Neural Comput & Applic (2011) 20:671–685
DOI 10.1007/s00521-010-0409-1

ORIGINAL ARTICLE

Relaxed constraints support vector machines for noisy data

Mostafa Sabzekar · Hadi Sadoghi Yazdi · Mahmoud Naghibzadeh

Received: 17 January 2010 / Accepted: 4 June 2010 / Published online: 20 June 2010

Abstract Real-world data collected for computer-based applications are frequently impure. Differentiation of outliers and noisy data from normal ones is a major task in data mining applications. On the other hand, elimination of noisy and outlier data from training samples of a dataset may lead to over-fitting or information loss. A fuzzy support vector machine (FSVM) provides an effective means to deal with this problem. It reduces the effect of the noisy data and outliers by using a fuzzy membership functions. In this paper, a new formation for SVMs is introduced that considers importance degrees for training samples. The constraints of the SVM are converted to fuzzy inequalities. The proposed method, RSVM, shows better efficiency in the classification of data in different domains. Especially, using the proposed RSVM for multi-class classification of arrhythmia disease is presented at the end of this paper as a practical case study to show the effectiveness of the proposed system.

Keywords Support vector machine · Noisy data · Importance degree · Fuzzy inequality · Multi-class classification

1 Introduction

Real-world data collected for computer-based applications are frequently impure. Therefore, noise handling should be a definite characteristic of any practical data mining method. A typical data mining application consists of four steps: data preparation, data transformation, pattern discovery, and pattern evaluation [1]. Since the results of the data mining applications rely on the quality of the prepared data, the collection and enhancement of data takes the majority of the project developing time circle [2]. Noisy information may exist for many reasons, such as data collection inaccuracy, device limitations, data transmission errors, man-made perturbation, and so on. There have been many efforts to solve this problem [3–5]. In data mining applications, some researchers believe that noises should be eliminated from the collected data [6–9]. This process is referred to as data cleansing. Although data cleansing methods are devised especially for this purpose and it is frequently used in many applications, it may cause negative effects in certain circumstances [1]:

1. Data cleansing only takes effect on certain types of errors.
2. Data cleansing cannot result in perfect data.
3. Data cleansing cannot be unconditionally applied to any data sources.
4. Eliminating noisy data items may lead to information loss.
5. The traditional data mining framework (without error awareness) isolates data cleansing from the actual mining process.

Novelty detection addresses the problem of detecting outliers from the rest of normal data patterns. There are many researches for novelty detection for example, estimation method. In these approaches, the noisy data and outliers (abnormal data patterns) are assumed to have different distribution levels from those of normal data patterns. Gaussian mixture model [10], Parzen density estimation model [11], and nearest neighbor method [12] are paradigm of these approaches. One drawback of such

M. Sabzekar (✉) · H. Sadoghi Yazdi · M. Naghibzadeh
Department of Computer Engineering,
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: sabzekar@wali.um.ac.ir

methods is that the noises or outliers are assumed to have different distribution levels from those of normal data patterns. But in real-world datasets, outliers may lie in the region of the normal data patterns. In this situation, the distribution estimation methods are unable to identify the outliers. Support vector novelty detector (SVND) [13] is another method, which estimates a sphere to contain all the normal data patterns with the smallest radius. The "smallest radius" means that the outliers will lie outside this sphere. For improving the performance of SVND, Tax [13] proposed the method of incorporating the known outliers in the development of SVND algorithm for improving the separation of the normal data patterns and the outliers. By not only letting the normal data patterns lie inside the sphere but also letting the existed outliers lie outside the sphere, Tax found that a more efficient description of the sphere can be obtained.

As a data mining method, support vector machines (SVMs) [14] suffer from the problem of noisy data and outliers. Since the classifier obtained by SVM depends on only a small part of the samples (support vectors), it is very easy for it to become sensitive to noises or outliers in the training set [15].

SVM is introduced originally by Vapnik within the area of statistical learning theory and structural risk minimization. It is a powerful tool for pattern classification and function estimation formulated as a convex optimization problem, usually quadratic programming (QP) problem, for which the dual problem is solved.

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to } y_i\left(w^T x_i + b\right) \geq 1 - \xi_i \\ &\quad \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

In the traditional SVM, formulized in (1), there is no possibility to handle noisy data and outliers. That is, the effect of different training samples on the obtained classifier is identical. Therefore, noisy data and outliers, similar to the real data, can affect the SVM results. However, in real-world applications, the effects of the training samples on the classification outcome may have to be different. That is, some training samples may be more important than others. There are some efforts to overcome these deficiencies of the SVM. Herbrich and Weston presented adaptive margin (AM) SVM in [16]. This approach pushed the idea of capacity control via margin maximization to its limit by allowing adapting margins at each training pattern. They showed, by experiment, that their method is robust against outliers. In [17], at first, a SVM is trained for training samples. Samples with large Lagrange multipliers are eliminated as outliers. Again a SVM is trained for training samples in the absence of the outliers. But one problem is that this method is rather empirical, and there is still no general method to distinguish outliers from those of normal data patterns. Since in the SVM methods, the separation margin is defined as the distances from the nearest samples to the classification boundary, the trained classifier depends on only support vectors. To avoid this in [15], another kind of margin is proposed. This margin is the distance from the two class centers to the separation boundary. Traditional SVM is then reformulated for the linear and nonlinear cases. It claimed that this method is less sensitive to noises and outliers. Also, Graf et al. [18] studied the classification performance of SVMs in a normalized feature space. They emphasized that the normalization of the input data can dramatically influence the results of the classification, as well as the convergence of the SV algorithm. Their proposed algorithm has the same optimal solution for as the standard support vector (SV) algorithm, but the considered correction is introduced in the final computation of the position of the offset b of the optimal separating hyperplane (OSH). They also, experimentally, showed that the new method is stable to noise on the input data and is optimal on the average.

Another famous version of SVM that considers noises and outliers was proposed by Lin and Wang [19, 20]. They applied a fuzzy membership to each input point. Then the SVMs are reformulated such that different input points can make different contributions to the learning of decision surface. This new method was called fuzzy SVM (FSVM). FSVM assigns a fuzzy membership to each input data pattern such that different data patterns can make different contributions to the trained SVM. FSVM has been successfully applied to different applications such as credit risk assessment [21], intrusion detection [22], face recognition [23], text categorization [24] etc.

In FSVM, a fuzzy membership value is assigned to each training sample such that noises or outliers have small membership values, and normal data patterns have membership values close to 1. FSVM provides a very good method to classify data with noises or outliers. It introduces a degree of importance for each sample in the cost-function of the SVM formulation (1).

In this paper, a different method to include a degree of importance to each sample in the SVM formulation is presented. Since the cost-function of SVM $(1/2\|w\|^2 + C\sum_{i=1}^{n}\xi_i)$ is very sensitive to changes and may lead to undesirable results when a modification is applied, modifications are applied to the constraints of the SVM. A new version of the SVM is proposed in which fuzzy memberships for each training sample is considered in the constraints of the SVM formulation (1). We call this new model relaxed constraints support vector machines

(RSVM). The tolerance for data patterns and the degree of uncertainty for a dataset is also considered in the RSVM.

The rest of this paper is organized as follows. A brief review of the architectures of SVMs and FSVMs is described in Sect. 2. The proposed RSVM and fuzzy RSVM (FRSVM) models are explained in detail in Sects. 3 and 4, respectively. Section 5 considers the extension of the proposed method for solving multi-class classification problems. In Sect. 6, the proposed system will be evaluated using different case studies. Some concluding remarks are given in Sect. 7.

2 SVM and fuzzy SVM

In this section, we describe the theory of SVM and FSVM for classification purposes and highlight their differences.

2.1 SVM architecture

Suppose that we have a training sample set $\{(x_1, y_1),(x_2, y_2),\ldots,(x_n, y_n)\}$ and each sample $x_i \in \Re^m$ belongs to either of two classes with given label $y_i = \{-1, 1\}$ for $i = 1,\ldots,n$. When the training samples are linear separable, the SVM separates the two classes with maximum margin between them, without any misclassification error. The optimal separating hyperplane (OSH) can be achieved by solving the following quadratic programming (QP) problem:

$$\begin{aligned} &\text{Minimize} \quad Q(w,b) = \frac{1}{2}\|w\|^2 \\ &\text{subject to } y_i\left(w^T x_i + b\right) \geq 1 \end{aligned} \tag{2}$$

where w is a weight vector of hyperplane, and b is the bias term. When input patterns are nonlinear separable, it is not possible to satisfy all the constraints of the SVM (2). Therefore, slack variables ξ_i are introduced to measure the violation amount of the constraints. Thus, the SVM is reformulated as:

$$\begin{aligned} &\text{Minimize} \quad Q(w,b,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to } y_i\left(w^T x_i + b\right) \geq 1 - \xi_i \\ &\quad \xi_i \geq 0, \quad i = 1,\ldots,n \end{aligned} \tag{3}$$

where parameter C controls the misclassification errors. If C is taken to be a large number, it will force the slack variables to become small numbers. On the contrary, if C is taken to be a small number, the slack variables will grow and training data that are far from others are allowed to be misclassified. For many datasets, finding a linear classifier is impossible. In order to classify nonlinearly, a solution is to map the input space into a higher dimension feature space and searching the OSH in this feature space. Therefore, the mapping function $z = \varphi(x)$ is introduced such that it satisfies Mercer's condition [25, 26]. To solve the QP problem, one needs to compute the scalar products of the form $\varphi(x_i) \cdot \varphi(x_j)$. The problem is that we do not know the shape of $\varphi(x)$. It is therefore convenient to introduce the kernel function $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) = z_i \cdot z_j$. By using the Lagrange multiplier method and kernel trick, the QP problem for finding the SVM is as follows:

$$\begin{aligned} &\text{Minimize} \quad Q(\alpha) = \frac{1}{2}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \sum_{i=1}^{n}\alpha_i \\ &\text{subject to } \sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \end{aligned} \tag{4}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the vector of non-negative Lagrange multipliers and solution of the QP problem (4). The point x_i with $\alpha_i \geq 0$ is called support vector (SV). The final classifier is to form

$$f(x) = w^T \cdot \varphi(x) + b = \sum_{i\in S}\alpha_i y_i K(x_i, x) + b \tag{5}$$

where S is the set of support vector indices, and b is given by

$$b = y_j - \sum_{i\in S}\alpha_i y_i K(x_i, x_j) \tag{6}$$

Here, x_j is an unbounded support vector ($0 < \alpha_i < C$). The decision function is

$$\begin{aligned} D(x) &= \text{sign}(f(x)) \\ &= \text{sign}\left(\sum_{i\in S}\alpha_i y_i K(x_i, x) + b\right) \end{aligned} \tag{7}$$

2.2 Fuzzy SVM architecture

In the standard SVM, the role of each input pattern in the trained classifier is identical. In the other words, each data point is fully assigned to one of the two classes. However, in many applications, some input points, such as the outliers, may not be exactly assigned to one of these two classes, and each point does not have the same meaning to the decision surface. To solve this problem, Lin and Wang [19, 20] proposed fuzzy SVM (FSVM). FSVM applies a fuzzy membership to each input point such that noisy data or outliers have small membership values in order to have less contribution to learning of decision surface.

Suppose we are given a set $\{(x_1, y_1, s_1),(x_2, y_2, s_2),\ldots, (x_n, y_n, s_n)\}$ of labeled training points with associated fuzzy membership. Each training sample x_i is given a label $y_i = \{-1, 1\}$ and a fuzzy membership s_i with $\sigma \leq s_i \leq 1$, and $\sigma > 0$ is a sufficiently small constant. The QP problem for finding the optimal hyperplane can be described as:

$$\begin{aligned} &\text{Minimize} \quad Q(w,b,\xi_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} s_i\xi_i \\ &\text{subject to } y_i\left(w^T x_i + b\right) \geq 1 - \xi_i \\ &\quad \xi_i \geq 0, \quad i = 1,\ldots,n \end{aligned} \tag{8}$$

The term $s_i\xi_i$ is a measure of error with different weights. A smaller s_i for input sample x_i can reduce the effect of corresponding ξ_i. Therefore, x_i can be treated as an outlier or a noisy data. Solving problem (8) is the same as that for standard SVM with little differences. However, choosing proper fuzzy memberships for a given classification problem is vital for FSVM. In [19] fuzzy membership, s_i for FSVM is introduced as follows:

$$s_i = \begin{cases} 1 - (\|x_+ - x_i\|/(r_+ + \delta)) & \text{if } x_i \in \text{class1} \\ 1 - (\|x_- - x_i\|/(r_- + \delta)) & \text{if } x_i \in \text{class2} \end{cases} \tag{9}$$

where x_+ and x_- is the mean of class 1 and class 2, respectively. Also, r_+ is the radius of class 1

$$r_+ = \max_{\{x_i : x_i \in \text{Class1}\}} \|x_+ - x_i\| \tag{10}$$

and radius of Class 2 is

$$r_- = \max_{\{x_i : x_i \in \text{Class2}\}} \|x_- - x_i\| \tag{11}$$

Figure 1 shows the decision functions obtained from standard SVM (Fig. 1a) and FSVM (Fig. 1b). For the pointed to training sample in the square class, FSVM ignores this outlier and assigns it to the sphere class while SVM attempts to train an accurate classifier.

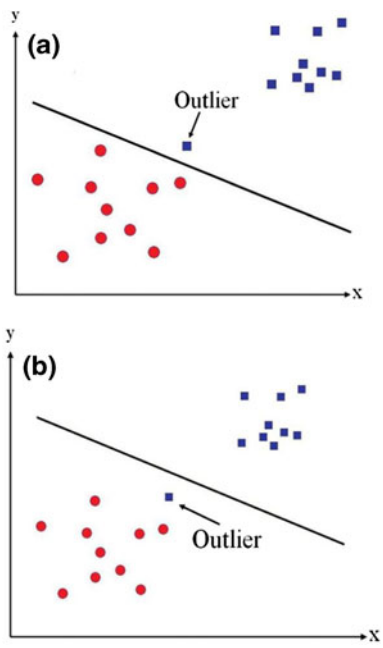

Fig. 1 Behavior of SVM (**a**) and FSVM (**b**) in the presence of an outlier

For the FSVM classifier, it is very important that a suitable model of fuzzy membership function is determined because choosing a proper fuzzy membership function can reduce the effect of noises and outliers. There are some efforts to propose proper fuzzy memberships such as the ones proposed in [20, 27, 28].

The FSVM provides a useful method to classify data with noises or outliers. The fuzzy membership values come into the penalty term of the SVM cost-function. The goal of this paper is that by changing the constraints of the SVM formulation (3), a new solution for handling noisy data is found. In the next section, the structure of our proposed method will be described in detail.

3 The structure of RSVM

The FSVM is so sensitive to changes that a minor modification in the cost-function may lead to inaccurate classifier. Thus, effectiveness of the FSVM is influenced by choosing fuzzy membership values s_i. This is the main motivation that we chose to work on the constraints of the SVM formulation.

As we know, each training sample x_i imposes a constraint to the learning system as follows:

$$y_i\left(w^T x_i + b\right) \geq 1 - \xi_i, \quad \text{for } i = 1,2,\ldots,n. \tag{12}$$

The non-negative slack variables ξ_i are introduced to allow inseparability. Figure 2 shows the effect of the slack variables in a two-dimensional space. If there are no slack variables, the margin could not exist. In fact the slack variables somewhat controls noisy data but they are unknown parameters of the learning system and thus cannot be manually assigned to each training sample, similar to fuzzy memberships values in the FSVM.

As mentioned before, each training sample imposes a constraint to the learning system. These constraints altogether construct the feasible region. The final results must be selected from this region. Figure 3 shows a typical feasible region that is created by four constraints as follows:

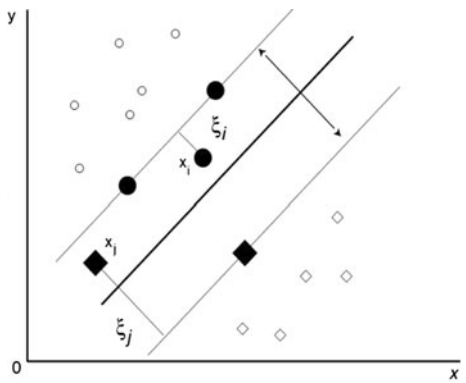

Fig. 2 Inseparable case in a 2-dimensional space

$$\text{Optimize} \quad f(x)$$
$$\text{subject to} \begin{cases} \text{constraint 1} \\ \text{constraint 2} \\ \text{constraint 3} \\ \text{constraint 4} \end{cases}$$

The SVM problem also is an optimization problem with n constraints where n is the number of samples. The unknown parameters of the SVM, $(w, b, \xi) \in \Re^{m+1+n}$, are selected from the feasible region, which is created by the constraints of the SVM.

We use fuzzy inequality in each constraint of the training samples in order to give extra relaxation to each constraint satisfaction. Therefore, the constraint associated with each training sample x_i is in the form of:

$$y_i\left(w^T x_i + b\right) \gtrsim 1 - \xi_i, \quad \text{for } i = 1, 2, \ldots, n. \tag{13}$$

With this scheme, constraints of SVMs provide more relaxation and flexibility. The symbol $\gtrsim$ means that we like to permit some constraint violations in the satisfaction of the constraints. The slack variables ξ_i cannot play this role because they are the unknowns of the system and are determined after the training phase of the SVM. For a normal data pattern, this violation is a very small amount. But for a noisy or outlier data, the violation is a greater value and the feasible region is extended for finding better results. Thus, the role of the noises or outliers, in training the optimal hyperplane, is decreased.

Now, the SVM can be reformulated as:

$$\begin{aligned} &\text{Minimize} \quad Q(w, b, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\ &\text{subject to } y_i\left(w^T x_i + b\right) \gtrsim 1 - \xi_i \\ &\xi_i \geq 0, \quad i = 1, \ldots, n \end{aligned} \tag{14}$$

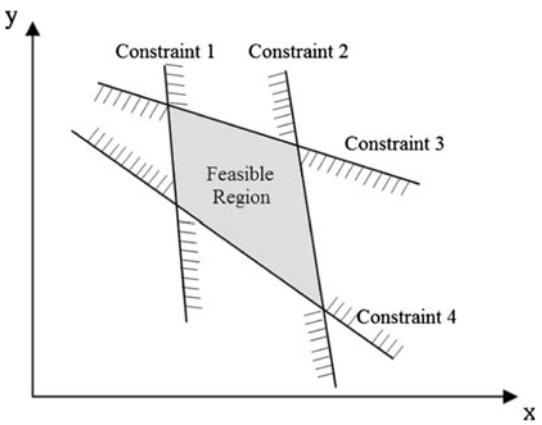

Fig. 3 Feasible region for a typical optimization problem with four constraints

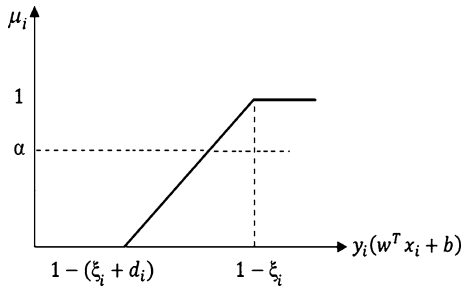

Fig. 4 Membership function μ_i

We call this new structure relaxed constraints support vector machine (RSVM).

Linear membership functions for fuzzy greater than or equal inequality can be defined as follows:

$$\mu_i : \Re^{m+1+n} \to [0, 1), \; i = 1, 2, \ldots, n,$$

$$\mu_i(w, b, \xi) = \begin{cases} 1, & \text{if } \; y_i(w^T x_i + b) \geq 1 - \xi_i \\ \frac{(w^T x_i + b) - 1 + \xi_i + d_i}{d_i}, & \text{if } \; 1 - (\xi_i + d_i) \leq y_i(w^T x_i + b) < 1 - \xi_i \\ 0, & \text{if } \; y_i(w^T x_i + b) < 1 - (\xi_i + d_i) \end{cases} \tag{15}$$

For each constraint i, i = 1,2,…,n, of (14),

$$P_i = \left\{(w, b, \xi) \in \Re^{m+1+n} \middle| y_i\left(w^T x_i + b\right) \gtrsim 1 - \xi_i, \xi_i \geq 0\right\}, \tag{16}$$

Taking $P = \cap_{i \in I} P_i$, where $I = \{1,2,\ldots,n\}$, then (14) can be written as:

$$\text{Minimize}\left\{Q(w, b, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \middle| (w, b, \xi) \in P\right\}. \tag{17}$$

It is clear that $\forall \alpha \in (0, 1]$, an α-cut of the constraint set will be the classical set

$$P(\alpha) = \left\{(w, b, \xi) \in \Re^{m+1+n} \middle| \mu_P(w, b, \xi) \geq \alpha\right\}, \tag{18}$$

where $\mu_P(x) = \inf\{\mu_i(x), i \in I\}$. In this case, $P_i(\alpha)$ will denote an α-cut of the ith constraint (13). The optimal solution of (14) for a given $\forall \alpha \in (0, 1]$ is as follows:

$$\begin{aligned} S(\alpha) &= \left\{(w, b, \xi) \in \Re^{m+1+n} \middle| \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i\right\} \\ &= \left\{\text{Min}\frac{1}{2}\|w'\|^2 + C\sum_{i=1}^{n} \xi_i', (w', b', \xi') \in P(\alpha)\right\} \end{aligned} \tag{19}$$

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

$$P(\alpha) = \bigcap_{i \in I}\left\{(w, b, \xi) \in \Re^{m+1+n} \middle| y_i\left(w^T x_i + b\right) \geq r_i(\alpha), \xi_i \geq 0\right\} \tag{20}$$

with $r_i(\alpha) = 1 - \xi_i - d_i(1-\alpha)$, we have the following problem:

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to } y_i\left(w^T x_i + b\right) \geq 1 - \xi_i - d_i(1-\alpha) \\ &\quad \xi_i \geq 0, \quad i = 1, 2, \ldots, n \end{aligned} \tag{21}$$

In [29], a new support vector machine (v-SVM) is proposed. The main focus of this method is on decreasing the number of SVs. The parameter C is replaced by a parameter $v \in [0, 1]$, which is the lower and upper bound on the number of examples that are support vectors and that lie on the wrong side of the hyperplane, respectively. Therefore, no constant C appears in this formulation; instead, there is a parameter v and also an additional variable ρ to be optimized. The value of v is just an upper bound on the fraction of margin errors and hence also on the fraction of training errors and is a lower bound on the fraction of SVs. The parameters v and ρ are fixed and have not different values for different samples. In RSVM, instead, different d_is are assigned to training samples. For less importance samples (for example, noisy data), more violation from their corresponding constraints is permitted. This means that we assign a larger d_i to outliers and noisy data.

Similar to the conventional SVM, we first convert this constrained problem into the equivalent unconstrained one. Introducing the nonnegative Lagrange multipliers β_i and γ_i, we obtain:

$$\begin{aligned} Q(w, b, \xi, \beta, \gamma) = &\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &- \sum_{i=1}^{n}\beta_i\left\{y_i\left(w^T x_i + b\right) - 1\right. \\ &\left. + \xi_i + d_i(1-\alpha)\right\} - \sum_{i=1}^{n}\gamma_i\xi_i \end{aligned} \tag{22}$$

where $\beta = (\beta_1, \beta_2, \ldots, \beta_n)^T$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)^T$. For the optimal solution, the following Karush–Kuhn–Tucker (KKT) conditions are satisfied:

$$\frac{\partial Q(w, b, \xi, \beta, \gamma)}{\partial w} = 0, \quad \text{i.e., } w = \sum_{i=1}^{n}\beta_i\gamma_i x_i, \tag{23}$$

$$\frac{\partial Q(w, b, \xi, \beta, \gamma)}{\partial b} = 0, \quad \text{i.e., } \sum_{i=1}^{n}\beta_i y_i = 0, \tag{24}$$

$$\frac{\partial Q(w, b, \xi, \beta, \gamma)}{\partial \xi} = 0, \quad \text{i.e., } \beta_i + \gamma_i = C, \tag{25}$$

$$\beta_i\left\{y_i\left(w^T x_i + b\right) - 1 + \xi_i + d_i(1-\alpha)\right\} = 0, \tag{26}$$

$$\gamma_i\xi_i = 0, \tag{27}$$

$$\xi_i \geq 0, \quad \beta_i \geq 0, \quad \gamma_i \geq 0, \tag{28}$$

where $i = 1,2,\ldots,n$.

Thus, substituting (23), (24), and (25) into (22), we obtain the following dual problem. Maximize

$$\begin{aligned} Q(\beta) &= \sum_{i=1}^{n}\beta_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j - \sum_{i=1}^{n}\beta_i d_i(1-\alpha) \\ &= \sum_{i=1}^{n}\beta_i(1 - d_i + d_i\alpha) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j \end{aligned} \tag{29}$$

subject to the constraints:

$$\sum_{i=1}^{n}\beta_i y_i = 0, \quad 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n.$$

The decision function is given by:

$$\begin{aligned} D(x) &= \text{sign}(w^T x + b) \\ &= \text{sign}\left(\sum_{i\in S}\beta_i y_i x_i^T x + b\right), \end{aligned} \tag{30}$$

where S is the set of support vector indices.

For nonlinear separable case, the original input space is mapped into high-dimensional dot-product feature space using a φ-function. Using the kernel function $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) = z_i \cdot z_j$, the dual problem in the feature space is given as follows:

$$\begin{aligned} \text{Maximize } Q(\beta) &= \sum_{i=1}^{n}\beta_i(1 - d_i + d_i\alpha) \\ &\quad - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j z_i z_j \\ &= \sum_{i=1}^{n}\beta_i(1 - d_i + d_i\alpha) \\ &\quad - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j K(x_i, x_j) \end{aligned} \tag{31}$$

subject to the constraints

$$\sum_{i=1}^{n}\beta_i y_i = 0, \quad 0 \leq \beta_i \leq C, \quad i = 1, 2, \ldots, n.$$

The decision function is given by

$$D(x) = \text{sign}\left(\sum_{i\in S}\beta_i y_i K(x_i, x) + b\right), \tag{32}$$

where b is given by

$$b = y_j - \sum_{i\in S}\beta_i y_i K(x_i, x_j) \tag{33}$$

where S is the set of support vector indices, and x_j is unbounded support vector ($0 < \alpha_i < C$).

Constraints of the proposed RSVM have more relaxation than traditional SVMs because of their fuzzy inequalities. In this model, d_i and α are user defined and meaningful

parameters. Each constraint in RSVM formulation (20, 21) is given a specific d_i that acts as a tolerance to the corresponding sample. In fact, by means of this parameter, the feasible region is extended for finding the unknown variables (w, b, ξ). Thus, d_i can be seen as the importance degree of each sample. A greater value for parameter d_i of the sample value x_i means violation from the constraints for this sample is increased. By doing so, the training sample is considered as a noisy data. Therefore, we can control the effect of noisy data or outliers on the classifier with the parameter d_i. Note that slack variables ξ_i are not user defined and are computed during the training phase. Therefore, we cannot control noisy or outlier samples directly or give importance degree to specific samples using ξ_i. If the same d_i is assigned to all constraints, the system can equally tolerate crossing over any sample. On the other hand, if different ds are assigned to different constraints, it means we have assumed a different degree of importance to samples similar to FSVM. Larger d_i causes the corresponding sample x_i to be less important and to be able to consider this data as noise or outlier. It then plays a less important role in determining the separating hyperplane. This fact will be discussed in the next sections.

Also, α is another user-defined parameter in RSVM formulation (20, 21). It is the level at which the membership degree of the fuzzy inequality of constraints, μ_i, is cut. A larger value for α means our certainty in the whole set of data is higher and vice versa. Note that, if we have high certainty in the training samples, we should not permit constraint violations. It means we should use a greater value for the parameter α. It is clear that $(1-\alpha)$ indicates the uncertainty of user in the accuracy of collected samples. In the next sections, we will study the effects of different values of this parameter on the obtained classifier.

This new SVM formulation as nonlinear optimization problem with fuzzy inequality constraints adds useful concepts to conventional SVMs. In the next section, we will combine our RSVM classification model with FSVM in order to be able to produce an SVM that has the ability to consider different importance degree for samples both in the cost-function and in constraints.

4 The proposed fuzzy RSVM

In the previous section, we proposed RSVM that considers importance degree of training samples in constraints of the SVM optimization problem against FSVM that use membership function s_i in the penalty term of the cost-function. In this section, we are going to combine these two methods. Our purposes are the following:

1. Presentation of a general method that considers the task of "giving degree of importance to each training pattern" in both cost-function and constraints of the SVM optimization problem.
2. Equipping FSVM with the new proposed concepts, namely, tolerance and certainty for data.

Thus, we reformulate FSVM with fuzzy inequality in constraints and call this method fuzzy RSVM:

$$\begin{aligned} &\text{Minimize} \quad Q(w,b,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} s_i\xi_i \\ &\text{subject to } y_i\left(w^T x_i + b\right) \gtrsim 1-\xi_i \\ &\quad \xi_i \geq 0, \quad i = 1,\ldots,n \end{aligned} \tag{34}$$

Similar to RSVM, a membership function is defined for fuzzy inequality, and an α-cut is performed to convert fuzzy inequality to a crisp one. So, fuzzy RSVM will be the following:

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} s_i\xi_i \\ &\text{subject to } y_i(w^T x_i + b) \geq 1-\xi_i - d_i(1-\alpha) \\ &\quad \xi_i \geq 0, \quad i = 1,\ldots,n \end{aligned} \tag{35}$$

Again we introduce the nonnegative Lagrange multipliers β_i and γ_i, we obtain from (35):

$$\begin{aligned} Q(w,b,\xi,\beta,\gamma) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} s_i\xi_i \\ &- \sum_{i=1}^{n}\beta_i\{y_i(w^T x_i + b) - 1 + \xi_i + d_i(1-\alpha)\} - \sum_{i=1}^{n}\gamma_i\xi_i \end{aligned} \tag{36}$$

where $\beta = (\beta_1, \beta_2, \ldots, \beta_n)^T$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)^T$.

Applying KKT conditions such as previous section and substituting in (36), we obtain the following dual problem:

$$\begin{aligned} \text{Maximize } Q &= \sum_{i=1}^{n}\beta_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j + C\sum_{i=1}^{n} s_i\xi_i \\ &- \sum_{i=1}^{n}(\beta_i+\gamma_i)\xi_i - \sum_{i=1}^{n}\beta_i d_i(1-\alpha) \\ &= \sum_{i=1}^{n}\beta_i(1-d_i+d_i\alpha) + C\sum_{i=1}^{n}(s_i-1)\xi_i \\ &- \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j \end{aligned} \tag{37}$$

subject to the constraints

$$\sum_{i=1}^{n}\beta_i y_i = 0, \quad 0 \leq \beta_i \leq s_i C, \quad i = 1,2,\ldots,n.$$

The decision function is

$$D(x) = \text{sign}(w^T x + b) = \text{sign}\left(\sum_{i \in S} \beta_i y_i x_i^T x + b\right), \quad (38)$$

where S is the set of support vector indices.

Similar to RSVM, we can extend our fuzzy RSVM to nonlinear separable case as

$$\begin{aligned} \text{Maximize} \quad Q &= \sum_{i=1}^{n} \beta_i (1 - d_i + d_i \alpha) + C \sum_{i=1}^{n} (s_i - 1)\xi_i \\ &\quad - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_i \beta_j y_i y_j z_i \cdot z_j \\ &= \sum_{i=1}^{n} \beta_i (1 - d_i + d_i \alpha) + C \sum_{i=1}^{n} (s_i - 1)\xi_i \\ &\quad - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_i \beta_j y_i y_j K(x_i, x_j) \end{aligned} \quad (39)$$

subject to the constraints

$$\sum_{i=1}^{n} \beta_i y_i = 0, \quad 0 \le \beta_i \le s_i C, \quad i = 1, 2, \ldots, n.$$

The role of the parameters d_i and α is similar to RSVM, as mentioned in previous section. These parameters give more robustness and generality to the FSVM. In the next sections, we consider the effectiveness of our proposed method in practice. We will show the prominence of RSVM and fuzzy RSVM in comparison with SVM and FSVM and emphasize that fuzzy RSVM is a comprehensive method that has the fitness of FSVM and also is equipped with other useful concepts. The extension of our proposed method to solve multi-class classification problems is also presented.

5 Multi-class RSVM

The basic SVM is designed to separate only two classes from each other. However, in many real applications, a method to deal with several classes is required. A solution is to decompose a multi-class problem into several two-class classification problems. At first, we briefly review these methods and then extend the proposed RSVM on these foundations.

5.1 Multi-class SVM classifiers

Two major decomposition implementations are "one-against-all" and "one-against one". The one-against-all [14] method constructs m SVMs where m is the number of classes. The ith SVM is trained to separate the ith class from the remaining classes. Let the ith decision function, with the maximum margin that separates Class i from the remaining classes, be

$$D_i(x) = w_i^T \varphi(x) + b_i, \quad (40)$$

where w_i is the l-dimensional vector, $\varphi(x)$ is the mapping function that maps x into the l-dimensional feature space, and b_i is the bias term. In classification, if for the input vector x, if there is only one i for which $D_i(x) > 0$, x is classified into Class i. Since only the sign of the decision function is used, the decision is of a discrete type as opposed to a continuous decision. If $D_i(x) > 0$ is satisfied for more than one i or there is no i for which $D_i(x) > 0$, x is unclassifiable. To avoid this, instead of discrete decision functions, continuous decision functions are proposed for the classification. In the continuous case, datum x is classified into the class

$$\arg \max_{i=1,2,\ldots,m} D_i(x). \quad (41)$$

The one-against-one [30] (pairwise SVMs) instead, constructs $m(m-1)/2$ decision functions for all the combinations of class pairs. In determination of a decision function for a class pair, we use the training data for the corresponding two classes. Thus, in each training session, the number of the training data is reduced considerably compared to one-against-all support vector machines, which use all the training data. Let the decision function for Class i against Class j, with the maximum margin, be

$$D_{ij}(x) = w_{ij}^T \varphi(x) + b_{ij}. \quad (42)$$

The regions

$$R_i = \{x | D_{ij}(x) > 0, \quad j = 1, 2, \ldots, m, \quad j \ne i\}. \quad (43)$$

do not overlap thus if x is in R_i, we classify x into Class i. If x is not in $R_i (i = 1,2,\ldots,m)$ for any i, x is classified by voting. In this case, for the input vector x, $D_i(x)$ calculates at follow:

$$D_i(x) = \sum_{i \ne j, j=1}^{n} \text{sign}(D_{ij}(x)), \quad (44)$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{for } x \ge 0, \\ -1 & \text{for } x < 0, \end{cases} \quad (45)$$

and x is classified into class:

$$\arg \max_{i=1,2,\ldots,m} D_i(x). \quad (46)$$

If $x \in R_i$, $D_i(x) = m - 1$ and $D_k(x) < m - 1$ for $k \ne i$. Thus, x is classified into Class i. But if any of $D_i(x)$ is not $m - 1$, (16) may be satisfied for plural is. In this case, x is unclassifiable. To resolve this problem, Vapnik [31] proposed to use continuous decision functions. To do so, a datum is classified into the class with maximum value of the decision functions.

Another popular solution is directed acyclic graph support vector machines (DAG SVM) [32] that uses a decision tree in the testing stage. Training of a DAG is the same as conventional pairwise SVMs. Classification by the original DAG is executed by list processing. First, we generate a list with class numbers as elements. Then, we calculate the value of the decision function for the input x. Let the two classes for which the classification decision is performed be i and j. If $D_{ij}(x) > 0$, we delete the element j from the list. We repeat the procedure until one element is left. Then we classify x into the class that corresponds to the element number. Figure 5 shows the decision tree for the three classes. In the figure, $\bar{i}$ shows that x does not belong to Class i. As the top-level classification, we can choose any pair of classes. Except for the leaf node, if $D_{ij}(x) \geq 0$, we consider that x does not belong to Class j, and if $D_{ij}(x) < 0$, it does not belong to Class i. Thus, if $D_{12}(x) > 0$, x does not belong to class II. Therefore, it belongs to either class I or class III, and the next classification pair is classes I and III.

There are some researches that have refined these classification methods. For example, in [33], a weighted multi-class classification technique divides the input space into several subspaces. In the training phase of the technique, for each subspace, a DAG SVM is trained and its probability density function (pdf) is guesstimated. In the test phase, fit in value of each input pattern to every subspace is calculated using the pdf of the subspace as the weight of each DAG SVM. Finally, a fusion operation is defined and applied to the DAG SVM outputs to decide the class label of the given input pattern.

5.2 RSVM for multi-class classification

We can use RSVM for both binary and multi-class classification problems. To do this, modifications should be applied to one-against-all, pairwise, and DAG SVM classifiers. In the one-against-all RSVM, we train m RSVM, where m is the number of classes. RSVM$_i$ separates Class i from the remaining classes. A testing sample x_t is assigned to the class with the maximum decision function value. Figure 6 shows the details of this method. Note that D_i is the value of ith decision function.

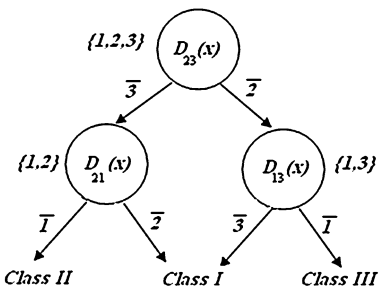

Fig. 5 Classification by DAG SVM for three classes

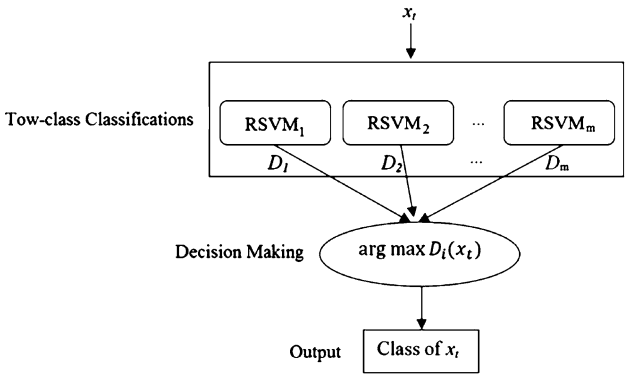

Fig. 6 Classification by one-against-all RSVM

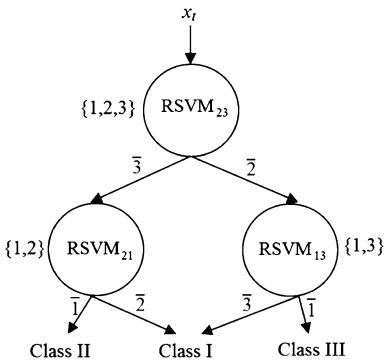

Fig. 7 Classification by DAG RSVM

In the pairwise RSVM and DAG RSVM, $m(m-1)/2$ RSVMs are trained. RSVM$_{ij}$ is the optimal separating hyperplane (OSH) between Class i and class j. In the pairwise RSVM, a testing sample x_t is assigned to class with maximum decision function represented by the (44). The DAG RSVM uses a decision tree in the testing stage. Figure 7 shows the decision tree for the case where there are three classes.

In the next section, we will evaluate our proposed methods using different case studies.

6 Case studies for effectiveness of the RSVM

In this section, we evaluate the proposed methods using different case studies. All methods for the comparison are implemented using MATLAB R2008a [34] running on a computer with an Intel processor (Core 2 Duo, 2.50 GHz) and 4 GB RAM.

6.1 Two-class problems

The effectiveness of the proposed RSVM is evaluated using different examples. The experiments are performed for two-class classification problems.

6.1.1 Example 1: Two classes with different importance

There are many real-world applications for which we would like to classify one of the classes with high precision. An intrusion detection system is an instance of these applications. It is designed to reliably separate attack patterns from normal ones. It is very important for such a system that an attack pattern is not classified into the normal class. However, it is not fatal to classify a normal pattern the attack class. In other words, the system must not detect an attack sample as a normal one but it can detect a normal sample as an attack. In fact, the importance of attack class is more than the normal class.

One of the main abilities of the FSVM is that it can assign different emphasis to each class. To realize this constraint, it is sufficient to assume a higher (close to 1) fuzzy membership for the premier class and a lower membership for the other one. This ability is not supported in the SVM model. This ability also exists in our proposed RSVM and fuzzy RSVM. As we illustrated in pervious sections, d_i can be used in order to give importance degree to each sample. Figure 8 shows the results of FSVM, RSVM, and fuzzy RSVM for a dataset that is created manually. All points x_i with $y_i = 1$ are indicated using a cross symbol, and all points x_i with $y_i = -1$ are indicated using a square symbol. We gave higher importance to class 1 (is indicated using a cross symbol). In FSVM, we set $s_i = 1$ (high importance degree) for class 1 and $s_i = 0.1$ (low importance degree) for class 2. In RSVM and fuzzy RSVM, we set $d_i = 0.1$ (high importance degree) for class 1, $d_i = 1$ (low importance degree) for class 2, and $\alpha = 0.8$. Note that, in RSVM, a lower d_i means a higher importance degree. Also, we used rbf kernel function with $\sigma = 1$.

As shown in Fig. 8, even though the most attention is paid to Class 1 ($s_i = 1$), the pointed sample is classified incorrectly by the FSVM. Although RSVM gets better result in comparison with SVM, it has a higher training error. Instead, fuzzy RSVM produces a precise and accurate result.

6.1.2 Example 2: Study of tolerance

One of the distinctive properties of the RSVM, in comparison with other known classifiers, is that it considers tolerance for the data. Suppose that we are going to give some degree of flexibility to the classifier. It means that the margin can be larger. It may be said that the parameter C in the SVM formulation (1) plays this role but increasing (decreasing) the parameter C higher (lower) than some value will have a small effect on the margin and the obtained classifier. One reason for this is that in the optimization problem (1) the goal is maximization of the margin (i.e., *Minimize* $1/2||w||^2$) and minimization of

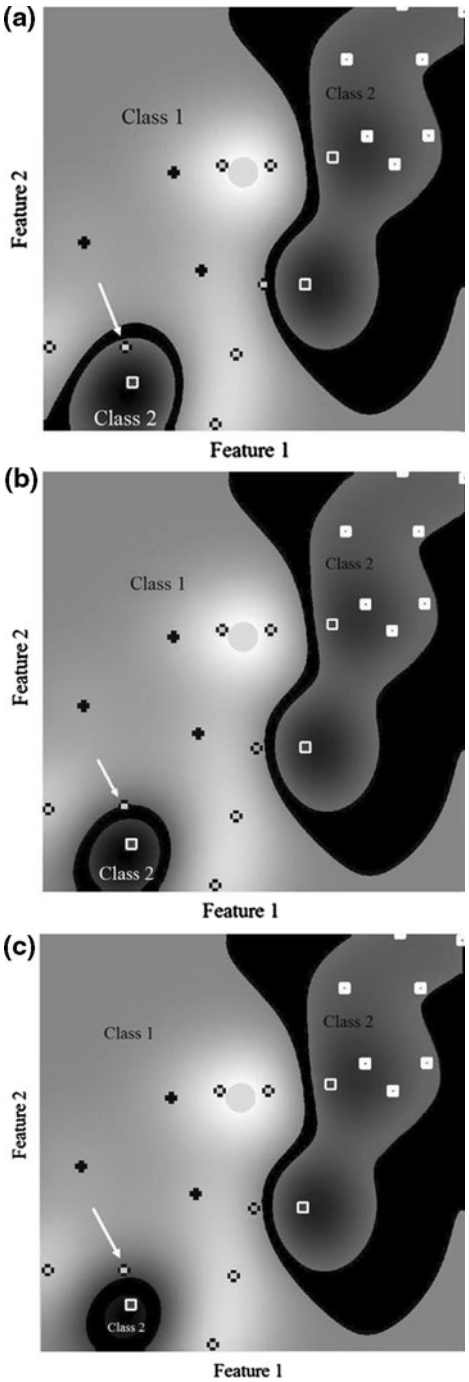

Fig. 8 Results of FSVM (**a**), RSVM (**b**), and fuzzy RSVM (**c**) in classification of two classes with different weight

overall errors (i.e., *Minimize* $C\sum_{i=1}^{n} \xi_i$), simultaneously, and there is a trade off. But in ther RSVM and its extension (fuzzy RSVM), we can assign a higher flexibility degree to the model. In fact, by applying changes to the constraints of SVM formulation, the feasible region (Fig. 3) changes. Note that each constraint poses a limitation on the system, and the overall constraints make a feasible region. Also, the slack variables ξ_i in the SVM classifier cannot be seen as

tolerance values because they are not user defined and are determined during solving the optimization problem.

Since the OSH obtained by SVM depends on only support vectors, it is very sensitive to noises and outliers in the training set. As noted in [17], the outliers tend to support vectors with large Lagrangian coefficients. By assigning tolerance to the training samples, as shown in Fig. 9, more samples become support vectors. Thus, the classifier is built using more samples. It is reasonable that the large part of the training samples is not affected by noise, or we can assume that although all samples are affected by noise with zero mean.

Figure 9 shows the effect of tolerance in the training samples and its role in determination of the SVM classifier. If we set all d_i to zero, the RSVM formulation (20, 21) is converted to the standard SVM formulation (1). Giving $d_i > 0$ increases the tolerance of the training samples and the margin will grow.

As shown in Fig. 9, the SVM classifier and the margin value are different for each value of d_i value. Note that here we give a tolerance to the entire training samples (i.e., all of d_is are identical and are equal to d). With the increase in the tolerance value, more samples become support vectors and contribute to finding the classifier.

6.1.3 *Example 3: Study of the certainty in the training samples*

As mentioned before, a typical data mining application consists of four major steps: data collection and preparation, data transformation and quality enhancement, pattern discovery, and interpretation and evaluation of patterns. In the Cross Industry Standard Process for Data Mining framework [35], this process is decomposed into six major phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. It is expected that the whole process starts with raw data and finishes with the extracted knowledge. Because of its data-driven nature, previous research efforts have concluded that data mining results crucially rely on the quality of the underlying data. For the most applications of data mining, the process of data collection, data preparation, and data enhancement cost the majority of the project budget and also the developing time circle [1].

Real-world data mining deals with noisy information sources where data collection inaccuracy, device limitations, data transmission and discretization errors, or man-made perturbations frequently result in imprecise or vague data. Therefore, each training dataset has a degree of uncertainty, in essence.

In RSVM, there is a parameter that handles the uncertainty of collected data. As we explained in the previous sections, we introduced a membership function μ_i (15) for the fuzzy inequality in the RSVM formulation (20, 21) and then apply α-cut to convert the fuzzy inequality to a crisp one. As shown in Fig. 4, whatever this cutting is closer to 1, it means that we have more confidence in our data and do

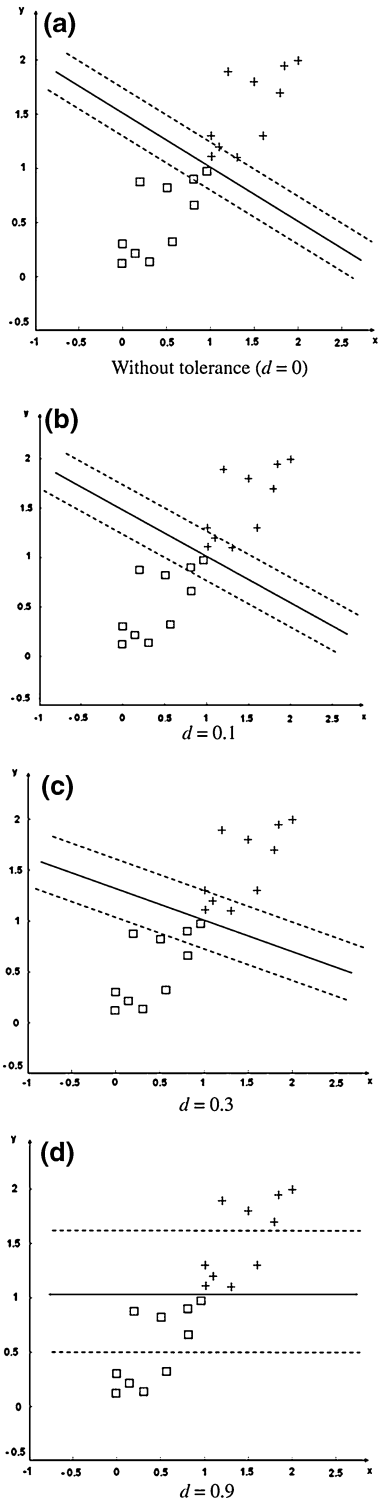

Fig. 9 Study of the tolerance in RSVM ($\alpha = 0.7$). **a** Without tolerance ($d = 0$), **b** $d = 0.1$, **c** $d = 0.3$, **d** $d = 0.9$

not permit violations in the accomplishment of the constraints. Therefore, α ($0 < \alpha \leq 1$) indicates the degree of certainty, and $(1 - \alpha)$ is our uncertainty about data. Plus, if we have no prior knowledge about our data, we can test the RSVM on our dataset with different α values and determine a degree of certainty for it. This matter will be discussed in the next example.

Now, we bring a simple test to illustrate the concept of certainty for training samples. Figure 10 shows the results of RSVM with different α values for a simple training data. When we choose a larger value for the certainty factor,α, the classifier is trained with more cautious. In fact, we trust our data with high certainty. Therefore, a classifier is trained with high precision.

As shown in Fig. 10, choosing a smaller value of α causes higher total soft error and provides a larger margin. The margin size m is calculated as:

$$m = 2/\|w\|^2$$

The margin sizes for this experiment are 0.0564 (Fig. 10a), 0.0587 (Fig. 10b), and 0.0599 (Fig. 10c).

6.1.4 Example 4: Experiment with real datasets

We evaluated our proposed methods by applying them to BUPA Liver Disorders, Statlog (Heart), and Haberman datasets. They were obtained from the UCI Repository of Machine Learning databases and domain theories [20]. All of them are two-class classification problems.

BUPA Liver Disorders dataset; liver is an effective organ in neutralizing toxics and throwing them from the body. If the amount of toxics reaches a level exceeding working capacity of the organ, the cells of related parts in organ are destroyed. Then, some substances and enzymes are appeared and interfere in blood. During diagnosis of the disease, the levels of these enzymes are analyzed. Because of the fact that effects of different alcohol dosages vary from one person to the other as well as the fact that there are many enzymes, there can be frequent possible errors in diagnosis [21]. BUPA Liver Disorders dataset is prepared by BUPA medical research company. It includes 345 samples with 6 attributes. The first five features for each sample are obtained from blood tests. The last feature is daily alcohol consumption. We selected 200 instances for training and 145 instances for testing in our experiment. Also, we used polynomial kernel function ($d = 3$) for training the classifier:

$$K(x, x') = (x^T x' + 1)^d. \tag{47}$$

The obtained results are summarized in Table 1. Note that α is considered for RSVM and fuzzy RSVM. Note that, for RSVM, we need a subsystem to determine d_i. We used

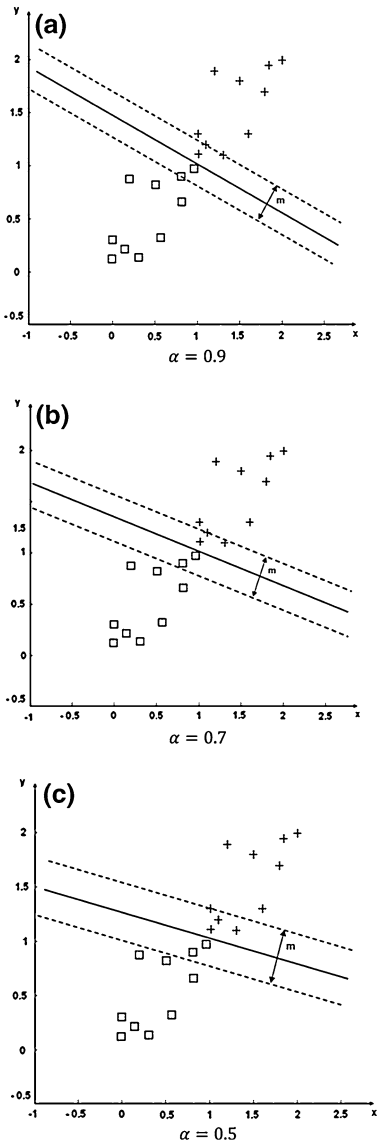

Fig. 10 Study of the certainty factor α in RSVM. **a** $\alpha = 0.9$, **b** $\alpha = 0.7$, **c** $\alpha = 0.5$

circle method [36], which is a geometric-based model for giving importance degree to each sample. It has led to good results for this purpose.

Statlog (Heart) dataset; it is a dataset for the recognition of absence (class 1) or presence (class 2) of heart disease in 270 observations. There are thirteen attributes for this dataset. We selected 180 random instances for training and 90 instances for testing. The kernel function for this experiment is radial basis function (RBF) kernel with $\sigma = 1$:

$$K(x, x') = \exp\left(-1/2\|x - x'\|^2 \Big/ 2\sigma^2\right). \tag{48}$$

Also, we set the parameter C to 100 for this experiment. The obtained results are summarized in Table 2.

Table 1 The recognition rate of SVM, FSVM, RSVM, and fuzzy RSVM on BUPA liver disorders dataset

	SVM	FSVM	RSVM	Fuzzy RSVM
$C = 100,\ \alpha = 0.9$	65.5556	68.8889	71.1111	**72.222**
$C = 1{,}000,\ \alpha = 0.9$	63.3333	68.8889	71.1111	**72.222**

The bold values represent the maximum value among different methods

Table 2 The recognition rate of SVM, FSVM, RSVM, and fuzzy RSVM on Statlog dataset

	SVM 81.1111 RSVM	FSVM 80 Fuzzy RSVM
$\alpha = 0.9$	82.2222	84.4444
$\alpha = 0.8$	84.4444	**86.6667**
$\alpha = 0.7$	82.2222	83.3333
$\alpha = 0.6$	78.8889	81.1111
$\alpha = 0.5$	78.8889	78.8889
$\alpha = 0.4$	67.7778	70
$\alpha = 0.3$	73.3333	74.4444
$\alpha = 0.2$	72.2222	73.3333
$\alpha = 0.1$	74.4444	75.5556

The bold values represent the maximum value among different methods

Table 3 The recognition rate of SVM, FSVM, RSVM, and fuzzy RSVM on Haberman dataset

Method	Linear kernel	Polynomial kernel ($d = 3$)	RBF kernel ($\sigma = 1$)
SVM	58.4906	70.7547	66.0378
FSVM	61.3208	**74.5283**	69.8132
RSVM	62.2642	71.6981	69.8132
Fuzzy RSVM	**64.1509**	**74.5283**	**72.6415**

The bold values represent the maximum value among different methods

Haberman dataset; it contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. In the Haberman dataset, there were two classes: patients who survived for up to 5 years and patients who died before that [37]. It includes 306 samples with three attributes. We selected 200 random samples for training and the rest of them for testing and performed this experiment with different kernel functions. Table 3 shows the results of this experiment.

Experimental results in Tables 1, 2, and 3 show that fuzzy RSVM outperforms other methods. We can run our method with different values of α and choose the best one with maximum recognition rate. As we mentioned in previous subsection, in this approach, we can assign a degree of certainty to underlying dataset. Indeed the value of α with maximum recognition rate indicates how much the process of data collection is reliable. Figure 11 shows the recognition rates of fuzzy RSVM with different values of α for Statlog dataset with respect to Table 2. We can say certainty of this dataset is 80%. So, in the test procedure of fuzzy RSVM for this dataset, α is set to 0.8.

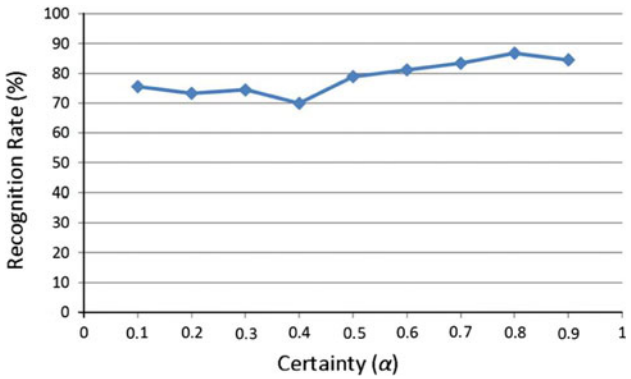

Fig. 11 The recognition rate of fuzzy RSVM with different values of α

Table 4 Details of datasets in experiments

Dataset	Number of classes	Number of attributes	Number of instances
Glass identification	6	10	214
Page blocks	5	10	5,473
Image segmentation	7	19	2,310
Statlog (Shttle)	7	9	58,000

Table 5 One-against-all SVM versus One-against-all RSVM recognition rates (%)

Dataset	One-against-all SVM	One-against-all RSVM
Glass identification	66.33	71.43
Page blocks	80.45	84.56
Image segmentation	70.62	77.12
Statlog (Shttle)	69.16	79.16

As it was illustrated in the previous examples, we increased the ability of the standard SVM and also fuzzy SVM with additional parameters. On the one hand, we introduced the concept of tolerance for training samples and, on the other hand, we added the concept of certainty for training samples and datasets to SVM. Therefore, we recommend RSVM and fuzzy RSVM to the researchers who work on datasets with noisy or low degree of certainty samples.

Table 6 Pairwise SVM versus pairwise RSVM recognition rates (%)

Dataset	Pairwise SVM	Pairwise RSVM
Glass identification	63.33	65
Page blocks	80.67	84.29
Image segmentation	68.04	73.04
Statlog (Shttle)	72.04	78.38

Table 7 DAG SVM versus DAG RSVM recognition rates (%)

Dataset	DAG SVM	DAG RSVM
Glass identification	66.67	68.33
Page blocks	85.06	90.33
Image segmentation	71.43	78.82
Statlog (Shttle)	65.18	75.70

6.2 Multi-class problems

In the Sect. 5, we have presented the RSVM classifier for multi-class classification problems. The evaluation of the proposed method is performed using real-world datasets. All datasets used in the following tests are obtained from the UCI Repository of Machine Learning Databases and Domain Theories [37]. Details of these datasets are summarized in Table 4.

We applied one-against-all SVM and one-against-all RSVM to datasets listed in Table 4. The results are summarized in Table 5. The one-against-all RSVM technique is similar to the one-against-all SVM with the difference that all of decision functions are trained using our proposed RSVM. In the same way, we compared the pairwise SVM to the pairwise RSVM and also the DAG SVM to the DAG RSVM and summarized the results in Tables 6 and 7, respectively. In these experiments, RBF kernel function (48) is used, with $C = 100$, and α (in the RSVM) being equal to 0.9. It is worth mentioning that 70% of the data in the dataset were randomly selected for the training phase and the rest left for the testing phase.

As shown in all of the experiments above, when RSVM is used for different multi-class SVM classifiers, better results are achieved. In some cases, we had up to 10% improvement.

7 Conclusion and future works

The FSVM is so sensitive to changes that a minor modification in the cost-function may lead to inaccurate classifier. Thus, effectiveness of the FSVM is influenced by choosing fuzzy membership values s_i. In this paper, we proposed a new model of support vector machines with emphasis on constraints of the optimization problem of the SVM formulation and named it RSVM. The constraints of RSVM have more relaxation and flexibility because of their fuzzy inequalities. Then we introduce a new method for SVM classification problems with a combination of RSVM and FSVM (fuzzy RSVM). In the new model, constraints of the standard SVM are converted to fuzzy inequalities. Solving SVM with fuzzy constraints leads to forming RSVM optimization problem that is equipped with new concepts that are not considered up until now. The experiments showed the superiority of the proposed methods in different case studies including two-class and multi-class classification problems. The superiority of these methods is in the following areas:

1. Handling data with tolerance. SVM is one of the most popular methods for patterns classification problems. The decision function is the solution of the optimization problem in which minimization of the total error and maximization of the margin are considered, simultaneously. In cases where there is a tendency toward having samples with tolerance, RSVM is the solution.
2. Handling uncertainty in data. If we have a prior knowledge about a dataset, and are aware of its accuracy and precision, there are no potentialities in SVM or FSVM to handle this knowledge. RSVM is the way to deal with. It is also capable to find certainty for the given datasets.
3. Better performance. The experiments we have performed showed that the quality of the RSVM classification on real data is higher than both that of SVM and FSVM.

For the future works, we will extend our proposed method to solve regression problems. So, we will propose relaxed constraints support vector regression. Also, support vector data description (SVDD) is another extension of SVMs that provides an effective tool for one-class classification problems. We will extend our method to find better results in this domain.

References

1. Wu X, Zhu X (2008) Mining with noise knowledge: error-aware data mining. IEEE Trans Syst Man Cybern Part A Syst Hum 38(4):917–932
2. Luebbers D, Grimmer U, Jarke M (2003) Systematic development of data mining-based data quality tools. In: Proceedings of 29th VLDB
3. Zhu X, Wu X (2004) Class noise vs. attribute noise: a quantitative study of their impacts. Artif Intell Rev 22(3/4):177–210

4. Quinlan JR (1986) The effect of noise on concept learning. In: Michalski RS, Carbonell JG, Mitchell TM (eds) Machine learning. Morgan Kaufmann, San Mateo
5. Quinlan JR (1983) Learning from noisy data. In: Proceedings of 2nd international machine learning workshop, Urbana-Champaign
6. Zhu X, Wu X, Chen Q (2003) Eliminating class noise in large datasets. In: Proceedings of ICML, pp 920–927
7. Gamberger D, Lavrac N, Groselj C (1999) Experiments with noise filtering in a medical domain. In: Proceedings of 16th ICML conference, San Francisco, pp 143–151
8. John GH (1995) Robust decision trees: removing outliers from databases. In: Proceedings of 1st international conference on knowledge discovery data mining, pp 174–179
9. Kubica J, Moore A (2003) A probabilistic noise identification and data cleaning. In: Proceedings of ICDM conference, Melbourne
10. Lauer M (2001) A mixture approach to novelty detection using training data with outliers. Lect Notes Comput Sci 2167:300–316
11. Parzen E (1962) On estimation of a probability density function and mode. Ann Math Stat 33:1065–1076
12. Ridder DD, Tax DMJ, Duin RPW (1998) An experimental comparison of one-class classification methods. In: Proceedings of fourth annual conference of the Advanced School for Computing and Imaging. ASCI
13. Tax DMJ One-class classification. Ph.D. Thesis, Delft University of Technology. Available from http://www.ph.tn.tudelft.nl/~davidt/papers.html
14. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
15. Zhang X (1999) Using class-center vectors to build support vector machines. In: Proceedings of the 1999 IEEE signal processing society workshop, pp 3–11
16. Herbrich R, Weston J (1999) Adaptive margin support vector machines for classification. In: Ninth international conference on artificial neural networks, vol 2, pp 880–885
17. Boser B, Guyon I, Vapnik V (1992) A training algorithm for optimal margin classifier. In: Presented at the 5th annual workshop on computational learning theory. ACM Press, Pittsburgh
18. Graf A, Smola A, Borer S (2003) Classification in normalized feature space using support vector machines. IEEE Trans Neural Netw 14:597–605
19. Lin CF, Wang SD (2002) Fuzzy support vector machine. IEEE Trans Neural Netw 13:464–471
20. Lin CF, Wang SD (2004) Training algorithms for fuzzy support vector machines with noisy data. Pattern Recogn Lett 25:1647–1656
21. Wang Y, Wang S, Lai KK (2005) A new fuzzy support vector machine to evaluate credit risk. IEEE Trans Fuzzy Systems 13:820–831
22. Hongle D, Shaohua T, Qingfang Z (2009) Intrusion detection based on Fuzzy support vector machines. In: International conference on networks security, wireless communications and trusted computing, pp 639–642
23. Liu YH, Chen YT (2007) Face recognition using total margin-based adaptive fuzzy support vector machines. IEEE Trans Neural Netw 18:178–192
24. Wang TY, Chiang HM (2007) Fuzzy support vector machine for multi-class text categorization. Inform Process Manag 43:914–929
25. Scholkopf B (1997) Support vector learning. Ph.D. dissertation, Technische Universitat Berlin, Germany
26. Burges CJC (1996) Simplified support vector decision rules. In: Proceedings of the 13th international conference on machine learning
27. Jiang X, Yi Z, Lu JC (2006) Fuzzy SVM with new fuzzy membership function. Neural Comput Appl 15:268–276
28. Tang H, Qu LS (2008) Fuzzy support vector machine with a new fuzzy membership function for pattern classification. In: Proceedings of the seventh international conference on machine learning and cybernetics, pp 768–773
29. Schölkopf B, Smola AJ, Williamson RC, Bartlett PL (2000) New support vector algorithms. Neural Comput 12:1207–1245
30. Hastie T, Tibshirani T (1998) Classification by pairwise coupling. Ann Sat 26:451–471
31. Vapnik V (1998) Statistical learning theory. Wiley, New York
32. Platt JC, Cristianini N, Shawe-Taylor J (2000) Large margin DAGs for multiclass classification. In: Advances in neural processing systems, vol 12. MIT press, Cambridge, pp 547–553
33. Sabzekar M, GhasemiGol M, Naghibzadeh M, Sadoghi Yazdi H (2009) Improved DAG SVM: a new method for multi-class SVM classification. In: Proceedings of the 2009 international conference on artificial intelligence (ICAI'09), vol 2, pp 548–553, Las Vegas
34. http://www.mathworks.com
35. Shearer C (2000) The CRISP-DM model: the new blueprint for data mining. J Data Wareh 5(4):13–22
36. Chu L, Wu C (2004) A fuzzy support vector machine based on geometric model. In: Proceedings of the fifth world congress on intelligent control and automation, Hangzhou, PR China, pp 1843–1846, June 15–19
37. Murphy PM, Aha KW (1994) UCI Repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html University of California, Department of Information and Computer Science, Irvine