

# Emphatic Constraints Support Vector Machines for Multi-class Classification

Mostafa Sabzekar

Department of Computer Engineering  
Ferdowsi University of Mashhad, Iran  
sabzekar@wali.um.ac.ir

Mahmoud Naghibzadeh

Department of Computer Engineering  
Ferdowsi University of Mashhad, Iran  
naghibzadeh@um.ac.ir

Hadi Sadoghi Yazdi

Department of Computer Engineering  
Ferdowsi University of Mashhad, Iran  
sadoghi@um.ac.ir

Sohrab Effati

Department of Applied Mathematics  
Ferdowsi University of Mashhad, Iran  
s-effati@um.ac.ir

**Abstract**—Support vector machine (SVM) formulation has been originally developed for binary classification problems. Finding the direct formulation for multi-class case is not easy but still an on-going research issue. This paper presents a novel approach for multi-class SVM by modifying the training phase of the SVM. First, we propose the Emphatic Constraints Support Vector Machines (ECSVM) as a new powerful classification method. Then, we extend our method to find efficient multi-class classifiers. We evaluate the performance of the proposed scheme by means of real world data sets. The obtained results show the superiority of our method.

**Keywords:** Support vector machines; multi-class classification; fuzzy inequality; emphatic constraints;

## I. INTRODUCTION

The theory of support vector machine (SVM), which is based on the idea of structural risk minimization, is a new classification technique and has drawn much attention due to its good performance and solid theoretical foundations [1, 2]. The good generalization ability of SVMs is achieved by finding a large margin between two classes [4]. In many real world applications, the theory of SVM has been shown to provide higher performance than traditional learning methods [5] and has been introduced as a powerful tool for solving classification problems.

SVMs work by implicitly (using the kernel trick) mapping all training data from input space into a higher dimensional feature space. An oriented linear hyperplane is constructed in this feature space such that it bisects the two classes of training vectors and maximizes the perpendicular distance between itself and those points lying closest to the support vectors. Maximizing this margin is a quadratic programming (QP) problem and can be solved from its dual problem by introducing Lagrangian multipliers.

In spite of all advantages, there are some limitations in using SVMs. One problem is that the SVM formulation has been initially developed for the binary (two-class) case and the direct formulation of the multiclass problems is seldom applied in practice, due to its complexity [6]. There are two ways of designing a multi-class classifier, one is to directly develop a multi-class algorithm; the other is to decompose a multi-class problem to multiple two-class problems.

Recently, the decomposition scheme has gained a lot of attention. The reason is twofold: First, binary classifiers are easier to implement; second, some powerful algorithms are inherently binary such as SVM [7].

Furthermore there are more and more applications using the SVM techniques. However, in many applications, some input points may not be exactly assigned to one of these two classes. Some are more important to be fully assigned to one class so that SVM can separate these points more correctly. Some data points corrupted by noises are less meaningful and the machine should better to discard them. SVM lacks this kind of ability and it is another problem with SVM [8]. Lin and Wang proposed Fuzzy SVM (FSVM) [8, 9] to overcome this problem.

In this paper we present a new model of SVM, namely Emphatic Constraints SVM (ECSVM) that considers an importance degree for each training sample in the constraints of SVM formulation. This is contrary to the FSVM that considers importance degrees in the cost function. Then we extend our proposed ECSVM to find multi-class classifiers.

The remainder of this paper is organized as follows. Section II reviews the SVM formulation and multi-class SVM classifiers. The structure of the proposed ECSVM is given in Section III. Section IV presents the empirical experiments to demonstrate the effectiveness of the proposed system. The conclusion is given in Section V.

## II. SUPPORT VECTOR MACHINES

### A. SVM Formulation

Support vector machine (SVM) is an advanced model for the classification of different sorts of data. In this section, the focus is on two-class classification problem. The training data is represented as:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \in \mathcal{R}^m \times \{\pm 1\},$$

where,  $y_i = +1, -1$  represent positive class and negative class, respectively. The geometrical interpretation of support vector classification is that the algorithm searches for the optimal separating hyperplane. First, the SVM is outlined for

the linearly separable case. The training data are linearly separable if there exists a pair  $(w, b)$  such that

$$\begin{cases} w^T \mathbf{x}_i + b \geq 1, & \text{for all } y_i = +1 \\ w^T \mathbf{x}_i + b \leq -1, & \text{for all } y_i = -1 \end{cases} \quad (1)$$

with the decision rule given by

$$f(\mathbf{x}) = \text{sign}(w^T \mathbf{x} + b) \quad (2)$$

where  $w$  is the normal vector and  $b$  is a scalar. The inequality constraints (1) can be combined to give

$$y_i(w^T \mathbf{x}_i + b) \geq 1 \quad (3)$$

The learning problem is hence reformulated as a convex quadratic programming (QP) problem

$$\begin{aligned} & \text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{subject to } y_i(w^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \quad \xi_i \geq 0, \quad i = 1, \dots, n \end{aligned} \quad (4)$$

This problem has a global optimum. The dual form of the SVM problem presented in (4) is to maximize the objective function

$$Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (5)$$

subject to the constraints:

$$\sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \dots, n.$$

The decision function is given by:

$$D(\mathbf{x}) = \text{sign}(w^T \mathbf{x} + b) = \text{sign}\left(\sum_{i \in S} \alpha_i y_i x_i^T \mathbf{x} + b\right) \quad (6)$$

where  $S$  is the set of support vector indices.  $D(\mathbf{x})$  is the desired hyperplane sought.

For nonlinear separable case, the original input space is mapped into high-dimensional dot-product feature space using a  $\varphi$ -function. Using the kernel function

$$K(\mathbf{x}, \mathbf{x}') = \varphi^T(\mathbf{x}) \cdot \varphi(\mathbf{x}'),$$

the dual problem in the feature space is to maximize the objective function

$$Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \varphi^T(x_i) \cdot \varphi(x_j)$$

$$= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (7)$$

subject to the constraints

$$\sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \dots, n.$$

The decision function is given by

$$D(\mathbf{x}) = \text{sign}\left(\sum_{i \in S} \alpha_i y_i K(x_i, \mathbf{x}) + b\right) \quad (8)$$

where  $b$  is given by

$$b = y_i - \sum_{i \in S} \alpha_i y_i K(x_i, x_j) \quad (9)$$

where  $S$  is the set of support vector indices. Here,  $D(\mathbf{x})$  is the desired hyperplane separating the two classes.

### B. Multi-class SVM Classifiers

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. The solution to the multi-class classification can be reconstructed from the outputs of the two-class classifiers. The following two strategies are mainly adopted: "one-against-all" and "one-against-one". Although, other methods exist, for instance, the error-correcting code techniques [10].

The one-against-all [1] method constructs  $n$  SVMs (where  $n$  is the number of classes). Let the  $i$ -th decision function, with the maximum margin that separates Class  $i$  from the remaining classes, be

$$D_i(\mathbf{x}) = w_i^T \varphi(\mathbf{x}) + b_i, \quad (10)$$

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

$$\arg \max_{i=1, \dots, n} D_i(\mathbf{x}). \quad (11)$$

The one-against-one [11] (pairwise), instead, constructs  $n(n-1)/2$  decision functions for all the combinations of class pairs, where  $n$  is the number of classes. Let the decision function for Class  $i$  against class  $j$ , with the maximum margin, be

$$D_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^T \boldsymbol{\varphi}(\mathbf{x}) + b_{ij}. \quad (12)$$

The regions

$$R_i = \{\mathbf{x} | D_{ij}(\mathbf{x}) > 0, j = 1, 2, \dots, n, j \neq i\}. \quad (13)$$

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

$$D_i(\mathbf{x}) = \sum_{i \neq j, j=1}^n \text{sign}(D_{ij}(\mathbf{x})), \quad (14)$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{for } x \geq 0, \\ -1 & \text{for } x < 0, \end{cases} \quad (15)$$

and  $\mathbf{x}$  is classified into class:

$$\arg \max_{i=1,2,\dots,n} D_i(\mathbf{x}). \quad (16)$$

If  $\mathbf{x} \in R_i$ ,  $D_i(\mathbf{x}) = n - 1$  and  $D_k(\mathbf{x}) < n - 1$  for  $k \neq i$ . Thus,  $\mathbf{x}$  is classified into  $i$ . But if any of  $D_i(\mathbf{x})$  is not  $n - 1$ , (16) may be satisfied for plural  $i$ s. In this case,  $\mathbf{x}$  is unclassifiable. To resolve this problem, Vapnik [3] 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) [12] that uses a decision tree in the testing stage. 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  $\mathbf{x}$ . Let the two classes for which the classification decision is performed be  $i$  and  $j$ . If  $D_{ij}(\mathbf{x}) > 0$  we delete the element  $j$  from the list. We repeat the procedure until one element is left. Then we classify  $\mathbf{x}$  into the class that corresponds to the element number.

In the next section we will present ECSVM for training SVM classifier and will use it for multi-class classification problem.

### III. THE PROPOSED EMPHATIC CONSTRAINTS SVM

In this section we propose a new structure for support vector machines and then use it for solving multi-class classification problems. Whereas in the training phase of the SVM (4) 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 and relaxation to each constraint satisfaction. Note that slack variables  $\xi_i$  in conventional SVM 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 (4) into the following formulation:

$$\begin{aligned} \text{Minimize } Q(\mathbf{w}, b, \boldsymbol{\xi}) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ \text{subject to } y_i(\mathbf{w}^T \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad i = 1, \dots, n \quad (17) \\ \mathbf{w} \in \mathcal{R}^m, \boldsymbol{\xi} &= (\xi_1, \xi_2, \dots, \xi_n), \xi_i \geq 0, i = 1, \dots, n \end{aligned}$$

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

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

According to the use of the representation theorem of fuzzy sets, consider a linear membership function for the  $i$ -th constraint (Figure 1),

$$\mu_i(\mathbf{w}, b, \boldsymbol{\xi}) = \begin{cases} 1, & \text{if } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ \frac{y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i + d_i}{d_i}, & \text{if } 1 - (\xi_i + d_i) \leq y_i(\mathbf{w}^T \mathbf{x}_i + b) \leq 1 - \xi_i \\ 0, & \text{if } y_i(\mathbf{w}^T \mathbf{x}_i + b) \leq 1 - (\xi_i + d_i) \end{cases} \quad (18)$$

Note that  $\mu_i$  is function of an  $m$ -dimensional vector  $\mathbf{w}$ , a scalar  $b$ , and an  $n$ -dimensional vector  $\boldsymbol{\xi}$ .

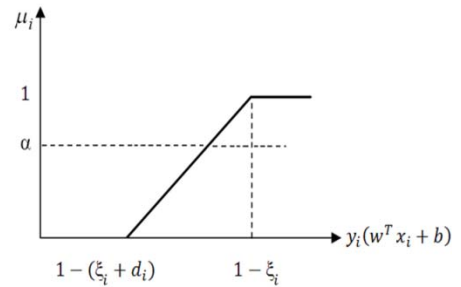


Figure 1. Membership function  $\mu_i$

For each constraint  $i, i=1, 2, \dots, n$ , of (17),

$$= \{(w, b, \boldsymbol{\xi}) \in \mathcal{R}^{m+1+n} | y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \dots, n\}, \quad (19)$$

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

Taking  $X = \bigcap_{i \in I} X_i$ , where  $I = \{1, \dots, n\}$ , then (17) can be written as

$$\text{Minimize } \left\{ Q(w, b, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \mid (w, b, \xi) \in X \right\}. \quad (20)$$

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

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

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

The optimal solution of (18) for a given  $\alpha \in (0, 1]$  is:

$$\begin{aligned} S(\alpha) &= \{(w, b, \xi) \in \mathcal{R}^{m+1+n} \mid \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ &= \{\text{Min } \frac{1}{2} \|w'\|^2 + C \sum_{i=1}^n \xi'_i, (w', b', \xi') \in X(\alpha)\} \end{aligned} \quad (21)$$

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

$$X(\alpha) = \bigcap_{i \in I} \{(w, b, \xi) \in \mathcal{R}^{m+1+n} \mid y_i(w^T x_i + b) \geq r_i(\alpha), \xi_i \geq 0, i = 1, \dots, n\} \quad (22)$$

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

$$\begin{aligned} &\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ &\text{subject to } y_i(w^T x_i + b) \geq 1 - \xi_i - d_i(1 - \alpha), \quad i = 1, \dots, n \\ &\quad \xi_i \geq 0, \quad i = 1, \dots, n \end{aligned} \quad (23)$$

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

$$\begin{aligned} Q(w, b, \xi, \beta, \gamma) &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ &- \sum_{i=1}^n \beta_i \{y_i(w^T x_i + b) - 1 + \xi_i + d_i(1 - \alpha)\} \\ &- \sum_{i=1}^n \gamma_i \xi_i \end{aligned} \quad (24)$$

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:

$$\frac{\partial Q(w, b, \xi, \beta, \gamma)}{\partial w} = 0, \quad i.e., w = \sum_{i=1}^n \beta_i y_i x_i, \quad (25)$$

$$\frac{\partial Q(w, b, \xi, \beta, \gamma)}{\partial b} = 0, \quad i.e., \sum_{i=1}^n \beta_i y_i = 0, \quad (26)$$

$$\frac{\partial Q(w, b, \xi, \beta, \gamma)}{\partial \xi} = 0, \quad i.e., \beta_i + \gamma_i = C. \quad (27)$$

$$\beta_i \{y_i(w^T x_i + b) - 1 + \xi_i + d_i(1 - \alpha)\} = 0 \quad (28)$$

$$\gamma_i \xi_i = 0 \quad (29)$$

$$\xi_i \geq 0, \quad \beta_i \geq 0, \quad \gamma_i \geq 0 \quad (30)$$

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

Thus substituting (25), (26), and (27) into (24), we obtain the following dual problem. Maximize

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

subject to the constraints:

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

The decision function is given by:

$$D(x) = \text{sign}(w^T x + b) = \text{sign}\left(\sum_{i \in S} \beta_i y_i x_i^T x + b\right) \quad (32)$$

and  $b$  is given by:

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

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 Emphatic Constraints Support Vector Machine (ECSVM). Constraints of ECSVM have more relaxation than

traditional SVMs because of their fuzzy inequalities. In this system,  $d_i$  and  $\alpha$  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  $(w, b, \xi_i)$ . Note that, slack variables  $\xi_i$  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  $\xi_i$ . 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  $d_s$  are assigned to different constraints, it means we have assumed a different degree of importance to samples; similar to Fuzzy SVM. 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. For ECSVM we need a subsystem to determine  $d_i$ . We used Circle Method [13] which is a geometric based model for giving importance degree to each sample.

Also,  $\alpha$  is another user defined parameter in RSVM formulation. It is the level at which the membership degree of the fuzzy inequality of constraints,  $\mu_i$ , is cut. This new SVM formulation as nonlinear optimization problem with fuzzy inequality constraints adds useful concepts to conventional SVMs.

We can use ECSVM 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 one-against-all ECSVM, we train  $n$  ECSVM, where  $n$  is the number of classes. ECSVM $_i$  separates Class  $i$  from the remaining classes. A testing sample  $x_t$  is assigned to the class with maximum decision function value. Figure 2 shows the details of this method. Note that,  $D_i$  is the value of  $i$ -th decision function. Figure 2 illustrates this method.

In the pairwise ECSVM and DAG ECSVM,  $n(n-1)/2$  ECSVMs are trained. ECSVM $_{ij}$  is the optimal separating hyperplane between Class  $i$  and class  $j$ . In the pairwise ECSVM, a testing sample  $x_t$  is assigned to class with maximum decision function represented by the Equation (14). The DAG ECSVM uses a decision tree in the testing stage. Figure 3 shows the decision tree for the case where there are three classes. In this figure,  $\bar{i}$  shows that  $x_t$  does not belong to Class  $i$ . For the top-level classification, we can choose any pair of classes. Except for the leaf node, if  $D_{ij}(x_t) \geq 0$  it means that  $x_t$  does not belong to class  $j$ . On the other hand, if  $D_{ij}(x_t) < 0$  it means that  $x_t$  does not belong to Class  $i$ . Thus, if  $D_{12}(x_t) > 0$ ,  $x$  does not belong to Class II. Therefore, it belongs to either Class I or Class III. Therefore, the next classification is between classes I and III.

In the next section we will evaluate our proposed method using real world data sets.

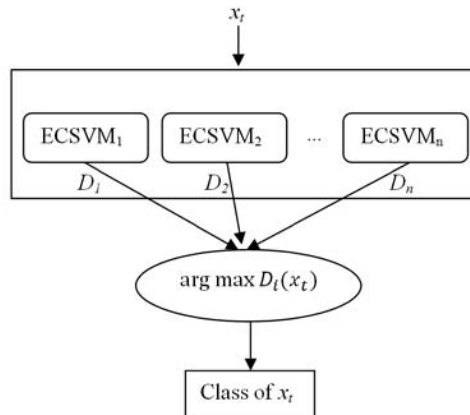


Figure 2. Details of one-against-all ECSVM

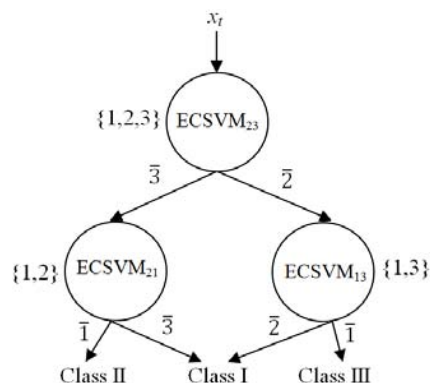


Figure 3. Classification by DAG ECSVM

#### IV. EXPERIMENTAL RESULTS

In the previous section, we presented the ECSVM classifier that was equipped with new concepts. In this section, our proposed method is tested using real world data sets. All data sets used in the following tests are obtained from the UCI Repository of Machine Learning Databases and Domain Theories [14]. Details of these data sets are summarized in Table I. We applied one-against-all SVM and one-against-all ECSVM to different data sets. The results are summarized in Table II. The one-against-all ECSVM technique is similar to the one-against-all SVM with the difference that all of decision functions are trained using our proposed ECSVM. In the same way, we compared the pairwise SVM to the pairwise ECSVM and also the DAG SVM to the DAG ECSVM and summarized the results in Table III and Table IV, respectively. In these experiments, RBF kernel function  $k(x, y) = e^{-\|x-y\|^2/2\sigma^2}$  is used, with  $C = 100$ , and  $\alpha$ , in the ECSVM, being equal to 0.9. It is worth mentioning that 70% of the data in the data set were randomly selected for the training phase and the rest for the testing phase.

TABLE I. DETAILS OF DATA SETS IN EXPERIMENTS

Data set	Number of classes	Number of attributes	Number of instances	Area
Glass Identification	6	10	214	Physical
Page Blocks	5	10	5473	Computer
Image Segmentation	7	19	2310	N/A
Statlog (Shttle)	7	9	58000	Physical

TABLE II. ONE-AGAINST-ALL SVM VS. ONE-AGAINST-ALL ECSVM RECOGNITION RATES

Data set	One-against-all SVM	One-against-all ECSVM
Glass Identification	66.33	71.43
Page Blocks	80.45	84.56
Image Segmentation	70.62	77.12
Statlog (Shttle)	69.16	79.16

TABLE III. PAIRWISE SVM VS. PAIRWISE ECSVM RECOGNITION RATES

Data set	Pairwise SVM	Pairwise ECSVM
Glass Identification	63.33	65
Page Blocks	80.67	84.29
Image Segmentation	68.04	73.04
Statlog (Shttle)	72.04	78.38

TABLE IV. DAG SVM VS. DAG ECSVM RECOGNITION RATES

Data set	DAG SVM	DAG ECSVM
Glass Identification	66.67	68.33
Page Blocks	85.06	90.33
Image Segmentation	71.43	78.82
Statlog (Shttle)	65.18	75.70

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

## V. CONCLUSION

SVM formulation has been originally developed for two-class classification problems. Finding the direct formulation for the multi-class classification case is not easy and it is an

on-going research issue. In this paper, we proposed a new approach for multi-class SVM by improving the training phase of the previous one. First, we proposed a new model of support vector machines with emphasis on constraints of the optimization problem of the SVM formulation and named it ECSVM. Using fuzzy inequalities, the constraints of the ECSVM are relaxed in order to offer a higher degree of flexibility. Then, we used the ECSVM for solving multi-class classification problems. In fact, we train all of the binary SVMs with our proposed ECSVM. The experimental results showed the superiority of our proposed methods over previous ones.

## ACKNOWLEDGMENT

This work has been supported by Iran Telecommunication Research Center (ITRC), Tehran, Iran (Contract number: 5075/500). This support is gratefully acknowledged.

## REFERENCES

- [1] V. Vapnik, "The Nature of Statistical Learning Theory," New York: Springer-Verlag, 1995.
- [2] J. C. Cortes, and V. Vapnik, "Support Vector Networks," in Machine Learning. Boston, MA: Kluwer, pp. 273–297, 1995.
- [3] V. Vapnik, "Statistical Learning Theory," John Wiley & Sons, New York, NY, 1998.
- [4] J. Shawe-Taylor and P.L., Bartlett, "Structural risk minimization over data-dependent hierarchies," IEEE Trans. Inform. Theor. 44 (5), 1926–1940, 1998.
- [5] J. Shawe-Taylor and P.L., Bartlett, "Structural risk minimization over data-dependent hierarchies," IEEE Trans. Inform. Theor. 44 (5), 1926–1940, 1998.
- [6] V. Navia-Vázquez, "Compact multi-class support vector machine," Neurocomputing 71, 400–405, 2007.
- [7] J. Zhou, H. Peng, and C.Y. Suen, "Data-driven decomposition for multi-class classification," Pattern Recognition 41, 67–76, 2008.
- [8] C.F. Lin, and S.D. Wang, "Fuzzy Support Vector Machine," IEEE Transactions on Neural Networks, Vol. 13, No. 2, 2002.
- [9] 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.
- [10] T. Dietterich, and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," J. Artif. Intell. Res. 2, 263–286, 1995.
- [11] T. Hastie, and R. Tibshirani, "Classification by pairwise coupling," Ann. Stat. 26 (2), 451–471, 1998.
- [12] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," Advances in Neural Information Processing Systems 12. MIT Press, 543–557, 2000.
- [13] 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.
- [14] Murphy PM, Aha KW. UCI Repository of machine learning databases, [http://www.ics.uci.edu/~mllearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science, 1994.