Emphatic Constraints Support Vector Machine

Mostafa Sabzekar, Hadi Sadoghi Yazdi, Mahmoud Naghibzadeh, and Sohrab Effati

***Abstract*— In this paper, a new support vector machine, ESVM, with more emphasis on constraints is presented. The constraints are fuzzy inequalities. With this scheme, two problems are solved: training samples with some degree of uncertainty, and samples with tolerance. Also, the fuzzy SVM (FSVM) model is modified with emphasis constraints. The new model is called fuzzy ESVM (FESVM), in this paper. With this scheme we will able to consider importance degree for samples both in the cost function and constraints, simultaneously. Necessary experiments are performed and the results show the superiority of the proposed methods. ESVM and fuzzy ESVM are strongly recommended to the researchers who work on data sets with noisy or low degree of certainty samples.**

***Index Terms*—Certainty, emphatic constraints, support vector machine, fuzzy inequality, tolerance in data.**

I. Introduction

Support vector machines (SVMs) are one of the most powerful methods that deliver state-of-the-art performance in real world pattern recognition and data mining applications. In the SVM solution, a pattern recognition problem is converted to a constraint quadratic programming. Support vector machine is originally introduced by Vapnik [1] within the area of statistical learning theory. It is bases on structural risk minimization (SRM) principle and finds a classifier with minimized Vapnik-Chervonenkis (VC) dimension. In pattern recognition, SVMs have been developed for data classification, feature reduction, and function estimation [2,3].

The SVM classifier is very sensitive to outliers and noisy samples since the penalty term of SVM treats every data point equally in the training process. Increasing the values of slack variables, help in reducing the effect of noisy support vectors. Without introducing these slack variables, in the presence of noisy data, SVM may not be able to determine a hyperplane between two classes. The generalized formulation of the SVM model which can tolerate noisy data close to separating hyperplane is as follows:

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

In this model, $X = \{(x_i, y_i)\}_{i=1}^{n}$ is a set of n training samples, where $x_i \in \mathcal{R}^m$ is an m-dimensional sample in the input space, and $y_i \in \{-1,1\}$ is the class label of x_i. SVM finds the optimal separating hyperplane (OSH) with the minimal classification errors. Let w_0 and b_0 denote the optimum values of the weight vector and bias, respectively. The hyperplane can be represented as: $w_0^T x + b_0 = 0$, where w is the normal vector of the hyperplane, and b is the bias which is a scalar.

Inputs to the SVM system are the training data and the constant C. The system will calculate proper slack variables $\xi_i, i = 1, \dots, n$, and will determine the separating hyperplane. ξ_i is the training error corresponding to data sample x_i. The parameter C in the SVM formulation (1) controls the misclassification error. 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. Therefore, by properly choosing a value for C, we can suppress outliers. In the SVM model we do not have the freedom of adjusting the slack variables ξ_i on each individual sample. This is considered by Lin and Wang [4,5]. The new model, called fuzzy SVM (FSVM), is shown in (2).

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

In this formulation, the error term ξ_i is scaled by $0 < s_i \leq 1$, which is called fuzzy membership value corresponding to the training sample x_i. The fuzzy membership values are used to weight the soft penalty term in the cost function of FSVM. The weighted soft penalty term reflects the relative fidelity of the training samples during training phases. Important samples with large membership value will have greater effect on the FSVM training procedure and hence greater effect on determination of the final classifier hyperplane. But some problems are seen, yet:

1) Although FSVM offers high flexibility on every sample in the cost functions, the lower bound of the constraints are of crisp type. By relaxing this lower bound we may be able to produce better results.
2) Data tolerance has not been mentioned in the SVMs while in many applications we obtain prescience (prior knowledge) about tolerance of data.
3) Measurement, noise processing, environmental conditions, deficiency in feature extraction and selection,

Mostafa Sabzekar, M.S. Student in Computer Engineering, Ferdowsi University of Mashhad, Iran. (Phone: +98-511-6084482; e-mail: sabzekar@wali.um.ac.ir).

Hadi Sadoghi Yazdi is with Computer Engineering Department, Ferdowsi University of Mashhad, Iran (e-mail: sadoghi@sttu.ac.ir).

Mahmoud Naghibzadeh is with Computer Engineering Department, Ferdowsi University of Mashhad, Iran (e-mail: naghibzadeh@um.ac.ir).

Sohrab Effati is with Applied Mathematics Department, Ferdowsi University of Mashhad, Iran (e-mail: s-effati@um.ac.ir).

can cause uncertainty in samples. Data uncertainty, also, has not been considered in SVMs.

The mentioned deficiencies are the motivation to do new research on SVM classifier. Using fuzzy inequalities in constraints give us the freedom to have noisy and uncertain samples amongst our training data. We have also considered the case of nonseparable data in our proposed SVM classifier. The system is examined on various synthetic as well as real data sets. We found the results to be remarkable. In the rest of this section we briefly review various ways for handling noisy samples in the SVM method in the literature.

A. *Some notes on noisy samples in the SVM*

Tipping [6] proposed the relevance vector machine (RVM) that the target (output) is independent and contaminated with mean-zero Gaussian noise with variance. Sun et al. [7] assume a noisy output in the least square SVM classifier and resolve noise reduction with function approximation techniques. Chen [8] assumes additive noise in the input signal for regression application of SVM and studies it in frequency domain. Zhang et al. [9] studies additive impulse and Gaussian noise in the input signal for filtering using SVM.

One of the reasons for presentation of FSVM [4,5] is decreasing the effect of noisy samples. In fact, because the noisy and error of measurement, the training examples are usually uncertain or fuzzy. Choosing an appropriate fuzzy membership for a given problem is very important for FSVM. Different membership functions have different effects on the algorithm. There are many methods to build fuzzy membership functions, but there is not a general rule to follow at present. We must choose an appropriate one according to the practical problem. In [10] authors present two new methods for calculation of membership function of s_i based on geometry distribution of the training samples. Those samples near to the optimal hyperplane have similar geometry property. The main idea of FSVM is that if an input is detected as an outlier or noisy sample, its membership function decreases so total error term decrease. In [11] another method for s_i of the FSVM algorithm is presented which follows the same idea that an input is assigned a low membership of the class if it is detected as an outlier. However, method of [11] treats each input as an input of the opposite class with higher membership and it makes full use of the data and achieves better generalization ability. Also, in two different works [12,13], authors try to determine membership function in multi-category data classification.

In Ji et al. [14] studied support vector machine with fuzzy training data and chance constrains which is modeled as:

$$\begin{aligned} &Minimize\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &subject\ to\quad Pos\{y_i\left(w^T\tilde{X}_i + b\right) + \xi_i \geq 1\} \geq \lambda \qquad (3) \\ &\xi_i \geq 0, \qquad i = 1, \dots, n \end{aligned}$$

They showed that $Pos\{\tilde{a} \leq 0\} \geq \lambda$ with triangular fuzzy number $\tilde{a} = (r_1, r_2, r_3)$ and for any given level $\lambda(0 < \lambda \leq 1)$ is equivalent to: $(1-\lambda)r_1 + r_2 \leq 0$. Thereupon, constraints of (3) are simplified and $\tilde{X}_i$ is a fuzzy input.

In our previous work [15], we applied probabilistic constraints for reducing noisy samples in maximization of margin. A Constraint is in the form of $Pr\{y_i(w^T x_i + b) \geq u_i\} \geq \delta_i$ where u_i is an independent random variable with known distribution functions and $0 \leq \delta_i \leq 1$ is the value of effect of *i*-th samples in fixation of the optimal hyperplane.

As we see, noisy samples have been discussed by very few researchers, in the SVM. Here, we will present an improved concept of noisy data which is a form of tolerance and uncertainty in the input samples.

The rest of this paper is organized as follows. The SVM is briefly introduced in Section 2 from viewpoints of noisy samples. The structure of the proposed ESVM is given in Section 3. Section 4 is devoted to derivation of fuzzy ESVM. The effectiveness of the proposed methods is illustrated by examples in Section 5. The conclusion is given in Section 6.

II. Support Vector Machine Briefly From Viewpoints Of Noisy Samples

A. *Support vector machine formulation*

Let $X = \{(x_i, y_i)\}_{i=1}^{n}$ be a set of *n* training samples, where $x_i \in \mathcal{R}^m$ is an *m*-dimensional sample in the input space, and $y_i \in \{-1,1\}$ is the class label of x_i. The SVM finds the optimal separating hyperplane (OSH) with the minimal classification errors. The linear separation hyperplane is to form of

$$f(x) = w^T x + b$$

where *w* and *b* are the weight vector and bias, respectively. The optimal hyperplane can be obtained by solving the optimization problem (1), where ξ_i is slack variable for obtaining a soft margin. But *C* controls the effect of slack variables and margin increases by decreasing the value of *C*.

In a support vector machine the optimal hyperplane is obtained while maximizing the generalization ability. But if the training data are not linearly separable, the obtained classifier may not have high generalization ability although the hyperplanes are determined optimally. Thus to enhance linear separability, the original input space is mapped into a high-dimensional dot-product space called the feature space. Now using the nonlinear vector function $\varphi(x) = (\varphi_1(x), \dots, \varphi_l(x))^T$ that maps the *m*-dimensional input vector *x* into the *l*-dimensional feature space, the OSH in the feature space is given by

$$f(x) = w^T\varphi(x) + b,$$

The decision function for a test data is:

$$D(x) = sign(w^T\varphi(x) + b).$$

The optimal hyperplane can be found by solving the following quadratic optimization problem:

$$Minimize\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$subject\ to\quad y_i(w^T\varphi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, \quad i = 1, \dots, n$$

B. *Soft margin in the SVM*

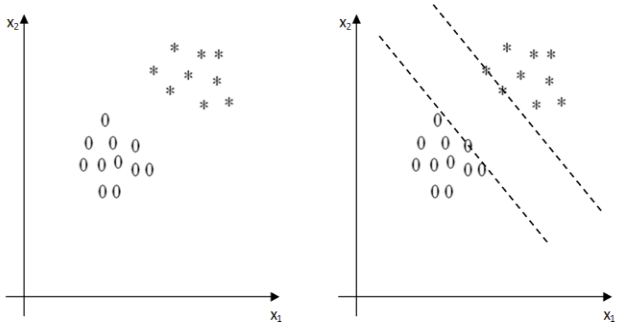
Adding weights or importance degree to each sample needs nonlinear constraints. For nonlinear and nonseparable case of SVM we can write,

$$J(w, b, \xi_i, \beta_i, \gamma_i, \theta_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$+\sum_{i=1}^{n}\beta_i g_i\left(\xi_i - 1 - y_i(w^T x_i + b) + \theta_i^2\right) - \sum_{i=1}^{n}\gamma_i \xi_i \qquad (4)$$

where β_i, γ_i are Lagrange multipliers, θ_i are values for achieving equal constraints. But $g_i(.)$ is a user defined function for obtaining the desired effect over constraints. $g_i(.)$ must be designed for suppression of noise or modeling of tolerance over samples. If user knows about importance degree of each constraint he can give different weights to different constraints, or present a degree of reliability for all samples, or determine a tolerance measure for data. This problem is discussed in the next section.

Consider the constraints modeling in the SVM. In the following problem (Fig. 1.a), with *C* approaching infinity in (1), classification is performed and the result is depicted in Fig. 1.b.



(a) (b)

Fig. 1: Classification task using SVM

Decreasing *C* can only increase margin as Fig. 2.

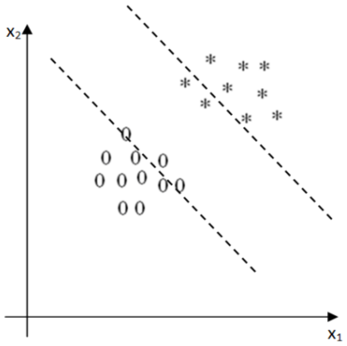


Fig. 2: Effect of decreasing *C*

As we know, existing parameters in the SVM formulation such as slack variables ξ_i and margin parameter *C* do not have direct control on training samples. Slack variables are not user defined and are determined during solving the optimization problem. Also, increasing (decreasing) the parameter *C* higher (lower) than some value will have small effect on the obtained classifier. But emphasis over specific training sample (pointed in Fig. 3) using FSVM (weighted SVM), with assigning larger s_i , gives us following classification. As we see, slope of optimum hyperplane changes in Fig. 3.

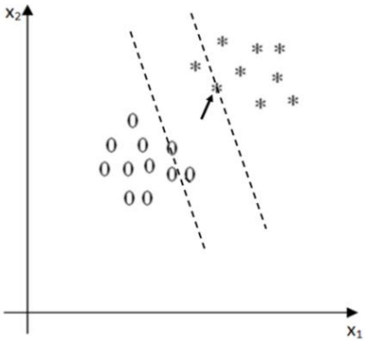


Fig. 3: Emphasis on the pointed sample using FSVM

As shown in Fig. 3, we can decrease the effect of noisy samples by inserting suitable weights, s_i, for each slack variable in the penalty term of FSVM. An improved version of SVM and FSVM are presented in the next sections, with new viewpoints of noise and uncertainty in training samples.

III. The proposed Emphatic constraints Support Vector Machine (ESVM)

In this section, we present a new viewpoint of importance degree and introduce tolerance and uncertainty for training samples as new concepts for support vector machines. Whereas in the training phase of the SVM a constraint is assigned to each sample, our primary question is that can we investigate the importance degree of samples in the constraint which is ascribed to each sample.

To answer this question we use fuzzy inequality in each constraint of the training samples in order to give more flexibility to each constraint satisfaction. Note that slack variables ξ_i cannot play this role because they are the unknowns of the system not the input variables. The proposed method is obtained by modifying the conventional SVM (1) into the following formulation:

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

$$\text{subject to}\quad y_i(w^T x_i + b) \gtrsim 1 - \xi_i, \quad i = 1, \dots, n \qquad (5)$$

$$w \in \mathcal{R}^m, \xi = (\xi_1, \xi_2, \dots, \xi_n), \xi_i \geq 0, i = 1, \dots, n$$

The symbol $\gtrsim$ means that we like to permit some violations in the satisfaction of the constraints. The fuzzy greater than or equal symbol defines membership functions

$$\mu_i : \mathcal{R}^{m+1+n} \to (0,1], \; i = 1, \dots, n.$$

According to the use of the representation theorem of fuzzy sets, consider a linear membership function for the *i*-th constraint (Fig. 4),

$$\mu_i(w,b,\xi)=$$

$$\begin{cases} 1, & if\ y_i(w^T x_i+b)\geq 1-\xi_i \\ \frac{y_i(w^T x_i+b)-1+\xi_i+d_i}{d_i}, & \text{if } 1-(\xi_i+d_i)\leq y_i(w^T x_i+b)\leq 1-\xi_i \\ 0, & \text{if } y_i(w^T x_i+b)\leq 1-(\xi_i+d_i) \end{cases} \quad (6)$$

Note that μ_i is function of an *m*-dimensional vector *w*, a scalar *b*, and an *n*-dimensional vector ξ.

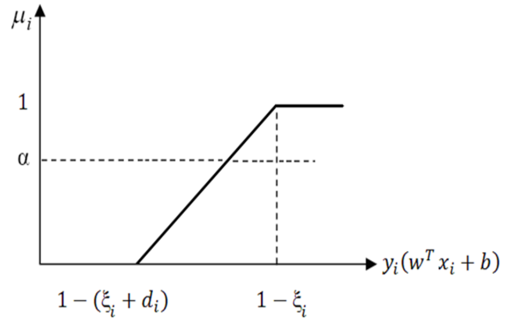


Fig. 4: membership function μ_i

For each constraint *i*, *i=1,2,…n*, of (5),

$$X_i=\{(w,b,\xi)\in\mathcal{R}^{m+1+n}\,|\,y_i(w^T x_i+b)\gtrsim 1-\xi_i,\xi_i\geq 0, i=1,\dots,n\},(7)$$

where $\xi=(\xi_1,\xi_2,\dots,\xi_n)$.

Taking $X=\cap_{i\in I}X_i$, where $I=\{1,\dots,n\}$, then (5) can be written as

$$Minimize\ \left\{Q(w,b,\xi)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i\,\middle|\,(w,b,\xi)\in X\right\}.\quad (8)$$

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

$$X(\alpha)=\{(w,b,\xi)\in\mathcal{R}^{m+1+n}\,|\,\mu_X(w,b,\xi)\geq\alpha\},$$

where $\mu_X(x)=\inf\{\mu_i(x),i\in I\}$. In this way $X_i(\alpha)$ will denote an α-cut of the *i*-th constraint.

The optimal solution of (5) for a given $\alpha\in(0,1]$ is:

$$S(\alpha)=\{(w,b,\xi)\in\mathcal{R}^{m+1+n}\,|\,\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i$$

$$=\{\text{Min}\ \frac{1}{2}\|w'\|^2+C\sum_{i=1}^{n}\xi'_i\},(w',b',\xi')\in X(\alpha)\}\quad (9)$$

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

$$X(\alpha)=$$

$$\bigcap_{i\in I}\{(w,b,\xi)\in\mathcal{R}^{m+1+n}\,|\,y_i(w^T x_i+b)\geq r_i(\alpha),\xi_i\geq 0, i=1,\dots,n\}\ (10)$$

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

$$Minimize\ \frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } y_i(w^T x_i+b)\geq 1-\xi_i-d_i(1-\alpha),\ \ i=1,\dots,n$$

$$\xi_i\geq 0,\ \ i=1,\dots,n\qquad (11)$$

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:

$$Q(w,b,\xi,\beta,\gamma)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i$$

$$-\sum_{i=1}^{n}\beta_i\{y_i(w^T x_i+b)-1+\xi_i+d_i(1-\alpha)\}$$

$$-\sum_{i=1}^{n}\gamma_i\,\xi_i\qquad (12)$$

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

For the optimal solution, the following Karush-Kuhn-Tucker (KKT) conditions are satisfied:

$$\frac{\partial Q(w,b,\xi,\beta,\gamma)}{\partial w}=0,\quad i.e., w=\sum_{i=1}^{n}\beta_i y_i x_i,\qquad (13)$$

$$\frac{\partial Q(w,b,\xi,\beta,\gamma)}{\partial b}=0,\quad i.e., \sum_{i=1}^{n}\beta_i y_i=0,\qquad (14)$$

$$\frac{\partial Q(w,b,\xi,\beta,\gamma)}{\partial \xi}=0,\quad i.e., \beta_i+\gamma_i=C.\qquad (15)$$

$$\beta_i\{y_i(w^T x_i+b)-1+\xi_i+d_i(1-\alpha)\}=0\qquad (16)$$

$$\gamma_i\xi_i=0\qquad (17)$$

$$\xi_i\geq 0,\quad \beta_i\ \geq 0,\quad \gamma_i\geq 0\qquad (18)$$

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

Thus substituting (13), (14), and (15) into (12), we obtain the following dual problem. Maximize

$$Q(\beta)=\sum_{i=1}^{n}\beta_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j-\sum_{i=1}^{n}\beta_i d_i(1-\alpha)$$

$$=\sum_{i=1}^{n}\beta_i(1-d_i+d_i\alpha)-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j \quad (19)$$

subject to the constraints:

$$\sum_{i=1}^{n}\beta_i y_i=0, \qquad 0\le\beta_i\le C, \qquad \text{for } i=1,\ldots,n.$$

The decision function is given by:

$$D(x)=sign(w^T x+b)=sign(\sum_{i\in S}\beta_i y_i x_i^T x+b) \quad (20)$$

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 $K(x,x')=\varphi^T(x).\varphi(x')$, the dual problem in the feature space is given as follows.

$$Maximize\ Q(\beta)=\sum_{i=1}^{n}\beta_i(1-d_i+d_i\alpha)-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j\varphi^T(x_i).\varphi(x_j)$$

$$=\sum_{i=1}^{n}\beta_i(1-d_i+d_i\alpha)-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j K(x_i,x_j) \quad (21)$$

subject to the constraints

$$\sum_{i=1}^{n}\beta_i y_i=0, \qquad 0\le\beta_i\le C, \qquad \text{for } i=1,\ldots,n.$$

The decision function is given by

$$D(x)=sign(\sum_{i\in S}\beta_i y_i x_i^T x+b) \quad (22)$$

where b is given by

$$b=y_i-\sum_{i\in S}\beta_i y_i K(x_i,x_j) \quad (23)$$

where S is the set of support vector indices.

In fact, we have changed constraints formulation of the SVM problem for our purposes and name this scheme as Emphatic constraints Support Vector Machine (ESVM). Constraints of ESVM have more flexibility than traditional SVMs because of their fuzzy inequalities. In this system, d_i and α are meaningful parameters. Each constraint is given a specific d_i which acts as a tolerance to the corresponding sample. In fact, the feasible region is extended for finding the unknown variables (w,b,ξ_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 Section 5. Fig. 5 depicts the concept of giving tolerance to samples, schematically.

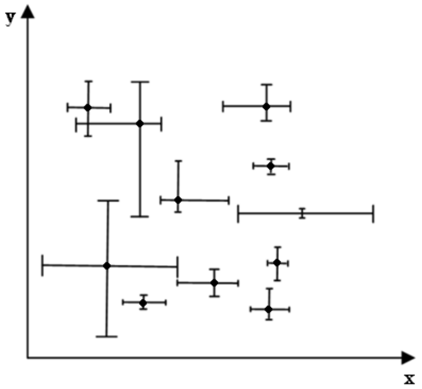


Fig. 5: Concept of data with tolerance

Also, α is another user defined parameter in ESVM formulation. 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 is clear that $(1-\alpha)$ indicates the uncertainty of user in the accuracy of collected samples. In the next sections we will study different effects of this parameter on 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 ESVM 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 constraints.

IV. The Proposed Fuzzy ESVM

In the previous section, we proposed EESVM 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. We follow two purposes from this idea:

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 (1).

2) Equipping FSVM with the new proposed concepts, namely, tolerance and certainty.

Therefore, we reformulate FSVM with fuzzy inequality in constraints and call this method fuzzy ESVM:

$$\mathit{Minimize}\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} s_i\xi_i$$

$$\text{subject to } y_i(w^T x_i + b) \gtrsim 1 - \xi_i, \quad i = 1, \dots, n \qquad (24)$$

$$\xi_i \geq 0, \qquad i = 1, \dots, n$$

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

$$\mathit{Minimize}\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} s_i\xi_i$$

$$\mathit{subject\ to}\ y_i(w^T x_i + b) \geq 1 - \xi_i - d_i(1-\alpha) \qquad (25)$$

$$\xi_i \geq 0, \qquad i = 1, \dots, n$$

Again we introduce the nonnegative Lagrange multipliers β_i and γ_i, we obtain from (25):

$$Q(w, b, \xi, \beta, \gamma) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} s_i\xi_i$$

$$-\sum_{i=1}^{n}\beta_i\{y_i(w^T x_i + b) - 1 + \xi_i + d_i(1-\alpha)\} - \sum_{i=1}^{n}\gamma_i\xi_i \qquad (26)$$

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

Applying KKT conditions such as previous section and substituting in (25), we obtain the following dual problem.

$$\mathit{Maximize}\ Q(\beta) = \sum_{i=1}^{n}\beta_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j + C\sum_{i=1}^{n} s_i\xi_i - \sum_{i=1}^{n}(\beta_i + \gamma_i)\xi_i - \sum_{i=1}^{n}\beta_i d_i(1-\alpha)$$

$$= \sum_{i=1}^{n}\beta_i(1 - d_i + d_i\alpha) + C\sum_{i=1}^{n}(s_i - 1)\xi_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j x_i^T x_j \qquad (27)$$

subject to the constraints

$$\sum_{i=1}^{n}\beta_i y_i = 0, \qquad 0 \leq \beta_i \leq s_i C, \qquad \text{for } i = 1, \dots, n.$$

Again, the decision function is:

$$D(x) = sign(w^T x + b) = sign(\sum_{i \in S}\beta_i y_i x_i^T x + b) \qquad (28)$$

where S is the set of support vector indices.

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

$$Q(\beta) = \sum_{i=1}^{n}\beta_i(1 - d_i + d_i\alpha) + C\sum_{i=1}^{n}(s_i - 1)\xi_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j \varphi^T(x_i).\varphi(x_j)$$

$$= \sum_{i=1}^{n}\beta_i(1 - d_i + d_i\alpha) + C\sum_{i=1}^{n}(s_i - 1)\xi_i$$

$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j y_i y_j K(x_i^T, x_j) \qquad (29)$$

subject to the constraints

$$\sum_{i=1}^{n}\beta_i y_i = 0, \qquad 0 \leq \beta_i \leq s_i C, \qquad \text{for } i = 1, \dots, n.$$

The role of the parameters d_i and α is similar to ESVM. These parameters give more robustness and generality to FSVM. In the next section we consider the effectiveness of our proposed method in practice. We will show the prominence of ESVM and fuzzy ESVM in comparison with SVM and FSVM and emphasize that fuzzy ESVM is a comprehensive method that has the fitness of FSVM and also equipped with other useful concepts.

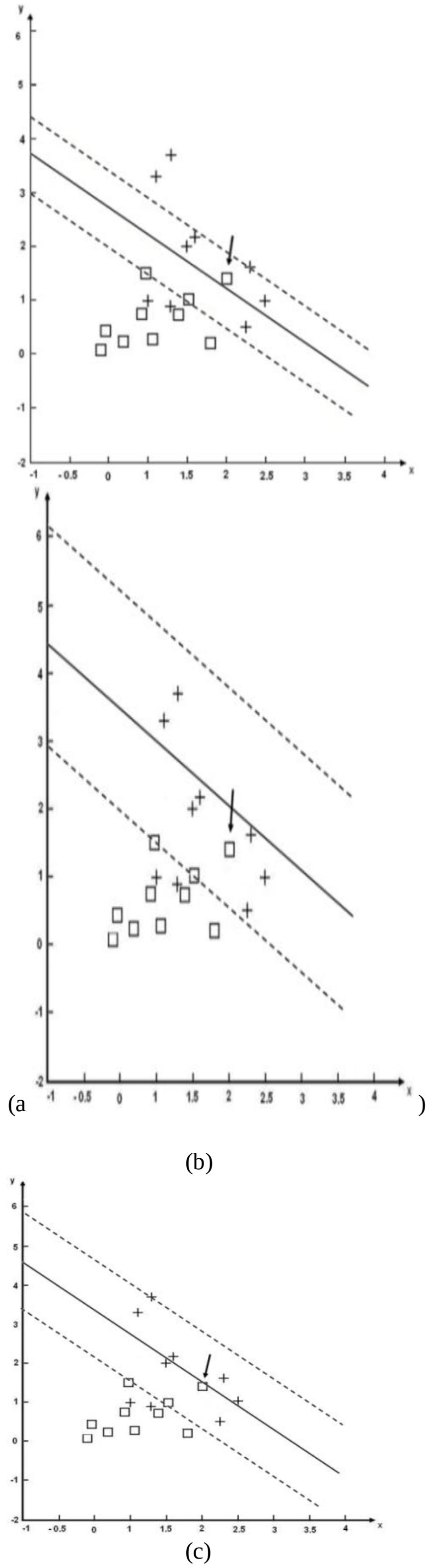
V. Effectiveness Of The Proposed Esvm And Fesvm

In this section we study the robustness and effectiveness of our proposed methods in practice with various experiments.

A. *Two classes with different weighting*

As we know, there may be some applications that we just want to focus on the accuracy of classifying one of the classes. One of the main aspects of the ability of FSVM is that it can give different emphasis to each class. It is sufficient to assume higher (close to 1) fuzzy membership for the premier class and lower membership for the other.

This ability also exists in our proposed ESVM and fuzzy ESVM. As we illustrate in pervious section, d_i can be used in order to give importance degree to each sample. Fig. 6 shows the results of an experiment with a data set that is created randomly. We gave higher importance to Class 1 (is indicated as square). In FSVM we set $s_i = 1$ for Class 1 and $s_i = 0.1$ for Class 2. In ESVM and fuzzy ESVM we set $d_i = 0.1$ for Class 1, $d_i = 1.2$ for Class 2, and $\alpha = 0.8$. Note that, in ESVM lower d_i means a higher importance degree.



(a)

(b)

(c)

Fig. 6: Results of FSVM (a), ESVM (b), and fuzzy ESVM (c) in classification of two classes with different weight

As we see in Fig. 6, even though the most attention is paid to Class 1 ($s_i = 1$), the pointed sample is classified incorrectly by FSVM. Although ESVM gets better result in comparison to SVM, but it has a higher training error. Instead, FESVM shows precise and accurate result.

B. A toy example

For synthetic training samples we run FSVM and ESVM algorithms, separately. Fuzzy membership s_i for FSVM is:

$$s_i = \begin{cases} 1-(\|x_+ - x_i\|/(r_+ + \delta)) & if\ x_i \in Class\ 1 \\ 1-(\|x_- - x_i\|/(r_- + \delta)) & if\ x_i \in Class\ 2 \end{cases} \qquad (30)$$

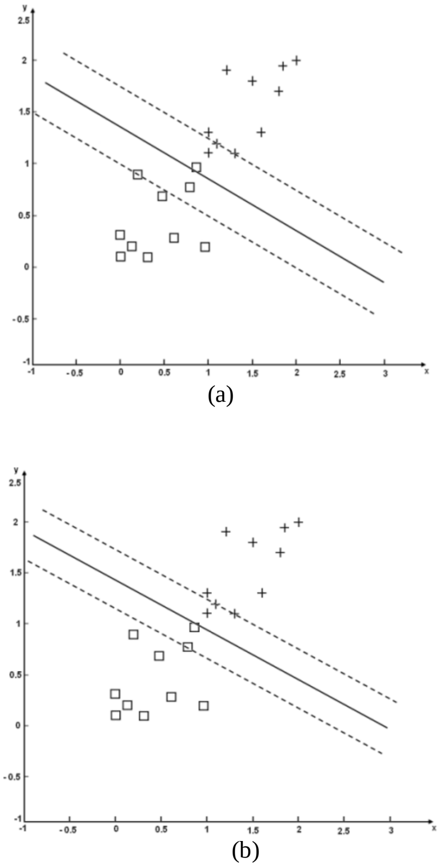
where x_+ and x_- is the mean of Class 1 and Class 2, respectively. Also, r_+ is the radius of Class 1

$$r_+ = \max_{\{x_i:\, x_i \in Class1\}} \|x_+ - x_i\| \qquad (31)$$

and radius of Class 2 is

$$r_- = \max_{\{x_i:\, x_i \in Class2\}} \|x_- - x_i\| \qquad (32)$$

For ESVM we need a subsystem to determine d_i. We used Circle Method [10] which is a geometric based model for giving importance degree to each sample. It has led to good results for this purpose. Fig. 7 shows the results of FSVM and ESVM (α=0.9) for a simple test satisfying above conditions. Obviously, ESVM learning is more effective.



(a)

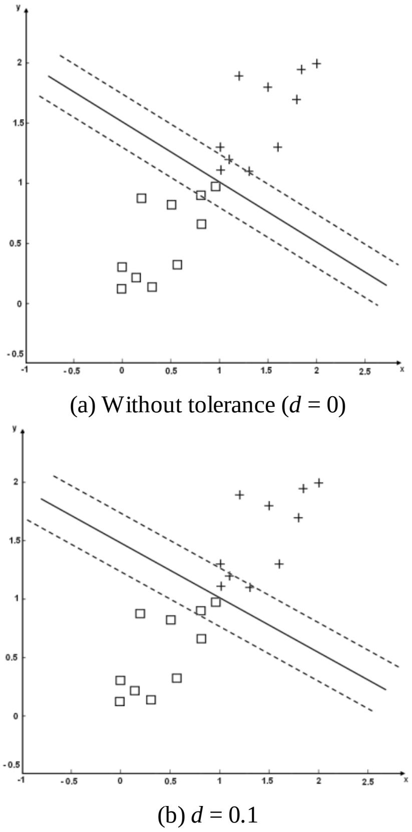
(b)

Fig. 7: The results of FSVM (a) and ESVM (b)

C. Study of the tolerance

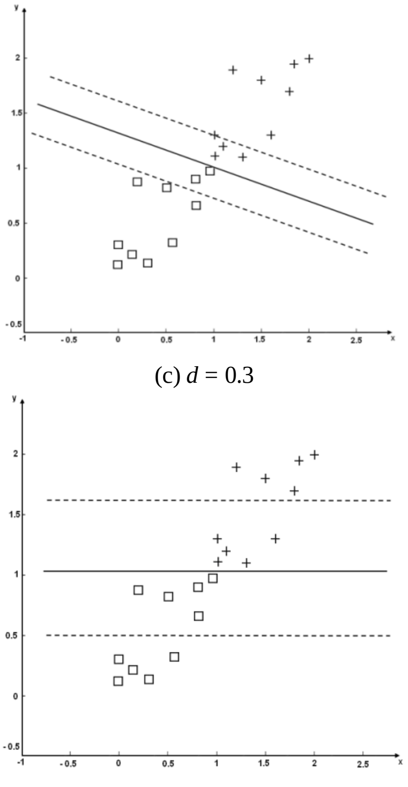
One of the distinctions of ESVM in comparison with other known classifiers is that it considers tolerance for the data. Suppose that we are going to give somewhat flexibility to the classifier. It means that the dash lines (in Fig. 7) can pass samples and the margin can be larger. It may be said that the parameter *C* in the SVM formulation (1) plays this role but as mentioned before increasing (decreasing) the parameter *C* higher (lower) than some value will have 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., $Minimize\ \frac{1}{2}\|w\|^2$) and minimization of overall errors (i.e., $Minimize\ C\sum_{i=1}^{n}\xi_i$) simultaneously and there is a trade off. But in ESVM and its extension (fuzzy ESVM) we can give more flexibility. In fact, by applying changes to the constraints of SVM formulation the feasible region changes. Note that, each constraint poses a limitation on the system and overall constraints make a feasible region. Also, the slack variables ξ_i cannot be seen as tolerance because they are not user defines and are determined during solving the optimization problem. In the situation that we have noisy and unreliable data, it is not correct that a tight classifier is trained. It is reasonable that we give tolerance to our samples.

Fig. 8 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 ESVM formulation (5) 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 we see in Fig. 8 the SVM classifier and the margin are different for each value of d_i. Not that here we give a tolerance to the entire training samples (all of d_is is identical and equal to d).



(a) Without tolerance (d = 0)

(b) d = 0.1



(c) d = 0.3

(d) d = 0.9

Fig. 8: Study of the tolerance in ESVM (α=0.7)

Note that the precision of the classifier in separating has not change, but training error is increased. Instead we create a safe margin.

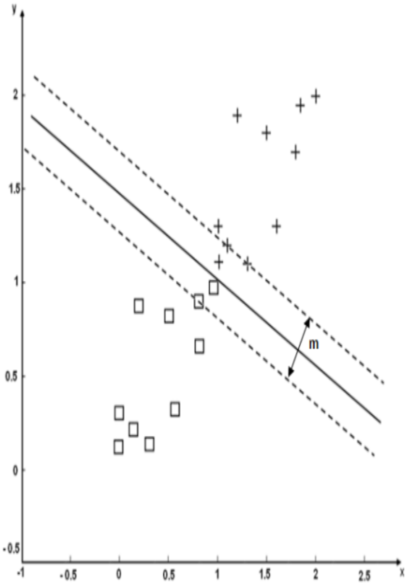
D. Study of the certainty in the training samples

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 [16]. In the Cross Industry Standard Process for Data Mining framework [17], 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, and for most of the data mining applications, the process of data collection, data preparation, and data enhancement cost the majority of the project budget and also the developing time circle [18].

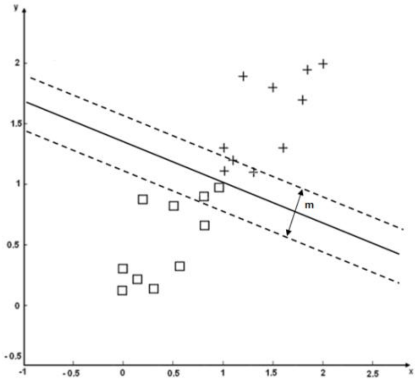
Real-world data mining deals with noisy information sources where data collection inaccuracy, device limitations, data transmission and discretization errors, or man-made perturbations frequently result in imprecise or vague data. Therefore, each training data set has a degree of uncertainty, in essence.

In ESVM 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 (6) for fuzzy inequality in the ESVM formulation (5) and then apply α-cut to convert the fuzzy inequality to a crisp one. As shown in Fig. 4, whatever this cutting is done closer to 1, it means that we have more confidence in our data and do 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 ESVM on our data set with different α values and determine a degree of certainty for it. This matter will be discussed in the next subsection.

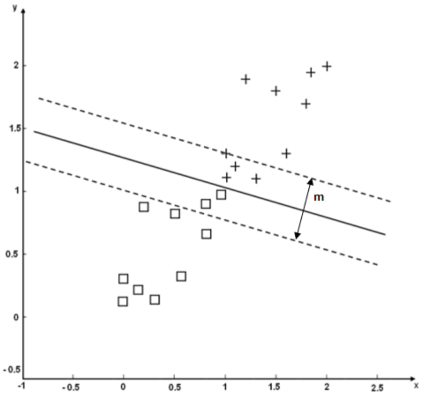
Now we bring a simple test to illustrate the concept of certainty for training samples. Fig. 9 shows the results of ESVM with different α values for a simple training data. When we choose a larger value for the certainty factor α, the classifier is trained with more cautious. In fact, we trust our data with high certainty. Therefore, a classifier is trained with high precision.



(a) α = 0.9



(b) α = 0.7



(c) α = 0.5

Fig. 9: Study of the certainty α in ESVM

As shown in Fig. 9 choosing a smaller value of α causes more training errors and larger margin. The margin size m is calculated as:

$$m = \frac{2}{\|w\|^2}$$

The margin sizes for this experiment are 0.0564 (Fig. 9.a), 0.0587 (Fig. 9.b), and 0.0599 (Fig. 9.c).

E. Experiments with real data

We evaluated our proposed methods by applying them to BUPA Liver Disorders, Statlog (Heart), and Haberman data sets. They were obtained from the UCI Repository of Machine Learning databases and domain theories [19]. All of them are two class classification problems.

BUPA Liver Disorders ***data set***; 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 analysed. 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 [20]. BUPA Liver Disorders data set 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:

$$K(x, x') = (x^T x' + 1)^d.$$

The obtained results are summarized in Table 1. Note that α is considered for ESVM and fuzzy ESVM.

Statlog (Heart) data set; it is a data set for recognition of absence (Class 1) or presence (Class 2) of heart disease in 270 observations. There are thirteen attributes for this data set. 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 $\sigma = 1$:

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

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, ESVM, and fuzzy ESVM on BUPA Liver Disorders data set

	SVM	FSVM	ESVM	fuzzy ESVM
C=100, α=0.9	65.5556	68.8889	71.1111	**72.222**
C=1000, α=0.9	63.3333	68.8889	71.1111	**72.222**

Table 2: The recognition rate of SVM, FSVM, ESVM, and fuzzy ESVM on Statlog data set

	SVM	FSVM
	81.1111	80
	ESVM	fuzzy ESVM
α=0.9	82.2222	84.4444
α=0.8	84.4444	**86.6667**
α=0.7	82.2222	83.3333
α=0.6	78.8889	81.1111
α=0.5	78.8889	78.8889
α=0.4	67.7778	70
α=0.3	73.3333	74.4444
α=0.2	72.2222	73.3333
α=0.1	74.4444	75.5556

Experimental results in Table 1 and Table 2 show that fuzzy ESVM 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 data set. Indeed the value of α with maximum recognition rate indicates how much the process of data collection is reliable. Fig. 10 shows the recognition rates of fuzzy ESVM with different values of α for Statlog data set with respect to Table 2. We can say certainty of this data set is 80%. So, in the test procedure of fuzzy ESVM for this data set, α is set to 0.8.

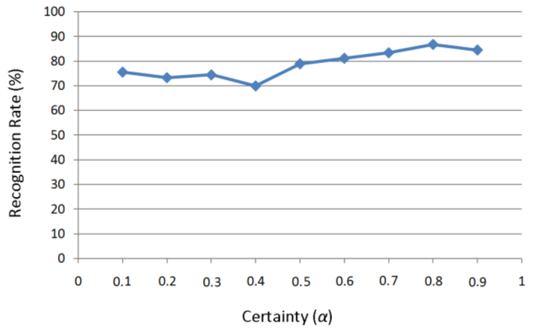


Fig. 10: The recognition rate of fuzzy ESVM with different values of α

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 data sets to SVM. We recommend ESVM and fuzzy ESVM to the researchers who work on data sets with noisy or low degree of certainty samples.

VI. Conclusion And Future Work

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 ESVM. The constraints of ESVM have more flexibility because of their fuzzy inequalities. Then we introduce a new method for SVM classification problems with combination of ESVM and FSVM (fuzzy ESVM). Constraints of the standard SVM is converted to fuzzy inequalities. Solving SVM with fuzzy constraints leads to forming ESVM optimization problem that is equipped with new concepts that are not considered up until now. All of the experimental results showed the power of our proposed methods. We can mention the superiority of these methods as follow:

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 towards having samples with tolerance ESVM works far better.
2) *Handling uncertainty in data*. If we have a prior knowledge about a data set, and are aware of its accuracy and precision, there are no potentialities in SVM or FSVM to handle this knowledge. ESVM is the solution. It is also capable to find certainty for unknown data sets.
3) *Better performance*. The experiments we have performed showed that the quality of classification on real data is higher than both SVM and FSVM.

For the future works, we will extend our proposed method to multi-class SVM classification and support vector regression problems.

Acknowledgement

This work has been partially supported by Iran Telecommunication Research Center (ITRC), Tehran, Iran. This support is gratefully acknowledged.

References

[1] V. Vapnik, “The Nature of Statistical Learning Theory,” New-York: Springer-Verlag, 1995.

[2] S. Abe, “Advances in Pattern Recognition,” Springer-Verlag London Limited, ISSN 1617-7916, 2005.

[3] L. Wang, “Support Vector Machines: Theory and Applications,” Springer-Verlag Berlin Heidelberg, ISSN print edition: 1434-9922, 2005.

[4] C. F. Lin and S. D. Wang, “Fuzzy Support Vector Machine,” IEEE Trans. on Neural Networks, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[5] C. F. Lin and S. D. Wang, “Training Algorithms for Fuzzy Support Vector Machines with Noisy Data”, Pattern Recognition Letters, vol.25, pp. 1647–1656, 2004.

[6] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” J. Mach. Learn. Res., vol. 1, pp. 211–244, 2001.

[7] J. Sun and Y. Zhou, “Noise Reduction of Chaotic Systems Based on Least Squares Support Vector Machines,” Communications, Circuits and Systems Proceedings, 2006 International Conference on, Vol. 1, pp. 336-339, June 2006.

[8] C. Chen, X. Qi, and M. Lin, “Research and Application of Noise Suppression Based on Support Vector Machine,” Proceedings ofISCIT, pp.346-349, 2005.

[9] J. Zhang, Q. Peng, H. Shao, and T. Shao, “Nonlinear Noise Filtering with Support Vector Regression,” Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, pp. 172-176, 2006.

[10] L. Chu, and C. Wu, “A Fuzzy Support Vector Machine Based on Geometric Model,” Proceedings of the fifth World Congress on Intelligent Control and Automation, Hangzhou, P.R. China, pp.1843-1846, June 15-19, 2004.

[11] Y. Wang, S. Wang, and K. K. Lai, “A New Fuzzy Support Vector Machine to Evaluate Credit Risk,” IEEE Trans. on Fuzzy Systems, vol.13 no.6, pp.820-831, Dec. 2005.

[12] [Jayadeva, R. Khemchandani, S. Chandra, “Fuzzy Linear Proximal Support Vector Machines for Multi-Category Data Classification,” Neurocomputing, vol.67, pp.426-435, 2005.

[13] T. Y. Wang and H. M. Chiang, “Fuzzy Support Vector Machine for Multi-Class Text Categorization,” Information Processing and Management, vol.43, pp.914 – 929, 2007.

[14] A. B. Ji, J. H. Pang, S. H. Li, and J. P. Sun, “Support Vector Machine for Classification Based on Fuzzy Training Data,” Proceedings of the Fifth Int. Conf. on Machine Learning and Cybernetics, Dalian, pp.1609-1614, 2006.

[15] H. Sadoghi Yazdi, S. Effati, and Z. Saberi, “The Probabilistic Constraints in the Support Vector Machine,” Applied Mathematics and Computation, vol.194, no.2, p.467-479, Dec 2007.

[16] M. Berry and G. Linoff, “Mastering Data Mining”, New York: Wiley, 1999.

[17] C. Shearer, “The CRISP-DM model: The new blueprint for data mining,” J. Data Warehous., vol. 5, no. 4, pp. 13-22, 2000.

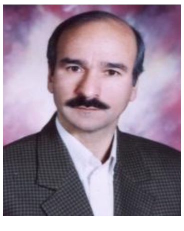
[18] X. Wu, X. Zhu, “Mining With Noise Knowledge: Error-Aware Data Mining,” IEEE Trans. on Systems, Man, and Cybernetics—Part A: Systems and Humans, vol. 38, no. 4, July 2008.

[19] Murphy PM, Aha KW. UCI Repository of machine learning databases, [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science, 1994.

[20] E. ComaK, K. Polat, S. Gunes, and A. Arslan, “A new medical decision making system: Least square support vector machine (LSSVM) with Fuzzy Weighting Pre-processing,” Expert Systems with Applications 32, pp. 409-414, 2007.

Mostafa Sabzekar was born in Birjand, Iran, in 1985. He received the B.S. degree in Computer Engineering from Tarbiat Moallem University of Tehran, Iran, in 2007. He is currently an M.S. student in Computer Engineering at Ferdowsi University of Mashhad, Iran. His research interests include knowledge-based system, machine learning, data mining, and optimization.

Hadi Sadoghi Yazdi received the B.S. degree in Electrical Engineering from Ferdowsi University of Mashhad, Iran, in 1994, and then he received to the M.S. and PhD degrees in Electrical Engineering from Tarbiat Modarres University of Tehran, in 1996 and 2005, respectively. He works in Computer Engineering Department as an assistant professor at Ferdowsi University of Mashhad, Iran. His research interests include optimization, adaptive filtering, image and video processing. He has more than 140 journal and conference publications in subject of interesting area.



Mahmoud Naghibzadeh received his B.S. degree in statistics and computer science from Ferdowsi University of Mashhad, Iran and M.S. and PhD degrees in computer science and Computer Engineering, respectively, from University of Southern California, USA. His past research interests include operating system concepts and techniques, especially process scheduling, distributed databases design concepts, real-time systems, and Grid computing with emphesis on scheduling. He has published many papers in international journals and conferences. He is currently researching in the area on knowledge engineering. He has published ten successful books in the area of Computer Engineering. He is supervising research activities in the fields of Grid computing, soccer simulation, real-time scheduling, and knowledge engineering. Prof. Naghibzadeh is currently a full professor at Ferdowsi University of Mashhad, Mashhad, Iran, where he is supervising M.S. and PhD students as well as teaching computer science courses.

Sohrab Effati received the B.S. degree in Applied Mathematics from Birjand University, Birjand, Iran, the M.S. degree in Applied Mathematics from Institute of Mathematics at Tarbiat Moallem Tehran University, Tehran, Iran, in 1992 and 1995, respectively. He received the PhD degree in control systems from Ferdowsi University of Mashhad, Mashhad, Iran, in April 2000. He is an Associate Professor with the Department of Applied Mathematics at Ferdowsi Univers ity of Mashhad in Iran. His research interests include control systems, fuzzy theory, and neural network models and its applications in optimization problems.