An Optimal Classification Method for Biological and Medical Data

Yao-Huei Huang,1 Yu-Chien Ko,2 and Hao-Chun Lu3

1 Institute of Information Management, National Chiao Tung University, Management Building 2, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan
2 Department of Information Management, Chu-Hua University, No. 707, Section 2, WuFu Road, Hsinchu 300, Taiwan
3 Department of Information Management, College of Management, Fu Jen Catholic University, No. 510, Jhongjhung Road, Sinjhuang, Taipei 242, Taiwan

Correspondence should be addressed to Yao-Huei Huang, yaohuei.huang@gmail.com

Received 25 October 2011; Revised 25 January 2012; Accepted 28 January 2012

Academic Editor: Jung-Fa Tsai

Copyright © 2012 Yao-Huei Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a union of hyperspheres by the mixed-integer nonlinear program to classify biological and medical datasets. A classifying program with nonlinear terms uses piecewise linearization technique to obtain a global optimum. The numerical examples illustrate that the proposed method can obtain the global optimum more effectively than current methods.

1. Introduction

Classification techniques have been widely applied in the biological and medical research domains [1–5]. Either objects classification or patterns recognition for biological and medical datasets necessarily demands an optimum accuracy for saving patients’ lives. However, cancer identification with the supervised learning technique does not take a global view in identifying species or predicting survivals. The improvement should cover the whole scope to give implications instead of only considering the efficiency for diagnosis. This research aims to extract features from whole datasets in terms of induction rules.

In the given dataset with several objects, in which each object has some attributes and belongs to a specific class, classification techniques are used to find a rule of attributes that appropriately describes the features of a specified class. The techniques have been studied over the last four decades, including decision tree-based methods [6–11], hyperplane-based methods [12–14], and machine learning-based methods [14–17].
To assess the effects of these classifying techniques, three criteria are used for evaluating the quality of inducing rules based on the study of Li and Chen [3].

(i) **Accuracy.** The rule fitting a class should not cover the objects of other classes. The accuracy of a rule should be the higher the better.

(ii) **Support.** A good rule of fitting a class should be supported by most of the objects of the same class.

(iii) **Compact.** A good rule should be expressed in a compact way. That is, the fewer the number of rules, the better the rules are.

This study proposes a novel method to induce rules with high rates of accuracy, support, and compactness based on global optimization techniques, which have become more and more useful in biological and medical researches.

The rest of this paper is organized as follows. Section 2 gives an overview of the related literatures. Two types of mathematical models and a classification algorithm are proposed in Section 3. The numerical examples demonstrate the effectiveness of the proposed method in Section 4. Finally, the main conclusions of this study and future work are drawn in Section 5.

### 2. Literature Review

Currently, two well-known methods are used to induce classification rules. The first method is the decision tree-based method, which has been developed in the last few decades [6–10]. It is widely applied to fault isolation of an induction motor [18] to classify normal or tumor tissues [19], skeletal maturity assessment [20], proteomic mass spectra classification [21], and other cases [22, 23]. Although the decision tree-based method assumes that all classes can be separated by linear operations, the inducing rules will suffer if the boundaries between the classes are nonlinear. In fact, the linearity assumption prohibits practical applications because many biological and medical datasets have complicated nonlinear interactions between attributes and predicted classes.

Consider the classification problem with two attributes as shown in Figure 1, where “○” represents a first-class object, and “●” represents a second-class object. Figure 1 depicts a situation in which a nonlinear relationship exists between the objects of two classes. Decision tree method focuses on inducing classification rules for the objects, as shown in Figure 1(b), in which the decision tree method requires four rectangular regions to classify the objects.

The second is the support vector hyperplane method, which conducts feature selection and rule extraction from the gene expression data of cancer tissue [24]; it is also applied in other applications [12–14, 25]. The technique separates observations of different classes by multiple hyperplanes. As the number of decision variables is required to express the relationship between each training datum and hyperplane, and the separating hyperplane is assumed a nonlinear programming problem, the training speed becomes slow for a large number of training data. Additionally, similar hypersphere support vector methods have been developed by Lin et al. [26], Wang et al. [27], Gu and Wu [28], and Hifi and M’Hallah [29] for classifying objects. In classification algorithms, they partition the sample space using the sphere-structured support vector machine [14, 30]. However, these methods need to form a classification problem as a nonlinear nonconvex program, which makes reaching an optimal solution difficult. Taking Figure 1 as an example, a hyperplane-based method requires four hyperplanes to discriminate the objects, as shown in Figure 2.
As previously mentioned, many biological and medical datasets have complicated boundaries between attributes and classes. Both decision tree-based methods and hyperplane-based methods find only the rules with high accuracy, which either cover only a narrow part of the objects or require numerous attributes to explain a classification rule. Although these methods are computationally effective for deducing the classifications rules, they have two limitations as follows.

(i) Decision tree-based methods are heuristic approaches that can only induce feasible rules. Moreover, decision tree-based methods split the data into hyperrectangular regions using a single variable, which may generate a large number of branches (i.e., low rates of compactness).

(ii) Hyperplane-based methods use numerous hyperplanes to separate objects of different classes and divide the objects in a dataset into indistinct groups. The method may generate a large number of hyperplanes and associated rules with low rates of compactness.
Therefore, this study proposes a novel hypersphere method to induce classification rules based on a piecewise linearization technique. The technique reformulates the original hypersphere model by a piecewise linearization approach using a number of binary variables and constraints in the number of piecewise line segments. As the number of break points used in the linearization process increases, the error in linear approximation decreases, and an approximately global optimal solution of the hypersphere model can be obtained. That is, the proposed method is an optimization approach that can find the optimal rules with a high rate of accuracy, support, and compactness. The concept of the hypersphere method is depicted in Figure 3, in which only one circle is required to classify the objects. All objects of class “•” are covered by a circle, and those not covered by this circle belong to class “○.”

3. The Proposed Models and Algorithm

As the classification rules directly affect the rates of accuracy, support, and compactness, we formulate two models to determine the highest accuracy rate and support rate, respectively. To facilitate the discussion, the related notations are introduced first:

- \( a_{i,j} \): \( j \)'th attribute value of the \( i \)'th object,
- \( h_{t,k,j} \): \( j \)'th center value of the \( k \)'th hypersphere for class \( t \),
- \( r_{t,k} \): radius of the \( k \)'th hypersphere for class \( t \),
- \( n(t) \): number of objects for class \( t \),
- \( c_i \): \( i \)'th object belonging to class \( c_i \in \{1,2,\ldots,g\} \),
- \( m \): number of attributes,
- \( R_t \): a rule describing class \( t \).

Based on these notations, we propose two types of classification models as follow.
3.1. Two Types of Classification Models

Considering the object \( x_i \) and hypersphere \( S_{t,k} \), and normalizing \( a_{i,j} \) (i.e., to express its scale easily), we then have the following three notations.

Notation 1. Normalization rescales all \( a_{i,j} \) as \( a'_{i,j} \). The following is a normalizing formula:

\[
a'_{i,j} = \frac{a_{i,j} - a_j}{a_j - a_j},
\]

where \( 0 \leq a'_{i,j} \leq 1 \), \( a_j \) is the largest value of attribute \( j \), and \( a_j \) is the smallest value of attribute \( j \).

Notation 2. A general form for expressing an object \( x_i \) is written as

\[
x_i = (a'_{i,1}, a'_{i,2}, \ldots, a'_{i,m}; c_i),
\]

where \( c_i \) is the class index of object \( x_i \).

Notation 3. A general form for expressing a hypersphere \( S_{t,k} \) is written as

\[
S_{t,k} = (h_{t,k,1}, h_{t,k,2}, \ldots, h_{t,k,m}; r_{t,k}),
\]

where \( S_{t,k} \) is the \( k' \) th hypersphere for class \( t \).

We use two and three dimensions (i.e., two attributes and three attributes) as visualizations to depict clearly a circle and a sphere, respectively (Figure 4). Figure 4(a) denotes the centroid of the circle as \( (h_{t,k,1}, h_{t,k,2}) \) and the radius of the circle as \( r_{t,k} \). They are extended to three dimensions called sphere (Figure 4(b)); in \( m \) dimensions (i.e., \( m \) attributes), \( m > 3 \), which are then called hyperspheres.

To find each center and the radius of the hypersphere, the following two nonlinear models are considered. The first model looks for a support rate as high as possible while the accuracy rate is fixed to 1, as shown in Model 1.
Model 1. One has the following:

Maximize $\sum_{i \in I} u_{i,j,k}$ 

subject to $\sum_{j=1}^{m} \left( \left( a'_{i,j} - h_{t,k,j} \right)^2 \right) \leq \left( 1 - u_{i,j,k} \right) M + r^2_{i,j,k}, \quad \forall i \in I^+$, \hspace{1cm} (3.4)

$\sum_{j=1}^{m} \left( \left( a'_{i',j} - h_{t,k,j} \right)^2 \right) > r^2_{i,k}, \quad \forall i' \in I^-$,

$u_{i,j,k} \in \{0, 1\} \quad \forall i \in I^+, r_{i,k} \geq 0$, and $M$ is big enough constant,

where $I^+$ and $I^-$ are the two sets for all objects expressed, respectively, by

$I^+ = \{ i \mid i = 1, 2, \ldots, n, \text{where object } i \in \text{class } t \}$, \hspace{1cm} (3.5)

$I^- = \{ i' \mid i' = 1, 2, \ldots, n, \text{where object } i' \notin \text{class } t \}$. \hspace{1cm} (3.6)

Referring to Li and Chen [3], the rates of accuracy and support of $R_t$ in Model 1 can be specified by the following definitions.

Definition 3.1. The accuracy rate of a rule $R_t$ for Model 1 is $AR(R_t) = 1$.

Definition 3.2. The support rate of a rule $R_t$ for Model 1 is specified as follows.

(i) If $\sum_{k \in K} u_{i,j,k} \geq 1$ for all $i$ belonging to class $t$, then $U_{i,j} = 1$; otherwise $U_{i,j} = 0$, where $K$ indicates the hypersphere set for class $t$.

(ii)

$$SR(R_t) = \frac{\sum_{i \in \text{class } t} U_{i,j}}{n(t)},$$ \hspace{1cm} (3.7)

where $n(t)$ indicates the number of objects belonging to class $t$.

The second model looks for an accuracy rate as high as possible while the support rate is fixed to 1, as shown in Model 2.

Model 2. One has the following:

Maximize $\sum_{i \in I^-} v_{i,j',k}$

subject to $\sum_{j=1}^{m} \left( \left( a'_{i,j} - h_{t,k,j} \right)^2 \right) \leq r^2_{i,j,k}, \quad \forall i \in I^+$,
\[
\sum_{j=1}^{m} \left( d_{i,j}^{t} - h_{t,k,j} \right)^{2} > (v_{t,i,k} - 1)M + r_{t,k}^{2}, \quad \forall i' \in I^-, \\
v_{t,i,k} \in \{0, 1\}, \quad \forall i' \in I^-, \quad r_{t,k} \geq 0,
\]

(3.8)

where \(I^+\) and \(I^-\) are the two sets expressed by (3.5) and (3.6), respectively.

Similarly, the rates of accuracy and support of \(R_{t}\) in Model 2 can be considered as follows.

**Definition 3.3.** The accuracy rate of a rule \(R_{t}\) of Model 2 is denoted as \(AR(R_{t})\) and is specified as follows.

1. If \(\sum_{k \in K} v_{t,i,k} = 0\) belongs to class \(t\), then \(V_{t,i} = 1\) for all \(i\); otherwise, \(V_{t,i} = 0\), where \(K\) represents the hypersphere set for class \(t\).
2. \[\text{AR}(R_{t}) = \frac{\|R_{t}\| - \sum_{i \in \text{class } t} V_{t,i}}{\|R_{t}\|},\] (3.9)

where \(\|R_{t}\|\) represents the number of total objects covered by \(R_{t}\).

**Definition 3.4.** The support rate of a rule \(R_{t}\) of Model 2 is denoted as \(SR(R_{t})\), and \(SR(R_{t}) = 1\).

**Definition 3.5.** The compactness rate of a set of rules \(R_{1}, \ldots, R_{g}\), denoted as \(CR(R_{1}, \ldots, R_{g})\), is expressed as follows:

\[CR(R_{1}, \ldots, R_{g}) = \frac{\sum_{t=1}^{g} US_{t}}{US_{t}},\] (3.10)

where \(US_{t}\) means the number of hyperspheres and unions of hyperspheres for class \(t\). A union of hyperspheres indicates that the object is covered by different hyperspheres, as shown in Figure 5. Take Figure 5 for an example, in which there are two classes. The objects of class “○” are covered by two unions of the circles (i.e., \(S_{1,1} \cup S_{1,2} \cup S_{1,3}\) and \(S_{1,4} \cup S_{1,5}\)), and the objects of class “•” are covered by one circle (i.e., \(S_{2,1}\)). Therefore, \(US_{1} = 2, US_{2} = 1,\) and \(CR(R_{1}, R_{2}) = 2/3\).

Moreover, Models 1 and 2 are separable nonlinear programs solvable to find an optimal solution by linearizing the quadratic terms \(h_{t,k,j}^{2}\). The piecewise linearization technique is discussed as follows.
Proposition 3.6 (referring to Beale and Forrest [31]). Denote approximate function \( L(f(x)) \) as a piecewise linear function (i.e., linear convex combination) of \( f(x) \), where \( b_l, l = 1, 2, \ldots, q \) represents the break points of \( L(f(x)) \). \( L(f(x)) \) is expressed as follows:

\[
f(x) \equiv L(f(x)) = \sum_{l=1}^{q} f(b_l)w_l,
\]

\[
x = \sum_{l=1}^{q} w_l b_l,
\]

\[
\sum_{l=1}^{q} w_l = 1,
\]

where \( w_l \geq 0 \), and (3.13) is a special-ordered set of type 2 (SOS2) constraint (reference to Beale and Forrest [31]).

Note that the SOS2 constraint is a set of variables in which at most two variables may be nonzero. If two variables are nonzero, they must be adjacent in the set.

Notation 4. According to Proposition 3.6, let \( f(x) = h_{i,k,j}^{2} \). \( f(x) \) is linearized by the Proposition 3.6 and is expressed as \( L(h_{i,k,j}^{2}) \).

### 3.2. Solution Algorithm

A proposed algorithm is also presented to seek the highest accuracy rate or the highest support rate, as described as follows.

Algorithm 3.7.

Step 1. Normalize all attributes (i.e., rescale \( a'_{i,j} = (a_{i,j} - a_j) / (\overline{a_j} - a_j) \) to be \( 0 \leq a'_{i,j} \leq 1 \)).

Step 2. Initialization: \( t = 1 \) and \( k = 1 \).
Step 3. Solve Model 1 (or Model 2) to obtain the $k'$th hypersphere of class $t$. Remove the objects covered by $S_{t,k}$ from the dataset temporarily.

Step 4. Let $k = k + 1$, and resolve Model 1 (or Model 2) until all objects in class $t$ are assigned to the hyperspheres of same class.

Step 5. Let $k = 1$ and $t = t + 1$, and reiterate Step 3 until all classes are processed.

Step 6. Check the independent hyperspheres and unions of hyperspheres $S_{t,k}$ in the same class $t$. 

Figure 6: Flowchart of the proposed algorithm.
### Table 1: Dataset of Example 1.

<table>
<thead>
<tr>
<th>Object</th>
<th>$a_{i,1}$</th>
<th>$a_{i,2}$</th>
<th>$c_i$</th>
<th>Object</th>
<th>$a_{i,1}$</th>
<th>$a_{i,2}$</th>
<th>$c_i$</th>
<th>Object</th>
<th>$a_{i,1}$</th>
<th>$a_{i,2}$</th>
<th>$c_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td>$x_2$</td>
<td>12</td>
<td>20</td>
<td>1</td>
<td>$x_3$</td>
<td>13</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>$x_4$</td>
<td>18</td>
<td>12.5</td>
<td>1</td>
<td>$x_5$</td>
<td>24</td>
<td>19</td>
<td>1</td>
<td>$x_6$</td>
<td>24</td>
<td>14.5</td>
<td>1</td>
</tr>
<tr>
<td>$x_7$</td>
<td>17.5</td>
<td>17.5</td>
<td>2</td>
<td>$x_8$</td>
<td>22</td>
<td>17</td>
<td>2</td>
<td>$x_9$</td>
<td>22</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>30</td>
<td>11</td>
<td>2</td>
<td>$x_{11}$</td>
<td>33.5</td>
<td>7.5</td>
<td>2</td>
<td>$x_{12}$</td>
<td>24.5</td>
<td>3.5</td>
<td>3</td>
</tr>
<tr>
<td>$x_{13}$</td>
<td>26.5</td>
<td>8</td>
<td>3</td>
<td>$x_{14}$</td>
<td>23.5</td>
<td>7.5</td>
<td>3</td>
<td>$x_{15}$</td>
<td>6</td>
<td>30</td>
<td>3</td>
</tr>
</tbody>
</table>

(Symbols of $c_i$: 1: $\bigcirc$, 2: $\triangle$, 3: $\times$.

---

**Step 7.** Calculate and record the number of independent hyperspheres and unions of hyperspheres in $US_t$, and iterate $t$ until all classes are done.

According to this algorithm, we can obtain the optimal rules to classify objects most efficiently. The process of the algorithm is depicted in Figure 6.

### 3.3. Operation of a Simple Example

Consider a dataset $T$ in Table 1 as an example, which has object $i$, two attributes ($a_{i,1}, a_{i,2}$), and an index of classes ($c_i$) for $i = 1, 2, \ldots, 15$. The dataset $T$ is expressed as $T = \{x_i \mid (a_{i,1}, a_{i,2}; c_i) \forall i = 1, 2, \ldots, 15\}$. There are the domain values of $c_i \in \{1, 2, 3\}$. As there are only two attributes, these 15 objects can be plotted on a two-dimensional space after normalizing them, as shown in Figure 7(a).

This example can be solved by the proposed algorithm as follows.

**Step 1.** Normalize all attributes (i.e., $a'_{i,1} = (a_{i,1} - 6)/(33.5 - 6)$ and $a'_{i,2} = (a_{i,2} - 3.5)/(30 - 3.5)$).

**Step 2.** Initialization: $t = 1$ and $k = 1$. 

---

**Figure 7:** Visualization of Example 1.
Step 3. The classification model (i.e., Model 1) is linearly formulated as follows:

\[
\begin{align*}
\text{Maximize} & \quad \sum_{i \in I^+} u_{i,j,k} \\
\text{subject to} & \quad \sum_{j=1}^{m} \left( a_{i,j}^2 - 2a_{i,j} h_{i,k,j} + L(h_{i,k,j}^2) \right) \leq (1 - u_{i,j,k})M + r_{i,k}^2, \quad \forall i \in I^+, \\
& \quad \sum_{j=1}^{m} \left( a_{i,j}^2 - 2a_{i,j} h_{i,k,j} + L(h_{i,k,j}^2) \right) > r_{i,k}^2, \quad \forall i' \in I^-,
\end{align*}
\]

where \( I^+ = \{x_1, x_2, \ldots, x_6\} \) and \( I^- = \{x_7, x_8, \ldots, x_{15}\} \). The optimal solution of the \((h_{1,k,1}, h_{1,k,2}, r_{1,k}) = (0.047, 0.265, 0.15749)\) for \( S_{1,1} \), where \( S_{1,1} \) covers objects 1–4. We then temporarily remove these objects covered by \( S_{1,1} \).

Step 4. \( k = k + 1 \): the optimal solution of the \((h_{1,k,1}, h_{1,k,2}, r_{1,k}) = (0.736, 0.5, 0.0138)\) for \( S_{1,2} \), where \( S_{1,2} \) covers objects 5–6. Class 1 is then done.

Step 5. As \( t = t + 1, k = 1 \), and Steps 3 and 4 are iterated, we then, respectively, have optimal solutions for \( S_{t,k} \) as follows. The results are shown in Figure 7(b).

(i) \((h_{2,1,1}, h_{2,1,2}, r_{2,1}) = (0.514, 0.469, 0.0127)\), where \( S_{2,1} \) covers objects 7–9.

(ii) \((h_{2,2,1}, h_{2,2,2}, r_{2,2}) = (0.929, 0.210, 0.0251)\), where \( S_{2,2} \) covers objects 10-11.

(iii) \((h_{3,1,1}, h_{3,1,2}, r_{3,1}) = (0.583, 0.188, 0.0436)\), where \( S_{3,1} \) covers objects 12–14.

Step 6. Check and calculate the unions of hypersphere \( S_{t,k} \) for all \( k \) in class \( t \) (i.e., Initial \( t = 1 \)).

Step 7. As \( t = t + 1 \), mark the number of unions of class \( t \) into \( US_t \) and iterate Step 6 until \( t = g \).
4. Numerical Examples

This study shows how the experimental results evaluate the performance, including accuracy, support, and compactness rates, and compares the proposed model with different methods using CPLEX [32]. All tests were run on a PC equipped with an Intel Pentium D 2.8 GHz CPU and 2 GB RAM. Three datasets were tested in our experiments as follows:

- (i) Iris Flower dataset introduced by Sir Ronald Aylmer Fisher (1936),
- (ii) European barn swallow (Hirundo rustica) dataset obtained by trapping individual swallows in Stirlingshire, Scotland, between May and July 1997 [1, 3],
- (iii) the highly selective vagotomy (HSV) patient dataset of F. Raszeja Memorial Hospital in Poland [3, 33, 34].

4.1. Iris Flower Dataset

The Iris Flower dataset contains 150 objects. Each object is described by four attributes (i.e., sepal length, sepal width, petal length, and petal width) and is classified by one of three classes (i.e., setosa, versicolor, and virginica). By solving the proposed method, we induced six hyperspheres (i.e., \( S_{1,1} \in \text{Class 1}, S_{2,1}, S_{2,2}, S_{2,3} \in \text{Class 2}, \) and \( S_{3,1}, S_{3,2} \in \text{Class 3} \)). The induced classification rules are reported in Table 2. Table 2 also lists a hypersphere and two unions (i.e., \( S_{1,1}, S_{2,1} \cup S_{2,2} \cup S_{2,3}, \) and \( S_{3,1} \cup S_{3,2} \)) of hyperspheres with centroid points and radii.

Rule \( R_1 \) in Table 2 contains a hypersphere \( S_{1,1} \), which implies that

(i) “if \((a_{i,1}' - 0.366)^2 + (a_{i,2}' - 0.807)^2 + (a_{i,3}' - 0)^2 + (a_{i,4}' - 0)^2 \leq 0.6205\), then object \( x_i \) belongs to class 1.”

Rule \( R_2 \) in Table 2 contains a union of three hyperspheres (i.e., \( S_{2,1} \cup S_{2,2} \cup S_{2,3} \)) which implies that

(i) “if \((a_{i,1}' - 0.557)^2 + (a_{i,2}' - 0.32)^2 + (a_{i,3}' - 0.205)^2 + (a_{i,4}' - 0.46)^2 \leq 0.254\), then object \( x_i \) belongs to class 2,” or

(ii) “if \((a_{i,1}' - 0.575)^2 + (a_{i,2}' - 0.581)^2 + (a_{i,3}' - 0.626)^2 + (a_{i,4}' - 0.515)^2 \leq 0.0612\), then object \( x_i \) belongs to class 2,” or

(iii) “if \((a_{i,1}' - 0.423)^2 + (a_{i,2}' - 0.261)^2 + (a_{i,3}' - 0.352)^2 + (a_{i,4}' - 0.49)^2 \leq 0.1388\), then object \( x_i \) belongs to class 2.”

Rule \( R_3 \) in Table 2 contains a union of two hyperspheres (i.e., \( S_{3,1} \cup S_{3,2} \)), which implies that

(i) “if \((a_{i,1}' - 0.248)^2 + (a_{i,2}' - 0)^2 + (a_{i,3}' - 2.226)^2 + (a_{i,4}' - 2.151)^2 \leq 4.8087\), then object \( x_i \) belongs to class 3,” or

(ii) “if \((a_{i,1}' - 0.329)^2 + (a_{i,2}' - 0.187)^2 + (a_{i,3}' - 0.65)^2 + (a_{i,4}' - 0.613)^2 \leq 0.033\), then object \( x_i \) belongs to class 3.”

Comparing the proposed method with both decision tree [3] and hyperplane methods [35] in deducing the classification rules for the Iris Flower dataset, Table 3 lists the experimental result.
Each object is described by eight attributes, and it belongs to one of two classes. The European barn swallow \( (Hirundo rustica) \) dataset was obtained by trapping individual swallows in Stirlingshire, Scotland, between May and July 1997. This dataset contains 69 swallows. Each object is described by eight attributes, and it belongs to one of two classes (i.e., the birds are classified by the gender of individual birds).

Here, we also used Model 1 to induce the classification rules. Table 4 lists the optimal solutions (i.e., centroid and radius) for both rules \( R_1 \) and \( R_2 \).

The result of the decision tree method, which is referred to in Li and Chen [3], is listed in Table 5, where \( AR(R_1, R_2) = (0.97, 1) \), \( SR(R_1, R_2) = (0.97, 1) \), and \( CR = 0.3 \).

### 4.2. Swallow Dataset

The European barn swallow \( (Hirundo rustica) \) dataset was obtained by trapping individual swallows in Stirlingshire, Scotland, between May and July 1997. This dataset contains 69 swallows. Each object is described by eight attributes, and it belongs to one of two classes (i.e., the birds are classified by the gender of individual birds).

<table>
<thead>
<tr>
<th>Rule number</th>
<th>Union of spheres</th>
<th>( S_{1,k} )</th>
<th>( h_{1,k,1} )</th>
<th>( h_{1,k,2} )</th>
<th>( h_{1,k,3} )</th>
<th>( h_{1,k,4} )</th>
<th>( h_{1,k,5} )</th>
<th>( h_{1,k,6} )</th>
<th>( h_{1,k,7} )</th>
<th>( h_{1,k,8} )</th>
<th>( r_{1,k} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_1 )</td>
<td>( S_{1,1} \cup S_{1,2} \cup S_{1,3} \cup S_{1,4} )</td>
<td>( S_{1,1} )</td>
<td>0.607</td>
<td>0.110</td>
<td>0.806</td>
<td>0.000</td>
<td>0.000</td>
<td>0.406</td>
<td>1.077</td>
<td>1.163</td>
<td>1.780</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{1,2} )</td>
<td>0.483</td>
<td>0.000</td>
<td>0.236</td>
<td>0.328</td>
<td>0.000</td>
<td>0.793</td>
<td>0.280</td>
<td>0.931</td>
<td>0.948</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{1,3} )</td>
<td>0.588</td>
<td>0.000</td>
<td>1.179</td>
<td>0.000</td>
<td>0.000</td>
<td>0.653</td>
<td>1.037</td>
<td>0.982</td>
<td>1.879</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{1,4} )</td>
<td>0.414</td>
<td>0.000</td>
<td>0.658</td>
<td>0.000</td>
<td>0.227</td>
<td>0.085</td>
<td>0.563</td>
<td>1.224</td>
<td>1.197</td>
</tr>
<tr>
<td>( R_2 )</td>
<td>( S_{2,1} \cup S_{2,2} \cup S_{2,3} \cup S_{2,4} \cup S_{2,5} )</td>
<td>( S_{2,1} )</td>
<td>0.358</td>
<td>0.667</td>
<td>0.371</td>
<td>1.403</td>
<td>0.360</td>
<td>1.530</td>
<td>0.634</td>
<td>0.114</td>
<td>2.825</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{2,2} )</td>
<td>0.198</td>
<td>1.102</td>
<td>0.697</td>
<td>0.000</td>
<td>0.000</td>
<td>1.734</td>
<td>0.110</td>
<td>0.595</td>
<td>2.632</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{2,3} )</td>
<td>0.532</td>
<td>0.422</td>
<td>0.323</td>
<td>1.676</td>
<td>1.009</td>
<td>0.000</td>
<td>0.180</td>
<td>0.020</td>
<td>2.458</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{2,4} )</td>
<td>0.528</td>
<td>0.000</td>
<td>0.430</td>
<td>0.876</td>
<td>0.579</td>
<td>0.461</td>
<td>0.435</td>
<td>0.040</td>
<td>0.694</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{2,5} )</td>
<td>0.659</td>
<td>1.408</td>
<td>0.516</td>
<td>0.000</td>
<td>0.382</td>
<td>0.000</td>
<td>0.173</td>
<td>0.064</td>
<td>1.570</td>
</tr>
</tbody>
</table>

The accuracy rates of \( (R_1, R_2, R_3) \) in the proposed method are \((1,1,1)\), as Model 1 has been solved. This finding indicates that none of objects in class 2 or class 3 are covered by \( S_{1,1} \), none of objects in classes 1 or 3 are covered by \( S_{2,1} \cup S_{2,2} \cup S_{2,3} \), and none of the objects in classes 1 or 2 are covered by \( S_{3,1} \cup S_{3,2} \). The support rate of \( (R_1, R_2, R_3) \) in the proposed method is \((1,0.98,0.98)\), indicating that all objects in class 1 are covered by \( S_{1,1} \), 98% of the objects in class 2 are covered by \( S_{2,1}, S_{2,2}, \) and \( S_{2,3} \), and 98% of the objects in class 3 are covered by \( S_{3,1} \) and \( S_{3,2} \). The compactness rate of rules \( R_1, R_2, \) and \( R_3 \) is computed as \( CR(R_1, R_2, R_3) = 3/3 = 1 \). Finally, we determine the following.

(i) Although all three methods perform very well in the rates of accuracy and support, the proposed method has the best performance for the accuracy of classes 2 and 3 (i.e., \( R_2 \) and \( R_3 \)).

(ii) The proposed method has the best compactness rate.

### Table 4: Centroid points for the Swallow data set by the proposed method.

<table>
<thead>
<tr>
<th>Items</th>
<th>Proposed method</th>
<th>Decision tree</th>
<th>Hyperplane support vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>( AR(R_1, R_2) )</td>
<td>((1, 1))</td>
<td>((0.97, 1))</td>
<td>((0.97, 1))</td>
</tr>
<tr>
<td>( SR(R_1, R_2) )</td>
<td>((1, 0.97))</td>
<td>((0.97, 1))</td>
<td>((0.97, 1))</td>
</tr>
<tr>
<td>( CR )</td>
<td>1</td>
<td>0.3</td>
<td>0.1</td>
</tr>
</tbody>
</table>
The result of the hyperplane method, referred to in Chang and Lin [35], is also listed in Table 5, where $AR(R_1, R_2) = (0.97, 1)$, $SR(R_1, R_2) = (0.97, 1)$, and $CR = 0.1$.

We compared the three methods in Table 5 to show that the proposed method can induce rules with better or equivalent values of $AR$ and $SR$. In fact, the proposed method also has the best compactness rate.

4.3. HSV Dataset

The HSV dataset contains 122 patients classified into four classes, with each patient having 11 preoperating attributes. To maximize the support rate with respect to the proposed method (i.e., Model 1), the proposed method generated seven hyperspheres and three unions of hyperspheres. The centroids and radii of the hyperspheres are reported in Table 6, and a comparison with other methods is reported in Table 7.

Using the decision tree method in the HSV dataset generates 24 rules. In addition, the hyperplane method deduces 45 hyperplanes for the HSV dataset. Table 7 also shows that the proposed method can find rules with the highest rates (i.e., $AR$, $SR$, and $CR$) compared with the other two methods.

4.4. Limitation of the Proposed Method

The hypersphere models are solved by one of the most powerful mixed-integer program software CPLEX [32] running in a PC. Based on optimization technique, the results of the numerical examples illustrate that the usefulness of the proposed method is better than that of the current methods, including the decision tree method and the hyperplane support vector method. As the solving time of the hypersphere model, which is linearized, mainly depends on the number of binary variables and constraints, solving the reformulated hypersphere model from the proposed algorithm takes about one minute for each dataset (i.e., in Sections 4.1 and 4.3), in which using eight piecewise line segments linearizes the nonlinear nonconvex term (i.e., $L(h^2_{i,k,j})$) of Model 1.

The computing time for solving a linearized hypersphere program grows rapidly as the numbers of binary variables and constraints increase. Also, the computing time of the proposed method is slower than that of the decision tree method and hyperplane method, especially for large datasets or a great number of piecewise line segments. In the further study, utilizing a mainframe-version optimization software [36–38], integrating meta-heuristic algorithms, or using distributed computing techniques can enhance solving speed to conquer this problem.

5. Conclusions and Future Work

This study proposes a novel method for deducing classification rules, which can find the optimal solution based on a hypersphere domain. The optimization technique for finding classification rules is approached to optimal. Results of the numerical examples illustrate that the usefulness of the proposed method is better than that of the current methods, including the decision tree method and the hyperplane method. The proposed method is guaranteed to find an optimal rule, but the computational complexity grows rapidly by increasing the problem size. More investigation and research are required to enhance further.
Table 6: Centroid points for the HSV data by the proposed method.

<table>
<thead>
<tr>
<th>Rule numbers</th>
<th>Union of spheres</th>
<th>$S_{1,1}$</th>
<th>$h_{1,1}$</th>
<th>$h_{1,2}$</th>
<th>$h_{1,3}$</th>
<th>$h_{1,4}$</th>
<th>$h_{1,5}$</th>
<th>$h_{1,6}$</th>
<th>$h_{1,7}$</th>
<th>$h_{1,8}$</th>
<th>$h_{1,9}$</th>
<th>$h_{1,10}$</th>
<th>$h_{1,11}$</th>
<th>$r_{1,k}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>$S_{1,1}$</td>
<td>0.504</td>
<td>0.528</td>
<td>0.366</td>
<td>0.288</td>
<td>0.850</td>
<td>0.848</td>
<td>0.351</td>
<td>−0.282</td>
<td>0.405</td>
<td>0.234</td>
<td>0.313</td>
<td>1.576</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1,2}$</td>
<td>0.469</td>
<td>0.395</td>
<td>0.312</td>
<td>0.133</td>
<td>−0.274</td>
<td>−0.309</td>
<td>0.134</td>
<td>0.590</td>
<td>0.813</td>
<td>1.000</td>
<td>−0.847</td>
<td>3.044</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1,3}$</td>
<td>0.287</td>
<td>0.867</td>
<td>0.458</td>
<td>0.189</td>
<td>0.977</td>
<td>0.267</td>
<td>0.017</td>
<td>−0.846</td>
<td>0.462</td>
<td>−1.000</td>
<td>1.000</td>
<td>4.249</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1,4}$</td>
<td>0.586</td>
<td>−0.400</td>
<td>0.605</td>
<td>0.467</td>
<td>−0.167</td>
<td>−0.631</td>
<td>0.268</td>
<td>1.000</td>
<td>0.526</td>
<td>1.000</td>
<td>0.120</td>
<td>3.365</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1,5}$</td>
<td>0.296</td>
<td>−0.511</td>
<td>−1.000</td>
<td>0.536</td>
<td>−0.263</td>
<td>0.552</td>
<td>0.954</td>
<td>0.635</td>
<td>−0.117</td>
<td>−1.000</td>
<td>0.600</td>
<td>4.608</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1,6}$</td>
<td>0.775</td>
<td>0.678</td>
<td>−0.092</td>
<td>0.340</td>
<td>0.366</td>
<td>0.455</td>
<td>0.451</td>
<td>1.000</td>
<td>0.422</td>
<td>0.295</td>
<td>0.470</td>
<td>1.560</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1,7}$</td>
<td>0.525</td>
<td>0.296</td>
<td>0.194</td>
<td>0.124</td>
<td>−0.945</td>
<td>−0.467</td>
<td>0.127</td>
<td>1.000</td>
<td>0.097</td>
<td>−0.897</td>
<td>0.654</td>
<td>4.113</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1,8}$</td>
<td>0.000</td>
<td>0.140</td>
<td>0.109</td>
<td>0.250</td>
<td>0.193</td>
<td>0.168</td>
<td>0.063</td>
<td>0.142</td>
<td>0.181</td>
<td>0.119</td>
<td>0.082</td>
<td>0.089</td>
<td></td>
</tr>
<tr>
<td>$R_2$</td>
<td>$S_{2,1}$</td>
<td>0.513</td>
<td>0.056</td>
<td>0.143</td>
<td>0.078</td>
<td>0.533</td>
<td>0.439</td>
<td>0.180</td>
<td>0.167</td>
<td>−0.284</td>
<td>−0.464</td>
<td>0.749</td>
<td>1.450</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{2,2}$</td>
<td>0.533</td>
<td>0.381</td>
<td>0.139</td>
<td>0.568</td>
<td>−0.373</td>
<td>−0.005</td>
<td>0.080</td>
<td>0.831</td>
<td>0.293</td>
<td>0.215</td>
<td>0.369</td>
<td>1.293</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{2,3}$</td>
<td>0.000</td>
<td>−0.450</td>
<td>0.709</td>
<td>0.553</td>
<td>0.896</td>
<td>1.000</td>
<td>0.157</td>
<td>−0.447</td>
<td>0.068</td>
<td>−0.580</td>
<td>1.000</td>
<td>3.579</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{2,4}$</td>
<td>0.000</td>
<td>0.862</td>
<td>0.543</td>
<td>−0.066</td>
<td>0.389</td>
<td>0.451</td>
<td>0.106</td>
<td>0.394</td>
<td>−0.004</td>
<td>0.042</td>
<td>−0.014</td>
<td>0.049</td>
<td></td>
</tr>
<tr>
<td>$R_3$</td>
<td>$S_{3,1}$</td>
<td>0.624</td>
<td>0.507</td>
<td>−1.000</td>
<td>0.147</td>
<td>0.840</td>
<td>1.000</td>
<td>0.827</td>
<td>−0.585</td>
<td>0.199</td>
<td>0.831</td>
<td>−0.380</td>
<td>4.277</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{3,2}$</td>
<td>0.534</td>
<td>0.483</td>
<td>−0.088</td>
<td>0.365</td>
<td>−0.364</td>
<td>0.270</td>
<td>0.437</td>
<td>0.785</td>
<td>0.786</td>
<td>0.682</td>
<td>−1.000</td>
<td>3.003</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{3,3}$</td>
<td>0.000</td>
<td>0.210</td>
<td>0.630</td>
<td>0.750</td>
<td>−1.000</td>
<td>0.244</td>
<td>0.388</td>
<td>1.000</td>
<td>0.474</td>
<td>−0.604</td>
<td>0.475</td>
<td>3.162</td>
<td></td>
</tr>
<tr>
<td>$R_4$</td>
<td>$S_{4,1}$</td>
<td>0.551</td>
<td>0.374</td>
<td>0.637</td>
<td>0.256</td>
<td>0.865</td>
<td>0.944</td>
<td>0.315</td>
<td>−0.831</td>
<td>−0.485</td>
<td>−0.676</td>
<td>0.979</td>
<td>4.330</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{4,2}$</td>
<td>0.717</td>
<td>0.254</td>
<td>0.548</td>
<td>−0.718</td>
<td>0.118</td>
<td>0.730</td>
<td>−0.547</td>
<td>0.498</td>
<td>−0.580</td>
<td>−0.464</td>
<td>0.821</td>
<td>3.287</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{4,3}$</td>
<td>0.527</td>
<td>1.000</td>
<td>0.533</td>
<td>0.089</td>
<td>−0.152</td>
<td>−1.000</td>
<td>0.366</td>
<td>0.046</td>
<td>0.547</td>
<td>0.947</td>
<td>−0.436</td>
<td>2.943</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{4,4}$</td>
<td>0.489</td>
<td>0.625</td>
<td>−0.209</td>
<td>0.522</td>
<td>0.644</td>
<td>0.582</td>
<td>0.615</td>
<td>1.000</td>
<td>0.306</td>
<td>−0.401</td>
<td>0.931</td>
<td>2.059</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{4,5}$</td>
<td>−0.011</td>
<td>0.491</td>
<td>−0.409</td>
<td>0.155</td>
<td>−0.766</td>
<td>−0.227</td>
<td>−0.107</td>
<td>1.000</td>
<td>0.303</td>
<td>−0.680</td>
<td>−0.185</td>
<td>2.843</td>
<td></td>
</tr>
</tbody>
</table>
the computational efficiency of globally solving large-scale classification problems, such as running mainframe-version optimization software, integrating meta-heuristic algorithms, or using distributed computing techniques.

**Acknowledgments**

The authors wish to thank the editor and the anonymous referees for providing insightful comments and suggestions, which have helped them improving the quality of the paper. This work was supported by the National Science Council of Taiwan under Grants NSC 100-2811-E-009-040-, NSC 99-2221-E-030-005-, and NSC 100-2221-E-030-009-.

**References**


