only find a feasible solution, instead of finding an optimal solution with respect to the maximal rates of accuracy, coverage, and compactness.

(ii) *Support vector hyper-plane methods*: Support vector hyper-plane methods [9–11] separate different classes by various hyper-planes, where the optimal separating hyper-plane is modeled as a convex quadratic programming problem. Since the number of variables must equal the number of training data, the training becomes tedious for a large dataset.

(iii) *Integer program hyper-plane methods*: Bertsimas and Shioda [12] recently used a mixed-integer optimization method [5] to solve the classical statistical problems of classification and regression. Their method separates data points into different regions by using hyper-planes. Each region is assigned a class during the classification. Solving this mixed-integer program, the rules with high rate of accuracy can be induced. However, this approach may generate too many polyhedral regions, which decrease the rate of compact in the induced rules. Using integer programming techniques, Li and Chen [5] developed a multiple criteria method to induce classification rules. Their method clusters data points into polyhedral regions, and yield highly accurate. However, since their approach is based on the concept of the separating hyper-planes, it may also generate many complicated hyper-planes, and especially for dataset containing a large number of attributes.

Some hyper-sphere methods [13–16] have been developed for classifying objects, which use a sphere-structured support vector machine to partition the sample space. This type of approach constructs a minimum bounding sphere for each class, and the smallest sphere encloses the training data as much as possible. However, these methods need to formulate a classification problem as a non-convex program which is hard to reach an optimal solution.

This study proposes a novel method called DIAMOND to improve current classification techniques. For a dataset with objects of various classes, the DIAMOND method clusters these objects into some sets of hyper-cubes. Each object is assigned to a cube by iteratively solving mixed 0–1 programs. This ensures that the most of objects are assigned to a proper set of cubes, where the number of total cubes is minimized.

The following list compares the features of the DIAMOND method with the decision tree methods, hyper-plane methods, and sphere methods mentioned above.

(i) Both hyper-plane methods and decision tree methods need to assume a linear boundary among various classes of objects. The DIAMOND method does not need this assumption.

(ii) Decision tree methods are heuristic approach which can only induce feasible rules. The DIAMOND method is an optimization approach which can find the optimal rules with high rates of accuracy, support and compact. In addition, decision tree methods split the data into hyper-rectangular regions using a single variable, which may generate large number of branches. The DIAMOND method clusters data into cubes based on multiple variables, where the number of cubes can be pre-specified. Thus, the rules induced by the DIAMOND method are more precise than the rules generated by decision tree methods.

(iii) Hyper-plane methods used numerous hyper-planes to separate objects of different classes, and divide the objects in a dataset into indistinct groups. Which may generate a large number of hyper-planes and associated rules with low rates of coverage. The DIAMOND method classifies objects into cubes, then to unify the related cubes as a class. Which is better able to induce rules with high rates of coverage.

(iv) Sphere methods can induce classification rules with better accuracy level than hyper-plane methods. However, these sphere methods need a non-convex form to express a sphere. This prevents the application in classifying large size data. The DIAMOND method is converted into a linear mixed-integer model, which is more convenient to find an optimal solution.

To examine the efficiency of DIAMOND method, this study tests three practical datasets: one of Iris flowers, another of HSV

![Fig. 1. Classifying the relationship between the objects of two classes. (a) Linear relationship. (b) Non-linear relationship.](image-url)
patients and a third of breast cancer patients. The results clearly illustrate the advantages of the DIAMOND method over current decision tree methods and separating hyper-plane methods.

This study is organized as follows. Section 2 uses an example to illustrate the basic idea of the DIAMOND method. Section 3 is the formulation of optimization program for the proposed model. Section 4 reports numerical experiments.

## 2. Basic concepts of the DIAMOND method

This section uses an example to express the basic concepts of the DIAMOND method.

**Example 1.** Considers the dataset \( T \) in Table 1 containing 15 objects \( (x_1, \ldots, x_{15}) \), two attributes \( (a_1, a_2) \), and an index of classes \((c)\). The dataset \( T \) is expressed as \( T = \{x_i|a_1, a_2, c_i\} | i = 1, \ldots, 15\). The domain values of \( c \) are \( \{1, 2, 3\} \). Since there are only two attributes, these 15 objects can be plotted on a plane (see Fig. 2). A hyper-plane method requires 14 hyper-planes to discriminate the objects of Table 1 shown in Fig. 3. This makes it more complicated to combine these 14 hyper-planes to form the regions for the objects of each class.

Alternatively, a sphere method can use “5 spheres” to classify these objects as Fig. 4(a) shows. Consider \( A_1 \) in Fig. 4(a) for instance, sphere \( A_1 \) contains three objects \( x_1, x_2, \) and \( x_3 \). Denote the centroid of \( A_1 \) as \( (b_1^0, b_2^0) \) and radius of \( A_1 \) as \( r_1^0 \) as Fig. 4(b) shows. The situation in which an object \( x_i|a_1, a_2, c_i\) is covered by \( A_1 \) is expressed as

\[
(a_1-b_1^0)^2 + (a_2-b_2^0)^2 \leq r_1^0 \quad \forall i = 1, 2, \ldots, 5. \tag{1}
\]

The situation that an object \( x_i \) is “not” covered by \( A_1 \) is expressed as

\[
(a_1-b_1^0)^2 + (a_2-b_2^0)^2 > r_1^0 \quad \forall i = 1, 2, \ldots, 5. \tag{2}
\]

Sphere methods can classify objects with better accuracy than hyper-plane methods. However, inequality (2) is non-convex, and is difficult to be linearized during the optimization process. Therefore, this study proposes another method, so called the DIAMOND method, to classify these objects.

Instead of using “hyper-planes”, DIAMOND uses “cubes” (shaped like diamonds) to classify these objects, where a rule is expressed by the union of cubes which belong to the same class. The DIAMOND method attempts to use the minimal number of cubes to classify these objects, subject to the constraints that a cube must cover as many objects of a target class as possible. Fig. 2 shows that a good way to classify these 15 objects is to cluster them using five cubes (see Fig. 5(a)), where Cube \( S_{1,1} \) contains \( (x_1, x_2, x_3, x_4) \); Cube \( S_{1,2} \) contains \( (x_4, x_5, x_6) \); Cube \( S_{2,1} \) contains \( (x_7, x_8, x_9) \); Cube \( S_{3,1} \) contains \( (x_{10}, x_{11}, x_{12}, x_{13}, x_{14}) \); and Cube \( S_{3,2} \) contains \( (x_{15}, x_{16}, x_{17}, x_{18}) \). Note that \( x_{15} \) is not covered by any cube, and is regarded as a noisy data. The terms \( S_{I,j}, b_{I,j}, \) and \( r_{I,j} \), respectively, denote the cube, centroid, and radius of the \( I \)th cube for class \( j \).

### Table 1

Dataset of Example 1.

<table>
<thead>
<tr>
<th>Object</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( c )</th>
<th>Symbol</th>
<th>Object</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( c )</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td></td>
<td>( x_2 )</td>
<td>12</td>
<td>20</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( x_3 )</td>
<td>13</td>
<td>8</td>
<td>1</td>
<td></td>
<td>( x_4 )</td>
<td>18</td>
<td>12.5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( x_5 )</td>
<td>21</td>
<td>19</td>
<td>1</td>
<td></td>
<td>( x_6 )</td>
<td>23.5</td>
<td>14.5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( x_7 )</td>
<td>17.5</td>
<td>17.5</td>
<td>2</td>
<td>( \Delta )</td>
<td>( x_8 )</td>
<td>22</td>
<td>17</td>
<td>2</td>
<td>( \Delta )</td>
</tr>
</tbody>
</table>

The radius of a cube is the distance between its centroid point and one of its corner points (Fig. 5(b)). The attribute values of \( b_{I,j} \) are denoted as \( (b^0_{I,j}, b^0_{I,j}) \). The situation that an object \( x_i|a_1, a_2, c_i\) is covered by a cube \( S_{I,j} \) is expressed as

\[
|a_1-b^0_{I,j}| + |a_2-b^0_{I,j}| \leq r_{I,j} \quad \forall i = 1, 2, \ldots, 5. \tag{3}
\]

The situation that an object \( x_i \) is not covered by a cube \( S_{I,j} \) is expressed as

\[
|a_1-b^0_{I,j}| + |a_2-b^0_{I,j}| > r_{I,j} \quad \forall i = 1, 2, \ldots, 5. \tag{4}
\]

Comparing (4) with (2), (4) is much easier to linearize by adding two binary variables, as described in Appendix A.

In this study, each cube should cover at least two objects. Since object \( x_{15} \) is not covered by any cubes, it is regarded as an outlier. A rule for class 1 can then be expressed as follows:

“If an object \( x_i \) is covered by a Cube \( S_{1,1} \) or \( S_{1,2} \) then \( x_i \) belongs to class 1”. This can be rewritten as

\[
R_1: \quad \text{if } x_i \text{ is covered by } S_{1,1} \cup S_{1,2} \text{ then } c_i = 1.
\]

Mathematically, \( R_1 \) can be expressed as

\[
|a_1-b^0_{1,1,1}| + |a_2-b^0_{1,1,2}| \leq r_{1,1} \quad \text{or} \quad |a_1-b^0_{1,2,1}| + |a_2-b^0_{1,2,2}| \leq r_{1,2}
\]

then \( x_i \) is covered by \( S_{1,1} \cup S_{1,2} \).

Fig. 4 shows that the objects \( x_1, \ldots, x_6 \) are covered by \( R_1 \). Similarly, rule 2 (for classifying class 2) and rule 3 (for classifying class 3) can be expressed as below.
R2: if \( x_i \) is covered by \( S_{2,1} \) or \( S_{2,2} \), then \( c_i = 2 \).
R3: if \( x_i \) is covered by \( S_{3,1} \), then \( c_i = 3 \).

Note that cubes \( S_{1,1} \) and \( S_{1,2} \).

According to Li and Chen [1], the rates of accuracy, support, and compactness in \( R_1, R_2 \) and \( R_3 \) can be specified below. These values are used to measure the quality of a rule.

The accuracy rate of a rule \( R_k \) is specified as
\[
AR(R_k) = \frac{\text{number of objects covered correctly by } R_k}{\text{number of objects covered by } R_k}.
\] (5)

For instance, \( AR(R_1) = \frac{6}{6} = 1 \).
An object \( x_i \) is called covered correctly by \( R_k \), if \( c_i = k \).
The support rate of a rule \( R_k \) is specified as
\[
SR(R_k) = \frac{\text{number of objects covered correctly by } R_k}{\text{number of objects of the class } k}.
\] (6)

For instance, \( SR(R_1) = \frac{6}{6} = 1 \), \( SR(R_2) = \frac{5}{5} = 1 \), but \( SR(R_3) = \frac{3}{4} = 0.75 \).
The compact rate for a set of rules is specified as
\[
CR = \frac{\text{number of classes}}{\text{total number of cubes and unions of cubes}},
\] (7)
where a union of cubes means the object is covered by different cubes as shown in Fig. 6.

Take Fig. 6 for instance, where there are three classes, three cubes (i.e., \( S_{2,1}, S_{2,2}, S_{2,3} \)) and one union of cubes (i.e., \( S_{1,1} \cup S_{1,2} \)) generated by rules \( R_1, R_2 \) and \( R_3 \). Therefore, \( CR(R_1, R_2, R_3) = \frac{3}{3} = 1 \).

3. Proposed DIAMOND method and algorithm
3.1. DIAMOND method

Consider a dataset \( T \) with \( n \) objects. Each object has \( m \) attributes \( \{a_1, \ldots, a_m\} \) and belongs to a class, expressed as \( T = (x_i(a_{i1}, \ldots, a_{im}, c_i) | i = 1, \ldots, n) \) where \( c_i \in \{1, \ldots, g\} \). Denote the number of objects at the \( k \)th class as \( \text{num}(k), 1 \leq k \leq g \).
Notation 1. An object $x_i$ in $T$ is specified as $x_i = (a_{i1}, a_{i2}, \ldots, a_{im}; c_i)$, where $a_{ij}$ is the value of the $j$th attribute for the $i$th object, and $c_i$ is the class to which the $i$th object belongs to $c_i \in \{1, \ldots, g\}$.

Notation 2. A rule $R_k$ is used to classify the objects of the $k$th class which is specified by the union of a set of $q_k$ cubes, expressed as $R_k = S_{k,1} \cup S_{k,2} \cup \ldots \cup S_{k,q_k}$.

Notation 3. A $l$th cube in the $k$th class, denoted as $S_{kl}$, is specified by its centroid and radius, expressed as $S_{kl} = (b_{kl,1}, \ldots, b_{kl,m}; r_{kl})$, where $b_{kl,j}$ is the centroid's value at the $j$th dimension, and $r_{kl}$ is its radius.

Remark 1. The total number of cubes is $\sum_{k=1}^g q_k$.

Referring to (3), this yields the following definitions.

Definition 1. An object $x_i = (a_{i1}, \ldots, a_{im}; c_i)$ is covered by a cube $S_{kj} = (b_{kj,1}, \ldots, b_{kj,m}; r_{kj})$ if
\[
\sum_{j=1}^m |a_{ij} - b_{kj,j}| \leq r_{kj}.
\]

Remark 2. An object $x_i$ is not covered by a cube $S_{kj} = (b_{kj,1}, \ldots, b_{kj,m}; r_{kj})$ if and only if
\[
\sum_{j=1}^m |a_{ij} - b_{kj,j}| > r_{kj}.
\]

Definition 4. Consider a cube $S_{kl}$ and two objects $x_i(a_{i1}, \ldots, a_{im}; c_i)$ and $x_j(a_{j1}, \ldots, a_{jm}; c_j)$, where $c_i = k$ and $c_j \neq k$. Denote $u_{kl,i}$ and $v_{kl,j}$ as the two binary variables specified below:

(i) $u_{kl,i} = 1$ if object $x_i$ is covered by $S_{kl}$, and $u_{kl,i} = 0$ otherwise.
(ii) $v_{kl,j} = 1$ if object $x_j$ is covered by $S_{kl}$, and $v_{kl,j} = 0$ otherwise.

That means if an object $x_i$ is covered by a cube $S_{kl}$ of the same class, then $u_{kl,i} = 1$. However, if the object $x_i$ is not covered by a cube $S_{kl}$, which is not the same class (i.e., $c_i \neq k$), then $v_{kl,j} = 1$.

Definition 2. The accuracy rate of a rule $R_k$ denoted as $AR(R_k)$ is specified by referring to (5):
\[
AR(R_k) = \frac{|R_k| - \sum_{k=1}^m q_k + \sum_{i=1}^g u_{kl,i}}{|R_k|},
\]
where $|R_k|$ indicates the number of total objects covered by $R_k$.

Definition 3. The support rate of a rule $R_k$, denoted as $SR(R_k)$, is specified by referring to (6):
\[
SR(R_k) = \frac{\sum_{k=1}^m q_k}{\sum_{i=1}^g u_{kl,i}}.
\]

Definition 4. The compact rate of a set of rules $R_1, \ldots, R_g$, denoted as $CR(R_1, \ldots, R_g)$, is expressed by referring to (7):
\[
CR(R_1, \ldots, R_g) = g \sum_{k=1}^g U_k,
\]
where $U_k$ represents the number of cubes and the unions of cubes for class $k$.

The DIAMOND model generates a set of diamonds (cubes) to induce a rule that maximizes the support rate subject to the constraint that the accuracy rate must exceed a threshold value. This study also design an iterative algorithm to keep the rate of compact as high as possible. The proposed model of classification is formulated below:

Model 1 (Non-linear DIAMOND model)

Maximize $\sum_{l=1}^m \sum_{i=1}^n u_{kl,i}$.

For a cube $S_{kl}$, the following constraints must be satisfied:
\[
\begin{align*}
\sum_{j=1}^m |a_{ij} - b_{kj,j}| &\leq r_{kj} + M(1 - u_{kl,i}), & v_{kl,i}, & c_i = k, \\
\sum_{j=1}^m |a_{ij} - b_{kj,j}| &> r_{kj} - Mv_{kl,i}, & v_{kl,i}, & c_i \neq k,
\end{align*}
\]

\[
AR(R_k) = \frac{|R_k| - \sum_{j=1}^m (a_{ij} - b_{kj,j}) + 2z_{kl,i,j} - 2r_{kj} + Mv_{kl,j}}{|R_k|} \geq \text{Threshold value},
\]

where $M = \max(a_{ij} \forall i \in \{1, \ldots, n\} \land j = \{1, \ldots, m\})$; $b_{kj,j} \geq 0, r_{kj} \geq 0, u_{kl,i}, v_{kl,i} \in \{0, 1\}$; and $a_{ij}$ and $c_i$ are constants.

The objective function (13) is to maximize the support rate. Constraints (14) and (15) come from (8) and (9). Constraint (16) ensures that the accuracy rate should exceed a threshold value. Constraint (14) implies that if a cube $S_{kl}$ covers an object $x_i$ of the same class, then $u_{kl,i} = 1$, and $v_{kl,j} = 0$ otherwise. Constraint (15) implies that if a cube $S_{kl}$ does not cover an object $x_i$ of another class, then $v_{kl,i} = 0$, and $u_{kl,j} = 1$ otherwise.

Inequalities (14) and (15) are non-linear, which need to be linearized. The related techniques in linearizing Model 1 are expressed by three propositions listed in Appendix A.

Model 1 can then be reformulated as the following linear mixed-binary program:

Model 2 (Linearized DIAMOND model)

Maximize (13)

subject to (16),

\[
\begin{align*}
\sum_{j=1}^m (a_{ij} - b_{kj,j}) + 2z_{kl,i,j} &\leq r_{kj} + M(1 - u_{kl,i}), \\
\sum_{j=1}^m (a_{ij} - b_{kj,j}) - 2z_{kl,i,j} + 2z_{kl,i,j} &> r_{kj} - Mv_{kl,j}, \\
2z_{kl,i,j} &\geq 0, \\
2z_{kl,i,j} &\leq 0.
\end{align*}
\]

3.2. A solution algorithm

The solution algorithm is listed below. This algorithm attempts to find the rules where the compact rate is as high as possible.

Step 1. Initialization: $k = 1$ and $l = 1$ specify the threshold value in (16).

Step 2. Solve Model 2 to obtain the $l$th cube of class $k$. Remove the objects covered by $S_{kl}$ from the dataset.

Step 3. Let $l = l + 1$, and resolve Model 2 until all objects in class $k$ are assigned to the cubes of same class.

Step 4. Let $k = k + 1$, and reiterate Step 2 until all classes are assigned.

Step 5. Check the unions of cubes $S_{kl}$; $k = 1$ and $l = 1$.

Step 6. Find the overlapped cubes $S_{kl}$ (i.e., $l = l + 1$) which cover the same objects for all $l$ in class $k$.

Step 7. Let $k = k + 1$ and $l = 1$, and reiterate Step 6 until all cubes containing same objects are merged into one.

According to the above algorithm, we can induce all rules for classifying objects in a dataset. Fig. 7 presents a flowchart of the algorithm.
4. Numerical examples

This section tests three datasets to assess the performance of the proposed method. One is the Iris flower dataset introduced by Sir Ronald Aylmer Fisher (1936) [17], another is the HSV (highly selective vagotomy) patients dataset of F. Raszeja Memorial Hospital in Poland [18,19], and the third is the breast cancer patients dataset of the University of Chicago’s Billings Hospital (1976) [20]. The following subsections compare the proposed model with related methods using IBM ILOG CPLEX (2009) [21]. All tests were run on a PC, equipped with an Intel Pentium (D) 2.8 GHz CPU and 2 GB RAM.

4.1. Iris flower dataset

The Iris flower dataset [17] contains 150 objects. Each object described by four attributes (1: sepal length; 2: sepal width; 3: petal length; 4: petal width) and classified by three classes (1: Setosa; 2: Versicolor; 3: Virginica). By utilizing DIAMOND method, the induced classification rules are reported in Table 2.

The Iris flower dataset contains three rules $R_1$, $R_2$, and $R_3$.

Rule $R_1$ is expressed by a cube $S_1, b_{1,1}$, which means that

- if $|\text{sepal length} - 5.1| + |\text{sepal width} - 3.2| + |\text{petal length} - 1.85| + |\text{petal width} - 0.5| \leq 2.45$ then the Iris belongs to Setosa.

Rule $R_2$ is the union of two cubes $S_2, b_{1,1}$ and $S_2, b_{1,2}$, which implies that

- if $|\text{sepal length} - 6.7| + |\text{sepal width} - 2.6| + |\text{petal length} - 3.5| + |\text{petal width} - 1.2| \leq 2.5$ or $|\text{petal length} - 6.9| + |\text{petal width} - 3.15| + |\text{petal length} - 1.3| \leq 1.55$ then the Iris belongs to Virginica.

Rule $R_3$ is also the union of two cubes $S_3, b_{1,1}$ and $S_3, b_{1,2}$, which shows that

- if $(\text{petal length} < 3)$ then the Iris belongs to Setosa;
- if $(\text{petal length} \geq 3)$ and $(\text{petal width} \geq 1.8)$ then the Iris belongs to Virginica.

<table>
<thead>
<tr>
<th>Rule #</th>
<th>Union of cubes</th>
<th>$b_{1,1}$</th>
<th>$b_{1,2}$</th>
<th>$b_{1,3}$</th>
<th>$b_{1,4}$</th>
<th>$r_{1,1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>$S_{1,1}$</td>
<td>5.1</td>
<td>3.2</td>
<td>1.85</td>
<td>0.5</td>
<td>2.45</td>
</tr>
<tr>
<td>$R_2$</td>
<td>$S_{2,1} \cup S_{2,2}$</td>
<td>6.7</td>
<td>2.6</td>
<td>3.5</td>
<td>1.2</td>
<td>2.5</td>
</tr>
<tr>
<td>$S_{2,2}$</td>
<td>5.9</td>
<td>3.15</td>
<td>4.0</td>
<td>1.3</td>
<td>1.55</td>
<td></td>
</tr>
<tr>
<td>$R_3$</td>
<td>$S_{3,1} \cup S_{3,2}$</td>
<td>6.2</td>
<td>2.9</td>
<td>6.6</td>
<td>2.4</td>
<td>2.7</td>
</tr>
<tr>
<td>$S_{3,2}$</td>
<td>5.3</td>
<td>2.45</td>
<td>4.9</td>
<td>1.6</td>
<td>1.05</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7. Flowchart of the proposed algorithm.

Table 2: Centroid points for the Iris dataset by the DIAMOND method.

Fig. 8. Classifying using the decision tree method. (a) Decision tree for the Iris dataset [17]. (b) Decision tree for the HSV dataset [5].
The HSV dataset contains 122 patients [5,17–19]. The patients are classified into four classes (1: excellent; 2: very good; 3: satisfactory; 4: unsatisfactory), and each patient has 11 pre-operating attributes (1: gender; 2: age; 3: duration of disease; 4: complication of ulcer; 5: HCL concentration; 6: volume of gastric juice per 1 h; 7: volume of residual gastric juice; 8: basic acid output (BAO); 9: HCL concentration; 10: volume of gastric juice per 1 h; 11: maximal acid output).

The details of the rules found by these three methods are listed in Tables 8–10 of Appendix B.

### 4.2. HSV dataset

The HSV dataset contains 122 patients [5,17–19]. The patients are classified into four classes (1: excellent; 2: very good; 3: satisfactory; 4: unsatisfactory), and each patient has 11 pre-operating attributes (1: gender; 2: age; 3: duration of disease; 4: complication of ulcer; 5: HCL concentration; 6: volume of gastric juice per 1 h; 7: volume of residual gastric juice; 8: basic acid output (BAO); 9: HCL concentration; 10: volume of gastric juice per 1 h; 11: maximal acid output).

Table 3 lists these results, which demonstrates that

(i) The accuracy rates for $R_1$, $R_2$, and $R_3$ are expressed as $AR(R_1, R_2, R_3) = (1,1,1)$. The accuracy rate of $R_1$ is 1, which means none of the objects in class 2 or class 3 are covered by $S_{1,1}$. The support rates for $R_1$, $R_2$, and $R_3$ are $SR(R_1, R_2, R_3) = (1,0.98,0.98)$. The compact rate of these three rules is $CR = 1$.

(ii) For the rule of class 1 (i.e., $R_1$), all three methods perform very well in the rates of accuracy and support. However, for the rules of classes 2 and 3 (i.e., $R_2$ and $R_3$), the DIAMOND method has the best performance.

(iii) The DIAMOND method achieves the highest rate of compact. Which means that the DIAMOND method can induce rules more compact than other.

The decision tree method was also applied to induce rules for the same dataset, creating 24 branches shown in Fig. 8(b). Fig. 8(b) is the partial HSV classification tree. For example, from the branches below know that

- if (maximal acid output < $12.2$) and (duration of disease < $0.83$) then the patient belongs to satisfactory;
- if (maximal acid output < $12.2$) and (duration of disease $\geq 0.83$) and (volume of gastric juice per 1 h $\geq 133$) and (HCL concentration < $5$) then the patient belongs to excellent.

The polynomial hyper-plane method [22] was also applied to find rules for HSV dataset, which has 45 hyper-planes. Table 5 also shows that the DIAMOND method can find rules with higher (or equal) rates of AR, SR and CR than the other two methods. These details are reported in Tables 11–13 of Appendix B. The experiments demonstrated that for all classes, the DIAMOND method generated rules with highest rates of accuracy, support, and compactness.

### 4.3. Breast cancer dataset

The breast cancer dataset used in this study contains 294 patients [20]. Surviving patients are classified into two classes (1: the patient survived 5 years or longer; 2: the patient died within 5 year), and each patient has three attributes (1: age of patient at time of operation; 2: patient’s year of operation; 3: number of positive auxiliary nodes detected). For this dataset, the DIAMOND method generates four unions of cubes for classifying 294 patients. The centroids and radiiuses of these cubes are listed in Tables 6 and 7 compare the results. Table 7 further indicates that the DIAMOND method achieves better performance is better than the other
classify biological datasets are discussed as follows: two methods. Detailed results for all three methods are reported in Tables 14–16 of Appendix B.

5. Implications and limitations of the DIAMOND method

The implications and limitations of using DIAMOND method to classify biological datasets are discussed as follows:

(i) The DIAMOND model in this paper is implemented by CPLEX (2009) [21], one of the most powerful mixed-integer programming packages. The program size for a linearized DIAMOND model (i.e., Model 2) is listed below:

- number of binary variables: \( nq \)
- number of continuous variables: \( mnq \)
- number of linear constraints: \( 5mnq \)

where \( n \) is the number of objects, \( m \) is the number of attributes, and \( q \) is the number of classes. A PC version CPLEX can typically solve a program containing around 1000 binary variables, 10,000 continuous variables and 100,000 linear constraints. Thus, using a PC version CPLEX, the DIAMOND method is capable of solving classification programs including 250 objects (\( m = 8 \)) and four classes (\( q = 4 \)), or solving the programs with \( n = 450, m = 10 \) and \( q = 10 \).

(ii) The computing time for solving a mixed-integer program grows rapidly as the number of binary variables increases. Therefore, the computing time of the DIAMOND method is slower than decision tree methods, especially for large size datasets. For instance, for running the breast cancer dataset [20] (294 patients, 450, \( m = 250 \), eight attributes (\( m = 8 \)) and four classes (\( q = 4 \)), or solving the programs with \( n = 450, m = 10 \) and \( q = 10 \).

(iii) Existing genomic fingerprinting techniques, such as single nucleotide polymorphisms (SNPs) and gene expression micro-arrays, often yield records with thousands of entries that are usually interpreted as binary. Therefore we need to use a...
mainframe version CPLEX to solve a large size classification problem. Some current bioinformatics or biological problems are formulated as a mixed-integer linear programs (MILP) and solved by CPLEX software carrying out on mainframe versions. Klau et al. [25] formed a linear program for solving minimal set of probe selection on a microarray for each biological sample; Li and Fu [26] and Deng et al. [27] proposed a MILP for solving DNA microarray. Their methods were to minimize the number of non-unique probes and can identify the algorithm complexity (i.e., O(n)) and error tolerance, and some of the experiments were carried out on Sun Fire 280 R with Solaris 8. Than et al. [28] and Rockville [29] used MILP to solve genome-scale multi-locus datasets and large scale biological datasets on mainframe computers (such as Linux). By referring to their reports on computation, we can estimate the problem size solvable by a DIAMOND model on mainframe system as

- number of binary variables: \( nq = 20,000 \)
- number of continuous variables: \( mnq = 100,000 \)
- number of linear constraints: \( 5mnq = 500,000 \)

which implies the DIAMOND method, operated under a mainframe system, can solve classification problems over 2000 object, 10 attributes and 10 classes.

(iv) The DIAMOND method uses mixed-integer techniques to find separated cubes of various classes, which is an optimization process of achieving an optimal solution. However, in connecting the cubes of the same class, the DIAMOND method uses a heuristic process which may only reach a feasible solution. How to use an optimal process to connect the cubes of the same class is an interesting issue for further study.

6. Conclusion

This study presents a method, called DIAMOND, to classify objects with various classes. In solving a mixed 0–1 linear program, DIAMOND generates a set of cubes to cluster objects of the same class. This approach achieves an accuracy rate (AR) higher than a threshold value, and maximizes the associated support rate (SR). The DIAMOND method also keeps the compact rate (CR) for all rules as high as possible via an iterative solution algorithm. Three commonly used datasets (iris, hiv, and the breast cancer) were tested to illustrate that, comparing with a decision tree method and a hyper-plane support vector method, the DIAMOND method can induce rules with higher AR, SR, and CR values. Owing to the capacity restriction of current mixed-integer programs, the DIAMOND method cannot solve a classification problem containing thousands of objects in reasonable time. More efforts are needed to accelerate the computation speed of the DIAMOND method.

Acknowledgement

This study has been supported partially by NSC 98-2221-E-009-050-MY3 of National Science Council of Taiwan, R.O.C.

Appendix A

Proposition 1. Inequality (14) is linearized as follows referring to Li [30]:

\[
\sum_{j=1}^{m} \left( a_{i,j} - b_{k,l,j} + 2e_{k,l,i,j} \right) \leq r_{k,l} + M(1 - u_{k,l,i,j}), \quad (17)
\]

where \( e_{k,l,i,j} \geq 0 \).

Proof.

(i) If \( a_{i,j} - b_{k,l,j} \geq 0 \) then \( e_{k,l,i,j} = 0 \). Which results in

\[
a_{i,j} - b_{k,l,j} \geq a_{i,j} - b_{k,l,j}. \quad (19)
\]

(ii) If \( b_{k,l,j} - a_{i,j} \geq 0 \) then \( e_{k,l,i,j} \geq b_{k,l,j} - a_{i,j} \geq 0 \). Which results in

\[
a_{i,j} - b_{k,l,j} + 2e_{k,l,i,j} \geq b_{k,l,j} - a_{i,j} = |a_{i,j} - b_{k,l,j}|. \quad (20)
\]

\[
\sum_{j=1}^{m} (1 - 2\lambda_{k,l,i,j})(a_{i,j} - b_{k,l,j}) \leq \sum_{j=1}^{m} (a_{i,j} - b_{k,l,j} - 2a_{i,j}\lambda_{k,l,i,j} + 2z_{k,l,i,j}) > r_{k,l} - M\tau_{k,l,i}. \quad (21)
\]

where

\[
a_{i,j} - b_{k,l,j} - 2a_{i,j}\lambda_{k,l,i,j} + 2z_{k,l,i,j} \geq 0, \quad (22)
\]

\[
\bar{b}_{j}(\lambda_{k,l,i,j}-1)+b_{k,l,j} \leq z_{k,l,i,j} \leq b_{k,l,j} +\bar{b}_{j}(1-\lambda_{k,l,i,j}). \quad (23)
\]

\[
0 \leq z_{k,l,i,j} \leq \bar{b}_{j}\lambda_{k,l,i,j}. \quad (24)
\]

\( \bar{b}_{j} \) is constant, \( \bar{b}_{j} = \max(a_{i,j}, \forall j \neq l) \) and \( \lambda_{k,l,i,j} \in (0,1) \).

Appendix B

Tables 8–16.

Table 8

<table>
<thead>
<tr>
<th>Rule</th>
<th>Unions of cubes</th>
<th>Covered objects (#)</th>
<th>Correctly</th>
<th>Incorrectly</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_1</td>
<td>S_1,3</td>
<td>1–50</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R_2</td>
<td>S_2,1 ∪ S_3,1</td>
<td>51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100</td>
<td>None</td>
<td>None</td>
<td>0.98</td>
<td></td>
</tr>
</tbody>
</table>

\( CR(R_1, R_2, R_3) = \frac{1}{2} = 1 \).
Table 9
Decision tree method for the Iris dataset.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Decision branch</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>If $(a_3 &lt; 3)$ then objects belong to class 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$R_2$</td>
<td>If $(a_3 \geq 3) \cap (a_4 &lt; 1.8) \cap (a_3 &lt; 5) \cap (a_4 = 1.7)$ or if $(a_3 \geq 3) \cap (a_4 &lt; 1.8) \cap (a_3 \geq 5) \cap (a_4 \geq 1.6)$ then objects belong to class 2</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>$R_3$</td>
<td>If $(a_3 \geq 3) \cap (a_4 &lt; 1.8) \cap (a_3 &lt; 5) \cap (a_4 = 1.7)$ or if $(a_3 \geq 3) \cap (a_4 &lt; 1.8) \cap (a_3 \geq 5) \cap (a_4 \geq 1.6)$ or if $(a_3 \geq 3) \cap (a_4 &lt; 1.8)$ then objects belong to class 3</td>
<td>0.98</td>
<td>0.98</td>
</tr>
</tbody>
</table>

$CR(R_1,R_2,R_3) = \frac{4}{10} = 0.4$.

Table 10
Hyper-plane method for the Iris dataset.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Support vectors (polynomial function)</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>$(y_0, y_1) = (0.008, 0.0004)$</td>
<td>(5.1, 3.3, 1.7, 0.5)</td>
<td>1</td>
</tr>
<tr>
<td>$R_2$</td>
<td>$(y_0, y_1) = (0.0006)$</td>
<td>(4.8, 3.4, 1.9, 0.2)</td>
<td>0.98</td>
</tr>
<tr>
<td>$R_3$</td>
<td>$(y_0, y_1) = (0.0005/0)$</td>
<td>(4.5, 2.3, 1.3, 0.3)</td>
<td>0.96</td>
</tr>
<tr>
<td>$R_4$</td>
<td>$(y_0, y_1) = (0.00006)$</td>
<td>(5.1, 3.8, 1.9, 0.4)</td>
<td>0.96</td>
</tr>
</tbody>
</table>

$CR(R_1,R_2,R_3) = \frac{4}{10} = 0.4$.

Table 11
Classification results for the HSV data by the DIAMOND method.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Unions of cubes</th>
<th>Covered objects (#)</th>
<th>Correctly</th>
<th>Incorrect (miss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>$S_{1,3} \cup S_{2,3} \cup \ldots S_{1,11}$</td>
<td>$(30,33, 1, 0.06$</td>
<td>$1$</td>
<td>0.96</td>
</tr>
<tr>
<td>$R_2$</td>
<td>$S_{2,3} \cup S_{2,6} \cup \ldots S_{2,6}$</td>
<td>$(44,73,96) 1$</td>
<td>$1$</td>
<td>0.83</td>
</tr>
<tr>
<td>$R_3$</td>
<td>$S_{1,3} \cup S_{3,3}$</td>
<td>$103,110/120,121$</td>
<td>$24$</td>
<td>$1$</td>
</tr>
<tr>
<td>$R_4$</td>
<td>$S_{4,1}$</td>
<td>$(118,95,105,107) 0.91$</td>
<td>$1$</td>
<td>0.79</td>
</tr>
</tbody>
</table>

$CR(R_1,R_2,R_3,R_4) = \frac{4}{10} = 0.29$.

Table 12
Decision tree method for the HSV dataset.

<table>
<thead>
<tr>
<th>Rules</th>
<th>Decision branch</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>If $(a_11 &lt; 12.2) \cap (a_3 \geq 0.83) \cap (a_6 &lt; 133) \cap (a_9 &lt; 5.7) \cap (a_7 \geq 88)$ or</td>
<td>0.93</td>
<td>0.93</td>
</tr>
</tbody>
</table>
Table 13 (continued)

<table>
<thead>
<tr>
<th>Rules</th>
<th>Decision branch</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>If (a11 ≥ 12.2) (\tau(a6 &lt; 166) \land \tau(a9 &lt; 14.2) \land \tau(a2 &lt; 37) \land \tau(a9 &lt; 11.7) \land \tau(a3 &lt; 11)) or</td>
<td>(0.81)</td>
<td>(0.72)</td>
<td></td>
</tr>
<tr>
<td>If (a11 ≥ 12.2) (\tau(a6 &lt; 166) \land \tau(a9 &lt; 14.2) \land \tau(a2 &lt; 37) \land \tau(a9 &lt; 11.7) \land \tau(a7 &lt; 27)) or</td>
<td>(0.7)</td>
<td>(0.78)</td>
<td></td>
</tr>
<tr>
<td>If (a11 ≥ 12.2) (\tau(a6 &lt; 166) \land \tau(a9 &lt; 14.2) \land \tau(a2 &lt; 37) \land \tau(a3 &lt; 11)) or</td>
<td>(0.71)</td>
<td>(0.71)</td>
<td></td>
</tr>
<tr>
<td>If (a11 ≥ 12.2) (\tau(a6 &lt; 166) \land \tau(a11 &lt; 39.1) \land \tau(a9 &lt; 249) \land \tau(a9 &lt; 8.7) \land \tau(a2 &lt; 26)) or</td>
<td>(0.9)</td>
<td>(0.69)</td>
<td></td>
</tr>
<tr>
<td>If (a11 ≥ 12.2) (\tau(a6 &lt; 166) \land \tau(a11 &lt; 39.1) \land \tau(a3 &lt; 0.83)) then objects belong to class 1</td>
<td>(0.92)</td>
<td>(0.92)</td>
<td></td>
</tr>
</tbody>
</table>

\(\text{CR}(R_1,R_2,R_3,R_4) = \frac{N}{D} = 0.17\).

Table 13
Hyper-plane method for the HSV dataset.

<table>
<thead>
<tr>
<th>Rule</th>
<th>#</th>
<th>Support vectors (polynomial function)</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(y_j)</td>
<td>((a_1,a_2,\ldots,a_{13}))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(R_1)</td>
<td>1</td>
<td>((0.229,0.114,0.164))</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>((0.229,0.114,0.164))</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>(R_2)</td>
<td>20</td>
<td>((-1.05,0.713))</td>
<td>1</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>((-1.05,0.713))</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>(R_3)</td>
<td>34</td>
<td>((-1,-1.1))</td>
<td>0.9</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>((-1,-1.1))</td>
<td>0.9</td>
<td>0.69</td>
</tr>
<tr>
<td>(R_4)</td>
<td>41</td>
<td>((-1,-1,0.7))</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>((-1,-1,0.7))</td>
<td>1</td>
<td>0.67</td>
</tr>
</tbody>
</table>

\(\text{CR}(R_1,R_2,R_3,R_4) = \frac{N}{D} = 0.09\).

Table 14
Classification results for the breast cancer dataset by the DIAMOND method.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Cube #</th>
<th>Covered objects (#)</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R_1)</td>
<td>(S_{1,1} \cup S_{1,4})</td>
<td>1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 47, 48, 49, 51, 52, 53, 56, 57, 58, 59, 60, 61, 62, 67, 68, 69, 71, 72, 73, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 94, 95, 96, 100, 101, 102, 103, 104, 105, 107, 111, 112, 114, 117, 118, 119, 120, 121, 122, 123, 124, 127, 128, 129, 130, 131, 132, 133, 135, 136, 139, 140, 141, 142, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 163, 164, 165, 166, 167, 172, 173, 174, 176, 179, 180, 183, 184, 185, 187, 193, 197, 203, 204, 205, 206, 207, 209, 210, 211, 213, 214, 215, 217, 218, 219, 221, 226, 235, 237, 243, 244, 247, 248, 249, 250, 251, 256, 257, 258, 264, 265, 267, 268, 273, 276, 277, 279, 280, 281, 284, 285, 289, 290, 292, 293, 295, 296, 297, 299, 301, 302, 303, 304, 188, 190, 194, 195, 196, 202, 208, 212, 223, 227, 201, 220, 222, 236, 242, 252, 254, 266</td>
<td>(0.92)</td>
<td>(0.92)</td>
</tr>
</tbody>
</table>
Table 14 (continued)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Cube #</th>
<th>Covered objects (#)</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S1,23</td>
<td>189, 228, 253, 255</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S1,25</td>
<td>272, 278, 283</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S1,27</td>
<td>178, 186</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>S2,1 ∪ S2,9</td>
<td>44, 63, 76, 93, 97, 108, 109, 137, 161, 169, 216</td>
<td>(8, 9, 25, 35, 45, 55, 83, 90, 98, 115, 126, 143, 145, 232, 240, 259, 269, 294)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 15

Decision tree method for the breast cancer dataset.

<table>
<thead>
<tr>
<th>Rules</th>
<th>Decision branch</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>If (a1 &lt; 9) (\cap) (a1 &lt; 78) (\cap) (a1 &lt; 48) (\cap) (a2 &lt; 48) (\cap) (a2 &lt; 43) or</td>
<td>0.92</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>If (a1 &lt; 9) (\cap) (a1 &lt; 78) (\cap) (a1 &lt; 3) (\cap) (a1 &lt; 48) (\cap) (a2 &lt; 48) (\cap) (a1 ≥ 43) (\cap) (a2 ≥ 60) or</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td></td>
<td>If (a1 ≥ 61) (\cap) (a2 ≥ 25) then objects belong to class 1</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>R2</td>
<td>If (a1 &lt; 9) (\cap) (a1 &lt; 78) (\cap) (a1 &lt; 3) (\cap) (a1 &lt; 48) (\cap) (a2 &lt; 48) (\cap) (a1 ≥ 43) (\cap) (a2 &lt; 60) or</td>
<td>0.92</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>If (a1 ≥ 9) (\cap) (a2 ≥ 61) (\cap) (a1 &lt; 25) (\cap) (a1 ≥ 65) then objects belong to class 2</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

CR(R1, R2) = \(\frac{1}{10}\) = 0.01.

Table 16

Hyper-plane method for the breast cancer dataset.

<table>
<thead>
<tr>
<th>Rule</th>
<th>#</th>
<th>Support vectors (linear function)</th>
<th>AR</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(y)</td>
<td>((a_1, a_2, a_3))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>((-1))</td>
<td>(31.65,4)</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>(\vdots)</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>((-1))</td>
<td>(76.67,0)</td>
<td>0.6</td>
</tr>
<tr>
<td>R2</td>
<td>1</td>
<td>((-1))</td>
<td>(38.69,21)</td>
<td>:</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>((-1))</td>
<td>(67.64,8)</td>
<td>:</td>
</tr>
</tbody>
</table>

CR(R1, R2) = \(\frac{1}{10}\) = 0.01.

References

Han-Lin Li obtained his PhD degree from University of Pennsylvania 1983. He has been Chair Professor at National Chain Tung University, Taiwan, since 2006. His research areas include global optimization, decision support system, and bioinformatics. He is also the outstanding researcher of National Science Council of Taiwan Government.

Yao-Huei Huang is a PhD student now. His research fields include global optimization and decision support system.