



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

ARTICLE INFO

Article history:

Received 17 August 2010

Accepted 2 May 2011

Keywords:

DIAMOND

Cubes

Classification rules

Integer program

ABSTRACT

Identifying the classification rules for patients, based on a given dataset, is an important role in medical tasks. For example, the rules for estimating the likelihood of survival for patients undergoing breast cancer surgery are critical in treatment planning. Many well-known classification methods (as decision tree methods and hyper-plane methods) assume that classes can be separated by a linear function. However, these methods suffer when the boundaries between the classes are non-linear. This study presents a novel method, called DIAMOND, to induce classification rules from datasets containing non-linear interactions between the input data and the classes to be predicted. Given a set of objects with some classes, DIAMOND separates the objects into different cubes, and assigns each cube to a class. Via the unions of these cubes, DIAMOND uses mixed-integer programs to induce classification rules with better rates of accuracy, support and compact. This study uses three practical datasets (Iris flower, HSV patients, and breast cancer patients) to illustrate the advantages of DIAMOND over some current methods.

© 2011 Published by Elsevier Ltd.

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 response 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 of 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 $\sum w_i x_i$, where x_i are the attributes and w_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 \circ 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

* Corresponding author.

E-mail addresses: hll0410@gmail.com, yaohuei.huang@gmail.com (H.-L. Li).

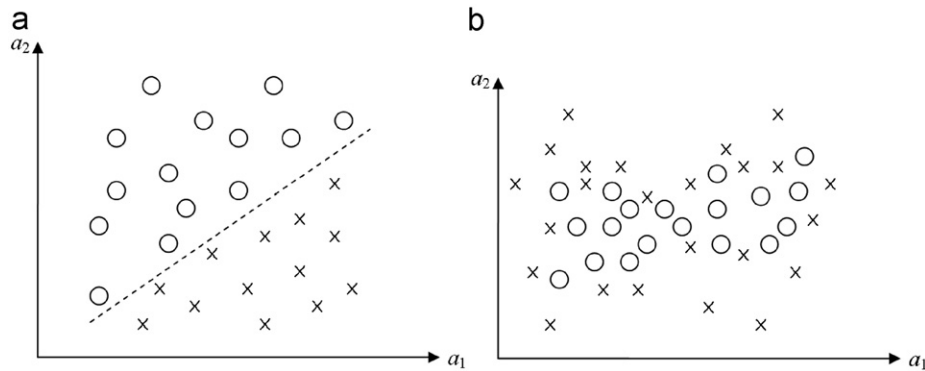


Fig. 1. Classifying the relationship between the objects of two classes. (a) Linear relationship. (b) Non-linear relationship.

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 performed 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

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, \dots, x_{15}), two attributes (a_1, a_2), and an index of classes (c). The dataset T is expressed as $T = \{x_i(a_{i,1}, a_{i,2}; c_i) | i = 1, \dots, 15\}$. 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 in 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 $x_i(a_{i,1}, a_{i,2}; c_i)$ is covered by A_1 is expressed as

$$(a_{i,1} - b_1^0)^2 + (a_{i,2} - b_2^0)^2 \leq r_1^0 \quad \forall i = 1, 2, \dots, 5. \tag{1}$$

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

$$(a_{i,1} - b_1^0)^2 + (a_{i,2} - b_2^0)^2 > r_1^0 \quad \forall i = 1, 2, \dots, 5. \tag{2}$$

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, subjected 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_{12}, x_{13}, x_{14}) ; and Cube $S_{2,2}$ contains (x_{10}, x_{11}) . Note that x_{15} is not covered by any cube, and is regarded as a noisy data. The terms $S_{k,l}$, $p_{k,l}$, and $r_{k,l}$, respectively, denote the cube, centroid, and radius of the l 'th cube for class k .

Table 1
Dataset of Example 1.

Object	a_1	a_2	c	Symbol	Object	a_1	a_2	c	Symbol
x_1	6	8	1	o	x_9	22	15	2	Δ
x_2	12	20	1	o	x_{10}	30	11	2	Δ
x_3	13	8	1	o	x_{11}	33.5	7.5	2	Δ
x_4	18	12.5	1	o	x_{12}	24.5	3.5	3	\times
x_5	21	19	1	o	x_{13}	26.5	8	3	\times
x_6	23.5	14.5	1	o	x_{14}	23.5	7.5	3	\times
x_7	17.5	17.5	2	Δ	x_{15}	6	30	3	\times
x_8	22	17	2	Δ					

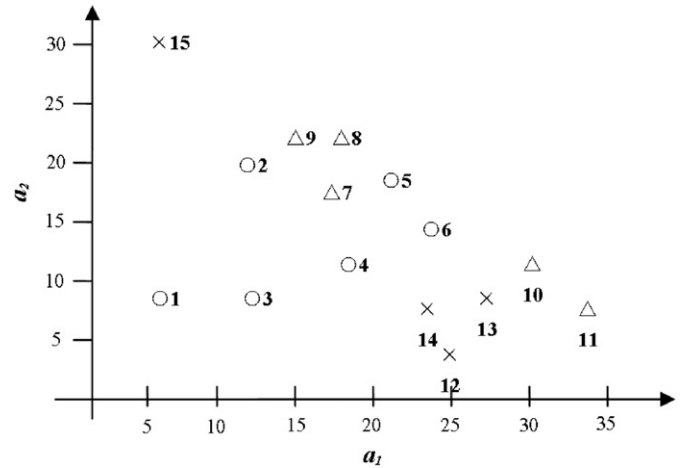


Fig. 2. Plot of objects.

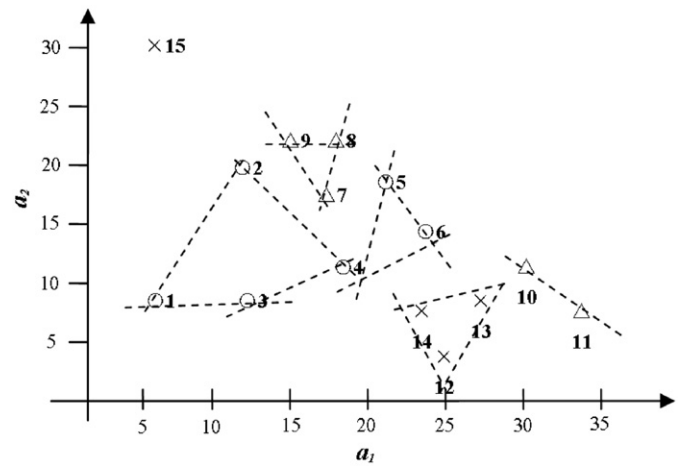


Fig. 3. Classify by separating hyper-planes.

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 $p_{k,l}$ are denoted as $(b_{k,l,1}, b_{k,l,2})$. The situation that an object $x_i(a_{i,1}, a_{i,2}; c_i)$ is covered by a cube $S_{k,l}$ is expressed as

$$|a_{i,1} - b_{k,l,1}| + |a_{i,2} - b_{k,l,2}| \leq r_{k,l} \quad \forall i = 1, 2, \dots, 5. \tag{3}$$

The situation that an object x_i is not covered by a cube $S_{k,l}$ is expressed as

$$|a_{i,1} - b_{k,l,1}| + |a_{i,2} - b_{k,l,2}| > r_{k,l} \quad \forall i = 1, 2, \dots, 5. \tag{4}$$

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 : if x_i is covered by $S_{1,1} \cup S_{1,2}$ then $c_i = 1$.

Mathematically, R_1 can be expressed as

R_1 : if $|a_{i,1} - b_{1,1,1}| + |a_{i,2} - b_{1,1,2}| \leq r_{1,1}$ or $|a_{i,1} - b_{1,2,1}| + |a_{i,2} - b_{1,2,2}| \leq r_{1,2}$ then x_i is covered by $S_{1,1} \cup S_{1,2}$.

Fig. 4 shows that the objects x_1, \dots, 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.

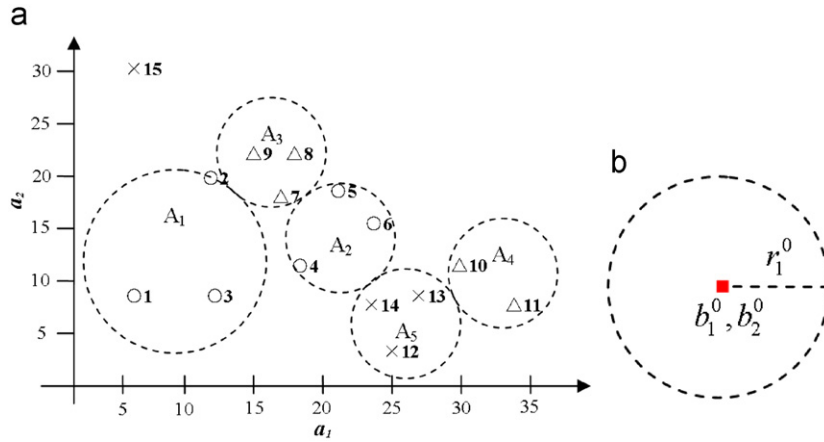


Fig. 4. Classify by the spheres. (a) Classify by the sphere method. (b) The radius of sphere $S_{k,l}$.

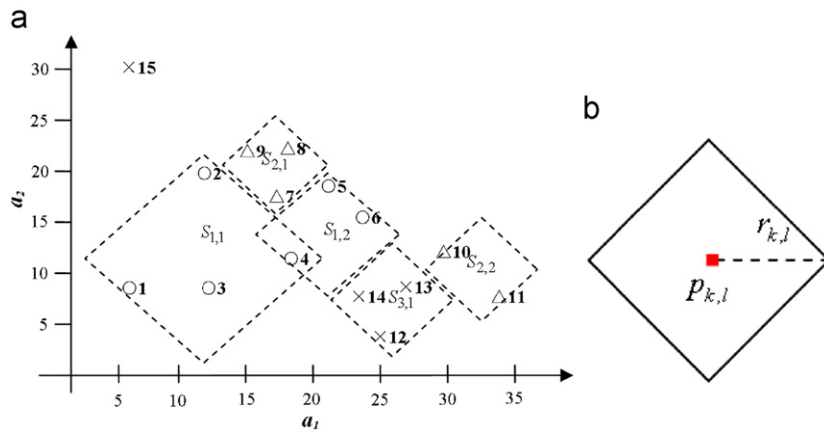


Fig. 5. Classify by proposed method. (a) Classify by cubes. (b) The radius of cube $S_{k,l}$.

R_2 : if x_i is covered by $S_{2,1} \cup S_{2,2}$, then $c_i = 2$.
 R_3 : 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

$$AR(R_k) = \frac{\text{number of objects covered correctly by } R_k}{\text{number of objects covered by } R_k} \quad (5)$$

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

$$SR(R_k) = \frac{\text{number of objects covered correctly by } R_k}{\text{number of objects of the class } k} \quad (6)$$

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

$$CR = \frac{\text{number of classes}}{\text{total number of cubes and unions of cubes}} \quad (7)$$

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., $S_{2,1}$, $S_{2,2}$, $S_{3,1}$) 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, $CR(R_1, R_2, R_3) = \frac{3}{4}$.

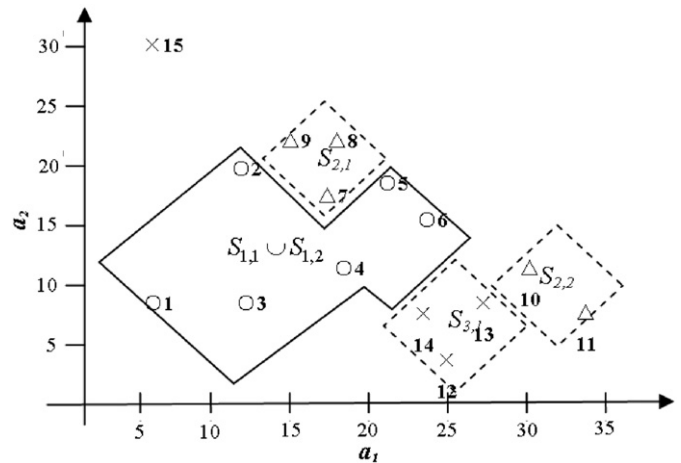


Fig. 6. Cubes and union of cubes.

3. Proposed DIAMOND method and algorithm

3.1. DIAMOND method

Consider a dataset T with n objects. Each object has m attributes $\{a_1, \dots, a_m\}$ and belongs to a class, expressed as $T = \{x_i(a_{i,1}, \dots, a_{i,m}; c_i) | i = 1, \dots, n\}$ where $c_i \in \{1, \dots, g\}$. Denote the number of objects at the k 'th class as $num(k)$, $1 \leq k \leq g$.

Notation 1. An object x_i in T is specified as $x_i = (a_{i,1}, a_{i,2}, \dots, a_{i,m}; c_i)$, where $a_{i,j}$ 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 $c_i \in \{1, \dots, g\}$.

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 $R_k = S_{k,1} \cup S_{k,2} \cup \dots \cup S_{k,q_k}$.

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

$$\sum_{j=1}^m |a_{i,j} - b_{k,l,j}| \leq r_{k,l}. \tag{8}$$

Remark 2. An object x_i is not covered by a cube $S_{k,l} = (b_{k,l,1}, \dots, b_{k,l,m}; r_{k,l})$ if and only if

$$\sum_{j=1}^m |a_{i,j} - b_{k,l,j}| > r_{k,l}. \tag{9}$$

Notation 4. Consider a cube $S_{k,l}$ and two objects $x_i(a_{i,1}, \dots, a_{i,m}; c_i)$ and $x_{i'}(a_{i',1}, \dots, a_{i',m}; c_{i'})$, where $c_i = k$ and $c_{i'} \neq k$. Denote $u_{k,l,i}$ and $v_{k,l,i'}$ as the two binary variables specified below:

- (i) $u_{k,l,i} = 1$ if object x_i is covered by $S_{k,l}$, and $u_{k,l,i} = 0$ otherwise.
- (ii) $v_{k,l,i'} = 1$ if object $x_{i'}$ is covered by $S_{k,l}$, and $v_{k,l,i'} = 0$ otherwise.

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

Definition 2. The accuracy rate of a rule R_k denoted as $AR(R_k)$ is specified by referring to (5):

$$AR(R_k) = \frac{\|R_k\| - \sum_{i'=1}^{num(k')} \sum_{l=1}^{q_k} v_{k,l,i'}}{\|R_k\|}, \tag{10}$$

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):

$$SR(R_k) = \frac{\sum_{i=1}^{num(k)} \sum_{l=1}^{q_k} u_{k,l,i}}{num(k)}. \tag{11}$$

Definition 4. The compact rate of a set of rules R_1, \dots, R_g , denoted as $CR(R_1, \dots, R_g)$, is expressed by referring to (7):

$$CR(R_1, \dots, R_g) = g / \sum_{k=1}^g U_k, \tag{12}$$

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)

$$\text{Maximize } \sum_{l=1}^{q_k} \sum_{i=1}^{n(k)} u_{k,l,i}. \tag{13}$$

For a cube $S_{k,l}$, the following constraints must be satisfied:

$$\sum_{j=1}^m |a_{i,j} - b_{k,l,j}| \leq r_{k,l} + M(1 - u_{k,l,i}), \quad \forall x_i, \text{ where } c_i = k, \tag{14}$$

$$\sum_{j=1}^m |a_{i',j} - b_{k,l,j}| > r_{k,l} - Mv_{k,l,i'} \quad \forall x_{i'}, \text{ where } c_{i'} \neq k, \tag{15}$$

$$AR(R_k) = \frac{\|R_k\| - \sum_{i'=1}^{num(k')} \sum_{l=1}^{q_k} v_{k,l,i'}}{\|R_k\|} \geq \text{Threshold value}, \tag{16}$$

where $M = \max\{a_{i,j} \mid \forall i = 1, \dots, n \text{ and } j = 1, \dots, m\}$; $b_{k,l,j} \geq 0, r_{k,l} \geq 0, u_{k,l,i}, v_{k,l,i'} \in \{0, 1\}$; and $a_{i,j}$ and $a_{i',j}$ 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 $S_{k,l}$ covers an object x_i of the same class, then $u_{k,l,i} = 1$, and $u_{k,l,i} = 0$ otherwise. Constraint (15) implies that if a cube $S_{k,l}$ does not cover an object $x_{i'}$ of another class, then $v_{k,l,i'} = 0$, and $v_{k,l,i'} = 1$ 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),

$$\begin{aligned} & \sum_{j=1}^m a_{i,j} - b_{k,l,j} + 2e_{k,l,i,j} \leq r_{k,l} + M(1 - u_{k,l,i}) \\ & a_{i,j} - b_{k,l,j} + e_{i,k,l,j} \geq 0 \\ & \sum_{j=1}^m (a_{i',j} - b_{k,l,j} - 2a_{i',j}\lambda_{k,l,i',j} + 2z_{k,l,i',j}) > r_{k,l} - Mv_{k,l,i'} \\ & a_{i',j} - b_{k,l,j} - 2a_{i',j}\lambda_{k,l,i',j} + 2z_{k,l,i',j} \geq 0 \\ & \bar{b}_j(\lambda_{k,l,i',j} - 1) + b_{k,l,j} \leq z_{k,l,i',j} \leq b_{k,l,j} + \bar{b}_j(1 - \lambda_{k,l,i',j}) \\ & 0 \leq z_{k,l,i',j} \leq \bar{b}_j\lambda_{k,l,i',j} \\ & \lambda_{k,l,i,j} \leq \lambda_{k,l,i',j} \quad \forall i \text{ and } i', \text{ where } a_{i,j} > a_{i',j}. \end{aligned}$$

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 $S_{k,l}$ 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 $S_{k,l}$; $k=1$ and $l=1$.
- Step 6.** Find the overlapped cubes $S_{k,l}$ (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.

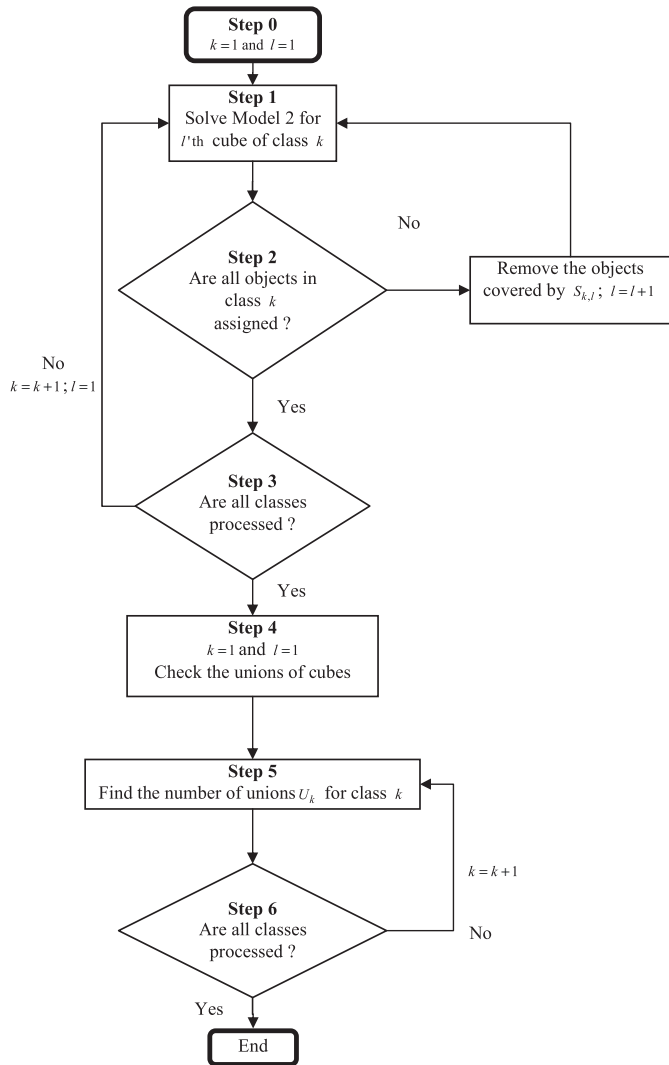


Fig. 7. Flowchart of the proposed algorithm.

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

Rule #	Union of cubes	$S_{k,l}$	$b_{k,l,1}$	$b_{k,l,2}$	$b_{k,l,3}$	$b_{k,l,4}$	$r_{k,l}$
R_1	$S_{1,1}$	$S_{1,1}$	5.1	3.2	1.85	0.5	2.45
R_2	$S_{2,1} \cup S_{2,2}$	$S_{2,1}$	6.7	2.6	3.5	1.2	2.5
		$S_{2,2}$	5.9	3.15	4	1.3	1.55
R_3	$S_{3,1} \cup S_{3,2}$	$S_{3,1}$	6.2	2.9	6.6	2.4	2.7
		$S_{3,2}$	5.3	2.45	4.9	1.6	1.05

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. Table 2 contains three rules R_1 , R_2 , and R_3 .

Rule R_1 is expressed by a cube $S_{1,1}$, 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 $S_{2,1}$ and $S_{2,2}$, 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 $|\text{sepal length}-5.9|+|\text{sepal width}-3.15|+|\text{petal length}-4|+|\text{petal width}-1.3| \leq 1.55$ then the Iris belongs to Versicolor.

Rule R_3 is also the union of two cubes $S_{3,1}$ and $S_{3,2}$, which shows that

- if $|\text{sepal length}-6.2|+|\text{sepal width}-2.9|+|\text{petal length}-6.6|+|\text{petal width}-2.4| \leq 2.7$ or $|\text{sepal length}-5.3|+|\text{sepal width}-2.45|+|\text{petal length}-4.9|+|\text{petal width}-1.6| \leq 1.05$ then the Iris belongs to Virginica.

Decision tree method [7] (see Fig. 8(a)) and polynomial hyper-plane support vector method [22] were also used to induce classification rules for the same data. Fig. 8(a) is the partial Iris classification tree, which only lists the best path for each class. For example, from the following branches know that

- if (petal length < 3) then the Iris belongs to Setosa;
- if (petal length ≥ 3) and (petal width ≥ 1.8) then the Iris belongs to Virginica.

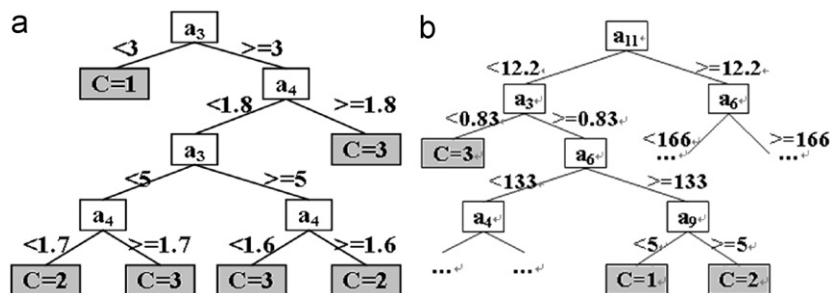


Fig. 8. Classifying using the decision tree method. (a) Decision tree for the Iris dataset [17]. (b) Decision tree for the HSV dataset [5].

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 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
Comparison results for the Iris dataset (R_1, R_2, R_3).

Items	DIAMOND	Decision tree	Hyper-plane support vector
$AR(R_1, R_2, R_3)$	(1,1,1)	(1,0.98,0.98)	(1,0.98,0.96)
$SR(R_1, R_2, R_3)$	(1,0.98,0.98)	(1,0.98,0.98)	(1,0.96,0.98)
CR	1	0.5	0.1875

Table 4
Centroid points for HSV data by the DIAMOND method.

Rule #	Union of cubes	$S_{k,l}$	$b_{k,l,1}$	$b_{k,l,2}$	$b_{k,l,3}$	$b_{k,l,4}$	$b_{k,l,5}$	$b_{k,l,6}$	$b_{k,l,7}$	$b_{k,l,8}$	$b_{k,l,9}$	$b_{k,l,10}$	$b_{k,l,11}$	$r_{k,l}$
R_1	$S_{1,1} \cup \dots \cup S_{1,11}$	$S_{1,1}$	0	63	14	2	14.7	86.5	180	13.8	23.3	627	61.8	565.5
		$S_{1,2}$	1	35.2	11	1	11.7	29	66.75	10.3	20.8	139.1	53.8	180.15
		$S_{1,3}$	1	38.1	8	3	4.1	159	118.45	21.6	5.3	115	49.8	205.85
		$S_{1,4}$	1	33	0	2	8.1	82	29.15	1.7	14.7	232	78.2	146.65
		$S_{1,5}$	0	59	16.05	2	2.5	34	32	12.8	16.7	81.5	16.95	142.5
		$S_{1,6}$	1	38	30	3	20.9	389.7	120	39.1	14.7	174.25	78.2	419.45
		$S_{1,7}$	1	40	5	2	8.6	122	5	8.7	34.5	336	8.4	221.2
		$S_{1,8}$	0	50	12	0	15.7	140	14.05	11.8	12.3	199	93.1	147.35
		$S_{1,9}$	1	35.1	4	3	4	149.35	38	15.7	12.5	128	8.45	100.4
		$S_{1,10}$	1	42	14	2.1	12.6	83.5	170.25	24.7	14.8	818.7	70.25	739.3
		$S_{1,11}$	1	60	10	3	4.2	97	112.75	26.8	13.9	163	104.2	241.75
R_2	$S_{2,1} \cup \dots \cup S_{2,6}$	$S_{2,1}$	0	27	16.45	4	11.7	198	88	11.4	34.5	172	10.9	152.35
		$S_{2,2}$	1	32	5	2	15.9	185	56.95	13.2	11	223	13.8	113.55
		$S_{2,3}$	1	50	32	3.9	10	191	6	1.1	12.3	199	13.7	146.1
		$S_{2,4}$	1	56	9	4	10.3	76	6	8.5	9.8	165.7	93.1	153.2
		$S_{2,5}$	1	32	4	1	8.3	118	60	9.2	27.5	163	13	99.3
		$S_{2,6}$	1	27	2	4	20.9	213	26	14.6	6.5	266	85.1	167.6
R_3	$S_{3,1} \cup \dots \cup S_{3,5}$	$S_{3,1}$	1	56	6	0	4	170.3	120	6.1	21	232	13.8	156.9
		$S_{3,2}$	1	27	20	3	6.8	91.25	67.15	5.2	19	87.2	12	103.8
		$S_{3,3}$	0	54	7	2	7.1	194.4	131.05	9.2	19	391.1	15.2	240.85
		$S_{3,4}$	1	56	4	2	14.1	212	78	14.6	16.7	41	6.3	201.5
		$S_{3,5}$	1	33	3	4	6.8	224.88	132	3.9	11	175.2	19.8	166
R_4	$S_{4,1} \cup \dots \cup S_{4,5}$	$S_{4,1}$	1	27	8	3	26.1	69	13	2.6	11.8	58.15	10.3	82.75
		$S_{4,2}$	1	51	11	4	21	474.2	50	3.6	38.7	387	151.4	527.3
		$S_{4,3}$	1	60	8	4	6	225	43.75	7.9	5.6	183	56.6	150.75
		$S_{4,4}$	1	28	11	1	7.5	143	32	36.1	16.7	202.85	17.2	95.85
		$S_{4,5}$	1	46	12	2	7.4	35.7	21.1	4.4	17.8	165	12.2	88.3

The details are expressed by [23]. To maximize the support rate with respect to the constraint that $AR \geq 0.9$ and to minimize the number of cubes, the DIAMOND method generates eight unions of cubes iteratively. Table 4 shows the centroids and radiuses of these cubes.

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 ≥ 0.83) and (volume of gastric juice per 1 h ≥ 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 auxillary nodes detected). For this dataset, the DIAMOND method generates four unions of cubes for classifying 294 patients. The centroids and radiuses 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

Table 5
Comparison of results for the HSV dataset (R_1, R_2, R_3, R_4).

Items	DIAMOND	Decision tree	Hyper-plane support vector
$AR(R_1, R_2, R_3, R_4)$	(1,1,1,1)	(0.93,0.81,0.7,0.71)	(0.9,1,1,0.9)
$SR(R_1, R_2, R_3, R_4)$	(0.98,0.89,0.89,0.79)	(0.93,0.72,0.78,0.71)	(0.9,0.72,0.67,0.69)
CR	0.5	0.17	0.09

Table 6
Centroid points for breast cancer data by the DIAMOND method.

Rule #	Union of cubes	$S_{k,l}$	$b_{k,l,1}$	$b_{k,l,2}$	$b_{k,l,3}$	$r_{k,l}$	$S_{k,l}$	$b_{k,l,1}$	$b_{k,l,2}$	$b_{k,l,3}$	$r_{k,l}$
R_1	$S_{1,1} \cup \dots \cup S_{1,32}$	$S_{1,1}$	54.503	60.998	0	6.495	$S_{1,2}$	37.5	61.498	4.003	9.995
		$S_{1,3}$	65.5	67	0	6.495	$S_{1,4}$	50.003	67.995	0.003	5.995
		$S_{1,5}$	48.5	60.003	1.503	4.995	$S_{1,6}$	58.498	60.498	0	5.995
		$S_{1,7}$	57.498	69	1.998	5.495	$S_{1,8}$	32	68.998	2.498	11.495
		$S_{1,9}$	40.003	66.003	15	10.995	$S_{1,10}$	41.998	65.498	2	4.495
		$S_{1,11}$	73.003	69	0.498	7.495	$S_{1,12}$	44.5	61	1	4.495
		$S_{1,13}$	64	58	6.503	8.498	$S_{1,14}$	49.503	61	6	5.498
		$S_{1,15}$	63	63.503	3	5.498	$S_{1,16}$	60	69	28.498	14.498
		$S_{1,17}$	38	58	1.5	6.495	$S_{1,18}$	76	59.003	2	7.998
		$S_{1,19}$	48.998	64	4.498	5.495	$S_{1,20}$	60	64	12.498	6.498
		$S_{1,21}$	69	61.5	0	4.495	$S_{1,22}$	38.998	61.998	12	11.995
		$S_{1,23}$	33	69	44.5	28.495	$S_{1,24}$	45.5	64	2	4.495
		$S_{1,25}$	55	60	19	7.995	$S_{1,26}$	48	58	5.998	4.998
		$S_{1,27}$	46.998	67.503	1	3.5	$S_{1,28}$	59.5	64	3.998	4.498
		$S_{1,29}$	59	61	10	6.995	$S_{1,30}$	34	58	19.498	12.498
		$S_{1,31}$	49.498	69	17.998	9.495	$S_{1,32}$	51	67.998	19.498	8.495
		R_2	$S_{2,1} \cup \dots \cup S_{2,24}$	$S_{2,1}$	53	64.5	13	8.495	$S_{2,2}$	48.503	58
$S_{2,3}$	61			62	16.003	9.998	$S_{2,4}$	54.5	65	5.003	5.498
$S_{2,5}$	44			64.5	9	7.495	$S_{2,6}$	44	63	38.5	21.495
$S_{2,7}$	52			58	3	2.995	$S_{2,8}$	41.003	68.998	3	5
$S_{2,9}$	67			64	7	7.995	$S_{2,10}$	61.503	60	3.997	4.494
$S_{2,11}$	52.503			66.003	3.5	4	$S_{2,12}$	71.003	62.003	5.003	6.997
$S_{2,13}$	83			62.003	2.497	9.494	$S_{2,14}$	55.503	61.494	8.003	4.994
$S_{2,15}$	44			58	1.003	2.997	$S_{2,16}$	45.997	65.003	2.5	4.494
$S_{2,17}$	45.003			68.503	7	5.494	$S_{2,18}$	66	65.997	10.003	7.994
$S_{2,19}$	56			62	5	5.994	$S_{2,20}$	46.003	65.494	3.003	4.494
$S_{2,21}$	72.003			63.003	6.503	7.497	$S_{2,22}$	83	58	3.503	13.497
$S_{2,23}$	43.503			59	5.5	4.997	$S_{2,24}$	55.503	58.503	4.5	4.494
$S_{2,25}$	$S_{2,25}$			43	64	0	0.994				
$S_{2,26}$	$S_{2,26}$			60.5	65.5	1	2.994				

Table 7
Comparison of results for the breast cancer dataset (R_1, R_2).

Items	DIAMOND	Decision tree	Hyper-plane support vector
$AR(R_1, R_2)$	(1,1)	(0.92,0.77)	(0.8,0.6)
$SR(R_1, R_2)$	(0.98,0.81)	(0.92,0.77)	(0.92,0.7)
CR	0.5	0.03	0.01

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 ($n=250$), eight attributes ($m=8$) 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, three attributes and two classes) by the DIAMOND method on a PC version CPLEX takes about 10 min. While a decision tree method takes only 5 min for solving the same problem. Recently, Li and Lu [24] developed a logarithmic method to accelerate the solution speed of solving an integer program, which may be helpful in enhancing the DIAMOND method.
- (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: $nq \approx 20,000$,
- number of continuous variables: $mnq \approx 100,000$,
- number of linear constraints: $5mnq \approx 500,000$,

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, HSV, 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]:

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

$$a_{i,j} - b_{k,l,j} + e_{k,l,i,j} \geq 0, \quad (18)$$

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 $a_{i,j} - b_{k,l,j} + 2e_{k,l,i,j} = a_{i,j} - b_{k,l,j} = |a_{i,j} - b_{k,l,j}|$.
- (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 $a_{i,j} - b_{k,l,j} + 2e_{k,l,i,j} \geq b_{k,l,j} - a_{i,j} = |a_{i,j} - b_{k,l,j}|$. \square

Proposition 2. Inequality (15) can be linearized as follows:

$$\sum_{j=1}^m |a_{i,j} - b_{k,l,j}| \quad (19)$$

$$= \sum_{j=1}^m (1 - 2\lambda_{k,l,i,j})(a_{i,j} - b_{k,l,j}) \quad (20)$$

$$= \sum_{j=1}^m (a_{i,j} - b_{k,l,j} - 2a_{i,j}\lambda_{k,l,i,j} + 2z_{k,l,i,j}) > r_{k,l} - Mv_{k,l,i}, \quad (21)$$

where

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

$$\bar{b}_j(\lambda_{k,l,i,j} - 1) + b_{k,l,j} \leq z_{k,l,i,j} \leq b_{k,l,j} + \bar{b}_j(1 - \lambda_{k,l,i,j}), \quad (23)$$

$$0 \leq z_{k,l,i,j} \leq \bar{b}_j\lambda_{k,l,i,j}, \quad (24)$$

$$\bar{b}_j \text{ is constant, } \bar{b}_j = \max\{a_{i,j}; \forall i \notin \{i\}\} \text{ and } \lambda_{k,l,i,j} \in \{0, 1\}. \quad (25)$$

Proof.

- (i) If $\lambda_{k,l,i,j} = 0$ then $z_{k,l,i,j} = 0$ from (23), which results in $|a_{i,j} - b_{k,l,j}| = a_{i,j} - b_{k,l,j}$.
- (ii) If $\lambda_{k,l,i,j} = 1$ then $z_{k,l,i,j} = b_{k,l,j}$ from (22), which results in $|a_{i,j} - b_{k,l,j}| = a_{i,j} - b_{k,l,j} - 2a_{i,j} + 2b_{k,l,j} = b_{k,l,j} - a_{i,j}$. \square

Appendix B

Tables 8–16.

Table 8
Classification results for the Iris dataset by the DIAMOND method.

Rule	Unions of cubes	Covered objects (#)		AR	SR
		Correctly	Incorrectly		
R ₁	S _{1,1}	1–50	None	1	1
R ₂	S _{2,1} ∪ S _{2,1}	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	None	1	0.98
R ₃	S _{3,1} ∪ S _{3,1}	101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150	None	1	0.98

$$CR(R_1, R_2, R_3) = \frac{3}{3} = 1.$$

Table 9
Decision tree method for the Iris dataset.

Rule	Decision branch	AR	SR
R_1	If ($a_3 < 3$) then objects belong to class 1	1	1
R_2	If ($a_3 \geq 3 \wedge a_4 < 1.8 \wedge a_3 < 5 \wedge a_4 < 1.7$) or if ($a_3 \geq 3 \wedge a_4 < 1.8 \wedge a_3 \geq 5 \wedge a_4 \geq 1.6$) then objects belong to class 2	0.98	0.98
R_3	If ($a_3 \geq 3 \wedge a_4 < 1.8 \wedge a_3 < 5 \wedge a_4 \geq 1.7$) or if ($a_3 \geq 3 \wedge a_4 < 1.8 \wedge a_3 \geq 5 \wedge a_4 < 1.6$) or if ($a_3 \geq 3 \wedge a_4 \geq 1.8$) then objects belong to class 3	0.98	0.98

$CR(R_1, R_2, R_3) = \frac{3}{8} = 0.5.$

Table 10
Hyper-plane method for the Iris dataset.

Rule	Support vectors (polynomial function)		AR	SR
	#	(Y_i, Y_j)		
R_1	1	(0.008, 0.0004)	1	1
	2	(0, 0.0006)		
	3	(0.0005, 0)		
	4	(0, 0.0006)		
R_2	5	(-0.1)	0.98	0.96
	6	(-0.0535)		
	7	(-0.0598)		
	8	(-0.1)		
	9	(-0.009, 0)		
R_3	10	(-0.0018, -0.0302)	0.96	0.98
	11	(-0, -0.1541)		
	12	(-0, -0.2262)		
	13	(-0, -0.6437)		
	14	(-0, -0.0793)		
	15	(-0, -1)		
	16	(-0, -1)		

$CR(R_1, R_2, R_3) = \frac{3}{16} = 0.1875.$

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

Rule	Unions of cubes	Covered objects (#)		AR	SR
		Correctly	Incorrect (miss)		
R_1	$S_{1,1} \cup S_{1,2} \cup \dots \cup S_{1,11}$	1, 2, 3, 4, 6, 7, 8, 9, 11, 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 28, 29, 31, 32, 34, 35, 36, 37, 38, 40, 46, 47, 48, 49, 50, 52, 53, 55, 56, 57, 58, 60, 61, 66, 67, 68, 69, 70, 71, 72, 74, 76, 77, 78, 83, 84, 85, 88, 89, 91, 93, 94, 97, 98, 99, 100, 102, 104, 106, 108, 111, 112, 113, 114, 115, 116, 117, 119, 122	(30,33)	1	0.96
R_2	$S_{2,1} \cup S_{2,2} \cup \dots \cup S_{2,6}$	10, 12, 20, 44, 45, 51, 54, 62, 73, 80, 86, 87, 90, 96, 103, 110, 120, 121	(44,73,96)	1	0.83
R_3	$S_{3,1} \cup S_{3,3}$	43, 79, 109, 118	-24	1	0.89
	$S_{3,2} \cup S_{3,4}$	5, 13, 65, 82			
R_4	$S_{4,1}$	18, 64	118 (95,105,107)	0.91	0.79
	$S_{4,2}$	39, 63, 75			
	$S_{4,3}$	42, 59			
	$S_{4,4}$	81, 92			
	$S_{4,5}$	41, 101			

$CR(R_1, R_2, R_3, R_4) = \frac{4}{14} = 0.29.$

Table 12
Decision tree method for the HSV dataset.

Rules	Decision branch	AR	SR
R_1	If ($a_{11} < 12.2 \wedge a_3 \geq 0.83 \wedge a_6 < 133 \wedge a_9 < 5.7 \wedge a_7 \geq 88$) or If ($a_{11} < 12.2 \wedge a_3 \geq 0.83 \wedge a_6 < 133 \wedge a_9 \geq 5.7 \wedge a_8 < 2.6$) or If ($a_{11} < 12.2 \wedge a_3 \geq 0.83 \wedge a_6 < 133 \wedge a_9 \geq 5.7 \wedge a_8 \geq 2.6 \wedge a_6 \geq 60 \wedge a_2 \geq 28$) or If ($a_{11} < 12.2 \wedge a_3 \geq 0.83 \wedge a_6 \geq 133 \wedge a_9 < 5$) or	0.93	0.93

Table 12 (continued)

Rules	Decision branch	AR	SR
R_2	If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 < 14.2) \cap (a_2 < 37) \cap (a_9 < 11.7) \cap (a_3 < 11)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 < 14.2) \cap (a_2 < 37) \cap (a_9 \geq 11.7) \cap (a_7 < 27)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 < 14.2) \cap (a_2 < 37) \cap (a_9 \geq 11.7) \cap (a_7 \geq 27) \cap (a_6 < 57)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 < 14.2) \cap (a_2 \geq 37) \cap (a_2 \geq 46)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 \geq 14.2)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_{11} < 39.1) \cap (a_6 < 249) \cap (a_9 < 8.7) \cap (a_2 < 26)$ or If $(a_{11} \geq 12.2) \cap (a_6 \geq 166) \cap (a_{11} < 39.1) \cap (a_6 \geq 249)$ or If $(a_{11} \geq 12.2) \cap (a_6 \geq 166) \cap (a_{11} \geq 39.1) \cap (a_3 < 0.83)$ then objects belong to class 1	0.81	0.72
	If $(a_{11} < 12.2) \cap (a_3 \geq 0.83) \cap (a_6 \geq 133) \cap (a_9 \geq 5)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 < 14.2) \cap (a_2 < 37) \cap (a_9 < 11.7) \cap (a_3 \geq 11)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 < 14.2) \cap (a_2 < 37) \cap (a_9 \geq 11.7) \cap (a_7 \geq 27) \cap (a_6 \geq 57)$ or If $(a_{11} \geq 12.2) \cap (a_6 \geq 166) \cap (a_{11} < 39.1) \cap (a_6 < 249) \cap (a_9 \geq 8.7) \cap (a_6 < 214)$ then objects belong to class 2		
R_3	If $(a_{11} < 12.2) \cap (a_3 < 0.83)$ or If $(a_{11} \geq 12.2) \cap (a_6 < 166) \cap (a_9 < 14.2) \cap (a_2 \geq 37)$ or $(a_2 < 46)$ or If $(a_{11} \geq 12.2) \cap (a_6 \geq 166) \cap (a_{11} < 39.1) \cap (a_6 < 249) \cap (a_9 < 8.7) \cap (a_2 \geq 26)$ then objects belong to class 3	0.7	0.78
R_4	If $(a_{11} \geq 12.2) \cap (a_6 \geq 166) \cap (a_{11} < 39.1) \cap (a_6 < 249) \cap (a_9 \geq 8.7) \cap (a_6 \geq 214)$ or If $(a_{11} < 12.2) \cap (a_3 \geq 0.83) \cap (a_6 < 133) \cap (a_9 < 5.7) \cap (a_7 < 88)$ or If $(a_{11} < 12.2) \cap (a_3 \geq 0.83) \cap (a_6 < 133) \cap (a_9 \geq 5.7) \cap (a_8 \geq 2.6) \cap (a_6 < 60)$ or If $(a_{11} < 12.2) \cap (a_3 \geq 0.83) \cap (a_6 < 133) \cap (a_9 \geq 5.7) \cap (a_8 \geq 2.6) \cap (a_6 \geq 60) \cap (a_2 < 28)$ or If $(a_{11} \geq 12.2) \cap (a_6 \geq 166) \cap (a_{11} \geq 39.1) \cap (a_3 \geq 0.83)$ then objects belong to class 4	0.71	0.71

$CR(R_1, R_2, R_3, R_4) = \frac{6}{24} = 0.17.$

Table 13
Hyper-plane method for the HSV dataset.

Rule	#	Support vectors (polynomial function)		AR	SR
		(y_i)	$(a_1, a_2, \dots, a_{11})$		
R_1	1	(0.229, 0.114, 0.164)	(0.22, 2.0, 8.3, 111, 28.9, 2.20, 8.192, 39.8)	0.9	0.9
	19	(0.229, 0.114, 0.164)	(0.35, 4.0, 3.8, 57, 116, 2.2, 10.4, 191, 19.8)		
R_2	20	(-1, 0.5, 0.713)	(0.33, 2.2, 8.7, 135, 54, 11.8, 29, 186, 53.8)	1	0.72
	33	(-1, 0.5, 0.713)	(0.28, 4.0, 8.9, 88, 28, 7.8, 12.3, 163, 20)		
R_3	34	(-1, -1, 1)	(0.54, 2.3, 5.3, 166, 124, 8.7, 6.8, 236, 16)	1	0.67
	40	(-1, -1, 1)	(0.45, 3.0, 5.2, 67, 128, 3.5, 11.8, 230, 27.1)		
R_4	41	(-1, -1, -0.7)	(1.40, 4.0, 8.1, 62, 17.5, 5.6, 41, 2.3)	0.9	0.69
	45	(-1, -1, -0.7)	(0.50, 8.4, 10.6, 185, 21, 19.6, 25.3, 224, 56.6)		

$CR(R_1, R_2, R_3, R_4) = \frac{4}{45} = 0.09.$

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

Rule	Cube #	Covered objects (#)		AR	SR
		Correctly	Incorrectly (miss)		
R_1	$S_{1,1} \cup S_{1,2} \cup$	1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,	(10, 43, 50, 70, 80, 84, 85, 89, 106, 113, 134, 175, 177, 245, 263, 271, 288, 291, 298)	1	0.92
	$S_{1,3} \cup S_{1,4} \cup$	21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36,			
	$S_{1,5} \cup S_{1,6} \cup$	37, 38, 39, 40, 41, 42, 47, 48, 49, 51, 52, 53, 56, 57,			
	$S_{1,8} \cup S_{1,9} \cup$	58, 59, 60, 61, 62, 67, 68, 69, 71, 72, 73, 77, 78, 79,			
	$S_{1,10} \cup S_{1,11} \cup$	86, 87, 88, 94, 95, 96, 100, 101, 102, 103, 104, 105,			
	$S_{1,13} \cup S_{1,14} \cup$	107, 111, 112, 114, 117, 118, 119, 120, 121, 122, 123,			
	$S_{1,16} \cup S_{1,17} \cup$	124, 127, 128, 129, 130, 131, 132, 133, 135, 136, 139,			
	$S_{1,18} \cup S_{1,19} \cup$	140, 141, 142, 147, 148, 149, 150, 151, 152, 153, 154,			
	$S_{1,20} \cup S_{1,21} \cup$	155, 156, 163, 164, 165, 166, 167, 172, 173, 174, 176,			
	$S_{1,22} \cup S_{1,24} \cup$	179, 180, 183, 184, 185, 187, 193, 197, 203, 204, 205,			
	$S_{1,26} \cup S_{1,28} \cup$	206, 207, 209, 210, 211, 213, 214, 215, 217, 218, 219,			
	$S_{1,29}$	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			
	$S_{1,7}$	188, 190, 194, 195, 196, 202, 208, 212, 223, 227			
$S_{1,12} \cup S_{1,15} \cup S_{1,30}$	201, 220, 222, 236, 242, 252, 254, 266				

Table 14 (continued)

Rule	Cube #	Covered objects (#)		AR	SR
		Correctly	Incorrectly (miss)		
	$S_{1,23}$	189, 228, 253, 255			
	$S_{1,25}$	272, 278, 283			
	$S_{1,27}$	178, 186			
R_2	$S_{2,1} \cup S_{2,9}$	44, 63, 76, 93, 97, 108, 109, 137, 161, 169, 216	(8, 9, 25, 35, 45, 55, 83, 90, 98, 115, 126, 143, 145, 232, 240, 259, 269, 294)	1	0.78
	$S_{2,2} \cup S_{2,3} \cup S_{2,4}$	116, 125, 146, 162, 168, 170, 171, 181, 182, 191, 192, 199, 224, 239, 241, 261, 262, 270			
	$S_{2,5}$	138, 144, 157			
	$S_{2,6}$	74, 81, 110			
	$S_{2,7}$	260, 274, 275			
	$S_{2,8}$	46, 54, 91			
	$S_{2,10}$	160, 198			
	$S_{2,11}$	282, 300			
	$S_{2,12}$	286, 287			
	$S_{2,13}$	65, 66			
	$S_{2,14}$	64, 75			
	$S_{2,15}$	158, 200			
	$S_{2,16}$	82, 99			
	$S_{2,17}$	305, 306			
	$S_{2,18}$	230, 246			
	$S_{2,19}$	225, 231			
	$S_{2,20}$	92			

$CR(R_1, R_2) = \frac{2}{23} = 0.09.$

Table 15

Decision tree method for the breast cancer dataset.

Rules	Decision branch	AR	SR
R_1	If $(a_3 < 9) \cap (a_1 < 78) \cap (a_3 < 3) \cap (a_1 < 48) \cap (a_2 < 64) \cap (a_1 < 43)$ or If $(a_3 < 9) \cap (a_1 < 78) \cap (a_3 < 3) \cap (a_1 < 48) \cap (a_2 < 64) \cap (a_1 \geq 43) \cap (a_2 \geq 60)$ or : or If $(a_2 \geq 61) \cap (a_3 \geq 25)$ then objects belong to class 1	0.92	0.77
R_2	If $(a_3 < 9) \cap (a_1 < 78) \cap (a_3 < 3) \cap (a_1 < 48) \cap (a_2 < 64) \cap (a_1 \geq 43) \cap (a_2 < 60)$ or : or If $(a_3 \geq 9) \cap (a_2 \geq 61) \cap (a_3 < 25) \cap (a_1 \geq 65)$ then objects belong to class 2	0.92	0.77

$CR(R_1, R_2) = \frac{2}{60} = 0.03.$

Table 16

Hyper-plane method for the breast cancer dataset.

Rule	#	Support vectors (linear function)		AR	SR
		(y_i)	(a_1, a_2, a_3)		
R_1	1	-1	(31,65,4)	0.8	0.92
		
R_2	81	-1	(76,67,0)	0.6	0.7
	1	(-1)	(38,69,21)		
		
	80	(-1)	(67,64,8)		

$CR(R_1, R_2) = \frac{2}{161} = 0.01.$

References

[1] N. Risch, Searching for genetic determinants in the new millennium, *Nature* 405 (6788) (2000) 847–856.
 [2] A. Fielding, Cluster and Classification Techniques for the Biosciences, Cambridge University Press, 2007 (ISBN 0521852811).
 [3] P. Royston, Choice of scale for cubic smoothing spline models in medical applications, *Statistics in Medicine* 19 (9) (2000) 1191–1205.
 [4] D. Altman, P. Royston, What do we mean by validating a prognostic model? *Statistics in Medicine* 19 (4) (2000) 453–473.
 [5] H. Li, M. Chen, Induction of multiple criteria optimal classification rules for biological and medical data, *Computers in Biology and Medicine* 38 (1) (2008) 42–52.

[6] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, vol. 1, Wadsworth, Belmont, CA, 1984.
 [7] J. Quinlan, C4. 5: Programs for Machine Learning, Morgan Kaufmann, 1993 (ISBN 1558602380).
 [8] H. Kim, W. Loh, Classification trees with unbiased multiway splits, *Journal of the American Statistical Association* 96 (454) (2001) 589–604.
 [9] V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, 2000 (ISBN 0387987800).
 [10] R. Rifkin, Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning, Ph.D. Thesis, MaSSachuSettS InStitute of Technology, 2002.
 [11] S. Katagiri, S. Abe, Incremental training of support vector machines using hyperspheres, *Pattern Recognition Letters* 27 (13) (2006) 1495–1507.
 [12] D. Bertsimas, R. Shioda, Classification and regression via integer optimization, *Operations Research* 55 (2) (2007) 252–271.
 [13] X. Wei, Y. Li, Linear programming minimum sphere set covering for extreme learning machines, *Neurocomputing* 71 (4–6) (2008) 570–575.
 [14] L. Gu, H. Wu, A kernel-based fuzzy greedy multiple hyperspheres covering algorithm for pattern classification, *Neurocomputing* 72 (1–3) (2008) 313–320.
 [15] Y. Lin, X. Wang, W. Ng, Q. Chang, D. Yeung, X. Wang, Sphere classification for ambiguous data, in: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, 2006, pp. 13–16.
 [16] H. Mhand, M. Rym, et al., A literature review on circle and sphere packing problems: models and methodologies, *Advances in Operations Research* 2009 (2009).
 [17] R. Fisher, et al., The use of multiple measurements in taxonomic problems, *Annals of Eugenics* 7 (1936) 179–188.
 [18] K. Slowinski, Rough classification of HSV patients, in: Intelligent Decision Support—Handbook of Applications and Advances of the Rough Sets Theory, 1992, pp. 77–94.

- [19] D. Dunn, W. Thomas, J. Hunter, An evaluation of highly selective vagotomy in the treatment of chronic duodenal ulcer, *Surgery, Gynecology & Obstetrics* 150 (6) (1980) 845.
- [20] S. Haberman, Generalized residuals for log-linear models, in: *Invited Papers: Proceedings of the 9th International Biometric Conference*, Biometric Society, Boston, August 22–27, 1976, p. 104.
- [21] IBM/ILOG, Cplex 12.0 reference manual, Software, 2009. Available at <<http://www.ilog.com/products/cplex/>>, 2009.
- [22] C. Chang, C. Lin, LIBSVM: a library for support vector machines, 2001.
- [23] J. Goligher, G. Hill, T. Kenny, E. Nutter, Proximal gastric vagotomy without drainage for duodenal ulcer: results after 5–8 years, *British Journal Surgery* 65 (3) (1978) 145–151.
- [24] H. Li, H. Lu, Global optimization for generalized geometric programs with mixed free-sign variables, *Operations Research* 57 (3) (2009) 701–713.
- [25] G. Klau, S. Rahmann, A. Schliep, M. Vingron, K. Reinert, Optimal robust non-unique probe selection using integer linear programming, *Bioinformatics* 20 (Suppl. 1) (2004) i186.
- [26] H. Li, C. Fu, A linear programming approach for identifying a consensus sequence on DNA sequences, *Bioinformatics* 21 (9) (2005) 1838.
- [27] P. Deng, M. Thai, Q. Ma, W. Wu, Efficient non-unique probes selection algorithms for DNA microarray, *BMC Genomics* 9 (Suppl. 1) (2008) S22.
- [28] C. Than, R. Sugino, H. Innan, L. Nakhleh, Efficient inference of bacterial strain trees from genome-scale multilocus data, *Bioinformatics* 24 (13) (2008) i123.
- [29] M. Rockville, Large Scale Computing and Storage Requirements for Biological and Environmental Research, Technical Report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2009.
- [30] H. Li, An efficient method for solving linear goal programming problems, *Journal of Optimization Theory and Applications* 90 (2) (1996) 465–469.

Han-Lin Li obtained his PhD degree from University of Pennsylvania 1983. He has been Chair Professor at National Tsing Hua 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.