Expert Systems

Article

DOI: 10.1111/j.1468-0394.2012.00645.x

Classification of fuzzy data based on the support vector machines

Yahya Forghani,[1] Hadi Sadoghi Yazdi[1] and Sohrab Effati[2]

(1) Computer Department, Ferdowsi University of Mashhad, Iran
Email: yahyafor2000@yahoo.com; h-sadoghi@um.ac.ir
(2) Applied Mathematics Department, Ferdowsi University of Mashhad, Iran
Email: s-effati@um.ac.ir

Abstract: *Data may be afflicted with uncertainty. Uncertain data may be shown by an interval value or in general by a fuzzy set. A number of classification methods have considered uncertainty in features of samples. Some of these classification methods are extended version of the support vector machines (SVMs), such as the Interval-SVM (ISVM), Holder-ISVM and Distance-ISVM, which are used to obtain a classifier for separating samples whose features are interval values. In this paper, we extend the SVM for robust classification of linear/non-linear separable data whose features are fuzzy numbers. The support of such training data is shown by a hypercube. Our proposed method tries to obtain a hyperplane (in the input space or in a high-dimensional feature space) such that the nearest point of the hypercube of each training sample to the hyperplane is separated with the widest symmetric margin. This strategy can reduce the misclassification probability of our proposed method. Our experimental results on six real data sets show that the classification rate of our novel method is better than or equal to the classification rate of the well-known SVM, ISVM, Holder-ISVM and Distance-ISVM for all of these data sets.*

Keywords: SVM, fuzzy number, linear separable data, non-linear separable data

1. Introduction

Clustering and classification are among the most important problem tasks in the realm of data analysis, data mining and machine learning. While clustering can be seen as the most popular representative of unsupervised learning, classification (together with regression) is arguably the most frequently considered task in supervised learning (Theodoridis & Koutroumbas, 1998; Webb, 2003).

The data, which we aim to cluster or to classify, may be afflicted with uncertainty. Uncertain data can take on many different forms (Ross, 2004). There is uncertainty that arises because of complexity; for example, the complexity in the reliability network of a nuclear reactor. There is uncertainty that arises from ignorance, from various classes of randomness, from the inability to perform adequate measurements, from lack of knowledge or from vagueness, such as the fuzziness inherent in our natural language.

Many clustering and classification algorithms have considered uncertainty, including, for example, fuzzy c-means (Masson & Denoeux, 2008), rule induction (Qin *et al*., 2010), nearest-neighbour estimation (Denoeux, 1995; Younes *et al*., 2009), naïve possibilistic network (Jenhani *et al*., 2008; Haouari *et al*., 2009), possibilistic decision tree (Jenhani *et al*., 2008; Haouari *et al*., 2009), possibilistic support vector machine (SVM) (Ji *et al*., 2010), Interval-SVMs (ISVMs) (Carrizosa *et al*., 2007), Holder-SVMs (Trafalis & Gilbert, 2006, 2007) and Distance-SVM (Do & Poulet, 2005). The later four methods are extended versions of well-known SVMs. In continue, we review the well-known SVM and these extended versions.

The well-known SVM (Cortes & Vapnik, 1995; Vapnik, 1995; Burges, 1998) is a pattern recognition technique and is indeed a fuzzy inference system (see Chen & Wang, 2003), and since it can be shown by a neural network (see Haykin, 1999), it is a neuro-fuzzy model (Mitra & Hayashi, 2000). On the basis of the idea of structural risk minimization in statistical learning theory, the SVM has been shown to provide higher performance in pattern recognition and function regression than traditional learning machines (Burges, 1998). It has been introduced as a powerful tool for solving classification problems in recent years. When the SVM is used for two-class classification of non-linear separable data, it first maps each sample of each class from the input space into a high-dimensional feature space in which the classes are linearly separable, and then separates them by a hyperplane with the widest symmetric margin in the high-dimensional feature space (see the following section for more information).

The well-known SVM has been proposed for classification of data whose features and class labels are known precisely. The fuzzy SVM or FSVM (Lin & Wang, 2002) is used for data classification when we know the features of training samples precisely, but we are uncertain about their class labels. The FSVM treats training samples with different degree of importance in its training process. The degree of importance of a training sample is inversely related to the degree of uncertainty of the class label of that sample.

The SVM also has been extended for classification of data whose features are imprecise or uncertain. In Ji *et al*. (2010), the features of data have been considered to be triangular fuzzy numbers (Zimmermann, 1996; Ross, 2004) and then the SVM has been extended for such training samples based

on the possibility concept (Dubois & Prade, 1988) in the input space (not in a high-dimensional feature space). Therefore, this method can obtain a classifier for linear separable fuzzy data. Moreover, this method is a non-convex program (Bazara *et al.*, 2006). Thus, obtaining its global optimal solution is very time consuming (Bazara *et al.*, 2006).

In some studies (Do & Poulet, 2005; Carrizosa *et al.*, 2007; Trafalis & Gilbert, 2007), each feature of data has been considered to be an interval value (Moore *et al.*, 2009). An interval value is a fuzzy set (Zimmermann, 1996; Ross, 2004). In Carrizosa *et al.* (2007), the SVM has been solved for such training samples in the input space. Thus, this method can be used only for classification of linear separable interval-valued data. We name this method as ISVM.

In Trafalis and Gilbert (2007), the ISVM in a high-dimensional feature space has been approximated by using the Holder inequality. Our experimental results show that the misclassification rate of this model for real data sets is high. We name this method as Holder-ISVM.

Finally, in Do and Poulet (2005), the SVM has been solved for classification of interval-valued data in a high-dimensional feature space by using the Gaussian kernel function and using the Hausdorff distance (Tian *et al.*, 2006) instead of the Euclidean distance in the kernel function. We name this method as Distance-ISVM. The Euclidean distance between two interval values is an interval value, but the Hausdorff distance between two interval values is a crisp value. Therefore, if the Hausdorff distance is used in the Gaussian kernel function, the SVM program for interval-valued data is simplified to a crisp program.

In this paper, we solve the SVM problem in the input space and also in high-dimensional feature space for robust classification of data whose features are fuzzy numbers. Fuzzy numbers are more general than interval values. Our proposed program is quadratic and convex (Bazara *et al.*, 2006) and our experimental results on six real data sets show that the classification rate of our novel method is better than or equal to the well-known SVM, ISVM, Holder-ISVM and Distance-ISVM for all of these data sets.

The organization of this paper is as follows. In Section 2, we pay to some preliminaries and in section 3, our novel method is explained. In Section 4, we show some numerical examples. In Section 5, we apply our novel method for classification of some real data sets. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. Some fuzzy concepts

Definition 2.1.1. (Zimmermann, 1996). Let $\tilde{x}$ be a fuzzy set. Support of $\tilde{x}$ is defined as follows:

$$\text{supp}(\tilde{x}) = \{x \mid \mu_{\tilde{x}}(x) > 0\}.$$

Definition 2.1.2. Each interval value $M = [a, b]$ with the lower bound a and the upper bound b can be shown by $\{e, f\}$, where $e = (a + b)/2$ is called the midpoint and $f = (b - a)/2$ is called the radius or spread.

Theorem 2.1.1. (Moore *et al.*, 2009). Let $M = \{e, f\}$ and $N = \{g, h\}$ be two interval values and $s \in \mathbb{R}$ be a crisp value. Then,

$$M + N = \{e + g, f + h\},$$
$$M - N = \{e - g, f + h\},$$
$$s \times M = \{se, |s|f\}$$

In general, we can obtain the midpoint of each function $f(.)$ for interval-valued input parameter(s) by using the above operations, but its spread can be obtained by using the above operations only if the terms of $f(.)$ are independent of each others. For example, Let x, y and z be three interval values. We can obtain the spread of $f(x, y, z) = 2x + yz$ by using the above operations because its first term is independent of its second term. But since the first term and the second term of $g(x, y, z) = xy + xz$ are dependent, its spread cannot be obtained by using the above operations.

Theorem 2.1.2. (Moore *et al.*, 2009). The spread of each function for interval-valued input parameter(s) obtained by using the basic operations on interval values is bigger than its actual spread which can be obtained by using Zadeh's extension principle (Zadeh, 1978).

Definition 2.1.3. (Zimmermann, 1996). The LR-type fuzzy number is a special type of representation for fuzzy number. It is defined by two functions L (and R) which map $\mathbb{R}^+ \to [0, 1]$ and are decreasing shape functions and $L(0) = 1$, $L(1) = 0$, $\forall x < 1 : L(x) > 0$ and $\forall x > 0 : L(x) < 1$. A fuzzy number $\tilde{x}$ is of LR-type if there exist reference functions L (for left) R (for right) and scalars $\alpha, \beta > 0$, with

$$\mu_{\tilde{x}}(x) = \begin{cases} L\left(\dfrac{m - x}{\alpha}\right) & x \le m, \\ R\left(\dfrac{x - m}{\beta}\right) & x \ge m \end{cases}$$

Here, m, called the mean value of $\tilde{x}$, is a real member and α and β are called the left and right spreads, respectively. Here $\mu_{\tilde{x}}(x)$ is membership function of LR-type fuzzy number $\tilde{x}$, denoted by $(m, \alpha, \beta)_{LR}$.

Definition 2.1.4. (Zimmermann, 1996). Let $\tilde{x} = (m, \alpha, \beta)_{LR}$ be a LR-type fuzzy number. If $L(.) = R(.) = T(.)$, where

$$T(x) = \begin{cases} 1 - x & 0 \le x \le 1, \\ 0 & \text{otherwise}, \end{cases}$$

then $\tilde{x}$ is called triangular fuzzy number, denoted by $(m, \alpha, \beta)_{TFN}$. Meanwhile, $\tilde{x}$ is called symmetric triangular fuzzy number, denoted by $(m, s)_{TFN}$, if $s = \alpha = \beta$.

Theorem 2.1.3. (Zimmermann, 1996). Let $\tilde{M} = (m_{\tilde{M}}, l_{\tilde{M}}, r_{\tilde{M}})_{LR}$ and $\tilde{N} = (m_{\tilde{N}}, l_{\tilde{N}}, r_{\tilde{N}})_{LR}$ be two LR-type fuzzy numbers, where $m_{\tilde{M}}$ and $m_{\tilde{N}}$ are the means, $l_{\tilde{M}}$ and $l_{\tilde{N}}$ are the left spreads and $r_{\tilde{M}}$ and $r_{\tilde{N}}$ are the right spreads of $\tilde{M}$ and $\tilde{N}$, respectively. Also, let $s \in \mathbb{R}$ be a crisp value. Then

$$\tilde{M} + \tilde{N} = (m_{\tilde{M}} + m_{\tilde{N}}, l_{\tilde{M}} + l_{\tilde{N}}, r_{\tilde{M}} + r_{\tilde{N}})_{LR},$$

$$\tilde{M} - \tilde{N} = \left(m_{\tilde{M}} - m_{\tilde{N}}, l_{\tilde{M}} + r_{\tilde{N}}, r_{\tilde{M}} + l_{\tilde{N}}\right)_{LR},$$

$$s \times \tilde{M} = \begin{cases} (sm_{\tilde{M}}, sl_{\tilde{M}}, sr_{\tilde{M}})_{LR} & s \geq 0, \\ (sm_{\tilde{M}}, -sr_{\tilde{M}}, -sl_{\tilde{M}})_{LR} & s < 0, \end{cases}$$

are also L-R fuzzy numbers.

Definition 2.1.5. (Klir & Yuan, 1995). For any fuzzy number $\tilde{M}$, $\tilde{N}$ and $\alpha \in (0, 1]$, $\tilde{M}_\alpha = [\tilde{M}_\alpha^L, \tilde{M}_\alpha^U]$ and $\tilde{N}_\alpha = [\tilde{N}_\alpha^L, \tilde{N}_\alpha^U]$ denoted the α-cut of $\tilde{M}$ and $\tilde{N}$, respectively. If we define the partial ordering of closed intervals in the usual way, that is

$$\left[\tilde{M}_\alpha^L, \tilde{M}_\alpha^U\right] \leq \left[\tilde{N}_\alpha^L, \tilde{N}_\alpha^U\right] \text{ if } \tilde{M}_\alpha^L \leq \tilde{N}_\alpha^L \text{ and } \tilde{M}_\alpha^U \leq \tilde{N}_\alpha^U,$$

then for any fuzzy number $\tilde{M}$, $\tilde{N}$, we have $\tilde{M} \leq_f \tilde{N}$ if $\tilde{M}_\alpha \leq \tilde{N}_\alpha$ for all $\alpha \in (0, 1]$, where $\leq_f$ denotes the fuzzy smaller than.

Theorem 2.1.4. (Ross, 2004). Suppose we have two fuzzy numbers, $\tilde{M}$ and $\tilde{N}$. We can use Zadeh's extension principle to calculate the truth value of the assertion that the fuzzy number $\tilde{M}$ is smaller than the fuzzy number $\tilde{N}$

$$T(\tilde{M} \leq \tilde{N}) = \sup_{m \leq n} \min[\mu_{\tilde{M}}(m), \mu_{\tilde{N}}(n)]$$

Since each fuzzy number is a convex fuzzy set, the fuzzy numbers $\tilde{M}$ and $\tilde{N}$ are convex. Therefore, it can be seen from Figure 1 that

$$T(\tilde{M} \leq \tilde{N}) = \begin{cases} 1 & \bar{m} \leq \bar{n}, \\ \text{height}(\tilde{M} \cap \tilde{N}) = \mu_{\tilde{M}}(d) = \mu_{\tilde{N}}(d) & \text{otherwise}, \end{cases}$$

where $\text{height}(\tilde{x}) = \max_x(\mu_{\tilde{x}}(x))$, d is the location of the highest intersection point, $\bar{m} = \arg\text{height}(\tilde{M})$ and $\bar{n} = \arg\text{height}(\tilde{N})$. For example, for the fuzzy numbers plotted in Figure 1, we have $T(\tilde{M} \leq \tilde{N}) = d$ and $T(\tilde{N} \leq \tilde{M}) = 1$.

Theorem 2.1.5. (Hao, 2008). Consider two symmetric triangular fuzzy numbers $\tilde{M} = (m_{\tilde{M}}, s_{\tilde{M}})_{TFN}$ and $\tilde{N} = (m_{\tilde{N}}, s_{\tilde{N}})_{TFN}$, where $m_{\tilde{M}}$ and $m_{\tilde{N}}$ are the means, and $s_{\tilde{M}}$ and $s_{\tilde{N}}$ are the radiuses (the left/right spreads) of $\tilde{M}$ and $\tilde{N}$, respectively. The degree that $\tilde{M}$ is smaller than $\tilde{N}$ is defined by the following membership function:

$$T(\tilde{M} \leq \tilde{N}) = \begin{cases} 1 & \eta > 0, \ \vartheta > 0, \\ 0 & \eta < 0, \ \vartheta < 0, \\ 0.5\left(1 + \dfrac{\eta + \vartheta}{\max\{|\eta|, |\vartheta|\}}\right) & \text{otherwise}, \end{cases}$$

where $\eta = (m_{\tilde{N}} + s_{\tilde{N}}) - (m_{\tilde{M}} + s_{\tilde{M}})$ and $\vartheta = (m_{\tilde{N}} - s_{\tilde{N}}) - (m_{\tilde{M}} - s_{\tilde{M}})$.

Definition 2.1.6. (Bortolan & Degani, 1985). Each defuzzifier is used to map a fuzzy set to a crisp point. For example, the Roubens defuzzification method is defined as follows:

$$\text{Defuzzify}(f(\tilde{x})) = 1/2 \int_0^1 (\inf(f(\tilde{x}))_\alpha + \sup(f(\tilde{x}))_\alpha)\, d\alpha$$

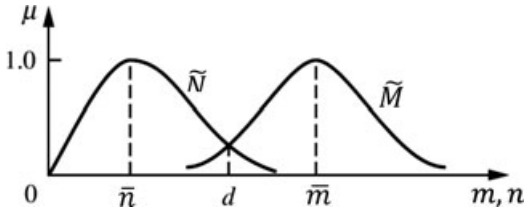

Figure 1: *Two fuzzy numbers as fuzzy sets on the real line.*

2.2. SVMs for two-class classification

The formulation of SVM for two-class classification is as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to } \begin{cases} y_i(w^T\varphi(x_i) + b) \geq 1 - \xi_i, & i = 1, 2, \ldots, n, \\ \xi_i \geq 0, & i = 1, 2, \ldots, n, \end{cases} \tag{1}$$

where n is the number of training samples, $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ is a p-dimensional sample in the input space, $\varphi(x)$ is a mapping function that maps x into a high-dimensional feature space and $y_i \in \{+1, -1\}$ is the class label of x_i. The SVM finds a hyperplane in the high-dimensional feature space, namely $w^T\varphi(x) + b = 0$, such that the samples are classified correctly, namely

$$y_i(w^T\varphi(x_i) + b) \geq 1, \quad i = 1, 2, \ldots, n, \tag{2}$$

where $w = (w_1, w_2, \ldots, w_h)^T$ is a weight vector, b is a bias term and h is the dimension of the high-dimensional feature space. There are an infinite number of hyperplane that satisfy equation (2). The generalization ability depends on the location of the separating hyperplane. The hyperplane with the maximum margin is called optimal hyperplane. The margin of a hyperplane is equal to $M = \frac{1}{\|w\|^2}$. The program (1) minimizes $\|w\|^2$ or equivalently maximizes the margin. ξ_i is the slack variable of i-th training sample, C is a penalty term that determines the trade-off between the maximization of the margin between two classes, namely $\frac{1}{\|w\|^2}$, and minimization of the classification error, namely $\sum_{i=1}^{n} \xi_i$ (For more information see (Cortes & Vapnik, 1995; Vapnik, 1995, 1998; Burges, 1998; Webb, 2003)). Figure 2 plots the optimal hyperplane and the slack parameters for a classification problem when $\varphi(x) = x$.

Where $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$ is a kernel function. The Lagrangian dual form of the program (1) can be restated as follows:

$$\max_{\beta} \sum_{i=1}^{n} \beta_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_i\beta_j y_i y_j K(x_i, x_j)$$

$$\text{subject to } \begin{cases} \sum_{i=1}^{n} \beta_i y_i = 0, \\ 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n \end{cases} \tag{3}$$

where $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$ is a kernel function and $0 \leq \beta_i \leq C (i = 1, 2, \ldots, n)$. From KKT conditions, if $0 < \beta_i < C$, the bias term can be obtained as follows:

$$b = y_i - \sum_{j=1}^{n} \beta_j y_j K(x_i, x_j)$$

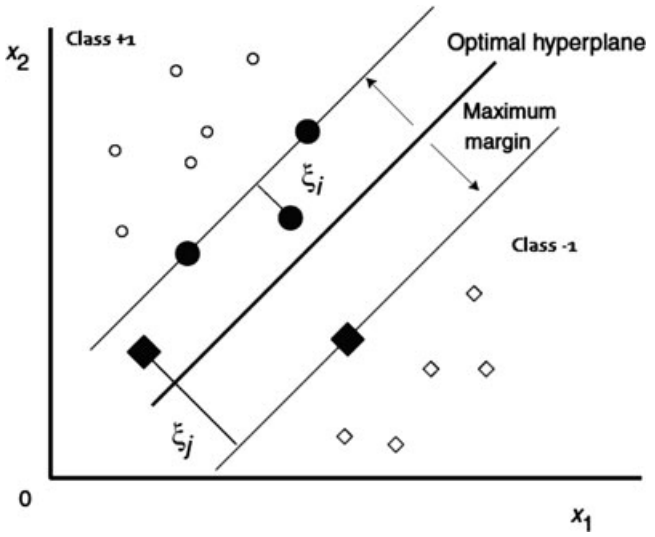

Figure 2: *Optimal separating hyperplane obtained by soft-margin SVM for* $\varphi(x) = x$.

Finally, the decision function for the test sample x is given by $\text{sign}(f(x))$, where

$$f(x) = \sum_{i=1}^{n} \beta_i y_i K(x_i, x) + b \tag{4}$$

3. Our proposed method

In this section, first, we extend the SVM for classification of linear separable samples whose features are fuzzy numbers and then extend it for non-linear separable case. Let $\tilde{x}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, \ldots, \tilde{x}_{ip})^T$ be i-th fuzzy training sample whose features, namely $\tilde{x}_{ij} (j = 1, 2, \ldots, p)$, are fuzzy numbers.

3.1. Linear separable fuzzy data

3.1.1. Training phase The SVM for classification of linear separable fuzzy samples $\tilde{x}_i (i = 1, 2, \ldots, n)$ can be reformulated as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \begin{cases} y_i(w^T \tilde{x}_i + b) \geq \tilde{1} - \xi_i, & i = 1, 2, \ldots, n \\ \xi_i \geq 0, & i = 1, 2, \ldots, n. \end{cases} \tag{5}$$

where $\tilde{1}$ is an optional fuzzy set for representation of approximately one. From Theorem (2.1.1), we have

$$(y_i(w^T \tilde{x}_i + b))_\alpha^L = y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) - |w|^T s_{(\tilde{x}_i)_\alpha},$$
$$(y_i(w^T \tilde{x}_i + b))_\alpha^U = y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) + |w|^T s_{(\tilde{x}_i)_\alpha},$$

where $|w| = (|w_1|, |w_2|, \ldots, |w_p|)^T$; $m_{(\tilde{x}_i)_\alpha} = \frac{(\tilde{x}_i)_\alpha^L + (\tilde{x}_i)_\alpha^U}{2}$ and $s_{(\tilde{x}_i)_\alpha} = \frac{(\tilde{x}_i)_\alpha^U - (\tilde{x}_i)_\alpha^L}{2}$ are the midpoint and the radius of $\tilde{x}_i$ at α-cut, respectively; and $(\tilde{x}_i)_\alpha^L$ is the lower bound and $(\tilde{x}_i)_\alpha^U$ is the upper bound of $\tilde{x}_i$ at α-cut. Thus, from Definition (2.1.5), the program (5) is transformed into the following program:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$\text{s.t.} \begin{cases} y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) - |w|^T s_{(\tilde{x}_i)_\alpha} \geq m_{(\tilde{1})_\alpha} - s_{(\tilde{1})_\alpha} - \xi_i, \\ \qquad\qquad i = 1, 2, \ldots, n, \quad \forall \alpha \in (0, 1], \\ y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) + |w|^T s_{(\tilde{x}_i)_\alpha} \geq m_{(\tilde{1})_\alpha} + s_{(\tilde{1})_\alpha} - \xi_i, \\ \qquad\qquad i = 1, 2, \ldots, n, \quad \forall \alpha \in (0, 1], \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \tag{6}$$

where $m_{(\tilde{1})_\alpha}$ and $s_{(\tilde{1})_\alpha}$ are the midpoint and the radius of $\tilde{1}$ at α-cut, respectively. If the support of $\tilde{1}$ is $\{1\}$, the program (6) is restated as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \begin{cases} y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) - |w|^T s_{(\tilde{x}_i)_\alpha} \geq 1 - \xi_i, \\ \qquad\qquad i = 1, 2, \ldots, n, \quad \forall \alpha \in (0, 1], \\ y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) + |w|^T s_{(\tilde{x}_i)_\alpha} \geq 1 - \xi_i, \\ \qquad\qquad i = 1, 2, \ldots, n, \quad \forall \alpha \in (0, 1], \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \tag{7}$$

Since for each i and for each $\alpha \in (0, 1]$, we have

$$y_i\left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) + |w|^T s_{(\tilde{x}_i)_\alpha} \geq y_i\left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) - |w|^T s_{(\tilde{x}_i)_\alpha}$$

the program (7) can be simplified as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \begin{cases} y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) - |w|^T s_{(\tilde{x}_i)_\alpha} \geq 1 - \xi_i, \\ \qquad\qquad i = 1, 2, \ldots, n, \quad \forall \alpha \in (0, 1], \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n. \end{cases} \tag{8}$$

Moreover, for each $\alpha \leq \beta \in (0, 1]$, we have

$$y_i \left(w^T m_{(\tilde{x}_i)_\alpha} + b\right) - |w|^T s_{(\tilde{x}_i)_\alpha} = (y_i(w^T \tilde{x}_i + b))_\alpha^L$$
$$\leq (y_i(w^T \tilde{x}_i + b))_\beta^L = y_i \left(w^T m_{(\tilde{x}_i)_\beta} + b\right) - |w|^T s_{(\tilde{x}_i)_\beta}$$

Therefore, the program (8) can be restated as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \begin{cases} y_i \left(w^T m_{(\tilde{x}_i)_{0^+}} + b\right) - |w|^T s_{(\tilde{x}_i)_{0^+}} \geq 1 - \xi_i, \\ \qquad\qquad i = 1, 2, \ldots, n, \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \tag{9}$$

The program (9) is a crisp, quadratic and convex program with linear constraints. Therefore, its global optimal

solution can be obtained easily. This program can be restated as follows:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to} \begin{cases} \min_{\tilde{x}_i}\{y_i(w^T\tilde{x}_i + b)\} \geq 1 - \xi_i, & i = 1, 2, \ldots, n, \\ \xi_i \geq 0, & i = 1, 2, \ldots, n \end{cases} \tag{10}$$

The support of each fuzzy training data can be shown by a hypercube. Indeed, the program (10) finds a hyperplane such that the nearest point of the hypercube of each training sample to the hyperplane, namely $x_i = \text{argmin}_{\tilde{x}_i}\{y_i(w^T\tilde{x}_i + b)\}, i = 1, 2, \ldots, n,$ is separated correctly, namely $\min_{\tilde{x}_i}\{y_i(w^T\tilde{x}_i + b)\} \geq 1 - \xi_i, i = 1, 2, \ldots, n.$ Moreover, this hyperplane have the widest symmetric margin to these nearest points because in the program (10), $\|w\|^2$ is minimized or equivalently the symmetric margin, namely $M = \frac{1}{\|w\|^2}$, is maximized.

3.1.2. Test phase Let w and b be the optimal solution of the program (9), and $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)^T$ be a fuzzy test sample whose components are fuzzy numbers. One can use sign(defuzzify($f(\tilde{x})$)) as a crisp decision, where $f(\tilde{x}) = w^T\tilde{x} + b$, and defuzzify(.) is a defuzzification method such as the Roubens method.

The membership function of $f(\tilde{x})$ can be obtained by using Zadeh's extension principle, but for example, when the components of $\tilde{x}$, namely $\tilde{x}_i = (m_{\tilde{x}_i}, l_{\tilde{x}_i}, r_{\tilde{x}_i})_{LR}$ $(i = 1, 2, \ldots, p)$, are LR-type fuzzy numbers, from Theorem (2.1.3), $f(\tilde{x})$ becomes LR-type fuzzy number and its mean, left spread and right spread can be obtained as follows:

$$m_{f(\tilde{x})} = \sum_{i=1}^{p} w_i m_{\tilde{x}_i} + b,$$
$$l_{f(\tilde{x})} = \sum_{w_i \geq 0} w_i l_{\tilde{x}_i} - \sum_{w_i < 0} w_i r_{\tilde{x}_i},$$
$$r_{f(\tilde{x})} = \sum_{w_i \geq 0} w_i r_{\tilde{x}_i} - \sum_{w_i < 0} w_i l_{\tilde{x}_i}$$

Since now, we explained necessary how to obtain a crisp decision for a fuzzy test sample. From Theorem (2.1.4), the fuzzy decision for the fuzzy test sample $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)^T$ is given by $T(f(\tilde{x}) \leq \tilde{0})$, where $\tilde{0}$ is fuzzy zero (an optional fuzzy set for representation of approximately zero). $T(f(\tilde{x}) \leq \tilde{0})$ calculates the truth value of the assertion that the fuzzy number $f(\tilde{x})$ is smaller than the fuzzy number $\tilde{0}$. In other words, $T(f(\tilde{x}) \leq \tilde{0})$ determines the membership value of the fuzzy test sample $\tilde{x}$ in the class -1 and $T(f(\tilde{x}) \geq \tilde{0})$ determines the membership value of the fuzzy test sample $\tilde{x}$ in the class $+1$.

When the components of the test sample $\tilde{x}$ are symmetric triangular fuzzy numbers, namely $\tilde{x}_i = (m_{\tilde{x}_i}, s_{\tilde{x}_i})_{TFN}$ $(i = 1, 2, \ldots, p)$, a simpler method can be used to obtain the value of $T(f(\tilde{x}) \leq \tilde{0})$. In this situation, from Theorem (2.1.3), $f(\tilde{x}) = (m_{f(\tilde{x})}, s_{f(\tilde{x})})_{TFN}$ becomes a symmetric triangular fuzzy number and its mean and its radius can be obtained as follows:

$$m_{f(\tilde{x})} = \sum_{i=1}^{p} w_i m_{\tilde{x}_i} + b,$$
$$s_{f(\tilde{x})} = \sum_{i=1}^{p} |w_i| s_{\tilde{x}_i}$$

and then, $T(f(\tilde{x}) \leq \tilde{0})$ can be obtained by using the method defined in Theorem 2.1.5.

3.2. Non-linear separable fuzzy data

3.2.1. Training phase The SVM for classification of non-linear separable fuzzy samples $\tilde{x}_i (i = 1, 2, \ldots, n)$ is formulated as follows:

$$\min_{\tilde{w},b,\xi} \frac{1}{2}w^T w + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to} \begin{cases} y_i(w^T\varphi(\tilde{x}_i) + b) \geq \tilde{1} - \xi_i, & i = 1, 2, \ldots, n, \\ \xi_i \geq 0, & i = 1, 2, \ldots, n \end{cases} \tag{11}$$

We showed in the Appendix that

$$w = \sum_{i=1}^{n} y_i \beta_i \varphi(\tilde{x}_i) \tag{12}$$

where $0 \leq \beta_i \leq C$ $(i = 1, 2, \ldots, n)$ and $\sum_{i=1}^{n} y_i\beta_i = 0$. Therefore, the program (11) can be restated as follows:

$$\min_{w,\beta,b,\xi} \frac{1}{2}w^T w + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to} \begin{cases} y_i(w^T\varphi(\tilde{x}_i) + b) \geq \tilde{1} - \xi_i, & i = 1, 2, \ldots, n, \\ w = \sum_{i=1}^{n} y_i\beta_i\varphi(\tilde{x}_i), & \\ 0 \leq \beta_i \leq C, & i = 1, 2, \ldots, n, \\ \sum_{i=1}^{n} y_i\beta_i = 0, & \\ \xi_i \geq 0, & i = 1, 2, \ldots, n \end{cases} \tag{13}$$

The fuzzy program (13) must be transformed into a crisp program. This crisp program must also be convex to be solved efficiently (Bazara *et al.*, 2006). This is a difficult task, if possible. Thus, in continue, we change the program (13) to a less accurate program such that the mentioned task becomes possible.

From equation (12), we conclude that the components of the weight vector w are fuzzy set. Let $m_{\tilde{x}_i} = (m_{\tilde{x}_{i1}}, m_{\tilde{x}_{i2}}, \ldots, m_{\tilde{x}_{ip}})^T$, where $m_{\tilde{x}_{ij}}$ is a member of the core of the fuzzy number $\tilde{x}_{ij}$, namely $\mu_{\tilde{x}_{ij}}(m_{\tilde{x}_{ij}}) = 1$. Thus, each component of the crisp vector $\bar{w} = \sum_{i=1}^{n} y_i\beta_i\varphi(m_{\tilde{x}_i})$ is a member of a component of the fuzzy vector $w = \sum_{i=1}^{n} y_i\beta_i\varphi(\tilde{x}_i)$, respectively. Therefore, we change the problem (13) as follows:

$$\min_{\bar{w},\beta,b,\xi} \frac{1}{2}\bar{w}^T\bar{w} + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \begin{cases} y_i(\bar{w}^T\varphi(\tilde{x}_i)+b) \geq \tilde{1} - \xi_i, & i = 1,2,\ldots,n, \\ \bar{w} = \sum_{i=1}^{n} y_i\beta_i\varphi(m_{\tilde{x}_i}), & \\ 0 \leq \beta_i \leq C, & i = 1,2,\ldots,n, \\ \sum_{i=1}^{n} y_i\beta_i = 0, & \\ \xi_i \geq 0, & i = 1,2,\ldots,n \end{cases} \tag{14}$$

which can be restated as follows:

$$\min_{\beta,b,\xi} \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} y_j y_k \beta_j\beta_k K\left(m_{\tilde{x}_j}, m_{\tilde{x}_k}\right) + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \begin{cases} \sum_{j=1}^{n} y_i y_j \beta_j K\left(\tilde{x}_i, m_{\tilde{x}_j}\right) + y_i b \geq \tilde{1} - \xi_i, & \\ \qquad\qquad i = 1,2,\ldots,n, & \\ 0 \leq \beta_i \leq C, \quad i = 1,2,\ldots,n, & \\ \sum_{i=1}^{n} y_i\beta_i = 0, & \\ \xi_i \geq 0, \quad i = 1,2,\ldots,n & \end{cases} \tag{15}$$

Let $\tilde{v}_i = \sum_{j=1}^{n} y_i y_j \beta_j K\left(\tilde{x}_i, m_{\tilde{x}_j}\right)$. Since each component of $\tilde{x}_i$ was considered to be a fuzzy set, $\tilde{v}_i$ is also a fuzzy set. Therefore, from Definition (2.1.5), the program (15) is transformed into the following program:

$$\min_{\beta,b,\xi} \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} y_j y_k \beta_j\beta_k K(m_{\tilde{x}_j}, m_{\tilde{x}_k}) + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \begin{cases} m_{(\tilde{v}_i)_\alpha} - s_{(\tilde{v}_i)_\alpha} + y_i b \geq m_{(\tilde{1})_\alpha} - s_{(\tilde{1})_\alpha} - \xi_i, & \\ \qquad i = 1,2,\ldots,n, \ \forall\alpha \in (0,1], & \\ m_{(\tilde{v}_i)_\alpha} + s_{(\tilde{v}_i)_\alpha} + y_i b \geq m_{(\tilde{1})_\alpha} + s_{(\tilde{1})_\alpha} - \xi_i, & \\ \qquad i = 1,2,\ldots,n, \ \forall\alpha \in (0,1], & \\ 0 \leq \beta_i \leq C, \quad i = 1,2,\ldots,n, & \\ \sum_{i=1}^{n} y_i\beta_i = 0, & \\ \xi_i \geq 0, \quad i = 1,2,\ldots,n & \end{cases} \tag{16}$$

where $m_{(\tilde{v}_i)_\alpha}$ and $s_{(\tilde{v}_i)_\alpha}$ are the midpoint and the radius of the fuzzy number $\tilde{v}_i$ at α-cut, respectively. Also, $m_{(\tilde{1})_\alpha}$ and $s_{(\tilde{1})_\alpha}$ are the midpoint and the radius of the fuzzy number $\tilde{1}$ at α-cut, respectively. If the support of $\tilde{1}$ is $\{1\}$, the program (16) is restated as follows:

$$\min_{\beta,b,\xi} \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} y_j y_k \beta_j\beta_k K(m_{\tilde{x}_j}, m_{\tilde{x}_k}) + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \begin{cases} m_{(\tilde{v}_i)_\alpha} - s_{(\tilde{v}_i)_\alpha} + y_i b \geq 1 - \xi_i, & \\ \qquad i = 1,2,\ldots,n, \ \forall\alpha \in (0,1], & \\ m_{(\tilde{v}_i)_\alpha} + s_{(\tilde{v}_i)_\alpha} + y_i b \geq 1 - \xi_i, & \\ \qquad i = 1,2,\ldots,n, \ \forall\alpha \in (0,1], & \\ 0 \leq \beta_i \leq C, \quad i = 1,2,\ldots,n, & \\ \sum_{i=1}^{n} y_i\beta_i = 0, & \\ \xi_i \geq 0, \quad i = 1,2,\ldots,n & \end{cases} \tag{17}$$

Since $\forall\alpha \in (0,1]$,

$$m_{(\tilde{v}_i)_\alpha} - s_{(\tilde{v}_i)_\alpha} = (\tilde{v}_i)_\alpha^L \leq (\tilde{v}_i)_\alpha^U = m_{(\tilde{v}_i)_\alpha} + s_{(\tilde{v}_i)_\alpha}, \quad i = 1,2,\ldots,n$$

The program (17) is simplified as follows:

$$\min_{\beta,b,\xi} \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} y_j y_k \beta_j\beta_k K(m_{\tilde{x}_j}, m_{\tilde{x}_k}) + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \begin{cases} m_{(\tilde{v}_i)_\alpha} - s_{(\tilde{v}_i)_\alpha} + y_i b \geq 1 - \xi_i, & \\ \qquad i = 1,2,\ldots,n, \ \forall\alpha \in (0,1], & \\ 0 \leq \beta_i \leq C, \quad i = 1,2,\ldots,n, & \\ \sum_{i=1}^{n} y_i\beta_i = 0, & \\ \xi_i \geq 0, \quad i = 1,2,\ldots,n & \end{cases} \tag{18}$$

and since $\forall\alpha \leq \beta \in (0,1]$,

$$m_{(\tilde{v}_i)_\alpha} - s_{(\tilde{v}_i)_\alpha} = (\tilde{v}_i)_\alpha^L \leq (\tilde{v}_i)_\beta^L = m_{(\tilde{v}_i)_\beta} - s_{(\tilde{v}_i)_\beta}, \ i = 1,2,\ldots,n$$

the program (18) is simplified as follows:

$$\min_{\beta,b,\xi} \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} y_j y_k \beta_j\beta_k K(m_{\tilde{x}_j}, m_{\tilde{x}_k}) + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \begin{cases} m_{(\tilde{v}_i)_{0^+}} - s_{(\tilde{v}_i)_{0^+}} + y_i b \geq 1 - \xi_i, & i = 1,2,\ldots,n, \\ 0 \leq \beta_i \leq C, & i = 1,2,\ldots,n, \\ \sum_{i=1}^{n} y_i\beta_i = 0, & \\ \xi_i \geq 0, & i = 1,2,\ldots,n \end{cases} \tag{19}$$

If we obtain the value of $m_{(\tilde{v}_i)_{0^+}}$ and $s_{(\tilde{v}_i)_{0^+}}$, the program (19) becomes a convex quadratic program. To do so, let $\tilde{z}_{ij} = K(\tilde{x}_i, m_{\tilde{x}_j})$. 0^+-cuts of the fuzzy number $\tilde{z}_{ij}$ can be obtained by using an optimization method (see the following subsection). Let $(\tilde{z}_{ij})_{0^+} = [(\tilde{z}_{ij})_{0^+}^L, (\tilde{z}_{ij})_{0^+}^U]$ be 0^+-cut of the fuzzy number $\tilde{z}_{ij}$. Then,

$$m_{(\tilde{z}_{ij})_{0^+}} = \frac{(\tilde{z}_{ij})_{0^+}^L + (\tilde{z}_{ij})_{0^+}^U}{2},$$

$$s_{(\tilde{z}_{ij})_{0^+}} = \frac{(\tilde{z}_{ij})^U_{0^+} - (\tilde{z}_{ij})^L_{0^+}}{2}$$

are the midpoint and the radius of $(\tilde{z}_{ij})_{0^+}$, respectively. From Theorem 2.1.1, $m_{(\tilde{v}_i)_{0^+}} = \sum_{j=1}^{n} y_i y_j \beta_j m_{(\tilde{z}_{ij})_{0^+}}$, but since the terms of $(\tilde{v}_i)_{0^+}$, namely $(\tilde{z}_{ij})_{0^+}$ for $j = 1, 2, \ldots, n$, are dependent on each other, from Theorem 2.1.2,

$$s_{(\tilde{v}_i)_{0^+}} \leq \sum_{j=1}^{n} \beta_j s_{(\tilde{z}_{ij})_{0^+}}$$

Clearly, we have

$$\forall j : s_{(\tilde{v}_i)_{0^+}} \geq \beta_j s_{(\tilde{z}_{ij})_{0^+}}$$

where $\beta_j s_{(\tilde{z}_{ij})_{0^+}}$ is the spread of one of the terms $(\tilde{v}_i)_{0^+}$. Therefore,

$$s_{(\tilde{v}_i)_{0^+}} \geq \frac{1}{n} \sum_{j=1}^{n} \beta_j s_{(\tilde{z}_{ij})_{0^+}} \tag{20}$$

Thus, the program (19) can be approximated as follows:

$$\min_{\beta,b,\xi} \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} y_j y_k \beta_j \beta_k K\left(m_{\tilde{x}_j}, m_{\tilde{x}_k}\right) + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \begin{cases} \displaystyle\sum_{j=1}^{n} y_i y_j \beta_j m_{(\tilde{z}_{ij})_{0^+}} - \frac{1}{n} \sum_{j=1}^{n} \beta_j s_{(\tilde{z}_{ij})_{0^+}} \\ \quad + y_i b \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n, \\ 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n, \\ \displaystyle\sum_{i=1}^{n} y_i \beta_i = 0, \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \tag{21}$$

which is a convex quadratic program and its global optimal solution can be obtained easily. Indeed, the program (21) approximates the following program:

$$\min_{\beta,b,\xi} \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} y_j y_k \beta_j \beta_k K\left(m_{\tilde{x}_j}, m_{\tilde{x}_k}\right) + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \begin{cases} \min\limits_{\tilde{x}_i} \left\{ \displaystyle\sum_{j=1}^{n} y_i y_j \beta_j K\left(\tilde{x}_i, m_{\tilde{x}_j}\right) + y_i b \right\} \geq 1 - \xi_i, \\ \qquad\qquad i = 1, 2, \ldots, n, \\ 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n, \\ \displaystyle\sum_{i=1}^{n} y_i \beta_i = 0, \\ \xi_i \geq 0, \qquad i = 1, 2, \ldots, n \end{cases} \tag{22}$$

or the following program:

$$\min_{w,\beta,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \begin{cases} \min\limits_{\tilde{x}_i} \{y_i (w^T \varphi(\tilde{x}_i) + b)\} \geq 1 - \xi_i, \ i = 1, 2, \ldots, n, \\ \xi_i \geq 0, \qquad i = 1, 2, \ldots, n \end{cases} \tag{23}$$

The support of each fuzzy training data can be shown by a hypercube. The program (23) finds a hyperplane in a high-dimensional feature space such that the nearest point of the hypercube of each training sample to the hyperplane, namely $x_i = \text{argmin}_{\tilde{x}_i} \{y_i (w^T \varphi(\tilde{x}_i) + b)\}$, $i = 1, 2, \ldots, n$, is separated, namely $\min_{\tilde{x}_i} \{y_i (w^T \varphi(\tilde{x}_i) + b)\} \geq 1 - \xi_i$, $i = 1, 2, \ldots, n$. Moreover, this hyperplane in the high-dimensional feature space have the widest symmetric margin to these nearest points because in the program (3.1.13), $\|w\|^2$ is minimized or equivalently the symmetric margin, namely $M = \frac{1}{\|w\|^2}$, is maximized.

3.2.2. Test phase Let $\beta_i (i = 1, \ldots, p)$ and b be the optimal solution of the program (21), and $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)^T$ be a fuzzy test sample whose components are fuzzy numbers. We can use $\text{sign}(f(\text{defuzzify}(\tilde{x})))$ as a crisp decision, where

$$f(x) = w^T \varphi(\text{x}) + b = \sum_{i=1}^{n} y_i \beta_i K(m_{\tilde{x}_i}, \text{x}) + b,$$

and defuzzify(.) is a defuzzification method such as the Roubens method.

3.3. Obtaining the lower bound and the upper bound of $(\tilde{z}_{ij})_{0^+}$

Without loss of generality, let $K(.,.)$ be a Gaussian kernel function. Thus, $\tilde{z}_{ij} = K(\tilde{x}_i, m_{\tilde{x}_j}) = \exp(\frac{-\|\tilde{x}_i - m_{\tilde{x}_j}\|^2}{2\sigma^2})$, where $\sigma > 0$ is a constant. The lower bound and the upper bound of $(\tilde{z}_{ij})_{0^+}$ can be obtained by using following two programs:

$$(\tilde{z}_{ij})^L_{0^+} = \min_{\text{x}} \exp\left(\frac{-\left\|x_i - m_{\tilde{x}_j}\right\|^2}{2\sigma^2}\right)$$

$$\text{subject to} (\tilde{x}_i)^L_{0^+} \leq x_i \leq (\tilde{x}_i)^U_{0^+}, \quad k = 1, 2, \ldots, p,$$

$$(\tilde{z}_{ij})^U_{0^+} = \max_{\text{x}} \exp\left(\frac{-\left\|x_i - m_{\tilde{x}_j}\right\|^2}{2\sigma^2}\right)$$

$$\text{subject to} (\tilde{x}_i)^L_{0^+} \leq x_i \leq (\tilde{x}_i)^U_{0^+}, \quad k = 1, 2, \ldots, p$$

In continue, we solve these two problems by using only some simple operations.

3.3.1. Solving the lower bound program Consider the following program:

$$U_{ij\alpha} = \max_{\text{x}} \left\|x_i - m_{\tilde{x}_j}\right\|^2$$

$$\text{subject to} (\tilde{x}_i)^L_{0^+} \leq x_i \leq (\tilde{x}_i)^U_{0^+}$$

This program can be restated as follows:

$$U_{ij} = \max_{\text{x}} \sum_{k=1}^{p} \left(x_{ik} - m_{\tilde{x}_{jk}}\right)^2$$

$$\text{subject to} (\tilde{x}_{ik})^L_{0^+} \leq x_{ik} \leq (\tilde{x}_{ik})^U_{0^+}, \quad k = 1, 2, \ldots, p$$

The value of U_{ij} can be calculated as follows:

$$U_{ij} = \sum_{\substack{|(\tilde{x}_{ik})_{0^+}^U - m_{\tilde{x}_{jk}}| \le |(\tilde{x}_{ik})_{0^+}^L - m_{\tilde{x}_{jk}}|, \\ k=1,2,\ldots,p}} \left((\tilde{x}_{ik})_{0^+}^L - m_{\tilde{x}_{jk}}\right)^2 + \sum_{\substack{|(\tilde{x}_{ik})_{0^+}^U - m_{\tilde{x}_{jk}}| > |(\tilde{x}_{ik})_{0^+}^L - m_{\tilde{x}_{jk}}|, \\ k=1,2,\ldots,p}} \left((\tilde{x}_{ik})_{0^+}^U - m_{\tilde{x}_{jk}}\right)^2$$

Finally, we have

$$(\tilde{z}_{ij})_{0^+}^L = \exp\left(\frac{-U_{ij}}{2\sigma^2}\right)$$

3.3.2. Solving the upper bound program Consider the following program:

$$L_{ij} = \min_{\mathrm{x}} \|x_i - m_{\tilde{x}_j}\|^2$$
$$\text{subject to } (\tilde{x}_i)_{0^+}^L \le x_i \le (\tilde{x}_i)_{0^+}^U$$

This program can be restated as follows:

$$L_{ij} = \min_{\mathrm{x}} \sum_{k=1}^{p} \left(x_{ik} - m_{\tilde{x}_{jk}}\right)^2$$
$$\text{subject to } (\tilde{x}_{ik})_{0^+}^L \le x_{ik} \le (\tilde{x}_{ik})_{0^+}^U, \quad k = 1, 2, \ldots, p$$

When $(\tilde{x}_{ik})_{0^+}^L \le m_{\tilde{x}_{jk}} \le (\tilde{x}_{ik})_{0^+}^U$, the minimum of $(x_{ik} - m_{\tilde{x}_{jk}})^2$ subject to $(\tilde{x}_{ik})_{0^+}^L \le x_{ik} \le (\tilde{x}_{ik})_{0^+}^U$ becomes zero. Therefore, we have

$$L_{ij} = \sum_{\substack{(|(\tilde{x}_{ik})_{0^+}^U - m_{\tilde{x}_{jk}}| \ge |(\tilde{x}_{ik})_{0^+}^L - m_{\tilde{x}_{jk}}|) \text{ and} \\ (((\tilde{x}_{ik})_{0^+}^L > m_{\tilde{x}_{jk}}) \text{ or } (m_{\tilde{x}_{jk}} > (\tilde{x}_{ik})_{0^+}^U)), \\ k=1,2,\ldots,p}} \left((\tilde{x}_{ik})_{0^+}^L - m_{\tilde{x}_{jk}}\right)^2 + \sum_{\substack{|(\tilde{x}_{ik})_{0^+}^U - m_{\tilde{x}_{jk}}| < |(\tilde{x}_{ik})_{0^+}^L - m_{\tilde{x}_{jk}}| \text{and} \\ (((\tilde{x}_{ik})_{0^+}^L > m_{\tilde{x}_{jk}}) \text{ or } (m_{\tilde{x}_{jk}} > (\tilde{x}_{ik})_{0^+}^U)), \\ k=1,2,\ldots,p}} \left((\tilde{x}_{ik})_{0^+}^U - m_{\tilde{x}_{jk}}\right)^2$$

Finally, we have

$$(\tilde{z}_{ij})_{0^+}^U = \exp\left(\frac{-L_{ij}}{2\sigma^2}\right)$$

4. Numerical examples and discussion

Consider the 10 interval-valued training samples plotted in Figure 3. In this Figure, each box determines the spreads of a sample and each +/* symbol shows the midpoint of a sample with the class label +1/–1. Figures 4(a) and 4(b) plot two classifiers obtained for these training samples by using the linear version of our proposed method, namely the program (9), and the well-known SVM for $\varphi(x) = x$, respectively. We used the midpoint of each interval-valued training sample for the train and the test phase of the well-known SVM. As can be seen, linear version of our proposed method has classified samples better or more robust than the well-known SVM. Our proposed method has found the classifier such that the nearest points of the box of training samples to the classifier are classified with the widest symmetric margin,

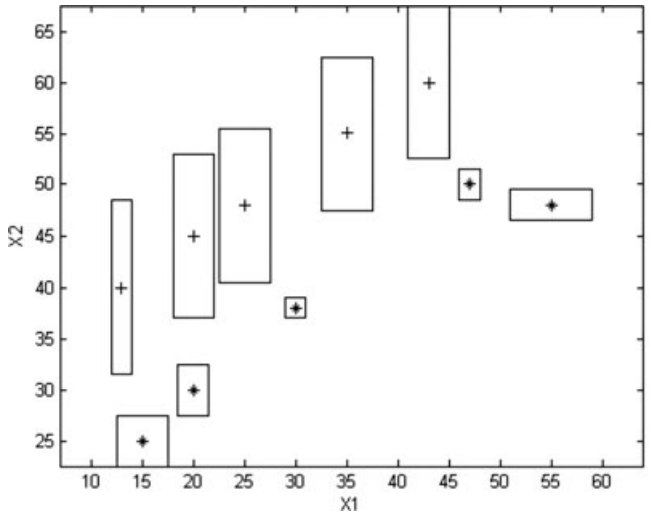

Figure 3: *Ten interval-valued samples; +: midpoint of sample of class +1; *: midpoint of sample of class –1.*

whereas the well-known SVM has found the classifier such that the midpoint of training samples are classified with the widest symmetric margin. In other words, we could classify the whole of the box of each sample by using our proposed method, whereas the well-known SVM could not classify the boxes completely. Therefore, each point of each box (as a test sample) is classified correctly when the classifier of our proposed method is used for data classification, whereas it may be not classified correctly when the classifier of the well-known SVM is used for that. Therefore, we can expect that the classification rate of our proposed method to be better than the well-known SVM.

Figures 5 and 6 plots four classifiers obtained for the same 10 training samples by using four classification methods (non-linear version of our proposed method, namely the program (21), the well-known SVM, the Holder-ISVM and the Distance-ISVM in a high-dimensional feature space) for the Gaussian kernel function and $\sigma = 5.0/3.5$. As can be seen, our proposed method has classified samples better than the three other classification methods. Our proposed method has been tried to obtain a hyperplane in the high-dimensional feature space such that the nearest point of each training sample to the hyperplane are separated with the widest symmetric margin (Figures 5(a) and 6(a)), whereas the well-known SVM (Figures 5(b) and 6(b)) has found the classifier such that only the midpoint of training samples are classified in the high-dimensional feature space. In other words, again, our proposed method has considered the spreads of training samples in its train phase whereas the well-known SVM (which is a fuzzy inference system (Chen & Wang, 2003)) has ignored the spreads and has used only the midpoints of training samples in its train phase.

As can be seen in Figures 5(c) and 6(c), the Holder-ISVM has not classified samples correctly for the mentioned values of σ (all points of the plane have classified as the members of only one class). The Holder-ISVM considers the spreads of training samples in its train phase, but some constraints of the Holder-ISVM have been substituted by using the Holder inequality for computation issues. The difference between the left and right side of the Holder inequality may be high which makes this substitution inaccurate. Therefore, sometimes the

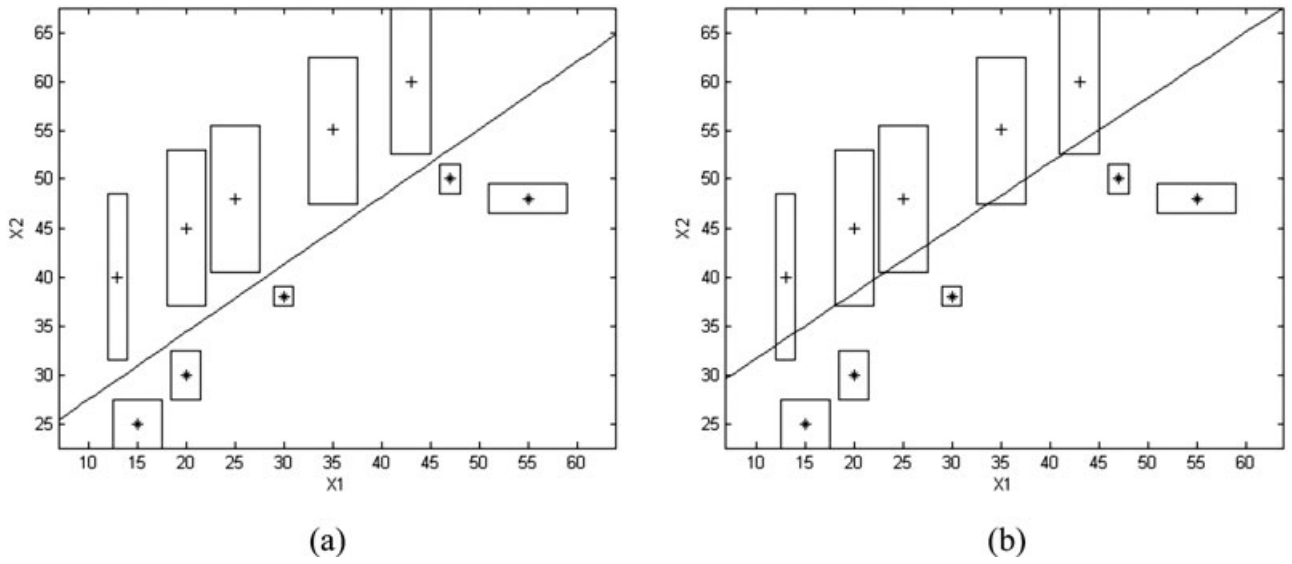

Figure 4: *Classification of some interval-valued data by using (a) linear version of our proposed method and(b) well-known SVM in input space, for C = 1000; +: midpoint of sample of class +1; *: midpoint of sample of class –1.*

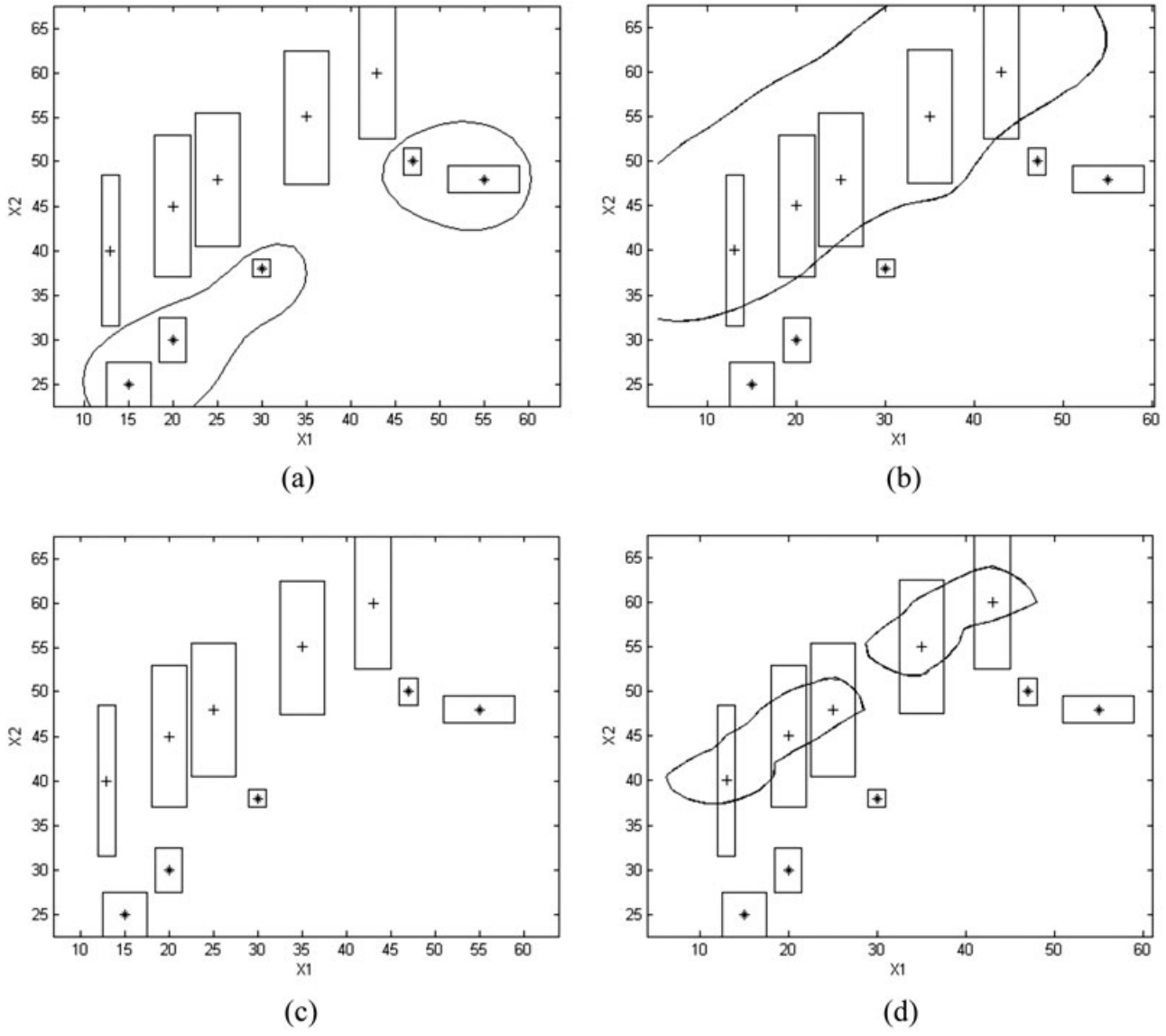

Figure 5: *Classification of some interval-valued data by using (a) non-linear version of our proposed method, (b) well-known SVM, (c) Holder-ISVM and (d) Distance-ISVM for $\sigma = 0.5$ and C = 1000. +: midpoint of sample of class +1; *: midpoint of sample of class –1.*

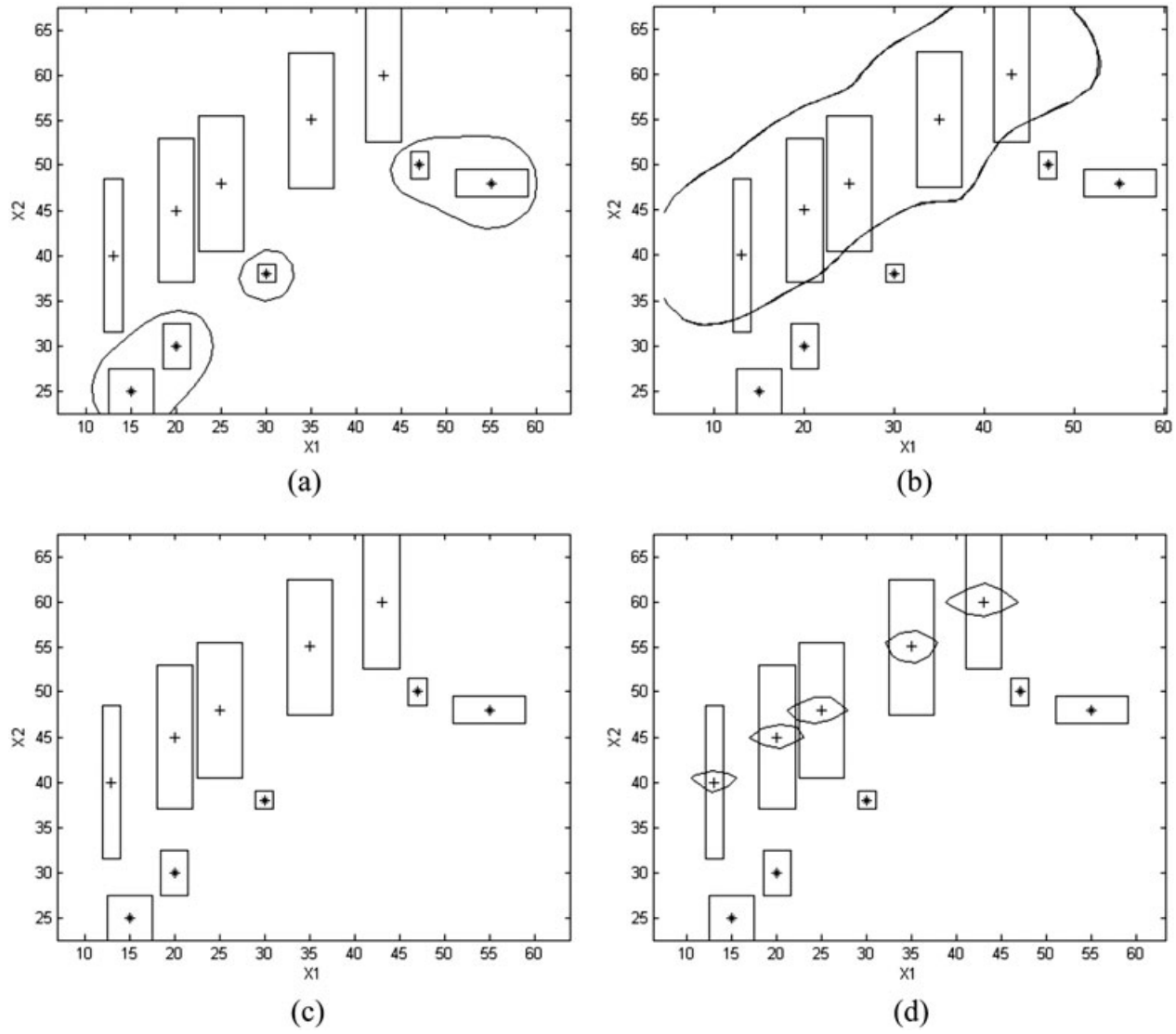

Figure 6: *Classification of some interval-valued data by using (a) non-linear version of our proposed method, (b) well-known SVM, (c) Holder-ISVM and (d) Distance-ISVM for* $\sigma = 0.35$ *and* $C = 1000$. *+: midpoint of sample of class +1; *: midpoint of sample of class –1.*

Holder-ISVM cannot classify interval-valued samples properly. Our experimental results on real data sets show that the probability of misclassification of this method is high (See the following section).

In the Distance-ISVM (which is also a fuzzy inference system), the Hausdorff distance is used instead of the Euclidean distance in the Gaussian kernel function. The Euclidean distance for two interval values is an interval value that makes the SVM problem a fuzzy problem (which must be transformed into an equivalent crisp problem). The Hausdorff distance for two interval values is a crisp value that makes the SVM problem, a crisp problem or a standard quadratic problem. The Hausdorff distance considers the spread of the interval values. Therefore, we expect that the Distance-ISVM get a lower probability of misclassification than the well-known SVM that uses the Euclidean distance between the midpoints of interval values and ignores the spread of interval values. But, since non-liner version of our proposed method tries to obtain a hyperplane in the high-dimensional feature space such that the nearest point of the box of each training sample to the hyperplane are separated with the widest symmetric margin, we expect to obtain a more accurate classifier than the Distance-ISVM (see Figures 5(d) and 6(d) and compare them by Figures 5(a) and 6(a)). Our experimental results on real data sets also confirm this issue (see section 5).

5. Application: multi-class classification of some multi-class real data sets

In this section, we compare the non-linear version of our proposed method, namely the program (21), by the ISVM, Distance-ISVM, Holder-ISVM and also the well-known SVM. To do so, we use some multi-class interval data sets, namely the 'car', 'fish' and 'temperature' data sets, and also some multi-class UCI data sets (Blake & Merz, 1998). The 'car' data set have been used previously in some studies (Duarte-Silva & Brito, 2006; Carrizosa *et al.*, 2007), and the UCI data sets is the most commonly used data sets for comparing classification methods (Tax, 2001; Chen & Wang, 2003; Webb, 2003; Do & Poulet, 2005, 2006; Wang, 2005; Lin *et al.*, 2006; Juang *et al.*, 2007). We use the midpoint of each

interval value when we use the well-known SVM for classification of interval-valued data.

We use the Gaussian kernel function for the Distance-ISVM, Holder-ISVM method and our novel method. To obtain the optimal value of σ for each of these three classification methods, we do some experiments and in each experiment we set σ to a different value, namely 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.3, 0.4, 0.5 and 1 for small data sets and 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.5 for larger data sets, and then we use the best result of each classification method for comparison. We also report the optimal value of σ for each experiment. We set the penalty term C to a large value, namely 1000. Setting a large value to the penalty term means that we want that all of training samples are separated by the objective classifiers (training samples of class $+1(-1)$ are positioned only in the left (right) side of objective classifier if possible). Meanwhile, we use the Roubens defuzzification method in the test phase of our proposed method.

Solving a multi-class classification problem (see Vapnik, 1998; Herbrich, 2002; Scholkopf & Smola, 2002) is, in general, a more difficult task than solving a two-class classification problem. Different strategies have been proposed. Most of these suggest transforming the multi-class problem in a series of two-class problems to be solved such as one-versus-one and one-versus-rest techniques (see e.g. Friedman, 1996; Hastie & Tibshirani, 1998; Weston & Watkins, 1998; Platt *et al.*, 2000; Hsu & Lin, 2002; Tax & Duin, 2002)). In this paper, we use the one-versus-rest (one-against-all) technique that is a commonly used technique, and is less time consuming than one-versus-one (one-against-one) (Carrizosa *et al.*, 2007). In this technique, for multi-class classification problem, N two-class classifiers are constructed, where N is the number of distinct class labels; k-th classifier is the result of solving a two-class classification problem where the elements of k-th group have a label $+1$ associated and the rest of elements have a label -1 associated. Then, the test sample x is assigned to class l if

$$f_l(x) = \max_{k=1,2,\ldots,N} f_k(x)$$

where $f_k(.)$ is k-th classifier (See equation (4)).

To reduce the time complexity of the procedure of measuring the probability of misclassification of each classification method for a large data set, 10-fold cross validation is usually used (Kohavi, 1995; Abe & Inoune, 2002), that is, the instances of the data set are grouped in 10 sets (these sets forming a partition) and, each one is used in turn as test set against all nine others taken together as training set, that is, the process is repeated 10 times and then the probability of misclassification is obtained as follows:

$$\text{probability of misclassification} = \frac{\#\text{of misclassifications}}{\#\text{of test samples}} \quad (24)$$

We do not use 10-fold cross validation strategy for small data sets because onefold (test fold) may contain all samples of a class, and therefore, nine other folds may contain no samples of that class. This can hurt the accuracy of equation (24). In order to measure the probability of misclassification of each classification method for a small data set, we use another commonly used strategy called the leave-one-out strategy (see e.g. Kohavi, 1995; Hand *et al.*, 2001; Abe & Inoune, 2002), that is, in turns, we consider only one element in the test sample set, we train the model with the remaining elements and we test this model with the unitary test sample. We repeat the process for every element of the data set and then obtain the probability of misclassifications by using equation (24).

5.1. *'car' data set*

The 'car' data set is a database with 33 car models described by eight interval variables (explaining the following characteristics of each car model: price, engine capacity, top speed, acceleration, step, length, width and height) and one nominal variable which represents one of the four following possible categories: utilitarian, berlina, sportive or luxury (see Duarte-Silva & Brito, 2006 for more details).

Table 1 shows probability of misclassification of different classification methods for the optimal value of σ and the leave-one-out strategy. As can be seen, our proposed method and the Distance-SVM are the most accurate classification methods for this data set.

5.2. *'Fish' data set*

Several studies realized in France have pointed out abnormal levels of mercury contamination in some Amerindian populations. This contamination is connected to their high consumption of contaminated freshwater fish (Bobou & Ribeyre, 1998). In order to get a better knowledge of this phenomenon, a data set has been collected by researchers from LEESA (Laboratoire d'Ecophysiologie et d'Ecotoxicologie des Systemes Aquatiques) laboratory. This data set concerns 12 fish species, each species being described by 13 interval variables and one categorical variable. These species are grouped in four classes: carnivorous, detritivorous, omnivorous and herbivorous. Table 2 shows probability of misclassification of different classification methods for the optimal value of σ and the leave-one-out strategy. As can be seen, our proposed method, the Distance-SVM and the well-known SVM are the most accurate classification methods for this data set.

Table 1: *Probability of misclassification of different classification methods for the optimal value of σ, the leave-one-out strategy and for the 'car' data set*

	Well-known SVM	*ISVM*	*Holder-ISVM*	*Distance-ISVM*	*Non-linear version of our proposed method*
Optimal value of σ	0.2	–	0.02	0.2	0.2
Probability of misclassification	0.21	0.18	0.24	0.15	0.15

Table 2: *Probability of misclassification of different classification methods for the optimal value of σ, the leave-one-out strategy and for the 'fish' data set.*

	Well-known SVM	*ISVM*	*Holder-ISVM*	*Distance-ISVM*	*Non-linear version of our proposed method*
Optimal value of σ	0.5	–	0.06	0.5	1.0
Probability of misclassification	0.17	0.25	0.18	0.17	0.17

Table 3: *Probability of misclassification of different classification methods for the optimal value of σ, the leave-one-out strategy and for the 'temperature' data set*

	Well-known SVM	*ISVM*	*Holder-ISVM*	*Distance-ISVM*	*Non-linear version of our proposed method*
Optimal value of σ	0.2	–	0.5	0.3	0.2
Probability of misclassification	0.05	0.05	0.09	0.05	0.05

5.3. 'Temperature' data set

This interval-valued data set (Guru *et al.*, 2004; Souza & De Carvalho, 2010) concerns 37 cities, each city is described by 12 interval-valued variables, which are minimum and the maximum temperatures of 12 months in degree centigrade. A priori classification given by a panel of human observers is as follows:

Class 1: Bahrain, Bombay, Cairo, Calcutta, Colombo, Dubai, Hong Kong, Kulalampur, Madras, Manila, Mexico, Nairobi, New Delhi, Sydney and Singapore.

Class 2: Amsterdam, Athens, Copenhagen, Frankfurt, Geneva, Lisbon, London, Madrid, Moscow, Munich, New York, Paris, Rome, San Francisco, Seoul, Stockholm, Tokyo, Toronto, Vienna and Zurich.

Class 3: Mauritius.

Class 4: Tehran.

We used this data set for multi-class classification. Table 3 shows probability of misclassification of different classification methods for the optimal value of σ and the leave-one-out strategy. As can be seen, four classification methods, namely the well-known SVM, ISVM, Distance-ISVM and our proposed method, have identical probability of misclassification for this data set.

Table 4: *The properties of three data sets of the UCI repository*

	No. of instances	*No. of features*	*No. of classes*
Glass data set	214	9	6
Wine data set	178	13	3
Iris data set	150	4	3

5.4. Some data sets of the UCI repository

We see in the previous sub-section that, first, the probability of misclassification of our proposed method is lower than or equal to the probability of misclassification of the other four classification methods for the three real interval data sets, and, second, the probability of misclassification of the Holder-ISVM is much higher than the probability of misclassification of the others. In continue, we apply our proposed method for classification of three data sets of the UCI repository (Blake & Merz, 1998), and then compare it only by the three comparable classification methods, namely the well-known SVM, ISVM and Distance-ISVM. The properties of these three data sets of the UCI repository have been shown in Table 4.

In order to construct the interval data set from a crisp data set, we compute the standard deviation of each feature of that data set, say $\sigma_j (j = 1, 2, \ldots, p)$. Then, j-th feature of i-th sample, x_{ij}, is replaced by the interval $[x_{ij} - kr\sigma_j, x_{ij} + k(2-r)\sigma_j]$, where r is a uniform distributed random variable between 0 and 1, and the values for k are 0, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75 or 1.0. The higher the value of k, the larger the uncertainty about the data. Observe that when $k = 0$, we obtain the original data set.

Tables 5–7 show probability of misclassification of different classification methods for the optimal value of σ, different values of k, 10-fold cross validation strategy and for the 'glass', 'wine' and 'iris' data sets, respectively. As can be seen, our proposed method is the most accurate classification method for all of the above different conditions.

Table 5: *Probability of misclassification of different classification methods for the optimal value of σ, for the 'glass' data set and for different values of k*

	Well-known SVM	*ISVM*	*Distance-ISVM*	*Non-linear version of our proposed method*	
Lowest probability of misclassification	0.37 ($\sigma = 0.1$)	0.49	0.37 ($\sigma = 0.1$)	0.27 ($\sigma = 0.1$)	$k = 0.0$
	0.33 ($\sigma = 0.1$)	0.46	0.33 ($\sigma = 0.1$)	0.29 ($\sigma = 0.5$)	k = 0.01
	0.33 ($\sigma = 0.1$)	0.52	0.33 ($\sigma = 0.1$)	0.29 ($\sigma = 0.2$)	$k = 0.05$
	0.30 ($\sigma = 0.1$)	0.48	0.30 ($\sigma = 0.1$)	0.29 ($\sigma = 0.2$)	$k = 0.1$
	0.30 ($\sigma = 0.1$)	0.45	0.30 ($\sigma = 0.1$)	0.29 ($\sigma = 0.1$)	$k = 0.25$
	0.39 ($\sigma = 0.1$)	0.52	0.39 ($\sigma = 0.1$)	0.33 ($\sigma = 0.2$)	$k = 0.5$
	0.47 ($\sigma = 0.1$)	0.63	0.47 ($\sigma = 0.1$)	0.38 ($\sigma = 0.2$)	$k = 0.75$
	0.51 ($\sigma = 0.1$)	0.63	0.51 ($\sigma = 0.1$)	0.42 ($\sigma = 0.08$)	$k = 1.0$

Table 6: *Probability of misclassification of different classification methods for the optimal value of σ, for the 'wine' data set and for different values of k*

	Well-known SVM	*ISVM*	*Distance-ISVM*	*Non-linear version of our proposed method*	
Lowest probability of misclassification	0.03 ($\sigma = 0.5$)	0.04	0.03 ($\sigma = 0.5$)	0.02 ($\sigma = 0.5$)	$k = 0.0$
	0.03 ($\sigma = 0.5$)	0.03	0.03 ($\sigma = 0.5$)	0.02 ($\sigma = 0.5$)	k = 0.01
	0.03 ($\sigma = 0.5$)	0.03	0.03 ($\sigma = 0.5$)	0.02 ($\sigma = 0.5$)	$k = 0.05$
	0.03 ($\sigma = 0.5$)	0.02	0.03 ($\sigma = 0.5$)	0.02 ($\sigma = 0.5$)	$k = 0.1$
	0.02 ($\sigma = 0.5$)	0.02	0.02 ($\sigma = 0.5$)	0.02 ($\sigma = 0.5$)	$k = 0.25$
	0.02 ($\sigma = 0.5$)	0.16	0.02 ($\sigma = 0.5$)	0.01 ($\sigma = 0.5$)	$k = 0.5$
	0.06 ($\sigma = 0.5$)	0.18	0.06 ($\sigma = 0.5$)	0.05 ($\sigma = 0.5$)	$k = 0.75$
	0.04 ($\sigma = 0.5$)	0.64	0.04 ($\sigma = 0.5$)	0.04 ($\sigma = 0.5$)	$k = 1.0$

Table 7: *Probability of misclassification of different classification methods for the optimal value of σ, for the 'iris' data set and for different values of k*

	Well-known SVM	*ISVM*	*Distance-ISVM*	*Non-linear version of our proposed method*	
Lowest probability of misclassification	0.06 ($\sigma = 0.1$)	0.40	0.06 ($\sigma = 0.1$)	0.03 ($\sigma = 0.5$)	$k = 0.0$
	0.05 ($\sigma = 0.5$)	0.22	0.05 ($\sigma = 0.5$)	0.03 ($\sigma = 0.5$)	$k = 0.01$
	0.05 ($\sigma = 0.5$)	0.21	0.05 ($\sigma = 0.5$)	0.03 ($\sigma = 0.5$)	$k = 0.05$
	0.05 ($\sigma = 0.5$)	0.23	0.05 ($\sigma = 0.5$)	0.03 ($\sigma = 0.5$)	$k = 0.1$
	0.08 ($\sigma = 0.08$)	0.18	0.08 ($\sigma = 0.08$)	0.03 ($\sigma = 0.5$)	$k = 0.25$
	0.15 ($\sigma = 0.5$)	0.19	0.15 ($\sigma = 0.5$)	0.05 ($\sigma = 0.5$)	$k = 0.5$
	0.18 ($\sigma = 0.5$)	0.28	0.18 ($\sigma = 0.5$)	0.10 ($\sigma = 0.2$)	$k = 0.75$
	0.23 ($\sigma = 0.5$)	0.57	0.24 ($\sigma = 0.5$)	0.14 ($\sigma = 0.04$)	$k = 1.0$

6. Conclusion

In this paper, we extended the SVM for robust classification of linear and also non-linear separable data whose features are fuzzy numbers. Each version of our proposed method, namely linear and non-linear version, consists of two phases: train and test.

The support of each fuzzy training data can be shown by a hypercube. Our proposed method tries to obtain a hyperplane in it train phase such that the nearest point of the hypercube of each fuzzy training sample to the hyperplane is separated with the widest symmetric margin. This strategy can reduce the misclassification probability of our proposed method.

We used our proposed method (non-linear version/crisp decision function) for classification of six real data sets. Our experimental results on these real data sets showed that the classification rate of our novel method is better than or equal to the classification rate of the well-known SVM, ISVM, Holder-ISVM and Distance-ISVM for all of these data sets.

Appendix

Theorem. The weight vector w in the following program can be replaced by equation (12):

$$\tilde{z} = \min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \begin{cases} y_i \left(w^T \varphi \left(\tilde{x}_i \right) + b \right) \geq 1 - \xi_i, & i = 1, 2, \ldots, n, \\ \xi_i \geq 0, & i = 1, 2, \ldots, n \end{cases} \tag{A.1}$$

Proof. Based on the extension principle, we have

$$\mu_{\tilde{z}}(z) = \sup_{x} \min\{\mu_{\tilde{x}_{ij}}(x_{ij}),\ \forall i, j \,|z = Z(\ x)\} \tag{A.2}$$

where $x = (x_1, \ldots, x_n)^T$ and $Z(x)$ is the function of the program of the following program:

$$Z = \min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \begin{cases} y_i \left(w^T \varphi \left(x_i \right) + b \right) \geq 1 - \xi_i, & i = 1, 2, \ldots, n, \\ \xi_i \geq 0, & i = 1, 2, \ldots, n \end{cases} \tag{A.3}$$

To drive $\mu_{\tilde{z}}$ by using equation (A.2) is hardly possible. To find $\mu_{\tilde{z}}$ it suffices to find the upper bound and the lower bound of objective function at each α-cut, named Z_α^U and Z_α^L, respectively (Liu, 2009). These bounds can be expressed as

$$Z_\alpha^L = \min_{x} \left\{ Z(x) \middle| \, (x_i)_\alpha^L \leq x_i \leq (x_i)_\alpha^U, \forall i \right\}, \tag{A.4}$$

$$Z_\alpha^U = \max_{x} \left\{ Z(x) \middle| \, (x_i)_\alpha^L \leq x_i \leq (x_i)_\alpha^U, \forall i \right\} \tag{A.5}$$

where $(x_i)_\alpha^L = ((x_{i1})_\alpha^L, (x_{i2})_\alpha^L, \ldots, (x_{ip})_\alpha^L)^T$ and $(x_i)_\alpha^U = ((x_{i1})_\alpha^U, (x_{i2})_\alpha^U, \ldots, (x_{ip})_\alpha^U)^T$. The bounds (A.4) and (A.5) can be determined from the following two-level mathematical programming models:

$$Z_\alpha^L = \min_x \begin{cases} \min\limits_{w,b,\xi} \dfrac{1}{2} w^T w + C \sum\limits_{i=1}^{n} \xi_i \\ \text{subject to} \begin{cases} y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i, \\ \qquad i = 1, 2, \ldots, n, \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \end{cases}$$
$$\text{subject to} (x_i)_\alpha^L \leq x_i \leq (x_i)_\alpha^U, i = 1, 2, \ldots, n \tag{A.6}$$

$$Z_\alpha^U = \max_x \begin{cases} \min\limits_{w,b,\xi} \dfrac{1}{2} w^T w + C \sum\limits_{i=1}^{n} \xi_i \\ \text{subject to} \begin{cases} y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i, \\ \qquad i = 1, 2, \ldots, n, \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \end{cases}$$
$$\text{subject to} (x_i)_\alpha^L \leq x_i \leq (x_i)_\alpha^U, i = 1, 2, \ldots, n \tag{A.7}$$

By using the Lagrangian dual form of the inner level program of (A.6) and (A.7), the program (A.6) and (A.7) can be restated as follows:

$$Z_\alpha^L = \min_x \begin{cases} \min\limits_{\beta,b,\xi} \dfrac{1}{2} \sum\limits_{j=1}^{n} \sum\limits_{k=1}^{n} y_j y_k \beta_j \beta_k K(x_j, x_k) + C \sum\limits_{i=1}^{n} \xi_i \\ \text{subject to} \begin{cases} y_i \left(\sum\limits_{j=1}^{n} y_j \beta_j K(x_i, x_j) + b \right) \\ \qquad \geq 1 - \xi_i, i = 1, 2, \ldots, n, \\ 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n, \\ \sum\limits_{i=1}^{n} y_i \beta_i = 0, \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \end{cases}$$
$$\text{subject to} (x_i)_\alpha^L \leq x_i \leq (x_i)_\alpha^U, \quad i = 1, 2, \ldots, n \tag{A.8}$$

$$Z_\alpha^U = \max_x \begin{cases} \min\limits_{\beta,b,\xi} \dfrac{1}{2} \sum\limits_{j=1}^{n} \sum\limits_{k=1}^{n} y_j y_k \beta_j \beta_k K(x_j, x_k) + C \sum\limits_{i=1}^{n} \xi_i \\ \text{subject to} \begin{cases} y_i \left(\sum\limits_{j=1}^{n} y_j \beta_j K(x_i, x_j) + b \right) \\ \qquad \geq 1 - \xi_i, i = 1, 2, \ldots, n, \\ 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n, \\ \sum\limits_{i=1}^{n} y_i \beta_i = 0, \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \end{cases}$$
$$\text{subject to} (x_i)_\alpha^L \leq x_i \leq (x_i)_\alpha^U, \quad i = 1, 2, , \ldots, n \tag{A.9}$$

where $\beta = \left(\beta_1, \beta_2, \ldots, \beta_n\right)^T$. Finally, the sub-problems (A.9) and (A.10) are the lower bound and the upper bound of the following program at α-cut:

$$\min_{\beta,b,\xi} \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} y_j y_k \beta_j \beta_k K(\tilde{x}_j, \tilde{x}_k) + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \begin{cases} y_i \left(\sum\limits_{j=1}^{n} y_j \beta_j K(\tilde{x}_i, \tilde{x}_j) + b \right) \\ \qquad \geq 1 - \xi_i, i = 1, 2, \ldots, n, \\ 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n, \\ \sum\limits_{i=1}^{n} y_i \beta_i = 0, \\ \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases} \tag{A.10}$$

By comparison of the programs (A.1) and (A.10), we can conclude

$$w = \sum_{i=1}^{n} y_i \beta_i \varphi(\tilde{x}_i) \tag{A.11}$$

where $0 \leq \beta_i \leq C (i = 1, 2, \ldots, n)$ and $\sum_{i=1}^{n} y_i \beta_i = 0$.

References

ABE, S. and T. INOUNE (2002) *Fuzzy Support Vector Machines for Multiclass Problems*, Belgium: European Symposium on Artificial Neural Networks.

BAZARA, M.S., H.D. SHERALI and C.M. SHETTY (2006) *Nonlinear Programming*, Hoboken, NJ: Wiley-Interscience.

BLAKE, C.L. and C.J. MERZ (1998) UCI repository of machine learning databases.

BOBOU, A. and F. RIBEYRE (1998) Mercury in the food web, in *Metal Ions in Biological Systems*, A. Sigrel and H. Sigrel (eds), New York: Marcel Dekker, 289–319.

BORTOLAN, G. and R. DEGANI (1985) A review of some methods for ranking fuzzy numbers, *Fuzzy Sets and Systems*, **15**, 1–19.

BURGES, C.J.C. (1998) A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, **2**, 121–167.

CARRIZOSA, E., J. GORDILLO and F. PLASTRIA (2007) *Classification Problems with Imprecise Data Through Separating Hyperplanes*, MOSI Department, Brussels, Belgium: Vrije University.

CHEN, Y. and J.Z. WANG (2003) Support vector learning for fuzzy rule-based classification systems, *IEEE Transactions on Fuzzy Systems*, **11**, 716–728.

CORTES, C. and V.N. VAPNIK (1995) Support vector networks, *Machine Learning*, **20**, 273–297.

DENOEUX, T. (1995) A k-nearest neighbor classification rule based on Dempster-Shafer theory, *IEEE Transactions on Systems, Man and Cybernetics*, **25**, 804–813.

DO, T.-N. and F. POULET (2005) Kernel methods and visualisation for interval data mining, in *Applied Stochastic Models and Data Analysis*, J. Janssen and P. Lenca (eds). Berlin/Heidelberg: Springer, 295–299.

DO, T.-N. and F. POULET (2006) Kernel Methods and Visualisation for Interval Data Mining, in *Proceedings of the 6th IEEE International Conference on Data Mining Workshops*, Hong Kong.

DUARTE-SILVA, A.P. and P. BRITO (2006) Linear discriminant analysis for interval data, *Computational Statistics*, **21**, 289–308.

DUBOIS, J.-H. and H. PRADE (1988) *Possibility Theory*, New York: Plenum Press.

FRIEDMAN, J.H. (1996) Another approach to polychotomous classification. Department of statistics and Standford linear accelerator center, Standford University, Stanford, CA.

GURU, D.S., B.B. KIRANAGI and P. NAGABHUSHAN (2004) Multivalued type dissimilarity measure and concept of mutual dissimilarity value for clustering symbolic patterns, *Pattern Recognition*, **38**, 1203–1213.

HAND, D.J., H. MANNILA and P. SMYTH (2001) *Principles of Data Mining*, Cambridge: MIT Press.

Hao, P.-Y. (2008) Fuzzy one-class support vector machines, *Fuzzy Sets and Systems*, **159**, 2317–2336.
Haouari, B., A.B. Amor, Z. Elouedi and K. Mellouli (2009) Naïve possibilistic network classifiers, *Fuzzy Sets and Systems*, **160**, 3224–3238.
Hastie, T. and R. Tibshirani (1998) Classification by pairwise coupling, *The Annals of Statistics*, **26**, 451–471.
Haykin, S. (1999) *Neural Networks, A Comprehensive Foundation*, Essex, UK: Pearson Education.
Herbrich, R. (2002) *Learning Kernel Classifiers: Theory and Algorithms*, Cambridge: MIT Press.
Hsu, C.-W. and C.-J. Lin (2002) A Comparison of methods for multiclass support vector machines, *IEEE Transactions on Neural Networks*, **13**, 415–425.
Jenhani, I., N.B. Amor and Z. Elouedi (2008) Decision trees as possibilistic classifiers, *Approximate Reasoning*, **48**, 784–807.
Ji, A.-B., J.-H. Pang and H.-J. Qiu (2010) Support vector machine for classification based on fuzzy training data, *Expert Systems with Applications*, **37**, 3495–3498.
Juang, C.F., S.H. Chiu and S.W. Chang (2007) A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems, *IEEE Transactions on Fuzzy Systems*, **15**, 998–1008.
Klir, G.J. and B. Yuan (1995) *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, New Jersey: Prentice-Hall.
Kohavi, R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*.
Lin, C. and S. Wang (2002) Fuzzy support vector machines, *IEEE Transactions on Neural Networks*, **13**, 464–471.
Lin, C.T., C.M. Yeh, S.F. Liang, J.F. Chung and N. Kumar (2006) Support-vector-based fuzzy neural network for pattern classification, *IEEE Transaction on Fuzzy Systems*, **14**, 31–41.
Liu, S.-T. (2009) A revisit to quadratic programming with fuzzy parameters, *Chaos, Solitons & Fractals*, **41**, 1401–1407.
Masson, M. and T. Denoeux (2008) ECM: An evidential version of the fuzzy c-means algorithm, *Pattern Recognition*, **41**, 1384–1397.
Mitra, S. and Y. Hayashi (2000) Neuro–fuzzy rule generation: survey in soft computing framework, *IEEE Transactions on Neural Networks*, **11**, 748–768.
Moore, R.E., R.B. Kearfott and M.J. Cloud (2009) *Introduction to Interval Analysis*, Philadelphia: Siam.
Platt, J.C., N. Cristianini and J. Shawe-Taylor (2000) Large margin DAGs for multiclass classification, in *Advances in Neural Information Processing Systems,* A. Sigrel and H. Sigrel (eds), Cambridge: MIT Press, Vol. 12, 547–553.
Qin, B., Y. Xia and S. Prabhakar (2010) Rule induction for uncertain data, *Knowledge and Information Systems*, **29**, 1–28.
Ross, T.J. (2004) *Fuzzy Logic with Engineering Applications*, San Francisco, CA: John Willey & Sons, Ltd.
Scholkopf, B. and A. Smola (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, Cambridge: MIT Press.
Souza, R.M.C.R. and F.A.T. De Carvalho (2010) Unsupervised pattern recognition models for mixed feature-type symbolic data, *Pattern Recognition Letters*, **31**, 430–443.
Tax, D.M.J. (2001) One-class classification: concept-learning in the absence of counter-examples, Doctoral Dissertation, University of Delft, The Netherlands.
Tax, D.M.J. and R.P.W. Duin (2002) Using Two-class Classifiers for Multiclass Classification, in *Proceedings of the 16th International Conference on Pattern Recognition*.
Theodoridis, S. and K. Koutroumbas (1998) *Pattern Recognition*, London: Academic press.
Tian, J., M.-H. Ha, J.-H. Li and D.-Z. Tian (2006) The fuzzy- number based key theorem of statistical learning theory, *Proceedings of International Conference on Machine Learning and Cybernetics*, 13–16 August, 2006, Dalian, China, pp. 3475–3479.
Trafalis, T.B. and R.C. Gilbert (2006) Robust classification and regression using support vector machines, *European Journal of Operational Research*, **173**, 893–909.
Trafalis, T.B. and R.C. Gilbert (2007) Robust support vector machines for classification and computational issues, *Optimization Methods and Software*, **22**, 187–198.
Vapnik, V.N. (1995) *The Nature of Statistical Learning Theory*, New York: Springer-Verlag.
Vapnik, V.N. (1998) *Statistical Learning Theory*, New York: John Wiley and Sons.
Wang, L. (2005) *Support Vector Machines: Theory and Applications*, New York: Springer.
Webb, A.R. (2003) *Statistical Pattern Recognition*, Hoboken, NJ: John Wiley & Sons, Ltd.
Weston, J. and C. Watkins (1998) *Multi-class Support Vector Machines. Department of Computer Science*, Royal Holloway: University of London.
Younes, Z., F. Abdallah and T. Denoeux (2009) An evidence-theoretic k-nearest neighbor rule for multi-label classification, in *LNCS*, L. Godo and A. Pugliese (eds), Heidelberg: Springer, 297–308.
Zadeh, L.A. (1978) Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, **100**(Suppl 1), 9–34.
Zimmermann, H.-J. (1996) *Fuzzy Set Theory and its Applications*, Norwell, MA: Kluwer Academic Publishers.

The authors

Yahya Forghani

Yahya Forghani was born in Mashhad, Iran, in 1979. He received the BS degree in computer engineering at Ferdowsi University of Mashhad, Iran, in 2001 and the MS degree in computer engineering at Isfehan University of Technology in 2006. Since September 2010, he has been pursuing the PhD degree in computer engineering at Ferdowsi University of Mashhad. His research interests include pattern recognition and optimization.

Hadi Sadoghi Yazdi

Hadi Sadoghi Yazdi was born in Sabzevar, Iran, in 1971. He received the BS degree in electrical engineering at Ferdowsi University of Mashhad, Iran, in 1994 and the MS and PhD degree in electrical engineering at Tarbiat Modarres University of Tehran, Iran, in 1996 and 2005, respectively. He works in Computer Engineering Department as an Associate Professor at Ferdowsi University of Mashhad. His research interests include pattern recognition, optimization, adaptive filtering and image and video processing.

Sohrab Effati

Sohrab Effati received the BS degree in applied mathematics from Birjand University, Iran, and the MS degree in applied mathematics from Tarbiat Moallem University of Tehran, Iran, in 1992 and 1995, respectively. He received the PhD degree in control systems from Ferdowsi University of Mashhad, Iran, in 2000. He works in Applied Mathematics Department as an Associate Professor at Ferdowsi University of Mashhad. His research interests include control systems, fuzzy theory, and neural network models and its application in optimization problems.