Intuitionistic Fuzzy Twin Support Vector Machines

Salim Rezvani, Xizhao Wang, *Fellow, IEEE,* Farhad Pourpanah, *Member, IEEE*

***Abstract*—Fuzzy twin support vector machine (FTSVM) is an effective machine learning technique which is able to overcome the negative impact of noise and outliers in tackling data classification problems. In FTSVM, the degree of membership function in the sample space just described the space between input data and class center, whilst ignored the position of input data in the feature space and simply miscalculated the ledge support vectors as noises. This paper presents an intuitionistic fuzzy twin support vector machine (IFTSVM) which combines the idea of intuitionistic fuzzy number with twin SVM (TSVM). An adequate fuzzy membership is employed to reduce the noise brought by the pollutant inputs. Two functions, i.e., linear and nonlinear, are used to formulate two non-parallel hyperplanes. IFTSVM not only reduces the influence of noises, it also distinguishes the noises from the support vectors. Further, this modification can minimize a newly formulated structural risk and improve the classification accuracy. Two artificial and eleven benchmark problems are employed to evaluate the effectiveness of the proposed IFTSVM model. To quantify the results statistically, the bootstrap technique with the 95% confidence intervals is used. The outcome shows that IFTSVM is able to produce promising results as compared with those from the original SVM, Fuzzy SVM (FSVM), FTSVM and other models reported in the literature.**

***Index Terms*—Intuitionistic Fuzzy Number, Kernel function, Quadratic Programming Problem, Twin support vector machines.**

I. INTRODUCTION

THE support vector machine (SVM) and its variants [1]–[5] are popular machine learning techniques which have shown astonishing results in various application domains such as regression [6]–[8], economy [9] [10], power system [11] and medical [12], just to name a few. In fact, SVM attempts to explore an optimal hyper-plane with the maximum margin, while, the generalization error of SVM mainly depends on the ratio of the radius and margin, i.e. radius-margin error bound [13]. For a given feature space, which radius is fixed, the SVM can minimize generalization error by only maximizing margin. Nonetheless, radius information becomes an important parameter for joint learning of feature transformation and classification algorithm which can not be ignored.

Traditional SVM builds two parallel support hyper-planes between which the area is first split into the two classes (i.e., $+$ and $-$), and then the margin is maximized. Therefore, the regularization term is achieved and the structural risk is minimized. Several research have been considered the radius-margin error [14]–[17]. However, most of these methods suffer from computational burden [18].

Apart from SVM with two parallel hyper-planes, several classifiers with non-parallel hyper-planes such as the generalized eigenvalue proximal SVM (GEPSVM) [19] and twin SVM (TSVM) [20]–[26] have been proposed. Both methods find two non-parallel proximal hyper-planes which locate hyper-plane as far as possible to one of the two classes and near to the other one. Unlike SVM which finds only one large quadratic programming problem (QPP), TSVM defines two small QPPs. As shown in [20], TSVM is four times faster than SVM. It also has shown promising results as compared those of the SVM and GEPSVM [27]. One important characteristic of SVM is the implementation of the structural risk minimization principle [28] [29], but, only the empirical risk is considered in TSVM. Although, the technique of organizing non-parallel hyper-planes has shown promising results [30], yet it is not always good enough from the theoretical viewpoint, and it needs further adjustments. On the other hand, it is known that the inverse matrices $(G^TG)^{-1}$ and $(H^TH)^{-1}$ appear in the dual problems, where $H = [A \quad e_1]$ and $G = [B \quad e_2]$. A and B represent training samples belonging to classes +1 and -1, respectively, e_1 and e_2 correspond to the unit vectors. To achieve dual problems, one of the following conditions must be satisfied: either the inverse matrices $(G^TG)^{-1}$ and $(H^TH)^{-1}$ occur or the matrices G^TG and H^TH are non-singular. Satisfying one of these conditions can improve the dual problems theoretically.

If the support vectors are mixed by noises, the SVM cannot find an optimal hyper-plane, which leads to produce inferior results. To alleviate this problem, fuzzy SVM (FSVM) has been proposed in [31]–[34], which uses a degree of membership function for each training sample. Even though, FSVM is able to reduce the effects of outliers and noises, but the degree of membership function only considers the distance between the training sample and the class center, which several outlier support vectors may be confused as noises. To solve this problem, an FSVM with dual memberships is suggested in [35]. However, this method improved the performance of FSVM, it also suffers from several problems. For example, those training samples which are located far away from the class center may produce better membership function as compared with those nearby the class center [35].

Coordinate descent methods have received increasing attention in the last years due to recent results in support vector machine [36] [37]. A new coordinate descent FTSVM for solving classification problems is introduced in [38], which is faster than TSVM. Later, a new FTSVM incorporated TSVM and fuzzy neural network to tackle binary classification problems in [39]. In [40], an SVM with an intuitionistic fuzzy

Salim Rezvani and Xizhao Wang are with the Big Data Institute, College of Computer Science and Software Engineering, Guangdong Key Lab. of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, Guangdong, China (e-mail: salim_rezvani@szu.edu.cn, xizhaowang@ieee.org.)

Farhad Pourpanah is with the College of Mathematics and Statistics Shenzhen University, Shenzhen 518060, Guangdong, China (e-mail: farhad.086@gmail.com.)

Xizhao Wang is the corresponding author.

Manuscript received April ??, ??; revised August ??, ??.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2019.2893863, IEEE Transactions on Fuzzy Systems

number (IFN) and kernel function is proposed to consider the situation of training samples in the feature space.

Building upon our newly proposed ranking method of trapezoidal intuitionistic fuzzy numbers and type-2 intuitionistic exponential triangular fuzzy number [41] [42], which is able to find the degrees of membership, non-membership, and hesitation, in this paper we propose a new classification model, called intuitionistic fuzzy twin support vector machine (IFTSVM), to solve binary classification problems. IFTSVM uses IFN to assign a pair of membership and non-membership functions to every training sample. The degree of membership function measures the distance between the training sample and class center, while the degree of non-membership function measures the relation among the number of inharmonic samples and the number of samples in its neighborhood. These two measurements help IFTSVM to reduce the effect of noise and identify support vectors from noises. In addition, it minimizes the structural risk and improves classification accuracy. Two artificial and eleven benchmark problems are employed to evaluate the effectiveness of IFTSVM. In summary, this paper proposes a new learning model which is called IFTSVM with the following contributions:

1) IFTSVM significantly alleviates the negative impact of noise and outliers on classification accuracy since it uses a pair of membership and non-membership functions for every training sample.

2) IFTSVM constructs a new structural risk function with regularization terms different from existing SVM models.

3) IFTSVM statistically shows a better performance on artificial and benchmark classification problems in comparison with other similar SVM models.

The rest of this paper is arranged as follows: Section II explains the details of Intuitionistic Fuzzy Set, SVM, FSVM, and TSVM. Section III describes the structure of the proposed IFTSVM model. The experimental results are reported in Section IV. Section V concludes and suggests the future research.

II. Preliminaries

In this Section, we firstly describe the intuitinistic fuzzy set. Then, the structure of SVM, FSVM and TSVM are explained in details. Suppose $S = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i)\}$ is a set of training samples where $x_i \in R^d$ and $y_i = \{-1, +1\}$, respectively, represent the *i*-th training sample and corresponding target class. The training samples can be separated into two matrices, i.e, X_+^S and X_-^S, where X_+^S (X_-^S) contains those samples which are belonging to positive (negative) class.

A. Intuitionistic Fuzzy Set

For a non-empty set X, a fuzzy set A in a universe X can be defined as:

$$A = \{(x, \mu_A(x))|x \in X\} \tag{1}$$

where $\mu_A : X \to [0, 1]$ and $\mu_A(x)$ is the degree of membership of $x \in X$. An intuitionistic fuzzy set is defined as:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x), \nu_{\tilde{A}}(x))|x \in X\} \tag{2}$$

where $\mu_{\tilde{A}}(x)$ and $\nu_{\tilde{A}}(x)$ define the degrees of membership and non-membership functions of $x \in X$, respectively, $\mu_{\tilde{A}} : X \to [0, 1]$, $\nu_{\tilde{A}} : X \to [0, 1]$ and $0 \leq \mu_{\tilde{A}}(x) + \nu_{\tilde{A}}(x) \leq 1$, and the hesitation degree of $x \in X$ can be presented as:

$$\pi_{\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x) - \nu_{\tilde{A}}(x) \tag{3}$$

An IFN can be defined as $\alpha = (\mu_\alpha, \nu_\alpha)$, where $\mu_\alpha \in [0, 1]$, $\nu_\alpha \in [0, 1]$, and $0 \leq \mu_\alpha + \nu_\alpha \leq 1$. The largest IFN is $\alpha^+ = (1, 0)$, and the smallest IFN is $\alpha^- = (0, 1)$. The IFN for a given $\alpha = (\mu_\alpha, \nu_\alpha)$ can be calculated as follows:

$$s(\alpha) = \mu_\alpha - \nu_\alpha. \tag{4}$$

where $s(\alpha)$ represents the score value of the IFN $\alpha = (\mu_\alpha, \nu_\alpha)$. However, it is impossible to determine the score value for some IFNs. To alleviate this problem, following function can be replaced:

$$h(\alpha) = \mu_\alpha + \nu_\alpha. \tag{5}$$

According to Eqs. 3 and 5, we have

$$h(\alpha) + \pi(\alpha) = 1 \tag{6}$$

If $s(\alpha_1) = s(\alpha_2)$ and $h(\alpha_1) < h(\alpha_2)$, then $\alpha_1 < \alpha_2$.

Based on Eq. 4, other score function can be determined as follows:

$$H(\alpha) = \frac{1 - \nu(\alpha)}{2 - \mu(\alpha) - \nu(\alpha)} \tag{7}$$

Therefore, the relationships between membership and nun-membership functions can be defined as follows:

(1) $s(\alpha_1) < s(\alpha_2) \Rightarrow H(\alpha_1) < H(\alpha_2)$;

(2) $s(\alpha_1) = s(\alpha_2),\ h(\alpha_1) < h(\alpha_2) \Rightarrow H(\alpha_1) < H(\alpha_2)$.

B. Support vector machines

Traditional SVM is able to solve binary classification problems. It attempts to find an optimal hyper-plane $w^T x + b = 0$, where $w \in \mathcal{R}^n$ is the weight, and $b \in \mathcal{R}$ is the bias term. This hyper-plane can be used to define the label of input sample x_i as follows:

$$\begin{cases} (w.x_i + b) \geq 0, & if \quad y_i \ is \ positive, \\ (w.x_i + b) \leq 0, & if \quad y_i \ is \ negative. \end{cases} \tag{8}$$

In a linear SVM an optimal hyper-plane can be achieved by solving the following primal quadratic programming problem (QPP):

$$\begin{cases} min \ \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i, \\ s.t. \ y_i(w^T x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, 2, ..., l. \end{cases} \tag{9}$$

where ξ_i $(i = 1, 2, ..., l)$, C and l are slack variables, penalty parameter and the number of training samples, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2019.2893863, IEEE Transactions on Fuzzy Systems

C. Fuzzy support vector machines

Suppose $\{(x_1, y_1, s_1), (x_2, y_2, s_2), .., (x_i, y_i, s_i)\}$ is a set of training data containing i samples with their corresponding fuzzy memberships (s_i), where $\sigma \leq s_i \leq 1$ and $\sigma > 0$ is a small positive value. Let $z = \phi(x)$ denote a mapping ϕ from $\mathcal{R}^N$ to a feature space $\mathcal{Z}$. The optimal hyper-plane cab be achieved by solving:

$$min\ \frac{1}{2}w^T.w + C\sum_{i=1}^{l} s_i\xi_i$$
$$s.t.\ y_i(w.z_i + b) \geq 1 - \xi_i,\quad \xi_i \geq 0,\ i = 1, ..., l \qquad (10)$$

where ξ_i is the measured error in the SVM and term $s_i\xi_i$ is measured error with different weighting, and C is a constant. A small C minimizes the efficacy of the ξ_i in Eq. 10.

The Lagrangian can be constructed to solve this problem as follows:

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}w^T.w + C\sum_{i=1}^{l} s_i\xi_i$$
$$- \sum_{i=1}^{l} \alpha_i(y_i(w.z_i + b) - 1 + \xi_i) - \sum_{i=1}^{l} \beta_i\xi_i \qquad (11)$$

and the following conditions must be satisfied to find the saddle point of $L(w, b, \xi, \alpha, \beta)$:

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial w} = w - \sum_{i=1}^{l} \alpha_i y_i z_i = 0 \qquad (12)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0 \qquad (13)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial \xi_i} = s_iC - \alpha_i - \beta_i = 0. \qquad (14)$$

Applying Eqs. 12-14 into the Eqs. 11 and 10 can be written as:

$$maximize\ W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

$$s.t.\ \sum_{i=1}^{l} y_i\alpha_i = 0,\quad 0 \leq \alpha_i \leq s_iC,\quad i = 1, ..., l \qquad (15)$$

and the Karush-Kuhn-Tucker (K.K.T) conditions [43] are described as:

$$\bar{\alpha}_i(y_i(\bar{w}.z_i + \bar{b}) - 1 + \bar{\xi}_i) = 0,\quad i = 1, ..., l \qquad (16)$$

$$(s_iC - \bar{\alpha}_i)\xi_i = 0,\quad i = 1, ..., l \qquad (17)$$

The point x_i with the corresponding $\bar{\alpha}_i > 0$ is known as a support vector. The FSVM can have two kinds of support vectors. The first one with $0 < \bar{\alpha}_i < s_iC$ lies on the margin of the hyper-plane, and the second one with $\bar{\alpha}_i = s_iC$ is misclassified. In contrast with SVM, TSVM may recognize a point with same $\bar{\alpha}_i$ into different kind of support vectors owing to the s_i.

D. Twin support vector machine

Unlike traditional SVM which uses only one hyper-plane to separate the positive samples from the negative samples, TSVM [20] obtains two non-parallel hyper-planes (as shown in Fig. 1). It finds a hyper-plane around which the data samples of the corresponding class get grouped [44]–[46] as follows:

$$w_{(1)}.x_i + b_{(1)} = 0,\qquad w_{(2)}x_i + b_{(2)} = 0 \qquad (18)$$

where $w_{(i)}$ and $b_{(i)}$ are the weight and bias term of the i-th hyper-plane, respectively. The two hyper-planes are achieved by solving the following QPPs:

$$\min_{w_{(1)}, b_{(1)}, \xi_2} \frac{1}{2}(Aw_{(1)} + e_1b_{(1)})^T(Aw_{(1)} + e_1b_{(1)}) + p_1e_2^T\xi_2$$

$$s.t.\quad -(Bw_{(1)} + e_2b_{(1)}) + \xi_2 \geq e_2,\ \xi_2 \geq 0 \qquad (19)$$

and

$$\min_{w_{(1)}, b_{(1)}, \xi_1} \frac{1}{2}(Bw_{(2)} + e_2b_{(2)})^T(Bw_{(2)} + e_2b_{(2)}) + p_2e_2^T\xi_1$$

$$s.t.\quad (Aw_{(2)} + e_1b_{(2)}) + \xi_1 \geq e_1,\ \xi_1 \geq 0 \qquad (20)$$

where A and B represent the data samples belonging to classes $+1$ and -1, respectively, ξ_1 and ξ_1 are the slack variables, e_1 and e_2 are the vector of ones with adequate length, and p_1 and p_2 are penalty parameters. Once optimal parameters, i.e., (w_1^*, b_1^*) and (w_2^*, b_2^*), are achieved, new input sample x can be labeled as follows:

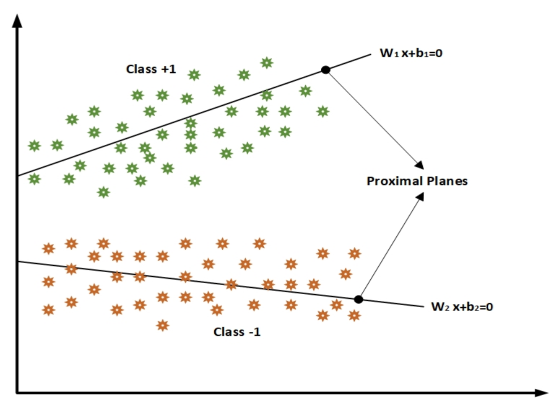


Fig. 1. The Geometric explanation of Twin SVM

$$f(x) = arg\ min\ _{i\ \in\ 1,2}\frac{|\ (w_i^*)^Tx + b_i^*\ |}{\|\ w_i^*\ \|}. \qquad (21)$$

III. Intuitionistic Fuzzy Twin Support vector machine

In this section, we first explain the proposed IFTSVM model. Then, the structures of two kernel functions, i.e., linear and non-linear, are discussed in detail.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2019.2893863, IEEE Transactions on Fuzzy Systems

A. Intuitionistic fuzzy membership Assignment

IFTSVM employs the degree of membership function which is proposed in [40]. To reduce the effect of noise and outliers, it is critical to select an appropriate membership function. For example, as shown in Fig. 2, those training samples which are located in the boundary areas of the two classes have the same membership degrees for both classes. This may lead to the wrong prediction. To alleviate this problem, IFTSVM assigns an IFN, i.e., (μ, ν), to each training sample, where μ defines the degree of membership function related to one class, and ν explains the degree of non-membership function related to other class. Obviously, the degrees of non-membership related to positive (negative) classes are not the same.

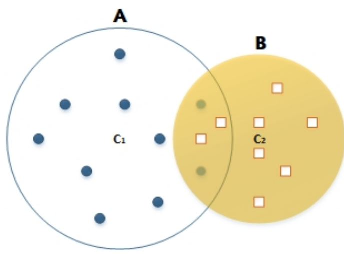


Fig. 2. Similar degree of membership for two training samples

The designed degrees of membership and non-membership functions for every training sample in the high-dimensional feature space are explained in the following subsections.

1) The membership function: the distance between training sample and the class center is used as membership function in the high dimensional feature space. For each training sample, the degree of membership can be described as:

$$\mu(x_i) = \begin{cases} 1 - \frac{\|\phi(x_i) - C^+\|}{r^+ + \delta} & y_i = +1, \\ \\ 1 - \frac{\|\phi(x_i) - C^-\|}{r^- + \delta} & y_i = -1. \end{cases} \tag{22}$$

where $\delta > 0$ is an adjustable parameter, r^+ (r^-) and C^+ (C^-) are the radius and class center of the positive (negative) class, and $||.||$ is the distance between input sample and the corresponding class center:

$$D(\phi(x_i), \phi(x_j)) = \|\phi(x_i) - \phi(x_j)\|, \tag{23}$$

where ϕ represents input sample in the high-dimensional feature space.

The class center of each class can be measured by:

$$C^{\pm} = \frac{1}{l_{\pm}} \sum_{y_i = \pm 1} \phi(x_i) \tag{24}$$

where l_+ (l_-) is the total number of positive (negative) samples.

The radius of each class can be calculated by:

$$r^{\pm} = \underset{y_i = \pm 1}{max} \|\phi(x_i) - C^{\pm}\| \tag{25}$$

2) The non-membership function: the relationship between all inharmonious points and the total number of training samples in its neighborhood (i.e., $\rho(x_i)$) is used as non-membership function, as follows:

$$\nu(x_i) = (1 - \mu(x_i))\rho(x_i). \tag{26}$$

where $0 \leq \mu(x_i) + \nu(x_i) \leq 1$, and $\rho(x_i)$ is defined as:

$$\rho(x_i) = \frac{|\{x_j | \|\phi(x_i) - \phi(x_j)\| \leq \alpha, \ y_j \neq y_i\}|}{|\{x_j | \|\phi(x_i) - \phi(x_j)\| \leq \alpha\}|}, \tag{27}$$

where $\alpha > 0$ is an adjustable parameter and $|.|$ denotes the cardinality.

The degrees of membership and non-membership functions of IFN are built based on the inner product distance in the feature space. Therefore, the kernel functions are used to make IFNs.

Theorem 1. [40] Suppose $K(x, x')$ be a kernel function. Hence, the inner product distance is presented by:

$$\|\phi(x) - \phi(x')\| = \sqrt{K(x,x) + K(x',x') - 2K(x,x')}$$

Proof:

$$\|\phi(x) - \phi(x')\| = \sqrt{(\phi(x) - \phi(x')).(\phi(x) - \phi(x'))}$$

$$= \sqrt{(\phi(x).\phi(x)) + (\phi(x').\phi(x')) - 2(\phi(x).\phi(x'))}$$

$$= \sqrt{K(x,x) + K(x',x') - 2K(x,x')}.$$

Theorem 2. With respect to theorem 1, the radiuses of both classes are respectively:

$$i)\ r^+ = \underset{y_i=+1}{max} \sqrt{K(x_i, x_i) + \frac{1}{l_+^2} \sum_{y_m=+1} \sum_{y_n=+1} K(x_m, x_n) - \frac{2}{l_+} \sum_{y_j=+1} K(x_i, x_j)},$$

$$ii)\ r^- = \underset{y_i=-1}{max} \sqrt{K(x_i, x_i) + \frac{1}{l_-^2} \sum_{y_m=-1} \sum_{y_n=-1} K(x_m, x_n) - \frac{2}{l_-} \sum_{y_j=-1} K(x_i, x_j)}.$$

Proof:

$i)$

$$r^+ = \underset{y_i=+1}{max} \|\phi(x_i) - C^+\| = max_{y_i=+1} \sqrt{(\phi(x_i) - C^+).(\phi(x_i) - C^+)}$$

$$= \underset{y_i=+1}{max} \sqrt{(\phi(x_i).\phi(x_i)) + (C^+.C^+) - 2(\phi(x_i).C^+)} =$$

$$\underset{y_i=+1}{max} \sqrt{K(x_i, x_i) + (\frac{1}{l_+} \sum_{y_i=+1} \phi(x_i))(\frac{1}{l_+} \sum_{y_i=+1} \phi(x_i)) - 2(\phi(x_i))(\frac{1}{l_+} \sum_{y_i=+1} \phi(x_i))}$$

$$= \underset{y_i=+1}{max} \sqrt{K(x_i, x_i) + \frac{1}{l_+^2} \sum_{y_m=+1} \sum_{y_n=+1} K(x_m, x_n) - \frac{2}{l_+} \sum_{y_j=+1} K(x_i, x_j)}.$$

$ii)$ Is similar to that of part (i).

Therefore, training samples can be converted into IFN as follows:

$$T = \{x_1, y_1, \mu_1, \nu_1\}, \{x_2, y_2, \mu_2, \nu_2\}, ..., \{x_l, y_l, \mu_l, \nu_l\}.$$

where μ_i and ν_i, respectively, indicate the degrees of membership function and non-membership functions of x_i. For a given IFN, the score function can be defined as:

$$s_i = \begin{cases} \mu_i & \nu_i = 0, \\ \\ 0 & \mu_i \leq \nu_i, \\ \\ \frac{1 - \nu_i}{2 - \mu_i - \nu_i} & others. \end{cases} \tag{28}$$

The score value can easily separate the support vector from outliers and noises [40]. For example, assume three training samples, i.e., A, B and C, in Fig. 3. When $\nu_i = 0$ (positive sample A in Fig.3), which has not negative samples as neighborhoods, the degree of membership function can correctly classify it. While $\mu_i \leq \nu_i$ (negative sample B in Fig.3), the degree of membership value is less than degree of non-membership value, it will be considered as noise. When $\mu_i > \nu_i$ and $\nu_i \neq 0$ (positive sample C in Fig.3), it is far away from the class center, there are few positive samples in its neighborhood. Thus, it may be considered as a support vector, instead of an outlier.

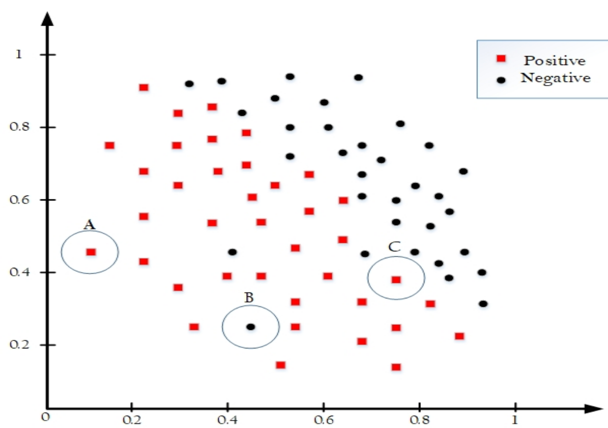


Fig. 3. Recognize samples

B. Linear IFTSVM

The linear kernel for IFTSVM can be considered as follows:

$$\min_{w_1,b_1,\xi_2} \frac{1}{2} \parallel Aw_1 + e_1b_1 \parallel^2 + \frac{1}{2}C_1 \parallel w_1 \parallel^2 + C_2 s_2^T \xi_2$$

$$subject\ to\ -(Bw_1 + e_2b_1) + \xi_2 \geq e_2,\ \xi_2 \geq 0 \tag{29}$$

and

$$\min_{w_2,b_2,\xi_1} \frac{1}{2} \parallel Bw_2 + e_2b_2 \parallel^2 + \frac{1}{2}C_3 \parallel w_2 \parallel^2 + C_4 s_1^T \xi_1$$

$$subject\ to\ (Aw_2 + e_1b_2) + \xi_1 \geq e_1,\ \xi_1 \geq 0 \tag{30}$$

where $C_1,\ C_2,\ C_3$ and C_4 are positive penalty parameters, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are column vectors of ones with desirable length, and $s_1 \in \mathcal{R}^{l+}$ and $s_2 \in \mathcal{R}^{l-}$ are the score values of class $+$ and $-$, respectively.

IFTSVM minimizes the structural risk by summing the regularization term with the opinion of maximizing margin. It will be shown that the structural risk is minimized in Eqs. 29 and 30. This pair of QPPs can be achieved by constructing the Lagrangian as follows:

$$L(w_1, b_1, \xi_2, \alpha, \beta) = \frac{1}{2} \parallel Aw_1 + e_1b_1 \parallel^2 \frac{1}{2}C_1 \parallel w_1 \parallel^2$$

$$+ C_2 s_2^T \xi_2 + \alpha[(Bw_1 + e_2b_1) - \xi_2 + e_2] - \beta\xi_2 \tag{31}$$

where α and β are Lagrangian multipliers. With K.K.T conditions, Eq. 31 can be obtained as follows:

$$\frac{\partial L}{\partial w_1} = A^T(Aw_1 + e_1b_1) + C_1w_1 + \alpha B = 0, \tag{32}$$

$$\frac{\partial L}{\partial b_1} = e_1^T(Aw_1 + e_1b_1) + \alpha e_2 = 0, \tag{33}$$

$$\frac{\partial L}{\partial \xi_2} = C_2 s_2^T - \alpha - \beta = 0. \tag{34}$$

By combining Eqs. 32 and 33, can achieve:

$$\begin{pmatrix} A^T \\ e_1^T \end{pmatrix} (A \quad e_1) \begin{pmatrix} w_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} B \\ e_2 \end{pmatrix} \alpha = 0 \tag{35}$$

Let $H_1 = (A \quad e_1)$, $G_2 = (B \quad e_2)$ and $u_1 = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix}$. The, Eq. 35 can be reformulated as

$$H_1^T H_1 u_1 + G_2^T \alpha = 0 \Rightarrow u_1 = -(H_1^T H_1)^{-1} G_2^T \alpha \tag{36}$$

It is hard to calculate the inverse of $H_1^T H_1$. This can be managed by attaching regularization unit $C_1 I$ in the Eq. 37, where I is an identity matrices with the appropriate dimension. Thus:

$$u_1 = -(H_1^T H_1 + C_1 I)^{-1} G_2^T \alpha \tag{37}$$

In a similar way, weight vector and bias for other class can be achieved by solving the following equation:

$$u_2 = (G_2^T G_2 + C_3 I)^{-1} H_1^T \beta \tag{38}$$

Using Eq. 29 and K.K.T conditions, the Wolfe dual Eq. 29 can be written as:

$$\max_{\alpha}\ e_2^T \alpha - \frac{1}{2}\alpha^T G_2(H_1^T H_1 + C_1 I)^{-1} G_2^T \alpha$$

$$subject\ to\ 0 \leq \alpha \leq C_2 s_2 \tag{39}$$

In the above equation, C_1 is a weighting factor which distinguishes the tradeoff between the regularization term and the empirical risk. Hence, choosing an appropriate C_1, whether small or large, reflects structure risk minimization principle.

Likewise, the Wolfe dual for Eq. 30 can be written as:

$$\max_{\beta}\ e_1^T \beta - \frac{1}{2}\beta^T G_1(G_2^T G_2 + C_3 I)^{-1} H_1^T \beta$$

$$subject\ to\ 0 \leq \beta \leq C_4 s_1 \tag{40}$$

Once optimal u_1^* and u_2^* are achieved, the two non-parallel hyper-planes (Eq. 18) are admitted. A new input data x can be categorized as a positive or negative class, as follows:

$$x \in W_k,\ k = arg \min_{i=1,2}\{\frac{|\ w_1^T x + b_1\ |}{\parallel w_1 \parallel}, \frac{|\ w_2^T x + b_2\ |}{\parallel w_2 \parallel}\} \tag{41}$$

where $|\ .\ |$ is the absolute value.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2019.2893863, IEEE Transactions on Fuzzy Systems

C. Nonlinear IFTSVM

In order to solve non-linear classification problems, the following kernel function is considered.

$$k(x, X^T)w_1 + b_1 = 0, \quad k(x, X^T)w_2 + b_2 = 0, \tag{42}$$

where $k(x_1, x_2) = (\phi(x_1), \phi(x_2))$ is a kernel function. The primal problem of nonlinear IFTSVM is defined as:

$$\min_{w_1,b_1,\xi_2} \frac{1}{2} \parallel k(A, X^T)w_1 + e_1b_1 \parallel^2 + \frac{1}{2}C_1 \parallel w_1 \parallel^2 + C_2 s_2^T \xi_2$$
$$subject\ to\ -(k(B, X^T)w_1 + e_2b_1) + \xi_2 \geq e_2,\ \xi_2 \geq 0 \tag{43}$$

and

$$\min_{w_2,b_2,\xi_1} \frac{1}{2} \parallel k(B, X^T)w_2 + e_2b_2 \parallel^2 + \frac{1}{2}C_3 \parallel w_2 \parallel^2 + C_4 s_1^T \xi_1$$
$$subject\ to\ (k(A, X^T)w_2 + e_1b_2) + \xi_1 \geq e_1,\ \xi_1 \geq 0 \tag{44}$$

Lagrangian of Eq. 44 is given as:

$$L(w_1, b_1, \xi_2, \alpha, \beta) = \frac{1}{2} \parallel k(A, X^T)w_1 + e_1b_1 \parallel^2 + \frac{1}{2}C_1 \parallel w_1 \parallel^2$$
$$+ C_2 s_2^T \xi_2 + \alpha[(k(B, X^T)w_1 + e_2b_1) - \xi_2 + e_2] - \beta\xi_2 \tag{45}$$

The K.K.T conditions are obtained as follows:

$$\frac{\partial L}{\partial w_1} = k(A, X^T)^T (k(A, X^T)w_1 + e_1b_1)$$
$$+ C_1w_1 + \alpha k(B, X^T) = 0, \tag{46}$$

$$\frac{\partial L}{\partial b_1} = e_1^T (k(A, X^T)w_1 + e_1b_1) + \alpha e_2 = 0, \tag{47}$$

$$\frac{\partial L}{\partial \xi_2} = C_2 s_2^T - \alpha - \beta = 0. \tag{48}$$

Combining Eqs. 46-48, can archive:

$$\begin{pmatrix} k(A, X^T)^T \\ e_1^T \end{pmatrix} (k(A, X^T)\ \ e_1) \begin{pmatrix} w_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} k(B, X^T) \\ e_2 \end{pmatrix} \alpha = 0 \tag{49}$$

Let $H_1^* = (k(A, X^T)\ \ e_1)$, $G_2^* = (k(B, X^T)\ \ e_2)$ and $u_1^* = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix}$. Then, Eq. 49 can be reformulated as:

$$u_1^* = -(H_1^{T*} H_1^* + C_1 I)^{-1} G_2^{T*} \alpha \tag{50}$$

With the K.K.T conditions and Lagrangian method, the corresponding Wolfe dual can be written as:

$$\max_{\alpha}\ e_2^T \alpha - \frac{1}{2}\alpha^T G_2^* (H_1^{T*} H_1^* + C_1 I)^{-1} G_2^{T*} \alpha$$
$$subject\ to\ 0 \leq \alpha \leq C_2 s_2 \tag{51}$$

and

$$\max_{\beta}\ e_1^T \beta - \frac{1}{2}\beta^T G_1^* (G_2^{T*} G_2^* + C_3 I)^{-1} H_1^{T*} \beta$$
$$subject\ to\ 0 \leq \beta \leq C_4 s_1 \tag{52}$$

According to Eqs. 42-52, the augmented vectors $u_1 = [w_1^T\ \ b_1]^T$ and $u_2 = [w_2^T\ \ b_2]^T$ can be obtained by:

$$u_1^* = -(H_1^{T*} H_1^* + C_1 I)^{-1} G_2^{T*} \alpha, \tag{53}$$

$$u_2^* = (G_2^{T*} G_2^* + C_3 I)^{-1} H_1^{T*} \beta, \tag{54}$$

Once the vectors u_1^* and u_2^* are achieved, the two non-parallel hyper-planes (Eq. 42) are obtained. A new input data x can be labeled as either positive or negative class, as follows:

$$k = \underset{i=1,2}{arg\ min} \{ \frac{\mid w_1^T k(x, X^T) + b_1 \mid}{\sqrt{w_1^T k(A, X^T) w_1}}, \frac{\mid w_2^T k(x, X^T) + b_2 \mid}{\sqrt{w_2^T k(B, X^T) w_2}} \}. \tag{55}$$

D. Complexity analysis of IFTSVM

In this section, the big-O notation [47] is employed for the analysis on time complexity of IFTSVM. Let n be the total number of training samples and $m = n/2$ be the number of samples in each class. IFTSVM measures the degrees of membership (Eq. 22) and non-membership (Eq. 26) functions to compute the score value (Eq. 28) of each sample. To measure the degree of membership function, it first computes the class center (Eq. 24) and radius of class (Eq. 25). Then computes the distance between each class center and sample (Eq.23), and measures the degree of membership function for each sample using Eq. 22, which requires $O(1) + O(1) + O(m) + O(m)$. On the other hand, to measure the degree of non-membership function (Eq. 26), IFTSVM requires to compute $\rho(x_i)$ (Eq. 27), which needs O(m)+O(m) operations. Therefore, IFTSVM involves $O(1)+O(1)+O(m)+O(m)+O(m)+O(m)$ operations to measure the score function of samples, which is $O(m)$ when m extends to infinity. Then, similar to TSVM, IFTSVM requires to solve two QPPs for both linear and non-linear functions. According to [48], the computational complexity of conventional SVM is $O(n^3)$, and the computational complexity of TSVM by considering $m = n/2$ is $O(2 \times (n/2)^3)$. The time complexity of IFTSVM is $O(2 \times (n/2)^3) + O(n/2)$, which is $O(2 \times (n/2)^3)$. Therefore, the time complexity of IFTSVM is almost same as TSVM which is four times faster than conventional SVM.

IV. Experimental Results

To evaluate the effectiveness and generalization capability of IFTSVM, eleven benchmark data sets from UCI machine learning repository [49] and two artificial, i.e., Ripleys [50] and circle-in-the-square [51], are conducted. Table I shows the details of the UCI data sets.

For each data set, the 10-fold cross-validation is repeated 10 times. For all data sets, 90% of data samples are used for training and the the remaining 10% for the test. The bootstrap method [52] with the 95% confidence intervals is employed to quantify the results statistically. All samples are normalized between 0 and 1. The IFTSVM parameters are set as follows: $c_i (i = 1, 2, 3, 4)$ are correctly explored in the grids $\{2^i | i = -10, -9, ..., 9, 10\}$ by setting $C_1 = C_3$, $C_2 = C_4$. Plus, Gaussian-kernel is applied to trade with the non-linear cases, i.e. $\mathcal{K}(x_1, x_2) = exp(-\|x_1 - x_2\|^2/\sigma^2)$ and $\sigma \in \{2^{\sigma_{min}:\sigma_{max}}\}$ with $\sigma_{min} = -10$, $\sigma_{max} = 10$. The entire experiments are performed using MATLAB 2018a under a desktop PC with Intel(R) Core i5 processor (3.30 GHz) and 12GB RAM.

Five performance indicators including accuracy, computational time, sensitivity/true positive rate, specificity/negative positive rate [53] and area under ROC (AUC) [54], are used to

TABLE I
DETAILS OF THE UCI DATE SETS

Data set	# of samples	# of Negative samples	# of Positive samples	# of features	# of classes
Ionosphere	351	126	225	34	2
Australian	690	383	307	14	2
WDBC	569	212	357	30	2
WPBC	198	151	98	33	2
Bupa	345	145	200	6	2
Sonar	208	97	111	60	2
Heart	270	150	120	14	2
Pima	768	268	500	8	2
Adult	48842	37155	11687	14	2
Advert	3279	459	2820	1558	2
Spam	4601	1813	2788	57	2

compare IFTSVM with those from the conventional SVM [35], FSVM [29], TSVM [12], CDFTSVM [38], gradient boosting (GB) [55], accelerated GB (AGB) [56], Lasso [57] and random forest (RF) [58] . The sensitivity or true positive rate (TPR) is the ratio of classified positive sample over all positive samples, while the specificity or true negative rate (TNR) is the ratio of correctly classified negative samples over all negative samples.

A. UCI data sets

In this Section, the performance of IFTSVM is evaluated with the UCI data sets. The results are compared with those of the original SVM [53], FSVM [31], TSVM [20] and CDFTSVM [38]. Note that all results related to SVM, FSVM, and TSVM are taken from [38]. Three experiments are conducted as follows:

In the first experiment, the effects of different setting of the kernel parameter σ and trade-off C are analyzed using the Ionosphere data set. The aim is to find optimal parameter(s), i.e., C for linear function, and C and σ for non-linear function, which yields high accuracy rate. Firstly, C is optimized for the linear function, it varied from -10 to 10. As shown in Fig. 4, IFTSVM produces better results for $C < 0$, specifically; when C is set to -1. Then, both C and σ are optimized for non-linear kernel function. C and σ can be varied from -10 to 10 and -10 to 10, respectively. From Fig. 5, it can be found that IFTSVM with $C = 1$ and $\sigma = 0.1$ outperforms other settings.

Finally, the performance of IFTSVM is compared with CDFTSVM for linear and non-linear functions. For both functions C varied from -10 to 10, and for non-linear function σ was set to 0.1. Figs. 6 and 7, respectively, show the accuracy rates of CDFTSVM and IFTSVM for linear and nonlinear functions. Except for linear function with $C = 2$, which both methods perform similar performance, IFTSVM outperforms CDFTSVM.

In the second experiment, the linear kernel with optimized C is evaluated. Table II shows the average accuracy rates along with the standard deviations (SD) and computational time (s) of IFTSVM and those methods reported in [38]. As can be seen, IFTSVM not only outperforms other methods, it also produces stable results owing to the small SD. In addition, IFTSVM and TSVM require shorter execution duration as compared with FSVM and SVM. However, CDFTSVM is the fastest method.

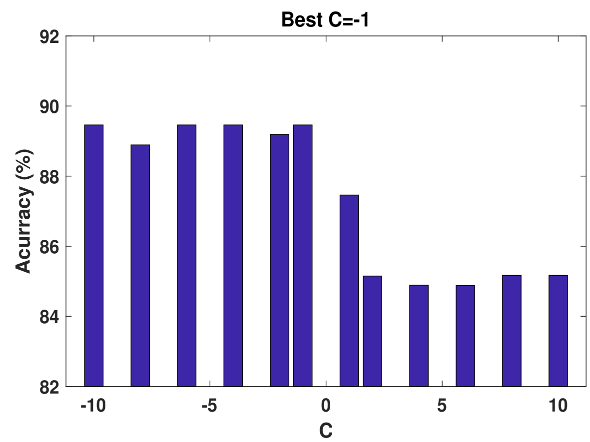


Fig. 4. The accuracy rates (%) of IFTSVM with linear function for Ionosphere data set with different C setting.

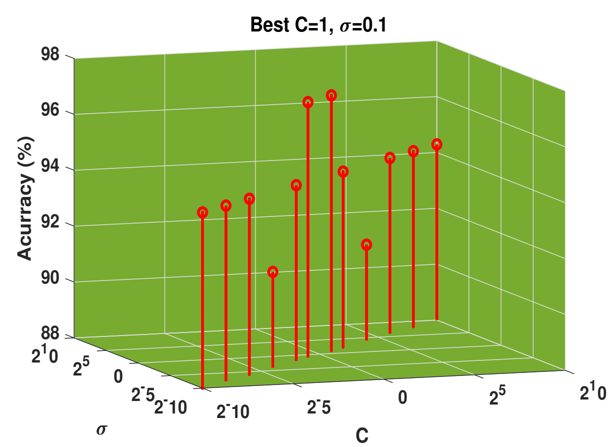


Fig. 5. The accuracy rates (%) of IFTSVM with non-linear kernel on Ionosphere data set with different C and σ settings.

Tables III and IV show the average sensitivity and specificity rates of IFTSVM and CDFTSVM for the linear kernel, respectively. As can be seen, IFTSVM achieves better results almost for all data sets. In overall, IFTSVM is able to achieve a balanced sensitivity and specificity rates for WDBC and Heart data sets, while CDFTSVM is able to achieve a balanced

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2019.2893863, IEEE Transactions on Fuzzy Systems

TABLE II
ACCURACY RATES (%) WITH STANDARD DEVIATIONS (SD) AND COMPUTATIONAL TIME (S) FOR UCI DATA SETS WITH LINEAR KERNEL

Data set	SVM		FSVM		TSVM		CDFTSVM		IFTSVM	
	ACC	Time(s)	ACC	Time(s)	ACC	Time(s)	ACC	Time(s)	ACC	Time(s)
Ionosphere	83.53±06.48	11.57	85.75±04.06	11.77	82.33±05.18	01.55	87.19±04.22	0.156	**89.46**±0.61	01.60
Australian	84.92±04.53	27.13	85.50±04.59	25.85	85.07±04.77	02.38	85.93±04.39	0.062	**86.70**±0.34	02.04
WDBC	95.34±05.17	0.148	95.87±03.21	0.147	93.84±05.86	0.047	96.39±03.52	0.016	**97.01**±0.28	0.055
WPBC	79.93±09.49	0.236	74.24±10.35	0.248	76.88±07.01	0.144	77.96±10.03	0.078	**80.21**±1.00	0.157
Bupa	66.36±06.04	01.08	67.51±07.36	01.10	61.72±05.96	0.087	64.38±06.24	0.062	**68.54**±0.62	0.089
Sonar	74.08±08.96	0.159	77.46±07.14	0.158	72.15±07.48	0.054	78.34±08.29	0.058	**81.68**±0.84	0.045
Heart	82.22±05.18	0.200	82.59±03.51	0.225	84.07±04.95	0.120	84.07±06.06	0.091	**84.81**±0.80	0.133
Pima	77.21±03.75	02.22	75.65±04.22	02.18	76.95±03.37	0.440	75.13±03.78	0.147	**79.85**±0.43	0.390

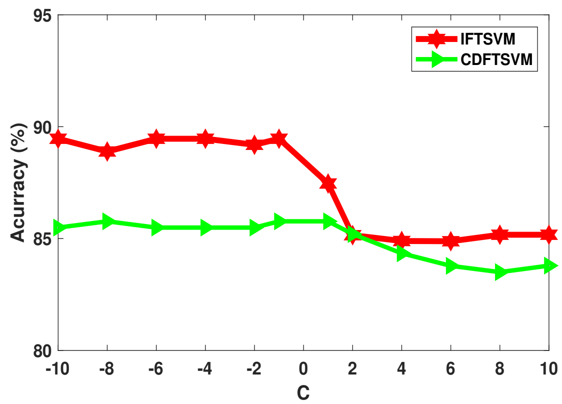


Fig. 6. Comparison of linear CDFTSVM and IFTSVM methods on Ionosphere data set.

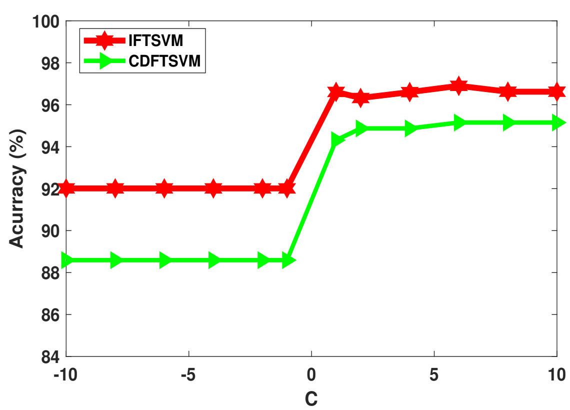


Fig. 7. Comparison of nonlinear CDFTSVM and IFTSVM on Ionosphere data set.

TABLE III
THE SENSITIVITY RATES OF CDFTDSVM AND IFTSVM ON UCI DATA SETS WITH LINEAR KERNEL

Data set	CDFTSVM	IFTSVM
Ionosphere	0.67	**0.76**
Australian	0.80	**0.81**
WDBC	0.90	**0.91**
WPBC	0.76	**0.90**
Bupa	0.65	**0.70**
Sonar	0.70	**0.75**
Heart	0.85	**0.90**
Pima	0.68	**0.72**

TABLE IV
THE SPECIFICITY RATES OF CDFTSVM AND IFTSVM ON UCI DATA SETS WITH LINEAR KERNEL

Data set	CDFTSVM	IFTSVM
Ionosphere	0.99	**0.998**
Australian	0.92	**0.93**
WDBC	**1**	**1**
WPBC	0.60	**0.70**
Bupa	0.75	**0.81**
Sonar	0.84	**0.90**
Heart	0.79	**0.81**
Pima	0.82	**0.92**

sensitivity and specificity rates only for Heart data set.

In the third experiment, non-linear kernel function with optimized parameters, i.e., C and σ, is evaluated. The average classification accuracy along with the SD and computational time of the IFTSVM and other methods is shown in Table V. For all data sets, IFTSVM outperforms other methods. Similar to the linear function, both IFTSVM and TSVM with non-linear functions need almost same execution durations.

Tables VI and VII show the average sensitivity and specificity rates for non-linear kernel function. Both IFTSVM and CDFTSVM are able to achieve balanced sensitivity and specificity rates for Ionosphere, Australian, WDBC and Heart data sets. IFTSVM also produces a balanced sensitivity and specificity rate for Sonar data set.

In the last experiment, the performance of IFTSVM with both linear and non-linear functions is compared with GB, AGB, Lasso, and RF. In this experiment, three data sets, i.e., Adult, Advert and Spam, are conducted. Following the same procedure in [56], 75% and 25% of data samples are used as a training and test samples, respectively. Table VIII shows the AUC rates along with the standard deviations (SD) of GB, AGB, Lasso, RF, and IFTSVM with both linear and non-linear functions. The performance of IFTSVM with the non-linear function for Adult is comparable to GB and better than AGB, LASSO, and RF. Also, IFTSVM with linear function performs better than other methods for Advert data set, while GB outperforms other methods for Adult and Spam data sets. However, IFTSVM is not able to produce better results for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2019.2893863, IEEE Transactions on Fuzzy Systems

TABLE V
ACCURACY RATES (%) WITH STANDARD DEVIATIONS (SD) AND COMPUTATIONAL TIME (S) FOR UCI DATA SETS WITH NON-LINEAR KERNEL

Data set	SVM		FSVM		TSVM		CDFTSVM		IFTSVM	
	ACC	Time(s)	ACC	Time(s)	ACC	Time(s)	ACC	Time(s)	ACC	Time(s)
Ionosphere	94.84±04.01	01.55	94.59±04.31	2.269	92.61±06.12	0.121	95.41±04.93	0.062	**96.90**±0.43	0.137
Australian	85.50±04.53	01.31	85.50±04.59	1.787	85.50±04.59	0.160	86.81±04.84	0.102	**87.81**±0.30	0.169
WDBC	94.84±04.23	0.779	95.34±03.80	1.407	95.34±03.80	0.116	96.39±03.52	0.078	**98.25**±0.24	0.102
WPBC	81.51±07.13	0.593	77.88±09.43	1.277	75.30±07.93	0.178	82.51±08.05	0.094	**82.87**±0.67	0.198
Bupa	70.68±08.28	0.391	72.71±07.93	0.551	71.86±05.71	0.110	71.84±05.67	0.047	**75.98**±0.55	0.089
Sonar	89.42±05.41	0.924	88.92±06.95	1.604	89.42±05.41	0.189	89.44±05.31	0.109	**92.23**±0.74	0.193
Heart	84.07±05.25	0.320	82.59±04.29	0.557	80.74±07.16	0.118	84.81±04.08	0.016	**86.67**±0.55	0.093
Pima	75.65±03.80	21.35	75.26±02.91	26.01	77.34±05.16	0.184	76.17±02.68	0.159	**79.17**±0.29	0.178

TABLE VI
THE SENSITIVITY RATES OF CDFTDSVM AND IFTSVM ON UCI DATA SETS WITH NON-LINEAR KERNEL

Data set	CDFTSVM	IFTSVM
Ionosphere	**0.93**	0.91
Australian	0.84	**0.89**
WDBC	**0.95**	0.94
WPBC	0.87	**0.95**
Bupa	**0.69**	**0.69**
Sonar	**0.93**	**0.93**
Heart	0.85	**0.89**
Pima	0.67	**0.72**

TABLE VII
THE SPECIFICITY RATES OF CDFTSVM AND IFTSVM ON UCI DATA SETS WITH NON-LINEAR KERNEL

Data set	CDFTSVM	IFTSVM
Ionosphere	0.94	**0.99**
Australian	0.87	**0.92**
WDBC	0.97	**0.99**
WPBC	0.69	**0.74**
Bupa	0.75	**0.82**
Sonar	0.78	**0.89**
Heart	**0.82**	0.80
Pima	0.83	**0.91**

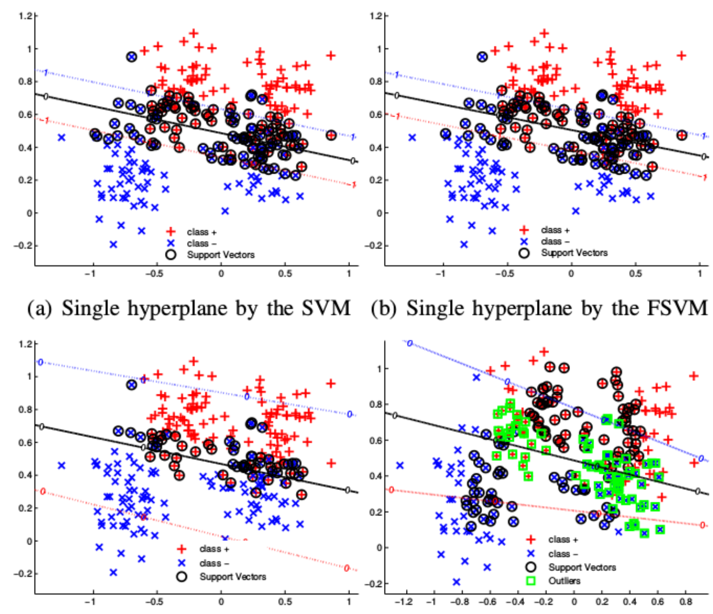
Adult and Spam data sets, but it performs more or less similar to Lasso, RF, and AGB.

B. Artificial data sets

In this Section, IFTSVM is evaluated with two artificial data problems i.e., Ripleys synthetic and Circle-in-the-square, as follows:

1) Ripleys data set: is a binary classification problem which has been generated by mixing two Gaussian distributions. Each data sample includes two features. Training and test sets include 250 and 1000 samples, respectively. In order to reduce the effect of outlier data on the hyper-plane, μ is set to 0.1. Table IX shows the results of SVM, FSVM, TSVM, CDFTSVM [38] and IFTSVM. The outcome indicates that IFTSVM outperform other methods for both linear and the non-linear functions.

Figs. 8 and 10 shows the linear and non-linear separating hyper-planes constructed by conventional SVM, TSVM, and CDFTSVM, respectively [38]. In addition, the linear and non-linear separating hyper-planes constructed by IFTSVM, respectively, are shown in Figs. 8 and 10. As can be seen, SVM (Figs. 8 (a) and 10 (a)) and FSVM (Figs. 8 (b) and 10 (b)) generate only one single hyper-plane, while, TSVM (Figs. 8(c) and 10(c)), CDFTSVM (Figs. 8(d) and 10(d)) and IFTSVM (Figs. 9 and 11) produce two proximal hyper-planes.



(a) Single hyperplane by the SVM (b) Single hyperplane by the FSVM

(c) Dual hyperplanes by TSVM (d) Dual hyperplanes by CDFTSVM

Fig. 8. Generated hyperplane(s) by linear SVM, FSVM, TSVM, CDFTSVM on Ripleys data set (adapted from [38]).

2) Circle-in-the-square: is also a binary classification problem, which requires a classifier to identify which samples within a unit square are placed inside or outside a circle. The location of the circle is center and covers half of the square. The performance of IFTSVM with non-linear function was compared with fuzzy ARTMAP (FAM) [59], Q-learning fuzzy ARTMAP (QFAM) [60] and CDFTSVM [38]. According to [59], two experiments were conducted. Each experiment is repeated ten times with different data samples.

In the first experiment, different numbers of training samples, i.e., 100, 1000 and 10000, are used, while 1000 samples are used for the test. Table X shows the accuracy rates of FAM, QFAM, CDFTSVM, and IFTSVM. It can be seen that the

TABLE VIII
AUC RATES WITH STANDARD DEVIATIONS (SD) AND COMPUTATIONAL TIME (S) FOR UCI DATA SETS

Data set	Gradient Boosting (GB)	Accelerated Gradient Boosting (AGB)	Lasso	Random Forest (RF)	IFTSVM Linear	IFTSVM Non-Linear
Adult	**0.920**±0.004	0.913±0.004	0.902±0.004	0.858±0.008	0.905±0.010	0.915±0.055
Advert	0.973±0.012	0.971±0.015	0.973±0.008	0.983±0.008	**0.985**±0.002	0.981±0.009
Spam	**0.980**±0.003	0.977±0.003	0.970±0.004	0.979±0.003	0.969±0.001	0.970±0.001

TABLE IX
THE ACCURACY RATES (%) FOR RIPLEYS DATA SET

Data set	SVM	FSVM	TSVM	CDFTSVM	IFTSVM
linear	89.70	88.80	89.20	89.10	**90.00**
Nonlinear	90.40	91.10	90.50	91.30	**91.50**

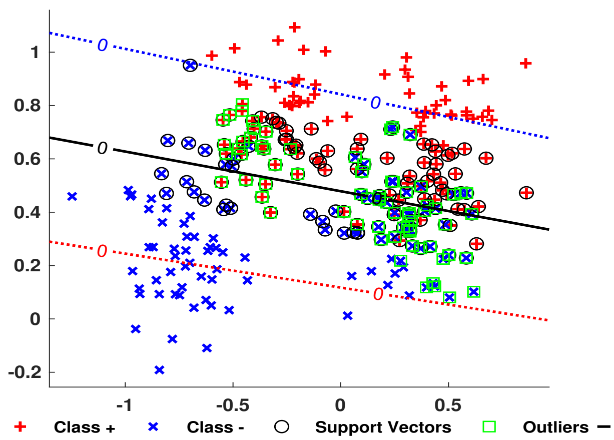
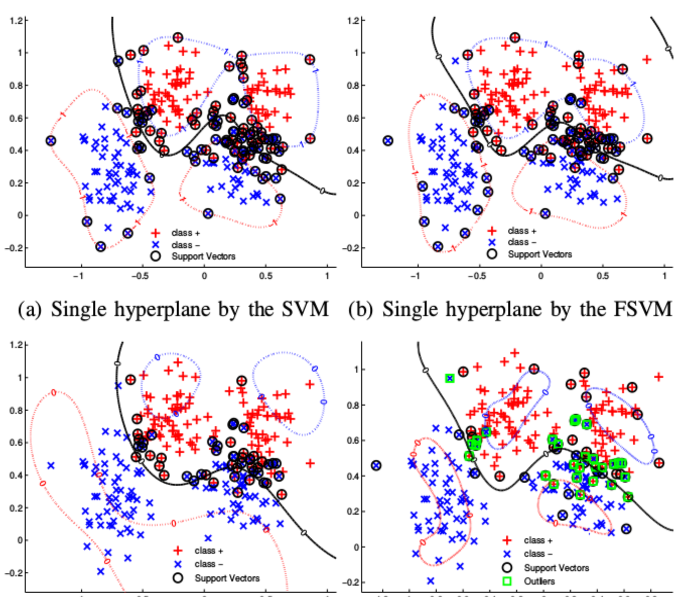


Fig. 9. Generated hyperplanes by linear IFTSVM on Ripleys data set.



(a) Single hyperplane by the SVM (b) Single hyperplane by the FSVM

(c) Dual hyperplanes by TSVM (d) Dual hyperplanes by CDFTSVM

Fig. 10. Generated hyperplane(s) by non-linear SVM, FSVM, TSVM, CDFTSVM on Ripleys data set (adapted from [38]).

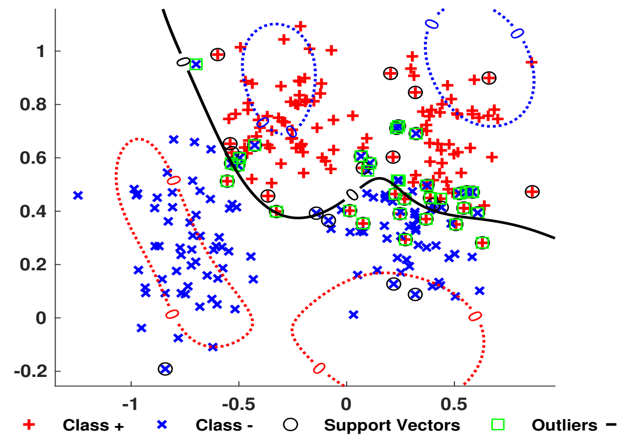


Fig. 11. Generated hyperplanes by nonlinear IFTSVM on Ripleys data set.

classification error of all methods except CDFTSVM reduce when the numbers of training samples increase from 100 to 10000. The CDFTSVM produces the inferior result for 1000 cases as compared with 100 and 10000 cases. In general, IFTSVM outperforms other methods for 1000 and 10000 statistically, as there is no overlap between the 95% confidence intervals of IFTSVM and other methods, while CDFTSVM and IFTSVM perform at the same level statistically for the 100 cases.

TABLE X
ACCURACY RATES (%) FOR THE CIRCLE-IN-THE-SQUARE PROBLEM WITH 95% CONFIDENCE INTERVALS ("MEAN", "UPPER" AND "LOWER" INDICATE THE MEAN ACCURACY, UPPER AND LOWER BOUNDS OF THE 95% CONFIDENCE INTERVALS, RESPECTIVELY)

Training Samples		100	1000	10000
FAM	Lower	88.63	93.26	95.39
	Mean	86.16	93.89	96.14
	Upper	89.89	94.51	96.54
QFAM	Lower	88.70	95.30	96.64
	Mean	90.31	96.04	97.15
	Upper	93.52	97.1	97.46
CDFTSVM	Lower	91.48	89.58	99.02
	Mean	93.60	90.06	99.19
	Upper	95.45	90.54	99.30
IFTSVM	Lower	**95.34**	**97.57**	**99.44**
	Mean	**96.38**	**97.93**	**99.50**
	Upper	**97.10**	**98.39**	**99.56**

In the second experiment, the performance of IFTSVM further tested by injecting noise into the training samples. The numbers of training and test samples, respectively, are fixed to 10000 and 1000, and different level of noise, i.e., 5% and 10%, is injected to the class of training samples. For example, 5% or 10% of the training samples are randomly picked and flipped their class. The accuracy rates are shown in Table XI. Obviously, the accuracy rates of all methods decrease by raising the noise level. IFTSVM achieves the highest accuracy rates for both noise-free and noisy data sets. The accuracy rates of both QFAM and CDFTSVM dramatically drop from 97.15% (for 0% noise) to 94.71% (for 10% noise) and 99.19% (for 0% noise) to 97.32% (for 10% noise), respectively. However, this trend is slow for IFTSVM, which drops from 99.50% (for 0% noise) to 98.54% (for 10% noise). In general, IFTSVM produces better results as compared with QFAM and CDFTSVM for both noise-free and noisy data sets. This is because of the capability of IFTSVM in reducing the effect of noise and outliers.

TABLE XI
ACCURACY RATES (%) FOR THE CIRCLE-IN-THE-SQUARE PROBLEM WITH THE 95% CONFIDENCE INTERVALS FOR DIFFERENT LEVEL OF NOISE

Noise(%)		0	5	10
QFAM	Lower	96.64	94.80	94.19
	Mean	97.15	95.30	94.71
	Upper	97.46	95	94.38
CDFTSVM	Lower	99.02	97.54	96.91
	Mean	99.19	97.81	97.32
	Upper	99.30	97.99	97.60
IFTSVM	Lower	**99.44**	**98.45**	**98.40**
	Mean	**99.50**	**98.65**	**98.54**
	Upper	**99.56**	**98.90**	**98.67**

V. CONCLUSION

In this paper, a new IFTSVM model, which is inspired by IFN and FTSVM, for solving binary classification problems has been proposed. The IFTSVM obtains two non-parallel hyper-planes by solving two QPPs instead of one as in traditional SVM. It classifies an input sample based on both degrees of membership and non-membership functions, which helps to decrease the effect of noise and outliers. The effectiveness of IFTSVM has been evaluated by eleven benchmarks and two artificially generated data sets. The results of IFTSVM were compared with those from the traditional SVM, FSVM, TSVM, CDFTSVM, FAM, and QFAM and other state-of-the-art classification algorithms. Overall, IFTSVM is able to produce astonishing results. However, it is sensitive to C, in which, if it is not chosen properly, IFTSVM produces inferior results. Our future work is focused on enhancing the structure of IFTSVM in order to solve imbalance classification problems.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grant 61772344 and Grant 61732011), in part by the Natural Science Foundation of SZU (Grant 827-000140, Grant 827-000230, and Grant 2017060).

REFERENCES

[1] V. N. Vapnik, *Statistical learning theory*, 1998.
[2] X. Gao, L. Fan, and H. Xu, "Multiple rank multi-linear kernel support vector machine for matrix data classification," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 2, pp. 251–261, Feb 2018.
[3] Y. Li, Q. Leng, and Y. Fu, "Cross kernel distance minimization for designing support vector machines," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 5, pp. 1585–1593, Oct 2017.
[4] J. Zhang, Q. Hou, L. Zhen, and L. Jing, "Locality similarity and dissimilarity preserving support vector machine," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 10, pp. 1663–1674, Oct 2018.
[5] S.-G. Chen and X.-J. Wu, "Multiple birth least squares support vector machine for multi-class classification," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 6, pp. 1731–1742, Dec 2017.
[6] C. Chuang, "Fuzzy weighted support vector regression with a fuzzy partition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 630–640, June 2007.
[7] L. Chen, M. Zhou, M. Wu, J. She, Z. Liu, F. Dong, and K. Hirota, "Three-layer weighted fuzzy support vector regression for emotional intention understanding in humanrobot interaction," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2524–2538, Oct 2018.
[8] P. Hao and J. Chiang, "Fuzzy regression analysis by support vector learning approach," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 428–441, April 2008.
[9] H. Do, A. Kalousis, and M. Hilario, "Feature weighting using margin and radius based error bound optimization in svms," in *Machine Learning and Knowledge Discovery in Databases*, 2009, pp. 315–329.
[10] Y. Wang, S. Wang, and K. K. Lai, "A new fuzzy support vector machine to evaluate credit risk," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 820–831, Dec 2005.
[11] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using lstm networks and svm," *IEEE Access*, vol. 6, pp. 7675–7686, 2018.
[12] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear svm," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 838–850, May 2017.
[13] V. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machines," *Neural Computation*, vol. 12, no. 9, pp. 2013–2036, 2000.
[14] Z. Wang, Y.-H. Shao, and T.-R. Wu, "Proximal parametric-margin support vector classifier and its applications," *Neural Computing and Applications*, vol. 24, no. 3, pp. 755–764, Mar 2014.
[15] Y. H. Shao and N. Y. Deng, "A coordinate descent margin based-twin support vector machine for classification," *Neural Networks*, vol. 25, pp. 114 – 121, 2012.
[16] H. Do and A. Kalousis, "Convex formulations of radius-margin based support vector machines," vol. 28, no. 1, pp. 169–177, 2013.
[17] Y. Liu and Y. Chen, "Face recognition using total margin-based adaptive fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 178–192, Jan 2007.
[18] X. Wu, W. Zuo, L. Lin, W. Jia, and D. Zhang, "F-svm: Combination of feature transformation and svm learning via convex relaxation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5185–5199, Nov 2018.
[19] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 69–74, Jan 2006.
[20] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, May 2007.
[21] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7535 – 7543, 2009.
[22] Z. Qi, Y. Tian, and Y. Shi, "Robust twin support vector machine for pattern classification," *Pattern Recognition*, vol. 46, no. 1, pp. 305 – 316, 2013.
[23] Y. Shao, C. Zhang, X. Wang, and N. Deng, "Improvements on twin support vector machines," *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 962–968, June 2011.
[24] R. Khemchandani, Jayadeva, and S. Chandra, "Optimal kernel selection in twin support vector machines," *Optimization Letters*, vol. 3, no. 1, pp. 77–88, Jan 2009.
[25] C.-N. Li, Y.-F. Huang, H.-J. Wu, Y.-H. Shao, and Z.-M. Yang, "Multiple recursive projection twin support vector machine for multi-class classification," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 5, pp. 729–740, Oct 2016.

[26] Z.-M. Yang, H.-J. Wu, C.-N. Li, and Y.-H. Shao, "Least squares recursive projection twin support vector machine for multi-class classification," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 3, pp. 411–426, Jun 2016.

[27] M. A. Kumar and M. Gopal, "Application of smoothing technique on twin support vector machines," *Pattern Recognition Letters*, vol. 29, no. 13, pp. 1842 – 1848, 2008.

[28] B. Scholkopf and A. Smola, *Learning with Kernels*, 2002.

[29] C. Zhang, Y. Tian, and N. Deng, "The new interpretation of support vector machines on statistical learning theory," *Science in China Series A: Mathematics*, vol. 53, no. 1, pp. 151–164, Jan 2010.

[30] R. Khemchandani, Jayadeva, and S. Chandra, "Fuzzy twin support vector machines for pattern classification," *Mathematical Programming and Game Theory for Decision Making*, pp. 131–142, 2008.

[31] C. F. Lin and S. D. Wang, "Fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 464–471, March 2002.

[32] R. Batuwita and V. Palade, "Fsvm-cil: Fuzzy support vector machines for class imbalance learning," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 558–571, June 2010.

[33] X. Yang, G. Zhang, J. Lu, and J. Ma, "A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 105–115, Feb 2011.

[34] R. K. Sevakula and N. K. Verma, "Compounding general purpose membership functions for fuzzy support vector machine under noisy environment," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1446–1459, 2017.

[35] M. M. Zhou, L. Li, and Y. L. Lu, "Fuzzy support vector machine based on density with dual membership," vol. 2, pp. 674–678, 2009.

[36] K. W. Chang, C. J. Hsieh, and C. J. Lin, "Coordinate descent method for large-scale l2-loss linear support vector machines," *J. Mach. Learn. Res.*, vol. 9, pp. 1369–1398, Jun. 2008.

[37] C. jui Hsieh, K. wei Chang, C. jen Lin, and S. S. Keerthi, "A dual coordinate descent method for large-scale linear svm," 2008.

[38] B. Gao, J. Wang, Y. Wang, and C. Yang, "Coordinate descent fuzzy twin support vector machine for classification," pp. 7–12, Dec 2015.

[39] S. G. Chen and X.-J. Wu, "A new fuzzy twin support vector machine for pattern classification," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 9, pp. 1553–1564, Sep 2018.

[40] M. Ha, C. Wang, and J. Chen, "The support vector machine based on intuitionistic fuzzy number and kernel function," *Soft Computing*, vol. 17, no. 4, pp. 635–641, Apr 2013.

[41] S. Rezvani, "Ranking method of trapezoidal intuitionistic fuzzy numbers," *Annals of Fuzzy Mathematics and Informatics*, vol. 5, no. 3, pp. 515–523, 2013.

[42] S. Rezvani and X. Wang, "A new type-2 intuitionistic exponential triangular fuzzy number and its ranking method with centroid concept and euclidean distance," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2018, pp. 1–8.

[43] D. S. R.O. Duda, P.E. Hart, *Pattern classification*, 2001.

[44] M. H. Li K, "A fuzzy twin support vector machine algorithm," *International Journal of Application or Innovation in Engineering and Management*, vol. 2, no. 3, pp. 459–465, 2013.

[45] Z. Zhang, L. Zhen, N. Deng, and J. Tan, "Sparse least square twin support vector machine with adaptive norm," *Applied Intelligence*, vol. 41, no. 4, pp. 1097–1107, Dec 2014.

[46] S. Ding, J. Yu, B. Qi, and H. Huang, "An overview on twin support vector machines," *Artificial Intelligence Review*, vol. 42, no. 2, pp. 245–252, Aug 2014.

[47] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.

[48] D. Tomar and S. Agarwal, "Twin support vector machine: A review from 2007 to 2014," *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 55 – 69, 2015.

[49] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[50] B. D. Ripley, *Pattern recognition and neural networks*, 1996.

[51] L. Y. Cai and H. K. Kwan, "Fuzzy classifications using fuzzy inference networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 334–347, June 1998.

[52] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, 1979.

[53] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[54] J. Huang and C. X. Ling, "Using auc and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, March 2005.

[55] J. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, pp. 1189–1232, 2001.

[56] G. Biau and B. Cadre, "Accelerated gradient boosting," *arXiv:1803.02042*, vol. 29, pp. 1–18, 2018.

[57] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Sta- tistical Society. Series B*, pp. 267–288, 1996.

[58] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[59] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, Sept 1992.

[60] F. Pourpanah, C. P. Lim, and Q. Hao, "A reinforced fuzzy artmap model for data classification," *International Journal of Machine Learning and Cybernetics*, Jun 2018.

Salim Rezvani is a PhD. student in the Big Data Institute, College of Computer Science and Software Engineering, Guangdong Key Lab. of Intelligent Information Processing, Shenzhen University (SZU), China.

Xizhao Wang (M'03−SM'04−F'12) served in Hebei University as a professor and the dean of school of Mathematics and Computer Sciences before 2014. After 2014 Prof. Wang worked as a professor in Big Data Institute of ShenZhen University. Prof. Wangs major research interests include uncertainty modeling and machine learning for big data. Prof. Wang has edited 10+ special issues and published 3 monographs, 2 textbooks, and 200+ peer-reviewed research papers. As a Principle Investigator (PI) or co-PI, Prof. Wang's has completed 30+ research projects. Prof. Wang has supervised more than 100 Mphil and PhD students.

Prof. Wang is an IEEE Fellow, the previous BoG member of IEEE SMC society, the chair of IEEE SMC Technical Committee on Computational Intelligence, the Chief Editor of Machine Learning and Cybernetics Journal, and associate editors for a couple of journals in the related areas. He was the recipient of the IEEE SMCS Outstanding Contribution Award in 2004 and the recipient of IEEE SMCS Best Associate Editor Award in 2006. He is the general Co-Chair of the 2002-2017 International Conferences on Machine Learning and Cybernetics, cosponsored by IEEE SMCS. Prof. Wang was a distinguished lecturer of the IEEE SMCS.

Farhad Pourpanah received his PhD degree in computational intelligence from the University Science Malaysia in 2015. He is currently an associate researcher at the College of Mathematics and Statistics, Shenzhen University (SZU), China. Before joining to SZU, he was a postdoctoral research fellow with the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China. His research interests include evolutionary algorithms, pattern recognition and deep learning.