Fuzzy Support Vector Machines

Chun-Fu Lin and Sheng-De Wang

***Abstract*—A support vector machine (SVM) learns the decision surface from two distinct classes of the input points. In many applications, each input point may not be fully assigned to one of these two classes. In this paper, we apply a fuzzy membership to each input point and reformulate the SVMs such that different input points can make different constributions to the learning of decision surface. We call the proposed method fuzzy SVMs (FSVMs).**

***Index Terms*—Classification, fuzzy membership, quadratic programming, support vector machines (SVMs).**

I. Introduction

THE theory of support vector machines (SVMs) is a new classification technique and has drawn much attention on this topic in recent years [1]–[5]. The theory of SVM is based on the idea of structural risk minimization (SRM) [3]. In many applications, SVM has been shown to provide higher performance than traditional learning machines [1] and has been introduced as powerful tools for solving classification problems.

An SVM first maps the input points into a high-dimensional feature space and finds a separating hyperplane that maximizes the margin between two classes in this space. Maximizing the margin is a quadratic programming (QP) problem and can be solved from its dual problem by introducing Lagrangian multipliers. Without any knowledge of the mapping, the SVM finds the optimal hyperplane by using the dot product functions in feature space that are called *kernels*. The solution of the optimal hyperplane can be written as a combination of a few input points that are called *support vectors*.

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 assinged to one class so that SVM can seperate 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.

In this paper, we apply a fuzzy membership to each input point of SVM and reformulate SVM into fuzzy SVM (FSVM) such that different input points can make different constributions to the learning of decision surface. The proposed method enhances the SVM in reducing the effect of outliers and noises in data points. FSVM is suitable for applications in which data points have unmodeled characteristics.

The rest of this paper is organized as follows. A brief review of the theory of SVM will be described in Section II. The FSVM will be derived in Section III. Three experiments are presented in Section IV. Some concluding remarks are given in Section V.

Manuscript received January 25, 2001; revised August 27, 2001.

C.-F. Lin is with the Department of Electrical Engineering, National Taiwan University, Taiwan (e-mail: genelin@hpc.ee.ntu.edu.tw).

S.-D. Wang is with the Department of Electrical Engineering, National Taiwan University, Taiwan (e-mail: sdwang@hpc.ee.ntu.edu.tw).

Publisher Item Identifier S 1045-9227(02)01807-6.

II. SVMs

In this section we briefly review the basis of the theory of SVM in classification problems [2]–[4].

Suppose we are given a set S of labeled training points

$$(y_1, \mathbf{x}_1), \ldots, (y_l, \mathbf{x}_l). \tag{1}$$

Each training point $\mathbf{x}_i \in \mathcal{R}^N$ belongs to either of two classes and is given a label $y_i \in \{-1, 1\}$ for $i = 1, \ldots, l$. In most cases, the searching of a suitable hyperplane in an input space is too restrictive to be of practical use. A solution to this situation is mapping the input space into a higher dimension feature space and searching the optimal hyperplane in this feature space. Let $\mathbf{z} = \varphi(\mathbf{x})$ denote the corresponding feature space vector with a mapping φ from $\mathcal{R}^N$ to a feature space $\mathcal{Z}$. We wish to find the hyperplane

$$\mathbf{w} \cdot \mathbf{z} + b = 0 \tag{2}$$

defined by the pair $(\mathbf{w}, b)$, such that we can separate the point $\mathbf{x}_i$ according to the function

$$f(\mathbf{x}_i) = \operatorname{sign}(\mathbf{w} \cdot \mathbf{z}_i + b) = \begin{cases} 1, & \text{if } y_i = 1 \\ -1, & \text{if } y_i = -1 \end{cases} \tag{3}$$

where $\mathbf{w} \in \mathcal{Z}$ and $b \in \mathcal{R}$.

More precisely, the set S is said to be *linearly separable* if there exist $(\mathbf{w}, b)$ such that the inequalities

$$\begin{cases} (\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1, & \text{if } y_i = 1 \\ (\mathbf{w} \cdot \mathbf{z}_i + b) \leq -1, & \text{if } y_i = -1, \end{cases} \quad i = 1, \ldots, l \tag{4}$$

are valid for all elements of the set S. For the linearly separable set S, we can find a unique optimal hyperplane for which the margin between the projections of the training points of two different classes is maximized. If the set S is not linearly separable, classification violations must be allowed in the SVM formulation. To deal with data that are not linearly separable, the previous analysis can be generalized by introducing some nonnegative variables $\xi_i \geq 0$ such that (4) is modified to

$$y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l. \tag{5}$$

The nonzero ξ_i in (5) are those for which the point $\mathbf{x}_i$ does not satisfy (4). Thus the term $\sum_{i=1}^{l} \xi_i$ can be thought of as some measure of the amount of misclassifications.

The optimal hyperplane problem is then regraded as the solution to the problem

$$\begin{aligned} &\text{minimize } \frac{1}{2}\mathbf{w}\cdot\mathbf{w}+C\sum_{i=1}^{l}\xi_i \\ &\text{subject to } y_i(\mathbf{w}\cdot\mathbf{z}_i+b)\geq 1-\xi_i, \quad i=1,\ldots,l \\ &\qquad \xi_i\geq 0, \quad i=1,\ldots,l \end{aligned} \tag{6}$$

where C is a constant. The parameter C can be regarded as a regularization parameter. This is the only free parameter in the SVM formulation. Tuning this parameter can make balance between margin maximization and classification violation. Detail discussions can be found in [4], [6].

Searching the optimal hyperplane in (6) is a QP problem, which can be solved by constructing a Lagrangian and transformed into the dual

$$\begin{aligned} &\text{maximize } W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{z}_i\cdot\mathbf{z}_j \\ &\text{subject to } \sum_{i=1}^{l}y_i\alpha_i=0 \quad 0\leq\alpha_i\leq C, \quad i=1,\ldots,l \end{aligned} \tag{7}$$

where $\alpha=(\alpha_1,\ldots,\alpha_l)$ is the vector of nonnegative Lagrange multipliers associated with the constraints (5).

The Kuhn–Tucker theorem plays an important role in the theory of SVM. According to this theorem, the solution $\bar{\alpha}_i$ of problem (7) satisfies

$$\bar{\alpha}_i(y_i(\bar{\mathbf{w}}\cdot\mathbf{z}_i+\bar{b})-1+\bar{\xi}_i)=0, \quad i=1,\ldots,l \tag{8}$$

$$(C-\bar{\alpha}_i)\bar{\xi}_i=0, \quad i=1,\ldots,l. \tag{9}$$

From this equality it comes that the only nonzero values $\bar{\alpha}_i$ in (8) are those for which the constraints (5) are satisfied with the equality sign. The point $\mathbf{x}_i$ corresponding with $\bar{\alpha}_i>0$ is called *support vector*. But there are two types of support vectors in a nonseparable case. In the case $0<\bar{\alpha}_i<C$, the corresponding support vector $\mathbf{x}_i$ satisfies the equalities $y_i(\bar{\mathbf{w}}\cdot\mathbf{z}_i+\bar{b})=1$ and $\xi_i=0$. In the case $\bar{\alpha}_i=C$, the corresponding ξ_i is not null and the corresponding support vector $\mathbf{x}_i$ does not satisfy (4). We refer to such support vectors as errors. The point $\mathbf{x}_i$ corresponding with $\bar{\alpha}_i=0$ is classified correctly and clearly away the decision margin.

To construct the optimal hyperplane $\bar{\mathbf{w}}\cdot\mathbf{z}+\bar{b}$, it follows that

$$\bar{\mathbf{w}}=\sum_{i=1}^{l}\bar{\alpha}_i y_i\mathbf{z}_i \tag{10}$$

and the scalar $\bar{b}$ can be determined from the Kuhn–Tucker conditions (8).

The decision function is generalized from (3) and (10) such that

$$f(\mathbf{x})=\operatorname{sign}(\mathbf{w}\cdot\mathbf{z}+b)=\operatorname{sign}\left(\sum_{i=1}^{l}\alpha_i y_i\mathbf{z}_i\cdot\mathbf{z}+b\right). \tag{11}$$

Since we do not have any knowledge of φ, the computation of problem (7) and (11) is impossible. There is a good property of SVM that it is not necessary to know about φ. We just only need a function $K(\cdot,\cdot)$ called *kernel* that can compute the dot product of the data points in feature space $\mathcal{Z}$, that is

$$\mathbf{z}_i\cdot\mathbf{z}_j=\varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}_j)=K(\mathbf{x}_i,\mathbf{x}_j). \tag{12}$$

Functions that satisfy the Mercer's theorem can be used as dot-products and thus can be used as kernels. We can use the polynomial kernel of degree d

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i\cdot\mathbf{x}_j)^d \tag{13}$$

to consturct a SVM classifier.

Thus the nonlinear separating hyperplane can be found as the solution of

$$\begin{aligned} &\text{maximize } W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \\ &\text{subject to } \sum_{i=1}^{l}y_i\alpha_i=0 \quad 0\leq\alpha_i\leq C, \quad i=1,\ldots,l \end{aligned} \tag{14}$$

and the decision function is

$$f(\mathbf{x})=\operatorname{sign}(\mathbf{w}\cdot\mathbf{z}+b)=\operatorname{sign}\left(\sum_{i=1}^{l}\alpha_i y_i K(\mathbf{x}_i,\mathbf{x})+b\right) \tag{15}$$

III. FSVMs

In this section, we make a detail description about the idea and formulations of FSVMs.

A. Fuzzy Property of Input

SVM is a powerful tool for solving classification problems [1], but there are still some limitataions of this theory. From the training set (1) and formulations discussed above, each training point belongs to either one class or the other. For each class, we can easily check that all training points of this class are treated uniformly in the theory of SVM.

In many real-world applications, the effects of the training points are different. It is often that some training points are more important than others in the classificaiton problem. We would require that the meaningful training points must be classified correctly and would not care about some training points like noises whether or not they are misclassified.

That is, each training point no more exactly belongs to one of the two classes. It may 90% belong to one class and 10% be meaningless, and it may 20% belong to one class and 80% be meaningless. In other words, there is a fuzzy membership $0<s_i\leq 1$ associated with each trainging point $\mathbf{x}_i$. This fuzzy membership s_i can be regarded as the attitude of the corresponding training point toward one class in the classification problem and the value $(1-s_i)$ can be regarded as the attitude of meaningless. We extend the concept of SVM with fuzzy membership and make it an FSVM.

B. Reformulate SVM

Suppose we are given a set S of labeled training points with associated fuzzy membership

$$(y_1, \mathbf{x}_1, s_1), \ldots, (y_l, \mathbf{x}_l, s_l). \tag{16}$$

Each training point $\mathbf{x}_i \in \mathcal{R}^N$ is given a label $y_i \in \{-1, 1\}$ and a fuzzy membership $\sigma \le s_i \le 1$ with $i = 1, \ldots, l$, and sufficient small $\sigma > 0$. Let $\mathbf{z} = \varphi(\mathbf{x})$ denote the corresponding feature space vector with a mapping φ from $\mathcal{R}^N$ to a feature space $\mathcal{Z}$.

Since the fuzzy membership s_i is the attitude of the corresponding point $\mathbf{x}_i$ toward one class and the parameter ξ_i is a measure of error in the SVM, the term $s_i\xi_i$ is a measure of error with different weighting. The optimal hyperplane problem is then regraded as the solution to

$$\begin{aligned} &\text{minimize } \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{l} s_i\xi_i \\ &\text{subject to } y_i(\mathbf{w}\cdot\mathbf{z}_i + b) \ge 1 - \xi_i, \quad i = 1, \ldots, l \\ &\qquad \xi_i \ge 0, \quad i = 1, \ldots, l \end{aligned} \tag{17}$$

where C is a constant. It is noted that a smaller s_i reduces the effect of the parameter ξ_i in problem (17) such that the corresponding point $\mathbf{x}_i$ is treated as less important.

To solve this optimization problem we construct the Lagrangian

$$\begin{aligned} &L(\mathbf{w}, b, \xi, \alpha, \beta) \\ &\quad = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{l} s_i\xi_i \\ &\qquad - \sum_{i=1}^{l}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{z}_i + b) - 1 + \xi_i) - \sum_{i=1}^{l}\beta_i\xi_i \end{aligned} \tag{18}$$

and find the saddle point of $L(\mathbf{w}, b, \xi, \alpha, \beta)$. The parameters must satisfy the following conditions:

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l}\alpha_i y_i \mathbf{z}_i = 0 \tag{19}$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial b} = -\sum_{i=1}^{l}\alpha_i y_i = 0 \tag{20}$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \xi_i} = s_i C - \alpha_i - \beta_i = 0. \tag{21}$$

Apply these conditions into the Lagrangian (18), the problem (17) can be transformed into

$$\begin{aligned} &\text{maximize } W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to } \sum_{i=1}^{l} y_i\alpha_i = 0 \quad 0 \le \alpha_i \le s_i C, \quad i = 1, \ldots, l \end{aligned} \tag{22}$$

and the Kuhn–Tucker conditions are defined as

$$\bar{\alpha}_i(y_i(\bar{\mathbf{w}}\cdot\mathbf{z}_i + \bar{b}) - 1 + \bar{\xi}_i) = 0, \quad i = 1, \ldots, l \tag{23}$$

$$(s_i C - \bar{\alpha}_i)\bar{\xi}_i = 0, \quad i = 1, \ldots, l. \tag{24}$$

The point $\mathbf{x}_i$ with the corresponding $\bar{\alpha}_i > 0$ is called a support vector. There are also two types of support vectors. The one with corresponding $0 < \bar{\alpha}_i < s_i C$ lies on the margin of the hyperplane. The one with corresponding $\bar{\alpha}_i = s_i C$ is misclassified. An important difference between SVM and FSVM is that the points with the same value of $\bar{\alpha}_i$ may indicate a different type of support vectors in FSVM due to the factor s_i.

C. Dependence on the Fuzzy Membership

The only free parameter C in SVM controls the tradeoff between the maximization of margin and the amount of misclassifications. A larger C makes the training of SVM less misclassifications and narrower margin. The decrease of C makes SVM ignore more training points and get wider margin.

In FSVM, we can set C to be a sufficient large value. It is the same as SVM that the system will get narrower margin and allow less miscalssifications if we set all $s_i = 1$. With different value of s_i, we can control the tradeoff of the respective training point $\mathbf{x}_i$ in the system. A smaller value of s_i makes the corresponding point $\mathbf{x}_i$ less important in the training.

There is only one free parameter in SVM while the number of free parameters in FSVM is equivalent to the number of training points.

D. Generating the Fuzzy Memberships

To choose the appropriate fuzzy memberships in a given problem is easy. First, the lower bound of fuzzy memberships must be defined, and second, we need to select the main property of data set and make connection between this property and fuzzy memberships.

Consider that we want to conduct the sequential learning problem. First, we choose $\sigma > 0$ as the lower bound of fuzzy memberships. Second, we identify that the time is the main property of this kind of problem and make fuzzy membership s_i be a function of time t_i

$$s_i = f(t_i) \tag{25}$$

where $t_1 \le \cdots \le t_l$ is the time the point arrived in the system. We make the last point $\mathbf{x}_l$ be the most important and choose $s_l = f(t_l) = 1$, and make the first point $\mathbf{x}_1$ be the least important and choose $s_1 = f(t_1) = \sigma$. If we want to make fuzzy membership be a linear function of the time, we can select

$$s_i = f(t_i) = at_i + b. \tag{26}$$

By applying the boundary conditions, we can get

$$s_i = f(t_i) = \frac{1-\sigma}{t_l - t_1}t_i + \frac{t_l\sigma - t_1}{t_l - t_1}. \tag{27}$$

If we want to make fuzzy membership be a quadric function of the time, we can select

$$s_i = f(t_i) = a(t_i - b)^2 + c. \tag{28}$$

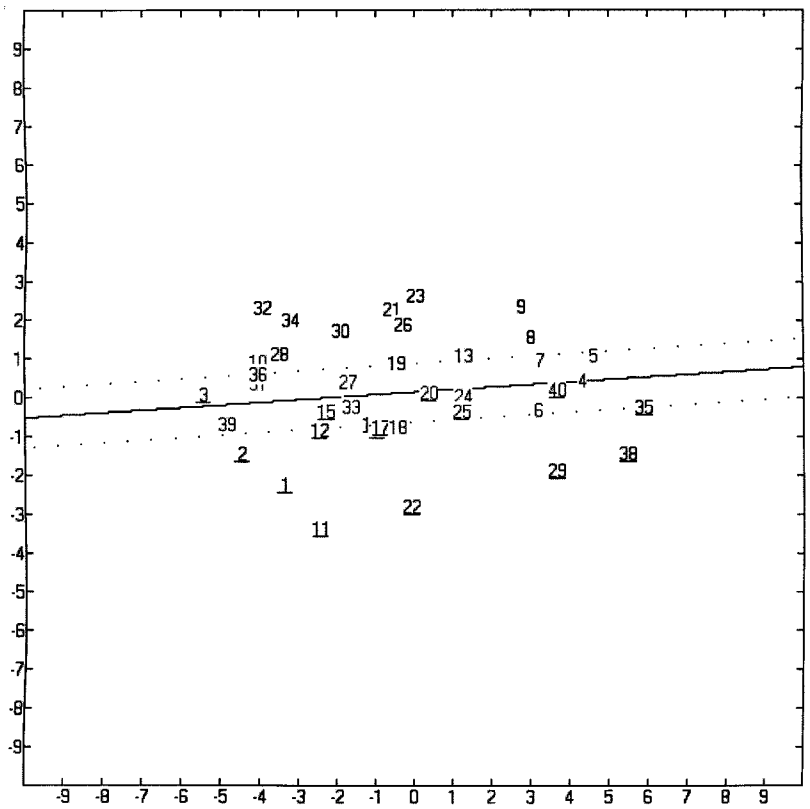

Fig. 1. The result of SVM learning for data with time property.

By applying the boundary conditions, we can get

$$s_i = f(t_i) = (1-\sigma)\left(\frac{t_i - t_1}{t_l - t_1}\right)^2 + \sigma. \tag{29}$$

IV. Experiments

There are many applications that can be fitted by FSVM since FSVM is an extension of SVM. In this section, we will introduce three examples to see the benefits of FSVM.

A. Data With Time Property

Sequential learning and inference methods are important in many applications involving real-time signal processing [7]. For example, we would like to have a learning machine such that the points from recent past is given more weighting than the points far back in the past. For this purpose, we can select the fuzzy membership as a function of the time that the point generated and this kind of problem can be easily implemented by FSVM.

Suppose we are given a sequence of training points

$$(y_1, \mathbf{x}_1, s_1, t_1), \ldots, (y_l, \mathbf{x}_l, s_l, t_l) \tag{30}$$

where $t_1 \leq \cdots \leq t_l$ is the time the point arrived in the system. Let fuzzy membership s_i be a function of time t_i

$$s_i = f(t_i) \tag{31}$$

such that $s_1 = \sigma \leq \cdots \leq s_l = 1$.

Fig. 1 shows the result of the SVM and Fig. 2 shows the result of FSVM by setting

$$s_i = f(t_i) = (1-\sigma)\left(\frac{t_i - t_1}{t_l - t_1}\right)^2 + \sigma. \tag{32}$$

The numbers with underline are grouped as one class and the numbers without underline are grouped as the other class. The value of the number indicates the arrival sequence in the same interval. The smaller numbered data is the older one. We can easily check that the FSVM classifies the last ten points with high accuracy while the SVM does not.

B. Two Classes With Different Weighting

There may be some applications that we just want to focus on the accuracy of classifying one class. For example, given a point, if the machine says 1, it means that the point belongs to this class with very high accuracy, but if the machine says -1, it may belongs to this class with lower accuracy or really belongs to another class. For this purpose, we can select the fuzzy membership as a function of respective class.

Suppose we are given a sequence of training points

$$(y_1, \mathbf{x}_1, s_1), \ldots, (y_l, \mathbf{x}_l, s_l). \tag{33}$$

Let fuzzy membership s_i be a function of class y_i

$$s_i = \begin{cases} s_+, & \text{if } y_i = 1, \\ s_-, & \text{if } y_i = -1. \end{cases} \tag{34}$$

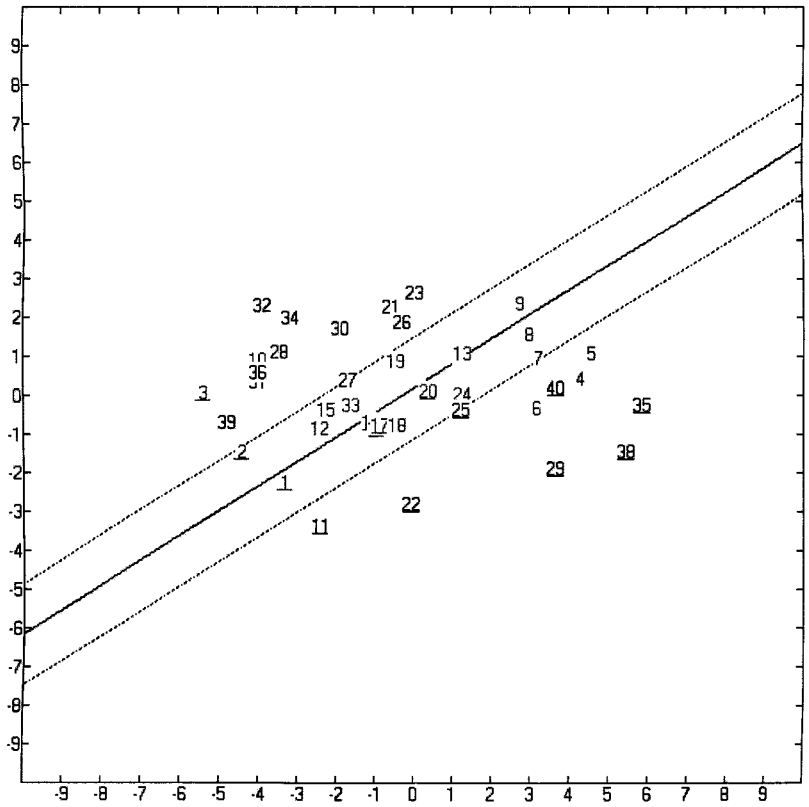

Fig. 2. The result of FSVM learning for data with time property.

Fig. 3 shows the result of the SVM and Fig. 4 shows the result of FSVM by setting

$$s_i = \begin{cases} 1, & \text{if } y_i = 1 \\ 0.1, & \text{if } y_i = -1. \end{cases} \tag{35}$$

The point $\mathbf{x}_i$ with $y_i = 1$ is indicated as cross, and the point $\mathbf{x}_i$ with $y_i = -1$ is indicated as square. In Fig. 3, the SVM finds the optimal hyperplane with errors appearing in each class. In Fig. 4, we apply different fuzzy memberships to different classes, the FSVM finds the optimal hyperplane with errors appearing only in one class. We can easily check that the FSVM classify the class of cross with high accuracy and the class of square with low accuracy, while the SVM does not.

C. Use Class Center to Reduce the Effects of Outliers

Many research results have shown that the SVM is very sensitive to noises and outliners [8], [9]. The FSVM can also apply to reduce to effects of outliers. We propose a model by setting the fuzzy membership as a function of the distance between the point and its class center. This setting of the membership could not be the best way to solve the problem of outliers. We just propose a way to solve this problem. It may be better to choose a different model of fuzzy membership function in different training set.

Suppose we are given a sequence of training points

$$(y_1, \mathbf{x}_1, s_1), \ldots, (y_l, \mathbf{x}_l, s_l). \tag{36}$$

Denote the mean of class $+1$ as $\mathbf{x}_+$ and the mean of class -1 as $\mathbf{x}_-$. Let the radius of class $+1$

$$r_+ = \max_{\{\mathbf{x}_i : y = 1\}} |\mathbf{x}_+ - \mathbf{x}_i| \tag{37}$$

and the radius of class -1

$$r_- = \max_{\{\mathbf{x}_i : y = -1\}} |\mathbf{x}_- - \mathbf{x}_i|. \tag{38}$$

Let fuzzy membership s_i be a function of the mean and radius of each class

$$s_i = \begin{cases} 1 - |\mathbf{x}_+ - \mathbf{x}_i|/(r_+ + \delta) & \text{if } y_i = 1 \\ 1 - |\mathbf{x}_- - \mathbf{x}_i|/(r_- + \delta), & \text{if } y_i = -1 \end{cases} \tag{39}$$

where $\delta > 0$ is used to avoid the case $s_i = 0$.

Fig. 5 shows the result of the SVM and Fig. 6 shows the result of FSVM. The point $\mathbf{x}_i$ with $y_i = 1$ is indicated as cross, and the point $\mathbf{x}_i$ with $y_i = -1$ is indicated as square. In Fig. 5, the SVM finds the optimal hyperplane with the effect of outliers, for example, a square at $(-3.5, 6.6)$ and a cross at $(3.6, -2.2)$. In Fig. 6, the distance of the above two outliers to its corresponding mean is equal to the radius. Since the fuzzy membership is a function of the mean and radius of each class, these two points are regarded as less important in FSVM training such that there is a big difference between the hyperplanes found by SVM and FSVM.

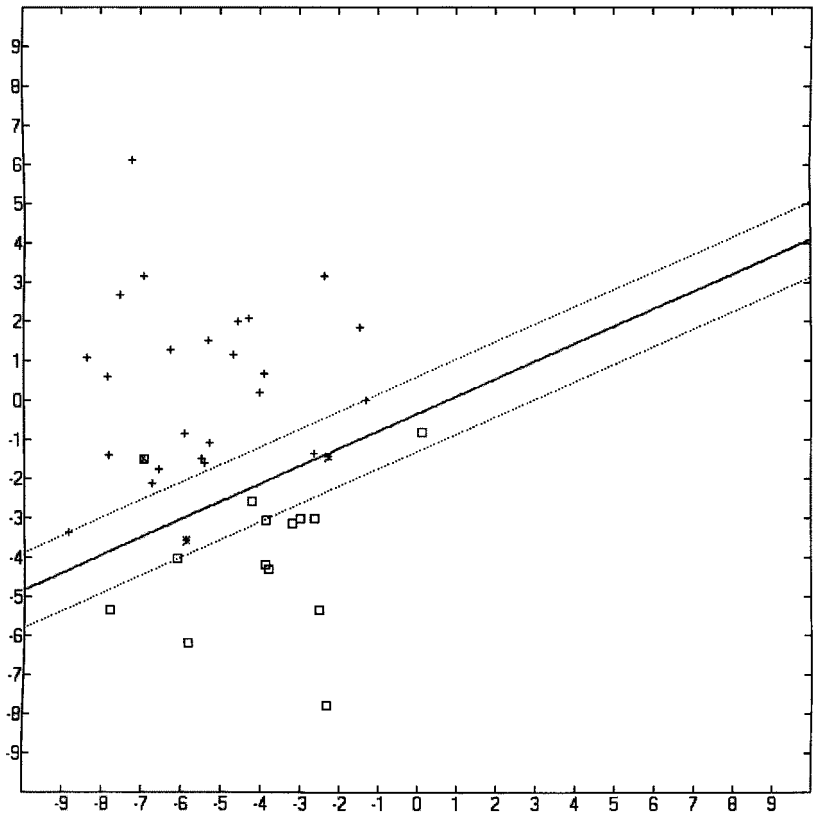

Fig. 3. The result of SVM learning for data sets with different weighting.

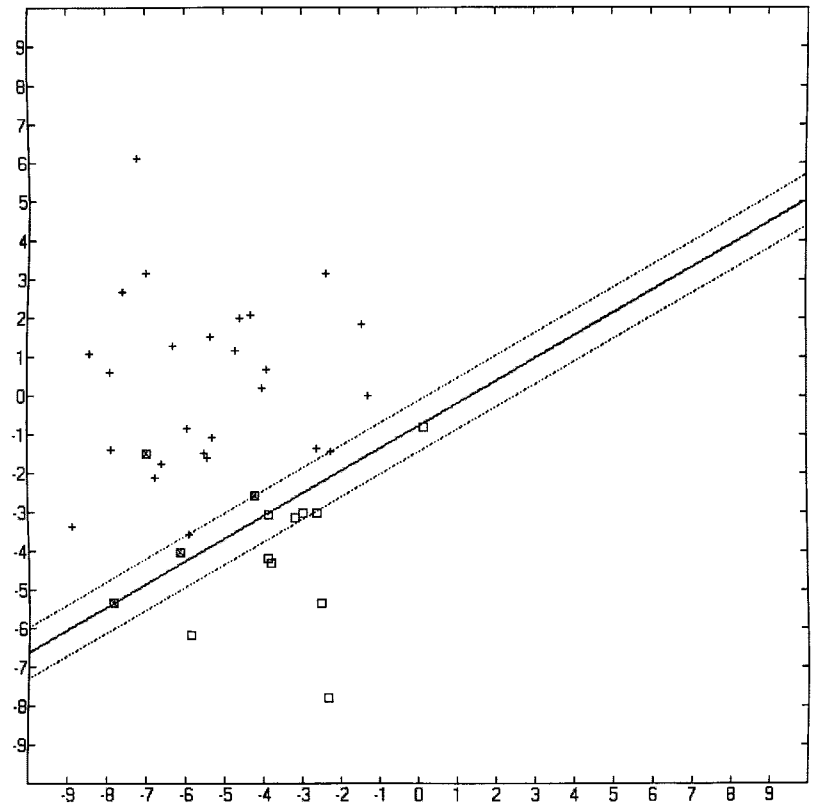

Fig. 4. The result of FSVM learning for data sets with different weighting.

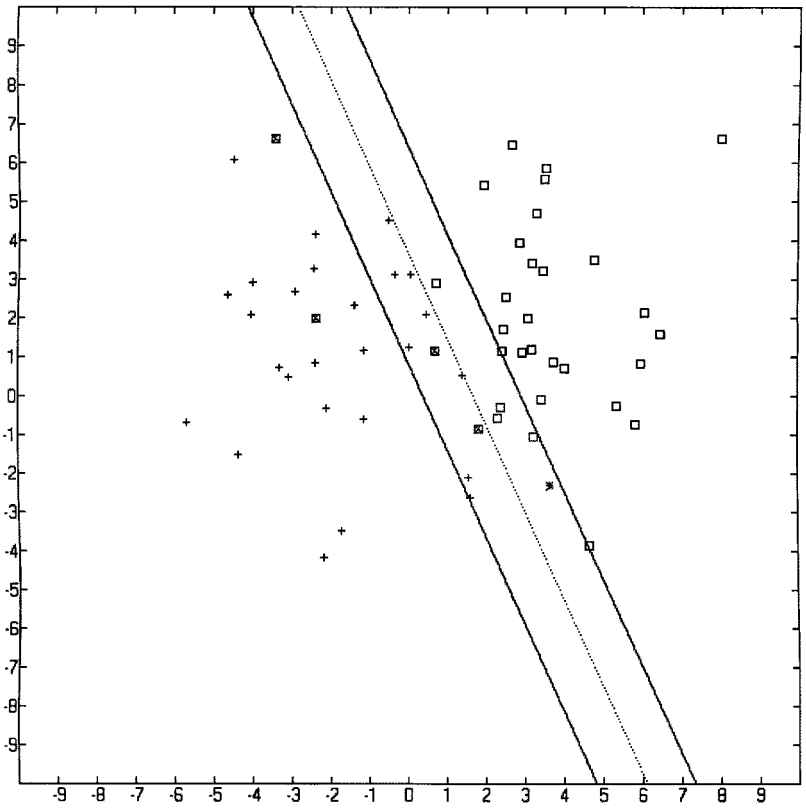

Fig. 5. The result of SVM learning for data sets with outliers.

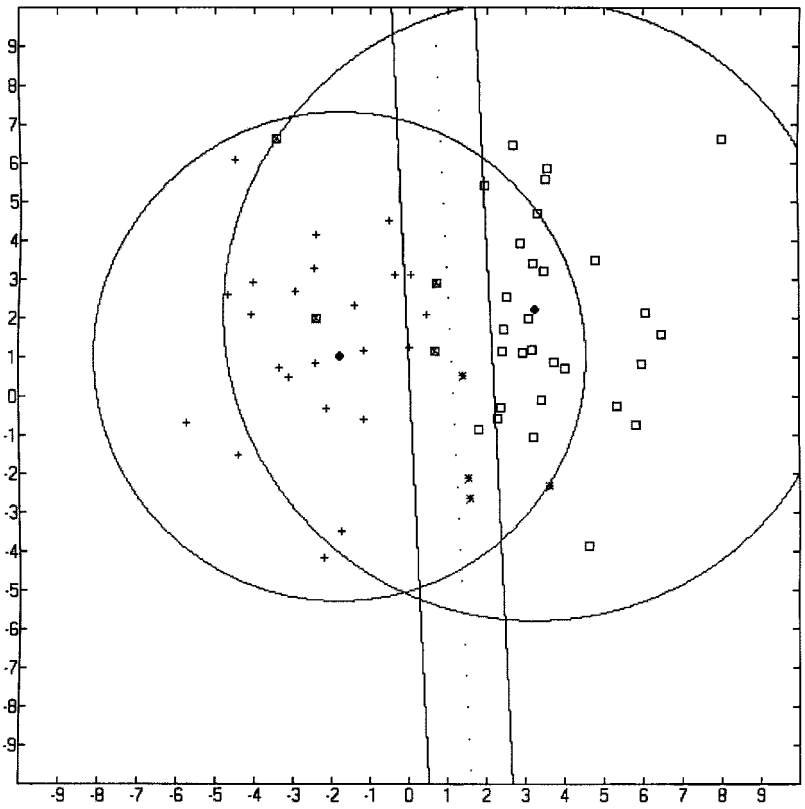

Fig. 6. The result of FSVM learning for data sets with outliers.

V. Conclusion

In this paper, we proposed the FSVM that imposes a fuzzy membership to each input point such that different input points can make different constributions to the learning of decision surface. By setting different types of fuzzy membership, we can easily apply FSVM to solve different kinds of problems. This extends the application horizon of the SVM.

There remains some future work to be done. One is to select a proper fuzzy membership function to a problem. The goal is to automatically or adaptively determine a suitable model of fuzzy membership function that can reduce the effect of noises and outliers for a class of problems.

References

[1] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, 1998.
[2] C. Cortes and V. N. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
[3] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[4] ——, *Statistical Learning Theory*. New York: Wiley, 1998.
[5] B. Schölkopf, C. Burges, and A. Smola, *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT Press, 1999.
[6] M. Pontil and A. Verri, *Massachusetts Inst. Technol.*, 1997, AI Memo no. 1612. Properties of support vector machines.
[7] N. de Freitas, M. Milo, P. Clarkson, M. Niranjan, and A. Gee, "Sequential support vector machines," *Proc. IEEE NNSP'99*, pp. 31–40, 1999.
[8] I. Guyon, N. Matić, and V. N. Vapnik, *Discovering Information Patterns and Data Cleaning*. Cambridge, MA: MIT Press, 1996, pp. 181–203.
[9] X. Zhang, "Using class-center vectors to build support vector machines," in *Proc. IEEE NNSP'99*, 1999, pp. 3–11.