# A New Multi-class Fuzzy Support Vector Machine Algorithm

Friedhelm Schwenker, Markus Frey, Michael Glodek, Markus Kächele,
Sascha Meudt, Martin Schels, and Miriam Schmidt

Ulm University, Institute of Neural Information Processing
89069 Ulm, Germany
friedhelm.schwenker@uni-ulm.de

**Abstract.** In this paper a novel approach to fuzzy support vector machines (SVM) in multi-class classification problems is presented. The proposed algorithm has the property to benefit from fuzzy labeled data in the training phase and can determine fuzzy memberships for input data. The algorithm can be considered as an extension of the traditional multi-class SVM for crisp labeled data, and it also extents the fuzzy SVM approach for fuzzy labeled training data in the two-class classification setting. Its behavior is demonstrated on three benchmark data sets, the achieved results motivate the inclusion of fuzzy labeled data into the training set for various tasks in pattern recognition and machine learning, such as the design of aggregation rules in multiple classifier systems, or in partially supervised learning.

## 1 Introduction

In real-world applications such as medical diagnosis or affective computing in an human-computer interaction scenario, the ground truth of the collected data is not always clearly defined, and even human experts have their difficulties to find a correct and unique class label, thus, labeling the collected data in such scenarios is not only expensive and time consuming [4], actually, in some cases it might be impossible to assign a unique label [10]. For instance, when asking a group of medical doctors one by one to categorize the status of a patient, these experts may disagree on the correct class label. Leaving out all such data when designing a training set may lead to small training sets and to classifiers of limited performance. One possible approach to avoid this, is to include all data into the training set, and to express the uncertainty of the class information in terms of fuzzy labels, so that a training set may be given by

$$\mathcal{S} = \big\{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in \mathbb{R}^d,\ \mathbf{y}_i \in \Delta^L,\ i = 1, \ldots, m\big\}$$

where $L$ is the number of classes and $\Delta^L = \{\mathbf{y} \in [0,1]^L \mid \sum_{j=1}^{L} y_j = 1\}$ is the set of possible fuzzy memberships. Components of $\mathbf{y}_i \in \Delta^L$ are interpreted as class memberships to the $L$ classes. In this paper the aim is to demonstrate how fuzzy

N. El Gayar et al. (Eds.): ANNPR 2014, LNAI 8774, pp. 153–164, 2014.

memberships can be incorporated into the overall learning process of support vector machines in multi-class classification.

The paper is organized in the following way: In Section 2 we review the standard SVM approach for binary classification (in Section 2.1) and the multi-class classification SVM (in Section 2.3). In Section 2.2) we report our previous work on two-class fuzzy-input fuzzy-output SVM ($F^2SVM$) (see [12]), and in Section 2.4 the $F^2SVM$ approach is extended to the multi-class classification setting. In Section 3 we present a statistical evaluation of fuzzy SVM on three data sets (two artificial and one from optical character recognition), finally we conclude in Section 4.

## 2 SVM Learning with Fuzzy Labels

### 2.1 Review on Binary SVM Classification

Basic principles of SVM classification will be introduced before we consider fuzzy SVM. Binary classification with crisp labels is the starting point for further investigations on learning from fuzzy labeled data sets. In the crisp classification framework, any given observation $\mathbf{x} \in X$ is associated with a corresponding target label $y \in Y$. It is assumed that $X$ is a compact subset of a real-valued vector space (i.e., $X \subseteq \mathbb{R}^d$), and that $Y = \{y_1, \ldots, y_L\}$ is the set of $L$ class labels. The training set is given by

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) \,|\, \mathbf{x}_i \in X, \mathbf{y}_i \in Y,\ i = 1, \ldots, m\}$$

In case of binary SVM classification we have $y_i \in \{-1, 1\}$. An introduction to SVM may be found in [13] or [1]. A generalized linear discriminant function with a fixed nonlinear transformation $\Phi : X \mapsto X'$

$$f(x) = \operatorname{sgn}\left(\mathbf{w}^T \Phi(\mathbf{x}) + w_0\right) \tag{1}$$

classifies all data points correctly if the following conditions are satisfied

$$y_i(\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0) \geq 1 \quad i = 1, \ldots, m. \tag{2}$$

Here $\mathbf{w}$ is a weight vector in $X'$ and $w_0 \in \mathbb{R}$ is a bias parameter. The distance of the transformed data points $\Phi(\mathbf{x}_i)$ to the separating hyperplane $H_{\mathbf{w},w_0} := \{\mathbf{x} \in X | \mathbf{w}^T \Phi(\mathbf{x}) + w_0 = 0\}$ is given by $1/\|\mathbf{w}\|_2$. In order to maximize this distance that is the *margin* between the data points and the separating hyperplane, we seek for a solution that is minimizing the cost function $\varphi(\mathbf{w}) := \|\mathbf{w}\|_2^2/2 = \mathbf{w}^T\mathbf{w}/2$ under the constraints given in Eq. (2). The original SVM the optimization problem is then formulated as *primal form*:

$$L_P(\mathbf{w}, w_0, \alpha) = \frac{\mathbf{w}^T\mathbf{w}}{2} - \sum_{i=1}^{m} \alpha_i(y_i(\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0) - 1) \tag{3}$$

with Lagrange multipliers $\alpha_i \geq 0$, $i = 1, \ldots, m$. Differentiating $L_P$ with respect to $\mathbf{w}$ and $w_0$ leads to the conditions $\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \Phi(\mathbf{x}_i)$ and $\sum_{i=1}^m \alpha_i y_i = 0$, respectively. Substituting these conditions in Equation (3) leads to the dual form

$$L_D(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \tag{4}$$

which must be maximized with respect to the constraints $\alpha_i \geq 0$, $i = 1, \ldots, m$ and $\sum_{i=1}^m \alpha_i y_i = 0$.

Once the multipliers $\alpha_i \geq 0$ have been computed, the weight vector is given by

$$\mathbf{w} = \sum_{i \in \mathcal{SV}} \alpha_i y_i \Phi(\mathbf{x}_i), \tag{5}$$

were $\mathcal{SV}$ is the set of indices of data points with $\alpha_j \neq 0$, the support vectors. From the Karush-Kuhn-Tucker conditions $\alpha_j(y_j(\mathbf{w}^T \Phi(\mathbf{x}_j) + w_0) - 1) = 0$, $i = 1, \ldots, m$, the value $w_0$ can be determined by averaging over all support vector equations, with $\alpha_j > 0$:

$$\sum_{j \in \mathcal{SV}} y_j (\mathbf{w}^T \Phi(\mathbf{x}_j) + w_0) = |\mathcal{SV}| =: n_{\mathcal{SV}} \tag{6}$$

and therefore

$$w_0 = \frac{1}{n_{\mathcal{SV}}} \left( \sum_{j \in \mathcal{SV}} y_j - \sum_{j \in \mathcal{SV}} \sum_{i \in \mathcal{SV}} \alpha_i y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \right). \tag{7}$$

The discriminant function is then determined by substituting Eqs. (5) and (7) into the discriminant function (1).

Since the separations constraints in Eq. (2) can not be fulfilled in realistic data sets they can be relaxed by introducing slack-variables $\xi_i$, $i = 1, \ldots, m$:

$$\begin{aligned} \mathbf{w}^T \Phi(\mathbf{x}_i) + w_0 &\geq 1 - \xi_i \quad \text{for } \; y_i = 1 \\ \mathbf{w}^T \Phi(\mathbf{x}_i) + w_0 &\leq -1 + \xi_i \quad \text{for } \; y_i = -1 \\ \xi_i \geq 0 \quad i &= 1, \ldots, m. \end{aligned} \tag{8}$$

These soft-constraints are incorporated into the cost function $\varphi(\mathbf{w})$ by adding $C \sum_{i=1}^m \xi_i$, with a positive regularization parameter $C > 0$,

$$\varphi(\mathbf{w}, \xi) := \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^m (\xi_i^+ + \xi_i^-). \tag{9}$$

The primal form is defined through

$$L_P(\mathbf{w}, w_0, \xi, \alpha, \mathbf{r}) = \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i (y_i (\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0) - 1 + \xi_i) - \sum_{i=1}^m r_i \xi_i \tag{10}$$

here $r_i \geq 0$ and $\alpha_i \geq 0$ are the Lagrange multipliers. Differentiating $L_P(\mathbf{w}, w_0, \xi, \alpha, \mathbf{r})$ with respect to $\mathbf{w}$ and $w_0$ leads again to $\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \Phi(\mathbf{x}_i)$ and $\sum_{i=1}^m \alpha_i y_i = 0$, differentiating with respect to $\xi_i$ gives the equations $C - \alpha_i - r_i = 0$, $i = 1, \ldots, m$. Substituting them into Eq. (10) yields the dual form:

$$L_D(\alpha, \mathbf{r}) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \tag{11}$$

with constraints $C \geq \alpha_i \geq 0$, $i = 1, \ldots, m$ and $\sum_{i=1}^m \alpha_i y_i = 0$. Here the upper bound $C \geq \alpha_i$ derived from the equations $C - \alpha_i - r_i = 0$.

The bias term $w_0$ can be computed as in Eq. (7) by averaging over all support vector equations satisfying $0 < \alpha_j < C$.

At this point it should be mentioned that the optimization of (11) as well as the discriminating function relies only on dot products $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ which can be replaced in many cases by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$. This so-called kernel-trick makes the use of SVM very appealing.

### 2.2 Fuzzy SVM for the Two Class Classification Problem

In the fuzzy-input fuzzy-output Support Vector Machine fuzzy class memberships of the training data are used during training, and a fuzzy output is generated by using a logistic function [9,8,12]. For instance, in a two class classification problem, the class memberships $y_i^+$ and $y_i^- := (1 - y_i^+)$ for a data point $\mathbf{x}_i$ are incorporated in the SVM training in the following way.

$$\varphi(\mathbf{w}, \xi) := \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^m (\xi_i^+ y_i^+ + \xi_i^- y_i^-) \tag{12}$$

using slack variables $\xi_i^-, \xi_i^+$ and constraints

$$\begin{gathered}
\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0 \geq 1 - \xi_i^+ \quad i = 1, \ldots, m \\
\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0 \leq -1 + \xi_i^- \quad i = 1, \ldots, m \\
\xi_i^- \geq 0, \quad \xi_i^+ \geq 0 \quad i = 1, \ldots, m
\end{gathered} \tag{13}$$

as in Eq.(8). This yields the primal form

$$\begin{aligned}
L_P(\mathbf{w}, w_0, \xi, \alpha, \mathbf{r}) = & \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^m (\xi_i^+ y_i^+ + \xi_i^- y_i^-) \\
& - \sum_{i=1}^m \alpha_i^+ (\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0 - 1 + \xi_i^+) - \sum_{i=1}^m r_i^+ \xi_i^+ \\
& + \sum_{i=1}^m \alpha_i^- (\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0 + 1 - \xi_i^-) - \sum_{i=1}^m r_i^- \xi_i^-
\end{aligned} \tag{14}$$

Differentiation of $L_P(\mathbf{w}, w_0, \xi, \alpha, \mathbf{r})$ with respect to $\mathbf{w}$ and $w_0$ yields

$$\mathbf{w} = \sum_{i=1}^{m}(\alpha_i^+ - \alpha_i^-)\Phi(\mathbf{x}_i) \quad \textbf{and} \quad \sum_{i=1}^{m}(\alpha_i^+ - \alpha_i^-) = 0,$$

and differentiation with respect to $\xi^+$ and $\xi^-$ gives $Cy_i^+ - r_i^+ - \alpha_i^+ = 0$ and $Cy_i^- - r_i^- - \alpha_i^- = 0$ for $i = 1, \ldots, m$. Thus the dual form is given by

$$L_D(\alpha) = \sum_{i=1}^{m}\alpha_i^+ + \sum_{i=1}^{m}\alpha_i^- - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i^+ - \alpha_i^-)\,(\alpha_j^+ - \alpha_j^-)\Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j) \quad (15)$$

subject to

$$\sum_{i=1}^{m}(\alpha_i^+ - \alpha_i^-) = 0, \quad \text{and} \quad 0 \le \alpha_i^+ \le Cy_i^+, \quad 0 \le \alpha_i^- \le Cy_i^-, \quad i = 1, \ldots, m.$$

The fuzzy SVM approach given in Eq. (12), (14), and (15) reduces to the crisp SVM Eq. (9), (10), and (11), in case of crisp labeled data.

### 2.3 Multi-class SVM for Crisp Labeled Data

The support vector optimization approach has been applied to the multi-class classification scenario, see for example [13,5,6,2]. In the case of $L$ classes one is considering discriminant functions

$$f_l(x) = \operatorname{sgn}\left(\mathbf{w}_l^T\Phi(\mathbf{x}) + w_{0l}\right) \quad l = 1, \ldots, L \quad (16)$$

with the aim to compute $\mathbf{w}_l^T$ and $w_{0l}$ for $l = 1, \ldots, L$ such that by using the **argmax**-decision rule the training data is separated without error. The **argmax**-decision rule says that a data point $\mathbf{x}$ is assigned to class $\omega$ if $\omega = \operatorname{argmax}_l f_l(\mathbf{x})$. Such a solution satisfies the crisp separation conditions

$$\mathbf{w}_k^T\Phi(\mathbf{x}_i) + w_{0k} - (\mathbf{w}_l^T\Phi(\mathbf{x}_i) + w_{0l}) \ge 1 \quad (17)$$

for all data points $\mathbf{x}_i$ where data point $\mathbf{x}_i$ is from class $k$ (denoted by $\mathbf{x}_i \in C_k$), and for all classes $l \in \{1, \ldots, L\}$ with $l \ne k$. The maximal margin solution is then computed by minimizing the cost function

$$\varphi(\mathbf{w}_1, \ldots, \mathbf{w}_L) = \frac{1}{2}\sum_{k=1}^{L}\mathbf{w}_k^T\mathbf{w}_k \quad (18)$$

For non-separable classification problems slack-variables $\xi_i^{k,l}$ for all data points $i = 1, \ldots, m$, and or all classes $l = 1, \ldots, L$ with $l \ne k$ are introduced into the separation constraints. This leads to pairwise soft-constraints:

$$(\mathbf{w}_k^T\Phi(\mathbf{x}_i) + w_{0k}) - (\mathbf{w}_l^T\Phi(\mathbf{x}_i) + w_{0l}) \ge 1 - \xi_i^{k,l} \quad (19)$$

for all data points $\mathbf{x}_i$ from class $k_i$, and for all classes $j \neq k_i$. These slack-variables $\xi_i^{k_i,l}$ are then introduced into the cost function:

$$\varphi(\mathbf{w}_1, \ldots, \mathbf{w}_L) = \frac{1}{2}\sum_{k=1}^{L} \mathbf{w}_k^T \mathbf{w}_k + C \sum_{k=1}^{L} \sum_{l=1, l\neq k}^{L} \sum_{\mathbf{x}_i \in C_k} \xi_i^{k,l} \tag{20}$$

this leads to a primal from, which is then the starting point for further developments, e.g. derivation of the dural form. We stop at this point and will provide more details for the multi-class SVM in in the fuzzy multi-class setting.

### 2.4 Fuzzy Multi-class SVM

In the next step we consider the multi-class classification problem with fuzzified class labels, here it is assumed that a training data set is given

$$\mathcal{S} = \big\{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in R^d,\ \mathbf{y}_i \in \Delta^L,\ i = 1, \ldots, m\big\}$$

where $\Delta^L = \{\mathbf{y} \in [0,1]^L \mid \sum_{j=1}^{L} y^j = 1\}$ and $L$ is the number of classes. Following the idea of the two-class fuzzy SVM [8,12] we incorporate the fuzzy class memberships into the cost function in the following form.

For any data point $\mathbf{x}_i$ the values of membership vector $\mathbf{y}_i$ are considered, we assume that they are given in descending order $y_i^{k_1} \geq y_i^{k_2} \cdots \geq y_i^{k_L}$. Fuzzy memberships can be incorporated into the multi-class optimization procedure by pairwise constraints in the following way: For a given data points $\mathbf{x}_i$ and all classes such that $j \neq k (= k_1)$ ($k = k_1$ denotes the class with the largest class membership for $\mathbf{x}_i$) the following constraints are introduced:

$$(\mathbf{w}_k^T \Phi(\mathbf{x}_i) + w_{0k}) - (\mathbf{w}_j^T \Phi(\mathbf{x}_i) + w_{0j}) \geq 1 - \xi_i^{k,j} \tag{21}$$

Overall, for each data point $L-1$ constraints are defined, so $m(L-1)$ constraints in total. The fuzzy memberships can be introduced directly into the cost function:

$$\varphi(\mathbf{w}, \xi) = \frac{1}{2}\sum_{k=1}^{L} \mathbf{w}_k^T \mathbf{w}_k + C \sum_{k=1}^{L} \sum_{\mathbf{x}_i \in C_k} \sum_{l=1, l\neq k}^{L} \xi_i^{k,l}(y_i^k - y_i^l) \tag{22}$$

note that $y_i^k - y_i^l \geq 0$ for all possible combinations, because $k$ denotes the class with the highest membership for data point $\mathbf{x}_i$. The primal form of the fuzzy multi-class SVM problem is then given by

$$\begin{aligned} L_P(\mathbf{w}, w_0, \xi, \alpha, \mathbf{r}) = {} & \frac{1}{2}\sum_{k=1}^{L} \mathbf{w}_k^T \mathbf{w}_k \\ & + C \sum_{k=1}^{L} \sum_{\mathbf{x}_i \in C_k} \sum_{l=1, l\neq k}^{L} \xi_i^{k,l}(y_i^k - y_i^l) - \sum_{k=1}^{L} \sum_{\mathbf{x}_i \in C_k} \sum_{l=1, l\neq k}^{L} \xi_i^{k,l} r_i^{k,l} \\ & + \sum_{k=1}^{L} \sum_{\mathbf{x}_i \in C_k} \sum_{l=1, l\neq k}^{L} \alpha_i^{k,l}(1 - \xi_i^{k,l} - ((\mathbf{w}_k^T \Phi(\mathbf{x}_i) + w_{0k}) - (\mathbf{w}_l^T \Phi(\mathbf{x}_i) + w_{0l}))) \end{aligned} \tag{23}$$

Considering the largest class membership is just one possible approach for the fuzzy multi-class classification scenario. Another way to to take advantage from the class member ships is to define a constraint for each pair of classes $k_p, k_q$ with $y_i^{k_p} \geq y_i^{k_q}$. But this yields $L(L-1)/2$ constraints per data point, so overall $mL(L-1)/2$ constraints. Differentiating with respect to $\mathbf{w}_k^T$ and $w_{0k}$ gives the same constraints for the crisp multi-class classification case.

Differentiating with respect to $\mathbf{w}_k^T$ gives

$$\mathbf{w}_k^T = \underbrace{\sum_{l=1, l\neq k}^{L} \overbrace{\left( \sum_{\mathbf{x}_i \in C_k} \alpha_i^{k,l} \Phi(\mathbf{x}_i) \right)}^{=:u_k^l}}_{=:u_k} - \underbrace{\sum_{l=1, l\neq k}^{L} \overbrace{\left( \sum_{\mathbf{x}_i \in C_l} \alpha_i^{l,k} \Phi(\mathbf{x}_i) \right)}^{=:v_l^k}}_{=:v^k} \qquad k = 1, \ldots, L. \tag{24}$$

Differentiating with respect to $w_{0k}$ leads to

$$0 = \sum_{l=1, l\neq k}^{L} \sum_{\mathbf{x}_i \in C_k} \alpha_i^{k,l} - \sum_{l=1, l\neq k}^{L} \sum_{\mathbf{x}_i \in C_l} \alpha_i^{l,k} \qquad k = 1, \ldots, L. \tag{25}$$

Differentiation with respect to $\xi_i^{k,l}$ gives the conditions

$$C(y_i^k - y_i^l) - \alpha_i^{k,l} - r_i^l = 0 \quad \text{for} i = 1, \ldots, m \text{with} l = 1, \ldots, L \text{and} l \neq k. \tag{26}$$

or as re-formulated as conditions to the $\alpha_i^{k,l}$

$$C(y_i^k - y_i^l) \leq \alpha_i^{k,l} \leq 0 \quad \text{for} i = 1, \ldots, m \text{with} l = 1, \ldots, L \text{and} l \neq k. \tag{27}$$

Now, substitution all these conditions and using shortcuts $u_k$, $v^k$ for $k = 1, \ldots, L$ and $u_k^l$, $v_l^k$ for $k = 1, \ldots, L$ and $l = 1, \ldots, L$ $l \neq k$ and $u_l^k = v_l^k$ yields to the corresponding dual from.

$$\begin{aligned} L_D(\alpha) = & \sum_{k=1}^{L} \sum_{l=1, l\neq k}^{L} \sum_{\mathbf{x}_i \in C_k} \alpha_i^{k,l} \\ & - \frac{1}{2} \sum_{k=1}^{L} \left( (u_k)^T u_k + (v^k)^T v^k \right) - \sum_{k=1}^{L} (v^k) u_k \end{aligned} \tag{28}$$

here dot products given through the following equations.

$$(u_k)^T u_k = \sum_{l=1, l\neq k}^{L} \sum_{\tilde{l}=1, \tilde{l}\neq k}^{L} \sum_{\mathbf{x}_i \in C_k} \sum_{\mathbf{x}_j \in C_k} \alpha_i^{k,l} \alpha_j^{k,\tilde{l}} \left( \Phi(\mathbf{x}_i) \right)^T \Phi(\mathbf{x}_j) \tag{29}$$

$$(v^k)^T v^k = \sum_{l=1, l\neq k}^{L} \sum_{\tilde{l}=1, \tilde{l}\neq k}^{L} \sum_{\mathbf{x}_i \in C_l} \sum_{\mathbf{x}_j \in C_{\tilde{l}}} \alpha_i^{l,k} \alpha_j^{\tilde{l},k} \left( \Phi(\mathbf{x}_i) \right)^T \Phi(\mathbf{x}_j) \tag{30}$$

$$(v^k)^T u_k = \sum_{l=1,l\neq k}^{L} \sum_{\tilde{l}=1,\tilde{l}\neq k}^{L} \sum_{\mathbf{x}_i\in C_l} \sum_{\mathbf{x}_j \in C_k} \alpha_i^{l,k}\alpha_j^{k,\tilde{l}} \left(\Phi(\mathbf{x}_i)\right)^T \Phi(\mathbf{x}_j) \tag{31}$$

The dual form (28) has to be maximized with respect to the constraints (25) and (27).

## 3 Numerical Evaluation on Benchmark Data Sets

### 3.1 Data Sets

In this section the numerical evaluation of the proposed fuzzy SVM approach is presented on a realistic benchmark data set consisting of 20,000 hand-written digits (2,000 instances for each class). These digits, normalized in height and width, are represented through a 16 × 16 matrix $G$ where the entries $G_{ij} \in \{0, \ldots, 255\}$ are values taken from an 8 bit gray scale, see Figure 1. Previously, this data set has been used for the evaluation of machine learning techniques in the Statlog project and many other studies (see for instance [11]).

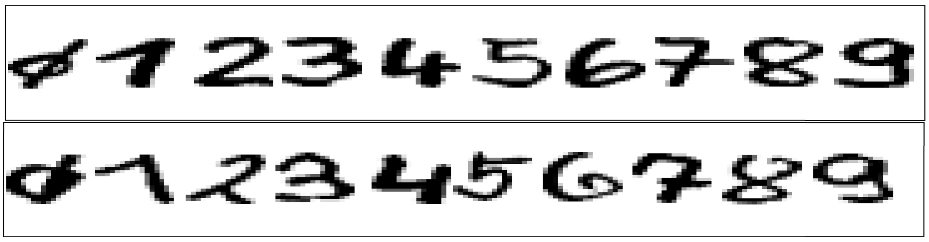

**Fig. 1.** Data set of hand-written digits. Each instance given through a 16 × 16 gray scale image (8-bit resolution).

In order to control the degree of fuzziness in the numerical experiments two different types of data sets have been prepared. For this, we define the ball of radius $r$ in $\mathbb{R}^d$ in $l_1$-norm by $B_d^1(r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_1 := \sum_{i=1}^d |x_i| \leq r\}$.

Data set **A** has been sampled according to the uniform distribution of set $B_2^1(2)$ and fuzzy labels for the data points are assigned in the following way: Given an instance $\mathbf{x} = (x_1, x_2) \in B_2^1(2)$ then its corresponding fuzzy class label $l$ is set to the following two-dimensional vector, representing the memberships of the two classes:

$$l := \left( \frac{e^{d(x_1+x_2)}}{1+e^{d(x_1+x_2)}}, \frac{1}{1+e^{d(x_1+x_2)}} \right).$$

The parameter $d \geq 0$ is used to control the degree of overlap between the data of the two classes: For small values of parameter $d$ the classes are overlapping, and for increasing $d$-values the data of the two classes becomes more more and more separated, thus $d$ is reflecting the distance between the data of the class distributions. This data set is used to demonstrate how the fuzzy SVM works in case of weak class memberships.

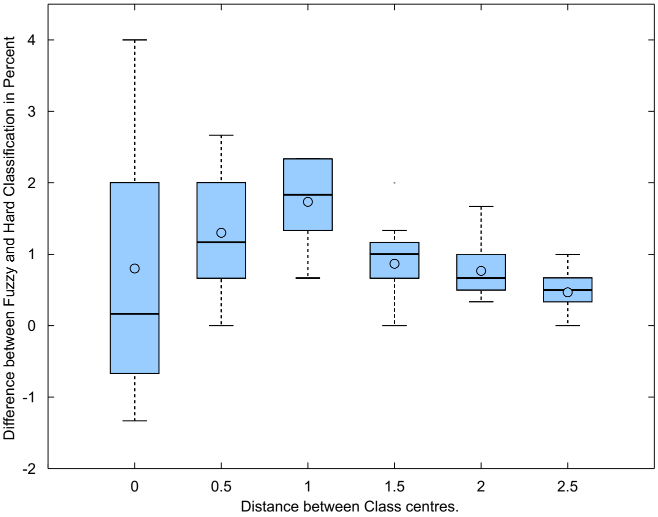

**Fig. 2.** Results for the artificially generated data set **A**, shown are differences between classification accuracy of crisp and fuzzy SVM for different values of distances $d$ (see text). A box plot shows the difference of classification accuracy between fuzzy SVM and standard SVM; positive difference means that fuzzy SVM performs better than standard SVM. For medium class overlap fuzzy labels are beneficial; for well separated classes ($d = 2.5$) and for highly overlapping classes ($d = 0$) the SVM can not benefit from the fuzzy labels.

Data set **B** is a four-class data set, and has been generated by four bi-variate Gaussian distributions of spherical shape ($\sigma^2 = 1$ in both directions), where each distribution is located in one of the four corners of $B_2^1(2)$. The fuzzy labels are generated by data clustering and fuzzification of the prototypes. The data set is used to show how data set reduction by vector quantization and prototype fuzzification can be applied in classification tasks of big data sets by utilizing fuzzy SVM.

Learning classifiers in a *big data* application is a time consuming task, and thus, instance selection or vector quantization might help to reduce the overall complexity. Clustering or vector quantization are common approaches to compute a small set of representative prototypes out of a larger data set. We applied fuzzy c-means clustering algorithm to compute representative prototypes, followed by Keller fuzzification [7] of the prototypes, then, the result of the procedure is a small set of prototypes, where each prototype has a fuzzy label derived from the crisp labels weighted by the cluster membership of the data points.

### 3.2 Numerical Results

First we present results for the artificial data set **A**. In Figure 2 classification results for standard SVM and fuzzy SVM are presented for different settings of the distance parameter $d$. A box plot shows the difference of the accuracy between fuzzy SVM and standard SVM, so positive values stand for the situation where the fuzzy SVM shows higher classification accuracy.

For very small $d$-values ($d = 0$) the data is hard to classify, both classifiers show the same, but very low accuracy. For small distance values $d$ ($x$-axis) the data is highly meshed and classification by using fuzzy labels and fuzzy SVM provides far better accuracy then crisp labels with standard SVM. All in all the fuzzy classifier works far better when the label of the data is weak and hard classifiers work better in the case of strong signals.

Result of data set **B** are given in Figure 3. Here the superior classification performance of fuzzy SVM in comparison to standard SVM using crisp labels is shown in settings where the data set is reduced to very few prototypes. The results were obtained by calculating fuzzy C-means and Keller fuzzification on the dataset to obtain fuzzy labels and after that the samples were reduced to a fraction of the normal size.

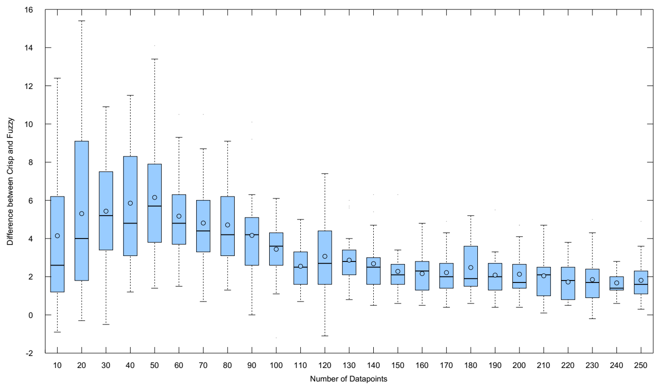

**Fig. 3.** Results for the artificially generated data set **B**, shown are differences between classification accuracy of crisp and fuzzy SVM for different numbers of prototypes $p = 10, \ldots, 250$. A box plot shows the difference of classification accuracy between fuzzy SVM and standard SVM; positive difference means that fuzzy SVM performs better than standard SVM. Fuzzy SVM using fuzzy labels is beneficial for a wide range of degree of data reduction.

Similar behavior of the classification performance can be observed in the digit dataset (see Figure 4 for the results). It shows the same behavior as dataset **B**

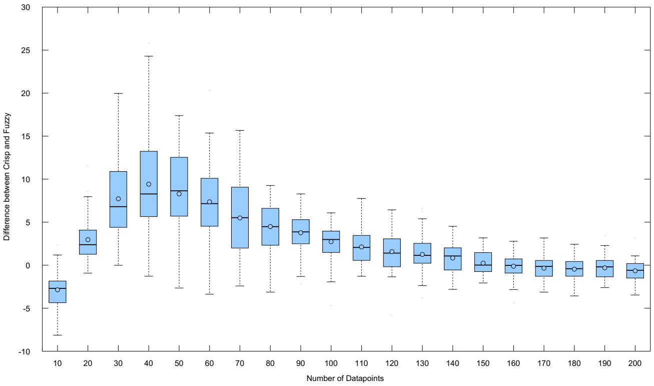

**Fig. 4.** Results for the digit data set, shown are differences between classification accuracy of crisp and fuzzy SVM for different numbers of prototypes $p = 10, \ldots, 250$. A box plot shows the difference of classification accuracy between fuzzy SVM and standard SVM; positive difference means that fuzzy SVM performs better than standard SVM. Fuzzy SVM using fuzzy labels is beneficial for a wide range of degree of data reduction.

in which for very few data samples the fuzzy approach has better generalization compared to the crisp one. As described above the dataset contains 256 features, corresponding to a grayscale image of a digit. The results were obtained by fuzzification of the labels with the fuzzy-c-means method and for labels which switched class we calculated the Keller algorithm. The digit data set is a real milt-class classification benchmark where a sub set of data points are difficult to classify, e.g. for instance data from the classes $0, 3, 8$ or 9.

## 4 Conclusion

We proposed a new SVM approach dealing with fuzzy or soft labels in multi-class classification applications. In contrast to other multi-class approaches we introduced a new technique where the fuzzy memberships of all classes are incorporated in an overall cost function. To gain results between the crisp and the fuzzy SVM we considered three datasets, in which two are artificial datasets. As shown above in dataset one the fuzzy approach has a better accuracy than the crisp one for some places were the signal level is weak. This could be helpful in cases where the crisp SVM has problems figuring out the seperation between classes. Furthermore the fuzzy SVM classifier has advantages over the crisp SVM in applications with very few samples as shown in the results for dataset 2 and 3. In these cases a good fuzzification approach can lead to better accuracy because each data point is optimized for each class present. This could also be useful for high dimension low sample size data, if the labels are fuzzified in a suitable way. This could happen either by applying the fuzzy-c-means algorithm or by obtaining the fuzzy labels by hand. In our benchmark data sets we could show that by using the fuzzy SVM one can benefit from fuzzy or soft labeled data in scenarios where the recognition accuracies are in intermediate range, this is a promising property for many machine learning applications, such as semi-supervised classification [3], multiple classifier systems, or in general information fusion systems [14].

**Acknowledgements.** The authors are supported by the Transregional Collaborative Research Centre SFB/TRR 62 *Companion-Technology for Cognitive Technical Systems* funded by the German Research Foundation (DFG); Markus Kächele is supported by a scholarship of the *Landesgraduiertenförderung Baden-Württemberg* at Ulm University.

## References

1. Abe, S.: Support Vector Machines for Pattern Classification (Advances in Pattern Recognition). Springer-Verlag New York, Inc., Secaucus (2005)
2. Bordes, A., Bottou, L., Gallinari, P., Weston, J.: Solving multiclass support vector machines with larank. In: Proceedings of the 24th International Conference on Machine Learning, ICML 2007, pp. 89–96. ACM, New York (2007)
3. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning, 1st edn. The MIT Press (2010)
4. Hady, M.F.A., Schwenker, F.: Semi-supervised learning. In: Bianchini, M., Maggini, M., Jain, L.C. (eds.) Handbook on Neural Information Processing. ISRL, vol. 49, pp. 215–239. Springer, Heidelberg (2013)
5. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE Transactions Neural Networks 13(2), 415–425 (2002)
6. Kahsay, L., Schwenker, F., Palm, G.: Comparison of multiclass SVM decomposition schemes for visual object recognition. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 334–341. Springer, Heidelberg (2005)
7. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Transactions on Systems, Man and Cybernetics 4, 580–585 (1985)
8. Lin, C.F., Wang, S.D.: Fuzzy support vector machines. IEEE Transactions on Neural Networks 13(2), 464–471 (2002)
9. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers, pp. 61–74 (1999)
10. Scherer, S., Kane, J., Gobl, C., Schwenker, F.: Investigating fuzzy-input fuzzy-output support vector machines for robust voice quality classification. Computer Speech & Language 27(1), 263–287 (2013)
11. Schwenker, F., Kestler, H.A., Palm, G.: Three learning phases for radial-basis-function networks. Neural Networks 14(4-5), 439–458 (2001)
12. Thiel, C., Scherer, S., Schwenker, F.: Fuzzy-input fuzzy-output one-against-all support vector machines. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part III. LNCS (LNAI), vol. 4694, pp. 156–165. Springer, Heidelberg (2007)
13. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons (1998)
14. Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms. Chapman Hall/CRC (2012)