

A Mixed-Integer Programming Approach to Multi-Class Data Classification

Problem

Fadime Üney and Metin Türkay*

funey@ku.edu.tr, mturkay@ku.edu.tr

*Department of Industrial Engineering, College of Engineering
Koç University, Rumelifeneri Yolu, 34450, Sariyer, Istanbul, Turkey*

Abstract

This paper presents a new data classification method based on mixed-integer programming. Traditional approaches that are based on partitioning the data sets into two groups perform poorly for multi-class data classification problems. The proposed approach is based on the use of hyper-boxes for defining boundaries of the classes that include all or some of the points in that set. A mixed-integer programming model is developed for representing existence of hyper-boxes and their boundaries. In addition, the relationships among the discrete decisions in the model are represented using propositional logic and then converted to their equivalent integer constraints using Boolean algebra. The proposed approach for multi-class data classification is illustrated on an example problem. The efficiency of the proposed method is tested on the well-known IRIS data set. The computational results on the illustrative example and the IRIS data set show that the proposed method is very accurate and efficient on multi-class data classification problems.

Keywords: Data Mining, Data Classification, Mixed-Integer Programming,
Boolean Algebra

* Corresponding author: Phone:+90-212-338-1586; Fax:+90-212-338-1548; E-mail: mturkay@ku.edu.tr

1. Introduction

Classification is a supervised learning strategy which emphasizes on building models able to assign new instances to one of a set of well-defined classes. Classification problems have been intensively studied by a diverse group of researchers including statisticians, engineers, biologists, computer scientists. There is variety of methods for solving classification problem in different disciplines. Some of these method include neural networks (NN), fuzzy logic, support vector machines (SVM), tolerant rough sets, principal component analysis (PCA), linear programming (Adem and Gochet, 2004).

Carpenter and Grossberg (1987) developed fuzzy adaptive resonance theory (ART), fast and reliable analog pattern clustering system. Lin and Lee (1991) carried out a general neural-network model for fuzzy logic control and decision systems. Furthermore, Simpson (1992) developed a fuzzy min-max classification neural network in which pattern classes are utilized as fuzzy sets. Rough set theory introduced by Pawlak (1982) is a mathematical tool to deal with vagueness and uncertainty in the areas of machine learning and pattern recognition. Banzan et al. (1994) proposed two applications of logic for classification by using rough set approach. Nguyen et al. (1998) used tolerance relation among the objects for pattern classification. Kim (2001) proposed a new data classification method based on the tolerant rough set that combines these two approaches. Chang et al. (2004) developed a novel filter-based greedy modular subspace technique using principal component analysis. Anthony (2004) analyzed theoretically the generalization properties of multi-class data classification techniques based on iterative linear partitioning.

In recent years, SVM has been considered as one of the most efficient methods for two-class classification problems (Yajima, 2003). SVM is a new classification technique developed by Vapnik and his group (Cortes and Vapnik, 1995). SVM is able to generate a separating hyper surface in order to maximize the margin and produce good generalization ability. However, the SVM has two important drawbacks. First, a combination of SVMs has to be used in order to solve the multi-class classification problems. Since it is originally a model for binary-class classification, proposed models for combination of SVMs doesn't have improved performance. Second, some approximation algorithms are used in order to reduce the computational time for SVMs while learning the large scale of data. On the other hand, this computational improvement could cause less efficient performance values (Kim et al., 2003). To overcome above problems, many variants of SVM are suggested. Zhan and Shen (2005) have suggested a four-step training method for increasing the efficiency of SVM. Kim et al. (2003) suggested using SVM ensemble with bagging or boosting rather than using a single SVM for a high classification performance.

There have been some attempts to solve classification problems using mathematical programming. Erenguc and Koehler (1990) give a good overview of these methods. Most of these methods modeled data classification as linear programming (LP) problems which optimize a distance function. Contrary to LP problems, MILP problems with minimizing the misclassifications on the design data set as an objective function are studied. There have been several attempts to formulate these problems as a mixed-integer programming problem (Bajgier and Hill, 1982; Gehrlein, 1986; Littschwager and Wang, 1978; Stam, and Joachimsthaler, 1990). Adem and Gochet (2004) proposed a heuristics extension of linear programming approach in order to improve the performance

of multi-class supervised classification. The logical analysis of data (LAD) is a combinatorics, optimization, and Boolean algebra-based methodology for extracting information from data including classification (Boros et al., 1997). This approach is very effective in binary classification; however, the method suffers from accuracy and efficiency when there are more than two classes.

The objective of this paper is to present an accurate and efficient mathematical programming method for multi-class data classification problems. We discuss our approach to multi-class data classification problem in Section 2. The mixed-integer programming model is presented in Section 3. The application of the approach on a sample problem is illustrated in Section 4 and the results for IRIS data set is given in Section 5. Finally, the paper is concluded by presenting the conclusions and discussion of the results.

2. Data Classification Approach

The objective in data classification is to assign data points that are described by several attributes into a predefined number of classes. Fig. 1.a shows the schematic representation of classification of multi-dimensional data using hyper-planes. Although the methods that are based on using hyper-planes to define the boundaries of classes can be efficient in classifying data into two sets, they are inaccurate and inefficient when data needs to be classified into more than two sets as shown in the Fig. 1.a. On the other hand, the use of hyper-boxes for defining boundaries of the sets that include all or some of the points in that set as shown in Fig. 1.b can be very accurate on multi-class problems.

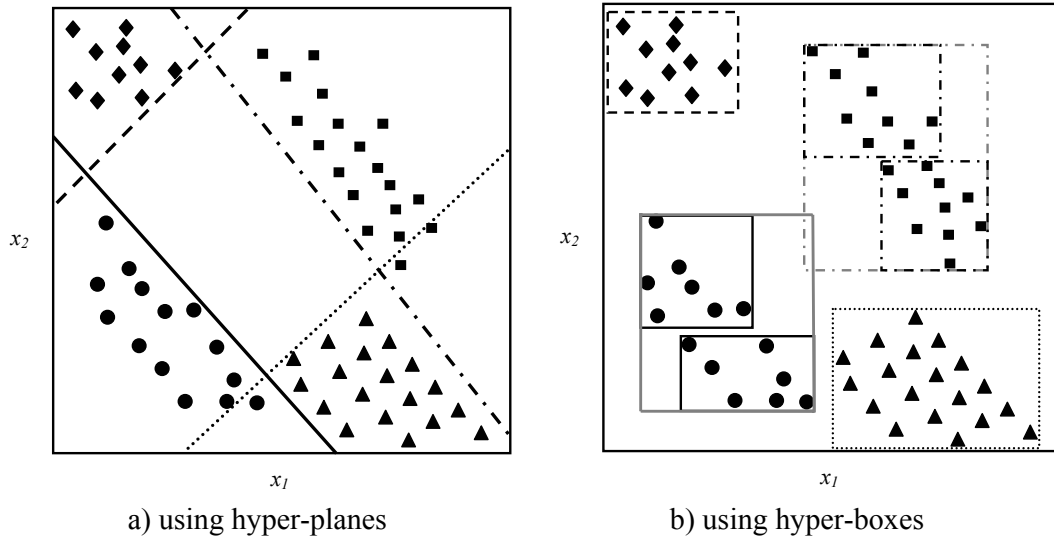


Fig. 1. Schematic representation of classification of data.

It may be necessary to use more than one hyper-box in order to represent a single class as shown in Fig. 1.b. When the classes that are indicated by square and circle data points are both represented by single hyper-box respectively, the boundaries of these hyper-boxes overlap. If a region of the attribute space is assigned to more than one classes, it is possible that a new data point is classified into more than a single class. In order to eliminate this possibility, more than one hyper-box must be used to include all of the data points that belong to a class into the same class. A very important consideration in using hyper-boxes is the number of boxes used to define a class. If the total number of hyper-boxes is equal to the number of classes (i.e., exactly one hyper-box classifies all data points of the same class), then the data classification is very efficient. On the other hand; if there are as many hyper-boxes of a class as the number of data points in a class (i.e., each data point of a particular class is represented by a separate hyper-box), then the data classification is inefficient.

We describe the modeling approach for the proposed method in the following section.

3. Problem Formulation for Multi-Class Data Classification

The data classification problem is considered in two parts: training and testing. The objective of the training part is to determine the characteristics of the data points that belong to a certain class and differentiate them from the data points that belong to other classes. After the distinguishing characteristics of the classes are determined, then the effectiveness of the classification must be tested.

3.1. Training Problem Formulation for Multi-Class Data Classification

The following indices are used to model the training problem of the data classification using hyper-boxes:

- i samples ($i=Sample1, Sample2, \dots, SampleI$)
- k class types ($k=Class1, Class2, \dots, ClassK$)
- l hyper-boxes that encloses a number of data points belonging to a class ($l=1, \dots, L$)
- m attributes ($m=1, \dots, M$)
- n bounds ($n=lo, up$)

The data points are represented by using the parameter a_{im} that denotes the value of attribute m for the sample i . The class k that the data point i belong to are given by the set D_{ik} . Given these parameters and the sets, the following Boolean variables are sufficient to model the data classification problem with hyper-boxes as depicted in Fig. 1b.:

- YB_l Boolean variable to indicate whether the box l is used or not
- YPB_{il} Boolean variable to indicate whether the data point i is in box l or not
- YBC_{lk} Boolean variable to indicate whether box l represent class k or not

YPC_{ik} Boolean variable to indicate whether the data point i is assigned to class k or not

$YPBN_{ilmn}$ Boolean variable to indicate whether the data point i is within the bound n with respect to attribute m of box l or not

$YPBM_{ilm}$ Boolean variable to indicate whether the data point i is within the bounds of attribute m of box l or not

YPI_{ik} type 1 Boolean variable to indicate the misclassification of data points

$YP2_{ik}$ type 2 Boolean variable to indicate the misclassification of data points

The relationships among these Boolean variables can be represented using the following propositional logic (Cavalier et al., 1990; Raman and Grossmann, 1991; Turkay and Grossmann, 1996):

$$\bigvee_l YPB_{il} \quad \forall i \quad (1)$$

$$\bigvee_k YPC_{ik} \quad \forall i \quad (2)$$

$$\bigvee_l YPB_{il} \Leftrightarrow \bigvee_k YPC_{ik} \quad \forall i \quad (3)$$

$$YB_l \Rightarrow \bigvee_k YBC_{lk} \quad \forall l \quad (4)$$

$$YBC_{lk} \Rightarrow \bigvee_i YPB_{il} \quad \forall lk \quad (5)$$

$$YBC_{lk} \Rightarrow \bigvee_i YPC_{ik} \quad \forall lk \quad (6)$$

$$\bigwedge_n YPBN_{ilmn} \Rightarrow YPBM_{ilm} \quad \forall ilm \quad (7)$$

$$\bigwedge_m YPBM_{ilm} \Rightarrow YPC_{ik} \quad \forall ilk \quad (8)$$

$$YPC_{ik} \Rightarrow YPI_{ik} \quad \forall ik \notin D_{ik} \quad (9)$$

$$\neg YPC_{ik} \Rightarrow YP2_{ik} \quad \forall ik \in D_{ik} \quad (10)$$

The Eqs. (1) and (2) states that every data point must be assigned to a single hyper-box and a single class respectively. The equivalence between Eqs. (1) and (2) is given in Eq. (3). The existence of a hyper-box implies the assignment of that hyper-box to a class as shown in Eq. (5). If a hyper-box represents a class, there must be a data point within that class as given in Eq. (6). The Eq. (7) represents the condition of a data point being within the bounds of a box in attribute m . If a data point is within the bounds of all attributes of a box, then it must be in the box as shown in Eq. (8). When a data point is assigned to a class that it is not a member of, type 1 penalty applies as indicated in Eq. (9). When a data point is not assigned to a class that it is a member of, type 2 penalty applies as given in Eq. (10). In mathematical programming applications, there is one-to-one correspondence between a Boolean variable and a binary variable (Cavalier et al., 1990). The value of True for a Boolean variable is equivalent to the value of 1 for the corresponding algebraic variable. The same argument applies between False and 0. In addition, it is possible to obtain exact algebraic equivalent of propositional logic expressions. The correspondence between propositional logic expressions and their equivalent algebraic representations are given in Table 1.

Table 1. The correspondence between propositional logic and algebraic inequalities.

Logical Relation	Boolean Expression	Algebraic Constraint
OR (\vee)	$Y_1 \vee Y_2 \vee \dots \vee Y_r$	$y_1 + y_2 + \dots + y_r \geq 1$
AND (\wedge)	$Y_1 \wedge Y_2 \wedge \dots \wedge Y_r$	$y_1 \geq 1$ $y_2 \geq 1$... $y_r \geq 1$
EXCLUSIVE OR ($\underline{\vee}$)	$Y_1 \underline{\vee} Y_2 \underline{\vee} \dots \underline{\vee} Y_r$	$y_1 + y_2 + \dots + y_r = 1$
IMPLICATION ($Y_1 \Rightarrow Y_2$)	$\neg Y_1 \vee Y_2$	$1 - y_1 + y_2 \geq 1$
EQUIVALLANCE ($Y_1 \Leftrightarrow Y_2$)	$(\neg Y_1 \vee Y_2) \wedge (Y_1 \vee \neg Y_2)$	$y_1 - y_2 = 0$

The algebraic equivalents of the propositional logic expressions in Eqs. (1)-(10) are given as follows:

$$\sum_l ypb_{il} = 1 \quad \forall i \quad (11)$$

$$\sum_k ypc_{ik} = 1 \quad \forall i \quad (12)$$

$$\sum_l ypb_{il} = \sum_k ypc_{ik} \quad \forall i \quad (13)$$

$$\sum_k ybc_{lk} \leq yb_l \quad \forall l \quad (14)$$

$$ybc_{lk} - \sum_i ypb_{il} \leq 0 \quad \forall lk \quad (15)$$

$$ybc_{lk} - \sum_i ypc_{ik} \leq 0 \quad \forall lk \quad (16)$$

$$\sum_n ypb_{ilmn} - ypb_{ilm} \leq N-1 \quad \forall ilm \quad (17)$$

$$\sum_m ypb_{ilm} - ypc_{ik} \leq M-1 \quad \forall ilk \quad (18)$$

$$ypc_{ik} - yp1_{ik} \leq 0 \quad \forall ik \notin D_{ik} \quad (19)$$

$$ypc_{ik} + yp2_{ik} \geq 1 \quad \forall ik \in D_{ik} \quad (20)$$

In order to define the boundaries of hyper-boxes, the following continuous variables are defined:

X_{lmn} the continuous variable that models bounds n for box l on attribute m

$XD_{l,k,m,n}$ the continuous variable that models bounds n for box l of class k on attribute m

The following mixed-integer programming problem models data classification problem with hyper-boxes:

$$\min z = \sum_i \sum_k (yp1_{ik} + yp2_{ik}) + c \sum_l yb_l \quad (21)$$

subject to

$$XD_{lkmn} \leq a_{im} ypb_{il} \quad \forall iklmn | n = lo \quad (22)$$

$$XD_{lkmn} \geq a_{im} ypb_{il} \quad \forall iklmn | n = up \quad (23)$$

$$XD_{lkmn} \leq M ybc_{lk} \quad \forall klmn \quad (24)$$

$$\sum_k XD_{lkmn} = X_{lmn} \quad \forall lmn \quad (25)$$

$$ypbn_{ilmn} \geq (1/M)(X_{lmn} - a_{im}) \quad \forall ilmn | n = up \quad (26)$$

$$ypbn_{ilmn} \geq (1/M)(a_{im} - X_{lmn}) \quad \forall ilmn | n = lo \quad (27)$$

$$\text{Eqs. (11)-(20)} \quad (28)$$

$$X_{lmn}, XD_{lkmn} \geq 0, yb_l, ybc_{lk}, ypb_{il}, ypc_{ik}, ypbn_{ilmn}, ypbm_{ilm}, yp1_{ik}, yp2_{ik} \in \{0,1\} \quad (29)$$

The objective function of the mixed-integer programming problem is to minimize the misclassifications in the data set with the minimum number of hyper-boxes. In order to eliminate unnecessary use of hyper-boxes, the unnecessary existence of a box is penalized with a small scalar c in the objective function. The lower and upper bounds of the boxes are given in Eqs. (22) and (23) respectively. The lower and upper bounds for the hyper-boxes are determined by the data points that are enclosed within the hyper-box. Eq. (4) enforces the bounds of boxes exist if and only if this box is assigned to a class. Eq. (5) is used to relate the two continuous variables that represent the bounds of the hyper-boxes. The position of a data point with respect to the bounds on attribute m for a hyper-box is given in Eqs. (26) and (27). The binary variable $ypbn_{ilmn}$ helps to identify whether the data point i is within the hyper-box l . Two constraints, one for the lower bound and one for the upper bound, are needed for this purpose (Eqs. (26) and (27)).

Since these constraints establish a relations between continuous and binary variables, a large parameter, M , is included in them. The model includes all of the algebraic constraints on binary variables that are constructed from propositional logic. Finally, last constraint gives non-negativity and integrality of decision variables.

3.2. Testing Problem for Multi-Class Data Classification

The testing problem for multi-class data classification using hyper-boxes is straight forward. If a new data point whose membership to a class is not known arrives, it is necessary to assign this data point to one of the classes. There are two possibilities for a new data point when determining its class:

- i. the new data point is within the boundaries of a hyper- box
- ii. the new data point is not enclosed in any of the hyper-boxes found in the training problem

When the first possibility is realized for the new data point, the classification is made by directly assigning this data to the class that was represented by the hyper-box enclosing the data point. In the case when the second possibility applies, the assignment of the new data point to a class requires some analysis; it is necessary to determine the closest hyper-box to the new data. If the data point is within the lower and upper bounds of all but one of the attributes (i.e., m') defining the box, then the shortest distance between the new point and the hyper-box is calculated using the minimum distance between hyper-planes defining the hyper-box and the new data point. The number of hyper-planes that must be evaluated for determining the minimum distance between the new data point and the hyper-box is given by $2(M-1)$ where M is the total number of attributes. The minimum distance between the new data point j and the hyper-box is

calculated using Eq. (30) considering the fact that the minimum distance is given by the normal of the hyper-plane.

$$\min_{l,m',n} \left\{ \left| \left(\sum_{m \neq m'} a_{jm} \right) + X_{lm'n} \right| / \sqrt{M-1} \right\} \quad (30)$$

If the data point is not within the lower and upper bounds of than one attributes defining the box, then the shortest distance between the new point and the hyper-box is calculated using the minimum distance between extreme points of the hyper-box and the new data point. The number of extreme points in a hyper-box is given by $2M$. Therefore, it is necessary to perform $2ML$ (L : total number of hyper-boxes) distance calculations for each new data point and select the minimum distance. The minimum distance between the new data point j and one of the extreme points of the hyper-box is calculated using Eq. (31).

$$\min_{l,n} \left\{ \sqrt{\sum_m (a_{jm} - X_{lmn})^2} \right\} \quad (31)$$

The following algorithm assign a new data point j with attribute values a_{jm} to class k :

Step 0: Initialize $inAtt(l,m)=0$.

Step 1: For each l and m , if

$$X_{lmn} \leq a_{jm} \leq X_{lmn'} \quad \forall n = lo, n' = up \quad (32)$$

Set $inAtt(l,m)=inAtt(l,m)+1$.

Step 2: If $inAtt(l,m)=N$, then go to Step 3. Otherwise, continue.

If $inAtt(l,m) \leq N-1$, then go to Step 4.

Step 3: Assign the new data point to class k where yc_{lk} is equal to 1 for the hyper-box in Step 2. Stop.

Step 4: Calculate the minimum given by Eq. (30) and set the minimum as $min1(l)$. Calculate the minimum given by Eq. (31) and set the minimum as $min2(l)$. Select the minimum between $min1(l)$ and $min2(l)$ to determine the hyper-box l that is closest to the new data point j . Assign the new data point to class k where ybc_k is equal to 1 for the hyper-box l . Stop.

The application of the proposed approach on an example is illustrated in the next section.

4. ILLUSTRATIVE EXAMPLE

We applied the mixed-integer programming method on a set of 16 data points in 4 different classes given in Figure 2. The data points can be represented by two attributes, 1 and 2.

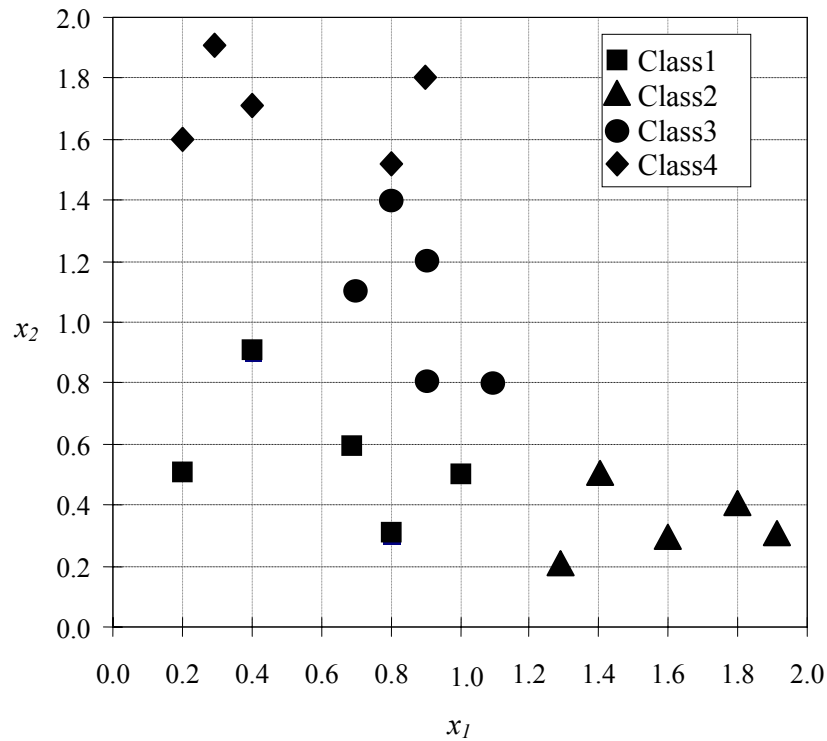


Fig. 2. Data points in the illustrative example and their graphical representation.

There are a total of 20 data points; 16 of these points were used in training and 4 of them used in testing. The training problem classified the data into 4 four classes using 5 hyper-boxes as shown in Fig. 3. It is interesting to note that Class1 requires two hyper-boxes while the other classes are represented with a single hyper-box only. The reason for having two hyper-boxes for Class1 is due to the fact that a single hyper-box for this class would include one of the data points that belong to Class3. In order to eliminate inconsistencies in training data set, the method included one more box for Class1.

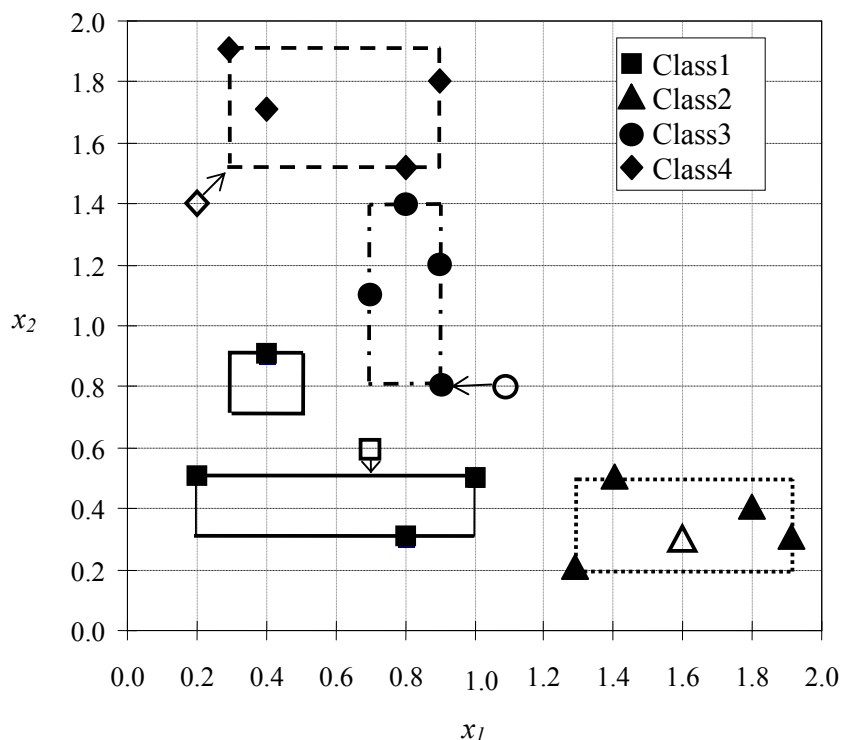


Fig. 3. Hyper-boxes that classify the data points in the illustrative example.

After the training is successfully completed, the test data is processed to assign them to hyper-boxes that classify the data perfectly. The assignment of the test data point to Class2 is straightforward since it is included in the hyper-box that classifies Class2 (i.e., $inAtt(l,m)=N$ for this data point). The test data in Class1 is assigned to one of the

hyper-boxes that classify Class1. Similarly, the test data in Class3 is also assigned to the hyper-box that classifies Class3. Since the test data in these classes are included within the bounds of one of the two attributes, the minimum distance is calculated as the normal to the closest hyper-plane to these data points. In the case of data point that belongs to Class4, it is assigned to its correct class since the closest extreme point of a hyper-box classifies Class4. This extreme point of the hyper-box 5 classifying Class4 is given by $(X_{5,1,lo}, X_{5,2,lo})$. The test problem also classified the data points with 100% accuracy.

The proposed data classification method is evaluated on the well-known IRIS data set in the next section.

5. EVALUATION OF THE METHOD ON IRIS DATA SET

In this part of the study, the efficiency of the proposed model is tested on the well-known IRIS data set. IRIS data published by Fisher (1936) is selected due to the reason that it has been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in centimeters on fifty iris specimens from each of three species, *Iris setosa*, *I. versicolor*, and *I. virginica*. This data set is one of the best known databases to be found in the pattern recognition (Schölkopf, Burges and Vapnik, 1996; Burges and Schölkopf, 1997; Bay, 1999; Kim et al., 2003).

We selected 24 data samples randomly for the training problem where each class is represented by exactly the same number of samples. Table 2 shows the training set used in order to solve the mixed-integer programming problem presented in Section 3.1.

Table 2. IRIS data training set.

SAMPLES	CLASS I (SETOSA)				CLASS II (VERSICOLOR)				CLASS III (VIRGINICA)			
	SL	SW	PL	PW	SL	SW	PL	PW	SL	SW	PL	PW
1	5.7	4.4	1.5	0.4	6.1	2.9	4.7	1.4	6.5	3.0	5.8	2.2
2	4.8	3.0	1.4	0.1	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
3	5.2	4.1	1.5	0.1	5.7	3.0	4.2	1.2	6.3	2.5	5	1.9
4	5.5	4.2	1.4	0.2	5.9	3.2	4.8	1.8	7.2	3.0	5.8	1.6
5	4.9	3.1	1.5	0.1	5.8	2.7	4.1	1.0	6.0	3.0	4.8	1.8
6	4.8	3.4	1.6	0.2	5.0	2.3	3.3	1.0	5.8	2.7	5.1	1.9
7	5.4	3.7	1.5	0.2	6.0	2.7	5.1	1.6	6.5	3.2	5.1	2.0
8	5.2	3.5	1.5	0.2	5.0	2.0	3.5	1.0	6.9	3.1	5.4	2.1

*SL: Sepal Length, SW: Sepal Width, PL: Petal Length, PW: Petal Width.

Using the above 24 samples, the MILP problem defined in Section 3.1 is modeled in GAMS (Brooke et al., 1998). The training problem is solved approximately in 15 seconds and without any misclassification of samples, 3 hyper-boxes are formed, one for each class. Table 3 gives the bounds on each attribute for constructed hyper-boxes and assigned classes of each box.

Table 3. Bounds of hyper-boxes constructed by training problem.

Hyper-Box and Classes	1 (SETOSA)		2 (VERSICOLOR)		3 (VIRGINICA)	
	Lower	Upper	Lower	Upper	Lower	Upper
Sepal Length	0	5.7	0	6.1	5.8	7.2
Sepal Width	2.7	4.4	2	3.2	2.3	3.2
Petal Length	1.4	1.6	3.3	5.1	4.8	5.8
Petal Width	0.1	0.4	1	1.8	1.6	2.2

After classifying the training data perfectly, the test set given in Table 4 is studied to assign them to constructed hyper-boxes by applying the method explained in Section 3.2. The assignment of data in the test set to classes is done without a prior knowledge on their membership to a class.

Table 4. IRIS data test set.

SAMPLES	CLASS I (SETOSA)				CLASS II (VERSICOLOR)				CLASS III (VIRGINICA)			
	SL	SW	PL	PW	SL	SW	PL	PW	SL	SW	PL	PW
1	4.5	2.3	1.3	0.3	4.9	2.4	3.3	1.0	6.7	3.3	5.7	2.1
2	5.0	3.5	1.6	0.6	6.2	2.2	4.5	1.5	7.3	2.9	6.3	1.8
3	4.3	3.0	1.1	0.1	5.5	2.6	4.4	1.2	4.9	2.5	4.5	1.7
4	5.0	3.5	1.3	0.3	6.0	3.4	4.5	1.6	6.7	3.1	5.6	2.4
5	5.1	3.8	1.9	0.4	5.8	2.7	3.9	1.2	5.8	2.8	5.1	2.4
6	5.0	3.4	1.5	0.2	5.6	3.0	4.1	1.3	6.5	3.0	5.5	1.8
7	5.1	3.7	1.5	0.4	5.7	2.9	4.2	1.3	7.7	3.8	6.7	2.2
8	5.1	3.8	1.5	0.3	5.9	3.0	4.2	1.5	5.7	2.5	5.0	2.0
9	4.6	3.4	1.4	0.3	6.9	3.1	4.9	1.5	6.8	3.0	5.5	2.1
10	5.4	3.4	1.7	0.2	5.2	2.7	3.9	1.4	7.7	3.0	6.1	2.3
11	5.8	4.0	1.2	0.2	7.0	3.2	4.7	1.4	6.9	3.2	5.7	2.3
12	4.9	3.0	1.4	0.2	5.7	2.6	3.5	1.0	7.4	2.8	6.1	1.9
13	5.0	3.2	1.2	0.2	6.6	2.9	4.6	1.3	7.2	3.2	6.0	1.8
14	5.0	3.0	1.6	0.2	6.0	2.9	4.5	1.5	6.4	2.7	5.3	1.9
15	5.1	3.8	1.6	0.2	6.6	3.0	4.4	1.4	7.9	3.8	6.4	2.0
16	4.7	3.2	1.3	0.2	6.1	2.8	4.0	1.3	6.2	2.8	4.8	1.8
17	5.4	3.4	1.5	0.4	6.4	3.2	4.5	1.5	6.4	3.2	5.3	2.3
18	4.6	3.2	1.4	0.2	5.5	2.5	4.0	1.3	6.7	3.3	5.7	2.5
19	4.9	3.1	1.5	0.2	5.8	2.6	4.0	1.2	6.4	2.8	5.6	2.1
20	5.0	3.4	1.6	0.4	6.4	2.9	4.3	1.3	6.3	2.8	5.1	1.5
21	4.4	2.9	1.4	0.2	5.5	2.4	3.8	1.1	6.5	3.0	5.2	2.0
22	4.6	3.6	1.0	0.2	5.1	2.5	3.0	1.1	6.8	3.2	5.9	2.3
23	5.1	3.3	1.7	0.5	5.6	2.5	3.9	1.1	7.7	2.8	6.7	2.0
24	5.0	3.3	1.4	0.2	6.3	3.3	4.7	1.6	6.1	3.0	4.9	1.8
25	5.1	3.5	1.4	0.2	5.4	3.0	4.5	1.5	6.1	2.6	5.6	1.4
26	5.0	3.6	1.4	0.2	6.7	3.1	4.4	1.4	7.1	3.0	5.9	2.1
27	5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	5.6	2.8	4.9	2.0
28	5.1	3.3	1.7	0.5	5.6	2.7	4.2	1.3	6.3	3.3	6.0	2.5
29	5.5	3.5	1.3	0.2	6.0	2.2	4.0	1.0	6.4	2.8	5.6	2.2
30	4.8	3.4	1.9	0.2	6.7	3.1	4.7	1.5	6.3	2.7	4.9	1.8
31	4.6	3.1	1.5	0.2	6.3	2.5	4.9	1.5	6.4	3.1	5.5	1.8
32	5.1	3.5	1.4	0.3	5.7	2.8	4.1	1.3	7.7	2.6	6.9	2.3
33	5.3	3.7	1.5	0.2	5.6	2.9	3.6	1.3	6.9	3.1	5.1	2.3
34	4.8	3.1	1.6	0.2	6.5	2.8	4.6	1.5	6.7	3.0	5.2	2.3
35	4.7	3.2	1.6	0.2	6.1	3.0	4.6	1.4	6.3	3.4	5.6	2.4
36	4.8	3.0	1.4	0.3	5.7	2.8	4.5	1.3	6.0	2.2	5.0	1.5
37	5.1	3.4	1.5	0.2	5.5	2.4	3.7	1.0	7.2	3.6	6.1	2.5
38	5.7	3.8	1.7	0.3	6.1	2.8	4.7	1.2	5.8	2.7	5.1	1.9
39	5.2	3.4	1.4	0.2	6.8	2.8	4.8	1.4	6.2	3.4	5.4	2.3
40	4.4	3.2	1.3	0.2	6.2	2.9	4.3	1.3	7.6	3.0	6.6	2.1
41	4.4	3.0	1.3	0.2	6.7	3.0	5.0	1.7	5.9	3.0	5.1	1.8
42	5.4	3.9	1.7	0.4	6.3	2.3	4.4	1.3	6.7	3.0	5.0	1.7

*SL: Sepal Length, SW: Sepal Width, PL: Petal Length, PW: Petal Width.

There are two possibilities for a new data: either it is enclosed in an existing hyper-box, or it is outside the region enclosed by a hyper-box. Data points like that are within the bounds of the hyper-box 1 are assigned to Setosa class. This assignment is the easiest one; however there are also some data points that are not enclosed in any of the hyper-

boxes found in the training problem. For example, test data point 4 in Class III (with the attribute values 6.7, 3.1, 5.6, 2.4) is a typical example for this type of data points. The closest hyper-box to this data is calculated using Eqs. (30) and (31). Then the hyper-box that is closest to this data point is (3); thus it is assigned to the class “Virginica”. For each of the data in the test set, the method is applied and the test data is assigned classes. Then, the accuracy of the proposed model is checked by comparing the assigned and original classes of samples. At the end of the comparisons, it is realized that 124 samples are assigned correctly to their original classes. On the other hand, 2 samples, (data 36 of (data 41 of Class II with attributes 6.7, 3.0, 5.0, 1.7) and (Class III with attributes 6, 2.2, 5.0, 1.5), are misclassified. Table 5 summarizes the above results by the help of confusion matrix.

Table 5. Classification performances.

Assigned Original	SETOSA	VERSICOLOR	VIRGINICA
SETOSA	42	0	0
VERSICOLOR	0	41	1
VIRGINICA	0	1	41

From this table it seen that, the overall accuracy of the proposed model to the IRIS data set is 98.4%. The previous models that use the IRIS data set have accuracy values of 98.0, 96.7, 90.7, and 98.0 in the case of the tolerant rough set (TRS), the back-propagation neural networks (BPNN), objective function-based unsupervised neural networks (OFUNN) and fuzzy C-means (FCM), respectively (Kim, 2001).

When we compare the classification performances of these different classification algorithms, the suggested mixed-integer programming (MIP) approach in this research is very accurate and efficient. Moreover, in the case of simplicity, MIP approach is simpler and easily understandable than the other approaches. Furthermore, the solution time of MIP approach is very short (15 CPU seconds).

6. CONCLUSIONS AND FUTURE WORK

The proposed data classification method based on mixed-integer programming allows the use of hyper-boxes for defining boundaries of the classes that enclose all or some of the points in that set. Traditional methods based using hyper-planes can be inaccurate and inefficient in classifying data more than two classes. Consequently, using hyper-boxes for multi-class data classification problems can be very accurate because of the well-construction of boundaries of each class.

In the training part of the proposed approach, the construction of hyper-boxes is done by representing the relationships between binary variables with propositional logic. Their equivalent integer constraints are obtained by the help of Boolean algebra. After solving the MIP problem, the characteristics of the data points that belong to a certain class are determined and differentiated from the ones that belong to other classes.

In the test part, by considering the bounds of hyper-boxes constructed in the training part, the test data is studied one by one with the suggested method in Section 3.2. If a sample's entire attribute values are between the bounds of a hyper-box, then it is directly assigned to that box's corresponding class. On the other hand, if not, the closest hyper-box to that sample is found out by investigating the distances to each of the hyper-boxes one by one, by the ideas of distance to hyper-plan and extreme points. This

procedure is explained in detail by a small illustrative example with two attributes and four classes.

The proposed multi-class data classification method was applied to the IRIS data and it was compared to other classification techniques such as TRS, BPNN, OFUNN, and FCM methods in terms of classification performances. The new data classification approach presented in this paper is very accurate and efficient on multi-class data classification problems. In addition to this, the simplicity and the understandability of the proposed model are preferable to the other methods.

In future, we will apply this proposed multi-class data classification problem to the field of bioinformatics, especially to classification of proteins. Due to importance of protein structure in understanding the biological and chemical activities in any biological system, protein structure determination and prediction has been a focal research subject in computational biology and bioinformatics. Proteins are classified into four main structural groups (folding types) according to their amino acid composition. There have been several methods proposed to exploit this theory for predicting folding type of a protein. The topics that will be explored in future include solution of this protein folding type problem.

Acknowledgements

The authors would like to thank the organizers and instructors at The EURO Summer Institute (ESI XXII): Optimization and Data Mining.

References

- Adem, J., Gochet, W., 2004. Mathematical programming based heuristics for improving LP-generated classifiers for the multi-class supervised classification problem, *European Journal of Operational Research*, In Press, Corrected Proof.
- Anthony, M., 2004. On data classification by iterative linear partitioning, *Discrete Applied Mathematics* 144 2-16.
- Bajgier, S.M., Hill, A.V., 1982. An experimental comparison of statistical and linear programming approaches to the discriminant problem, *Decision Sciences* 13 604-618.
- Bay, B., 1999. The UCI KDD Archive [<http://kdd.ics.uci.edu>], Department of Information and Computer Science, University of California, Irvine, CA.
- Banzan, J., Nguyen, H., Skowron, A., Stepaniuk, J., 1994. Some logic and rough set applications for classifying objects, ICS Research Report No. 34, Warsaw University of Technology.
- Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., 1997. Logical analysis of numerical data, *Mathematical Programming* 79, 163-190.
- Brooke, A., Kendrick, D., Meeraus, A., Raman, R., 1998. *GAMS: A User's Guide*, GAMS Development Co., Washington, DC.
- Burges, C.J.C., Schölkopf, B., 1997. Improving the accuracy and speed of support vector learning machines, *Adv. Neural Inf. Process. Systems* 9 375-380.
- Carpenter, G., Grossberg, S., 1987. A massively parallel architecture for a self-organizing neural pattern recognition machine, *Comput. Vision Graphics Image Understanding* 37 (2) 54-115.

- Cavalier, T.M., P.M. Pardalos and A.L. Soyster (1990). Modeling and integer programming techniques applied to propositional calculus, *Comp. Oper. Res.*, **17**(6), 561-570.
- Chang, Y., Han, C., Jou, F., Fan, K., Chen, K.S., Chang, J., 2004. A modular eigen subspace scheme for high-dimensional data classification, *Future Generation Computer Systems* 20 1131-1143.
- Cortes, C., Vapnik, V., 1995. Support vector network, *Machine Learning* 20 273-297.
- Erenguc, S.S., Koehler, G.J., 1990. Survey of mathematical programming models and experimental results for linear discriminant analysis, *Managerial and Decision Economics* 11 215-225.
- Fisher, R., 1936. The use of multiple measurements in taxonomic problems, *Ann. Eugenics* 7 (2) 179-188.
- Gehrlein, W.V., 1986. General mathematical programming formulations for the statistical classification problem, *Operations Research Letters* 5 (6) 299-304.
- Kim, D., 2001. Data classification based on tolerant rough set, *Pattern Recognition* 34 1613-1624.
- Kim, H., Pang, S., Je, H., Kim, D., Bang, S.Y., 2003. Constructing support vector machine ensemble, *Pattern Recognition* 36 2757-2767.
- Littschwager, J.M., Wang, C., 1978. Integer programming solution of a classification problem, *Management Science* 24 (14) 1515-1525.
- Lin, C.T., Lee, C.S.G., 1991. Neural-network-based fuzzy logic control and decision system, *IEEE Trans. on Comput.* 40 (12) 1320-1336.

- Nguyen, H., Skowron, A., Synak, P., 1998. Discovery of data patterns with applications to decomposition and classification problems, *Rough Sets in Knowledge Discovery*, Physica-Verlag, Wurzburg.
- Pawlak, Z., 1982. Rough set, *Int. J. Inform. Comput. Sci.* 11 341-356.
- Raman, R. and I.E. Grossmann (1991). Relation between MILP modeling and logical inference for chemical process synthesis, *Comp. Chem. Eng.*, **15**(2), 73-84.
- Schölkopf, B., Burges, C., Vapnik, V., 1995. Extracting support data for a given task, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Montreal, Canada.
- Simpson, P., 1992. Fuzzy min-max neural networks- Part 1: Classification, *IEEE Trans. Neural Networks* 3 (5) 776-786.
- Stam, A., Joachimsthaler, E.A., A comparison of a robust mixed-integer approach to existing methods for establishing classification rules for the discriminant problem, *European Journal of Operations Research* 46 (1) 113-122.
- Turkay, M., Grossman, I.E., 1996b. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions, *Ind. Engng Chem.* 35 2611-2623.
- Yajima, Y., 2003. Linear programming approaches for multi category support vector machines, *European Journal of Operational Research*, In Press, Corrected Proof.