Weighted support vector machine for data classification

Conference Paper *in* International Journal of Pattern Recognition and Artificial Intelligence · August 2007

DOI: 10.1109/IJCNN.2005.1555965 · Source: IEEE Xplore

3 authors, including:

Xulei Yang
Institute for Infocomm Research
91 PUBLICATIONS **683** CITATIONS
SEE PROFILE

Q. Song
Nanyang Technological University
129 PUBLICATIONS **1,568** CITATIONS
SEE PROFILE

International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 21, No. 5 (2007) 961–976

World Scientific
www.worldscientific.com

A WEIGHTED SUPPORT VECTOR MACHINE FOR DATA CLASSIFICATION

XULEI YANG*, QING SONG and YUE WANG
School of Electrical and Electronic Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639898
**yangxulei@pmail.ntu.edu.sg*

This paper presents a weighted support vector machine (WSVM) to improve the outlier sensitivity problem of standard support vector machine (SVM) for two-class data classification. The basic idea is to assign different weights to different data points such that the WSVM training algorithm learns the decision surface according to the relative importance of data points in the training data set. The weights used in WSVM are generated by a robust fuzzy clustering algorithm, kernel-based possibilistic c-means (KPCM) algorithm, whose partition generates relative high values for important data points but low values for outliers. Experimental results indicate that the proposed method reduces the effect of outliers and yields higher classification rate than standard SVM does when outliers exist in the training data set.

Keywords: Support vector machines; possibilistic C-means; Kernel based learning; data classification; robust fuzzy clustering.

1. Introduction

Support vector machine (SVM) was first introduced to solve the pattern classification and regression problems by Vapnik and his colleagues.[1,6] It can be seen as a new training algorithm for the traditional polynomial, radial basis function (RBF) and multi-layer perceptron (MLP) classifiers by defining relevant kernel functions. The interesting property of SVM is that it is an approximate implementation of the structural risk minimization principle in statistical learning theory[31] rather than the empirical risk minimization method. In recent years, SVM has drawn considerable attention due to its high generalization ability for a wide range of applications and better performance than other traditional leaning machines.[7,20,23]

In many practical engineering applications, the obtained training data is often contaminated by noises. Furthermore, some points in the training data set are misplaced far away from the main part of data cluster or even on the wrong side in feature space. These *atypical* points are known as outliers. As noted in Refs. 1 and 32, outliers tend to become support vectors with large Lagrangian coefficients during the training process. It is well known that the decision boundary obtained

by SVM only depends on the support vectors, if there exist outliers in the training data set and the outliers become the support vectors during the training procedure, then the decision boundary will deviate severely from the optimal hyperplane, such that, the SVM is very sensitive to outliers.

Some techniques have been found to tackle this problem for SVM in the literature of data classification.[a] In Ref. 32, a central SVM (CSVM) is proposed by using the class centers to build the support vector machine. In Ref. 12, an adaptive margin SVM (AMSVM) is developed based on the utilization of adaptive margins for each training point. Motivated by these two methods, a robust SVM (RSVM) proposed in Refs. 13 and 25 presents a general way to use the distance between the center of each class of the training data and the data point to form an adaptive margin. This approach, however, has the drawback that the penalty parameter λ is difficult to be tuned. In addition, RSVM deals with outliers by *averaging* method which is practically sensitive to outliers and noises in some sense. Different from the methods using the *averaging* algorithm to reduce the effect of outliers, the proposed approach in Refs. 18 and 19, resulting in a fuzzy SVM (FSVM), is to apply fuzzy memberships to the training data to relax the effect of the outliers and noises. However, the selection of membership function remains a problem for FSVM so far. Another interesting method, which is called support vector novelty detector (SVND) [or support vector data description (SVDD)],[2,22,29] has been shown effectiveness to detect outliers from the normal data points, however, it is well-known that SVND (or SVDD) is particular in the case of one-class classification problem. More recently, a hybrid technique is proposed in Ref. 17 involving symbolization of data to remove the noises (and outliers) and use of entropy minima to feature selection in conjunction with support vector machines to obtain a more robust classification algorithm.

In this paper, we use a robust fuzzy clustering technique to improve the outlier sensitivity problem of standard support vector machine (SVM) for two-class data classification. The basic idea is to assign different weights to different data points by the robust fuzzy clustering technique such that the induced weighted support vector machine (WSVM) learns the decision surface according to the relative importance of data point in the training set. As a result, the effect of the less important points (outliers and noises) to the decision surface is reduced during the training process. We use the kernel-based possibilistic c-means (KPCM) algorithm to generate the weights for WSVM. The original PCM algorithm[15,16] has been shown to have satisfactory capacity of handling noises and outliers. In addition, the induced partition assigns each data point a relative value according to the compatibility of the points with the class prototypes: important data points have high values but outliers and noises have low values. In this paper, we extend the original PCM algorithm into the kernel space by using kernel methods and develop

[a]This paper focuses on SVM classification, interested readers can refer to Refs. 3–5, 26–28 for discussions on outlier problems for SVM regression.

the KPCM algorithm whose partitioned relative values are served as the weights for the proposed WSVM training.

The rest of this paper is organized as follows. Section 2 briefly reviews the basic theory and outlier sensitivity problem of standard SVM. In Sec. 3, we derive the formulation of the proposed WSVM, and present the weights generating algorithm. The detailed experimental results are reported in Sec. 4. In Sec. 5, we present some remarks and discussions on algorithms, experiments and further investigations. Finally, the conclusion is given in Sec. 6.

2. Standard SVM

2.1. *Formulation of SVM*

The main idea behind SVM technique is to derive a unique separating hyperplane (i.e. the optimal margin hyperplane) that maximizes the margin between the two classes. Given l training data points

$$\{(x_i, y_i)\}_{i=1}^{l}, \quad x_i \in R^N, \quad y_i \in \{-1, 1\}$$

the support vector technique requires the solution of the following optimization problem:

$$\text{Minimize } \Phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{1}$$

subject to

$$\begin{aligned} y_i(\langle w, \phi(x_i)\rangle + b) &\geq 1 - \xi_i, \quad i = 1, \ldots, l \\ \xi_i &\geq 0, \qquad\quad i = 1, \ldots, l \end{aligned} \tag{2}$$

where the training vectors x_i are mapped into a higher-dimensional space by the function ϕ. C is a user-specified positive parameter, which controls the tradeoff between classification violation and margin maximization. The existing common method to solve (1) is through its dual, a finite quadratic programming problem:

$$\begin{aligned} W(\alpha) &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j (\langle \phi(x_i), \phi(x_j)\rangle) \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \end{aligned} \tag{3}$$

subject to

$$\sum_{i=1}^{N} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l \tag{4}$$

where α is the lagrangian parameter. Note the kernel trick $K(x, y) = \langle \phi(x), \phi(y)\rangle$ is used in the last equality in (3). The Kuhn–Tucker conditions of SVM are defined by

$$\begin{aligned} \alpha_i[y_i(\langle w, \phi(x_i)\rangle + b) - 1 + \xi_i] &= 0, \quad i = 1, \ldots, l \\ (C - \alpha_i)\xi_i &= 0, \quad i = 1, \ldots, l. \end{aligned} \tag{5}$$

The point x_i with the corresponding $\alpha_i > 0$ is called a support vector. The optimal value of weight vector w_0 is obtained by $w_0 = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i) = \sum_{i=1}^{l_s} \alpha_i y_i \phi(x_i)$, where l_s is the number of support vectors. The optimal value of bias b_0 can be computed from the Kuhn–Tucker conditions (5). Once the optimal pair (w_0, b_0) is determined, the decision function is obtained by

$$f(x) = \text{sign}(\langle w_0, \phi(x) \rangle + b_0) = \text{sign}\left(\sum_{i=1}^{l_s} \alpha_i y_i K(x, x_i) + b_0 \right). \tag{6}$$

2.2. *Outlier sensitivity problem of SVM*

It can be observed from Eq. (6) that the decision boundary is decided by only the support vectors: the points with $\alpha_i > 0$. In many practical engineering applications, the training data is often affected by noises or outliers, which tend to become the support vectors during the SVM training process.[1,32] In this case, the decision boundary will be deviated from the optimal boundary severely. This phenomenon is well-known as the outlier sensitivity problem of standard SVM algorithm.

Figure 1 gives an illustration of the outlier sensitivity problem of SVM. When no outlier exists in the training data, SVM can find an optimal hyperplane (denoted by solid line) with maximum margin to separate the two classes. If there exists two outliers (located on the wrong side and denoted by numbers 1 and 2), for example, in class 1, the decision hyperplane (denoted by dotted line) is severely deviated from

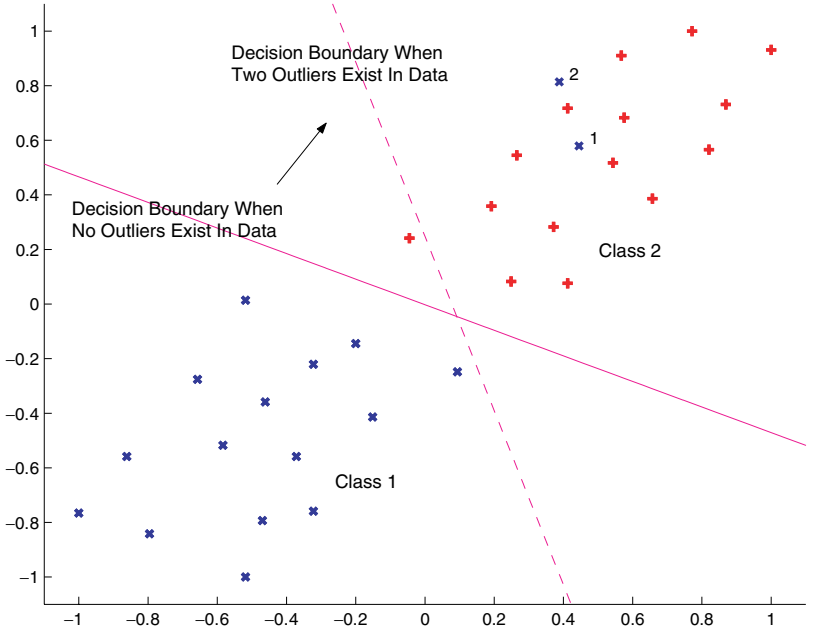


Fig. 1. Outlier sensitivity problem of standard SVM.

the optimal one due to the over-fitting problem. Therefore, the training process of standard SVM is very sensitive to outliers. This paper aims to improve the SVM training, especially focuses on the over-fitting problem with outliers that make the two classes nonseparable.

3. The Proposed Weighted SVM

The basic idea of weighted support vector machine (WSVM) is to assign each data point a different weight according to its relative importance in the class such that different data points have different contribution to the learning of the decision surface. Suppose the weights are given, then the training data set becomes

$$\{(x_i, y_i, W_i)\}_{i=1}^{l}, \quad x_i \in R^N, y_i \in \{-1, 1\}, W_i \in R$$

where the scalar $0 \leq W_i \leq 1$ is the weight assigned to data point x_i by kernel-based possible c-means (KPCM) algorithm, which will be discussed later in this section.

3.1. *Formulation of WSVM*

Starting with the construction of a cost function, the WSVM wants to maximize the margin of separation and minimize the classification error such that a good generalization ability can be achieved. Unlike the penalty term in standard SVM, where the value of C is fixed and all training data points are equally treated during the training process, WSVM weighs the penalty term in order to reduce the effect of less important data points (such as outliers and noises). The constrained optimization problem is formulated as

$$\text{Minimize } \Phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{l} W_i \xi_i \tag{7}$$

subject to

$$\begin{aligned} y_i(\langle w, \phi(x_i)\rangle + b) &\geq 1 - \xi_i, \quad i = 1, \ldots, l \\ \xi_i &\geq 0, \qquad\quad i = 1, \ldots, l. \end{aligned} \tag{8}$$

Note that we assign the weight W_i to the data point x_i in the above formulation. Accordingly, the dual formulation becomes

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{9}$$

subject to

$$\sum_{i=1}^{l} y_i \alpha_i = 0 \quad 0 \leq \alpha_i \leq C W_i \quad i = 1, \ldots, l \tag{10}$$

and the KT conditions of WSVM become

$$\begin{aligned} \alpha_i[y_i(\langle w, \phi(x_i)\rangle + b) - 1 + \xi_i] &= 0, \quad i = 1, \ldots, l \\ (C W_i - \alpha_i)\xi_i &= 0, \quad i = 1, \ldots, l. \end{aligned} \tag{11}$$

It is clear that the only difference between SVM and the proposed WSVM is the upper bounds of Lagrange multipliers α_i in the dual problem. In SVM, the upper bounds of α_i are bounded by a constant C while they are bounded by dynamical boundaries that are weight values CW_i in WSVM. Suppose a data point x_i is an outlier that is located on the wrong side in kernel space. Then the outlier may become a support vector and the upper bound of Lagrange multiplier α_i is reached. For WSVM training, we have $\alpha_i = CW_i$. The weight W_i assigned by KPCM algorithm to this support vector (outlier) is much smaller than other support vectors (nonoutliers), such that the effect of this support vector to the training procedure will be reduced and the decision boundary will be less distorted by the outlier. While for the standard SVM training, the value of α_i is equal to C, which will affect the decision surface more significantly than WSVM does. We have to point out that in the cases of $C \to \infty$ and $C \to 0$, we have $CW_i = C$, then the performance of the proposed WSVM becomes identical to the standard SVM. That means the effect of the proposed WSVM for reducing outlier effect can only be observed for the cases where the value of C is finitely positive.

3.2. *Weights generating algorithm*

The weights used in WSVM are generated by kernel-based possibilistic c-means (KPCM) algorithm. Possibilistic c-means (PCM) algorithm, which was first proposed in Ref. 15 and further explored in Refs. 8 and 16 to overcome the relative membership problem of fuzzy c-means (FCM) algorithm, has been shown to have satisfactory capacity of handling noises and outliers. In addition, the induced partition assigns each data point a relative value (possibility) according to the compatibility of the points with the class prototypes: important data points have high values but outliers and noises have low values. Therefore, we can choose the partitioned relative values as the weight values in the proposed WSVM training process. However, the WSVM training procedure is processed in kernel space by using different kernel functions. The weights generated by PCM in input space may be not suitable for WSVM training. To make use of the PCM algorithm for the proposed WSVM, we have to extend the original PCM into the kernel space by using kernel methods and develop the KPCM algorithm to generate weights in kernel space instead of in input space.

Assume the training data are partitioned into c clusters in kernel space, the set of cluster centers denoted by $\{\phi_1(x), \ldots, \phi_c(x)\} \subset F$ (F represents the transformed high-dimensional kernel space) has a similar presentation like the K-means clustering in kernel space as in Ref. 24

$$\phi_k(x) = \sum_{j=1}^{l} \gamma_{kj} \phi(x_j), \quad k = 1, 2, \ldots, c \tag{12}$$

where γ_{kj} is the parameters to be decided by KPCM algorithm as shown below. Note that unlike the *hard* clustering, where each data point is assigned to one certain

cluster, KPCM is a *soft* clustering approach. It means that all the training data with a total number l could have relationship with all the c clusters as shown in (12). Then the squared Euclidian distance[b] between a cluster center $\phi_k(x)$ and a mapped point $\phi(x)$ can be stated as

$$\begin{aligned} D_k(x) &= \left\| \phi(x) - \sum_{i=1}^{l} \gamma_{ki} \phi(x_i) \right\|^2 \\ &= K(x,x) - 2\sum_{i=1}^{l} \gamma_{ki} K(x_i, x) + \sum_{i,j=1}^{l} \gamma_{ki}\gamma_{kj} K(x_i, x_j). \end{aligned} \tag{13}$$

Note the kernel trick $K(x,y) = \langle \phi(x), \phi(y) \rangle$ is used in the above equation.

Similar to the objective function of PCM (which is formulated by modifying the objective function of standard FCM, details referring to Refs. 8 and 15), the objective function of KPCM is formulated as follows by using the distance definition (13).

$$J = \sum_{k=1}^{c} \sum_{j=1}^{l} (p_{kj})^m D_k(x_j) + \sum_{k=1}^{c} \eta_k \sum_{j=1}^{l} (1 - p_{kj})^m \tag{14}$$

where $D_k(x_j)$ is the squared Euclidian distance between the transformed point $\phi(x_j)$ and cluster prototype $\phi_k(x)$ in kernel space as defined in (13), m $(1 < m < \infty)$ determines the fuzziness, and p_{kj} is the possibility of transformed point $\phi(x_j)$ in the (k)th cluster, and there are no constraints on the possibility p_{kj} other than the requirement that they should be in $[0, 1]$. Note the second term forces p_{kj} to be as large as possible, thus avoiding the trivial solution of standard FCM, more details can be found in Ref. 8.

Assuming that the parameter η_k is specified, the minimization of (14) gives the updating equations for possibility p_{kj} and cluster prototype $\phi_k(x)$ as follows:

$$p_{kj} = \frac{1}{1 + \left[\frac{D_k(x_j)}{\eta_k} \right]^{1/(m-1)}} \tag{15}$$

$$\phi_k(x) = \sum_{j=1}^{l} \gamma_{kj} \phi(x_j) = \frac{\sum_{j=1}^{l} (p_{kj})^m \phi(x_j)}{\sum_{j=1}^{l} (p_{kj})^m}. \tag{16}$$

Note the explicit expression of γ_{kj} is given in (16). The positive parameter η_k is a critical parameter which should be determined in the above algorithm. Referring to

[b]The calculation of the kernel-induced distance, i.e. Eq. (13), may suffer from expensive computation for large data sets. However, compared to the expensive training time of standard SVM, it is acceptable and even can be negligible. In the literature, several efficient methods, especially the decomposition algorithms,[33,34,35] have been proposed to make large-scale SVM practical in the past years. For this case, some modification has to be made to combine the proposed method to these decomposition algorithms.

the recommendation in Ref. 15, η_k can be obtained from the average possibilistic intra-cluster distance of cluster k as

$$\eta_k = K \frac{\sum_{j=1}^{l} (p_{kj})^m D_k(x_j)}{\sum_{j=1}^{l} (p_{kj})^m}. \tag{17}$$

The KPCM algorithm is implemented by iteratively updating (15) and (16) until the stopping criterion is satisfied. The partition results are not very sensitive to the range of the values of K.[15] In our experiments, we set K a constant with the value 1 for benchmark data set and 2 for artificial data set.

We can now use the above KPCM algorithm to generate weights for the training of WSVM algorithm. Let the set of possibility values of positive class ($y_i = 1$) and negative class ($y_i = -1$) be denoted by p_i^+ and p_i^- after partitioning, respectively. Then the weights of WSVM are finally obtained by setting $W_i^+ = p_i^+$ and $W_i^- = p_i^-$. (Here W^+ denote the weights of points in positive class and W^- denote the weights of points in negative class.) After partitioning, the weights of outliers and noises will be very small or even nearly zero, such that the effect of outliers and noises to decision hyperplane is significantly reduced. Figure 2 shows the weights of a given class (class 1 from Fig. 1, where two outliers exist) generated by KPCM. As we can see, the weights of training data points in the main part of data cluster are much larger (nearly one) and the weights of the outliers are much smaller (nearly zero).

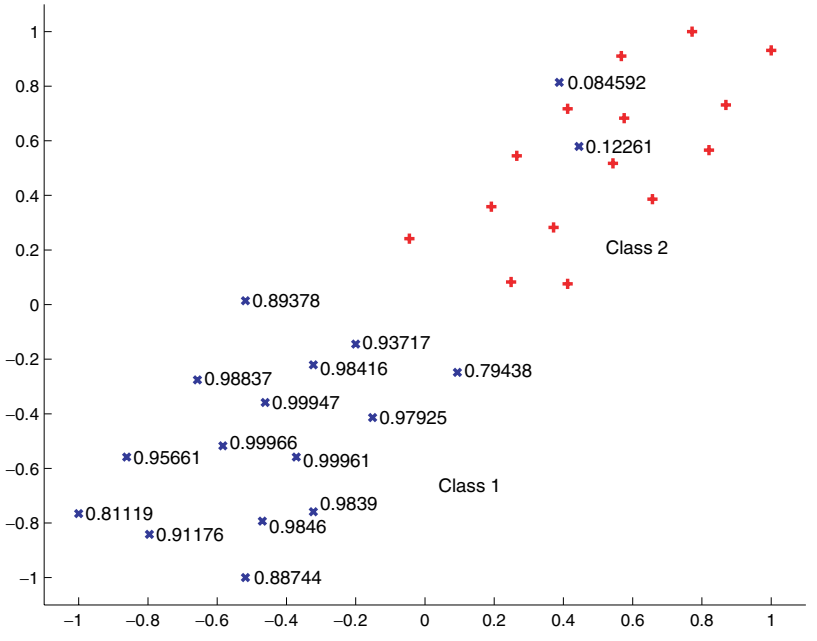


Fig. 2. Weights generated by KPCM for class 1.

Accordingly, the effect of the outliers to the decision surface will be significantly reduced, as observed in the section of experimental results.

4. Experimental Results

The effectiveness of the proposed WSVM algorithm to classification problems is tested on several artificial and benchmark data sets. The WSVM program[c] is developed based on the standard SVM training program.[11]

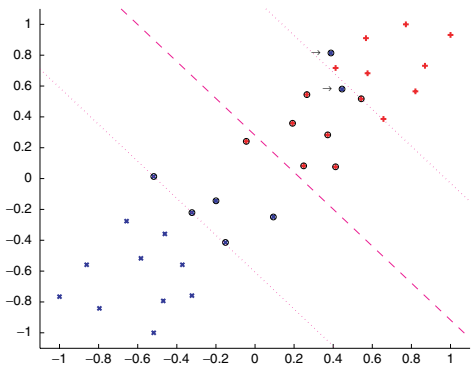
4.1. *Artificial data set*

We generate a two class data set in R^2 (represented by crosses and pluses), where positive class has two outliers (marked by arrows nearby) located on the wrong side. The training results of SVM and WSVM using linear kernel function with different values of C are shown in Fig. 3. Where "x", and "+" denote the positive class and negative class respectively, the data points with circle symbols surrounding them are support vectors, the distance between the two dotted lines is the margin of separation, the line between the two dotted lines is the optimal hyperplane i.e. the decision surface. As we can see, the decision surface obtained by standard SVM is significantly distorted by outliers [as shown in (a1), (a2) and (a3)], which will lead to poor generalization performance, i.e. the test error is large. In contrast, the proposed WSVM has much better performance [as shown in (b1), (b2) and (b3)], the obtained decision surface is hardly affected by outliers compared to SVM. Actually, the decision boundary obtained by WSVM with $C = 100$ is almost identical to the solid line in Fig. 1, which is the optimal hyperplane assuming the outliers are removed. Thus, the proposed method successfully reduces the effect of outliers and accordingly yields better generalization performance than standard SVM does. In addition, it can be obtained from Fig. 3 that the proposed WSVM is less sensitive to the value of the penalty parameter C than standard SVM does for the data set. Note if the outliers are removed, SVM and WSVM will have similar performances.

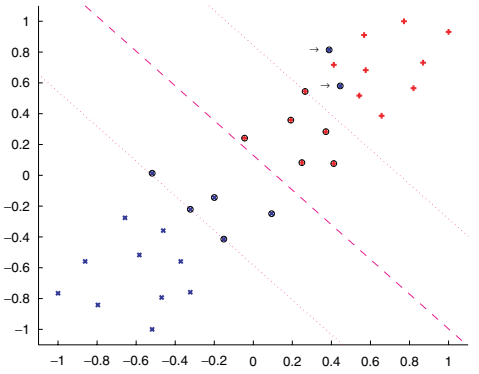
4.2. *Benchmark data set*

The benchmark data set used to test the proposed method is "Twonorm" from IDA benchmark repository,[14] which has been used to evaluate kernel and boosting methods in recent years.[20,21] This data set is originally from DELVE benchmark repository.[9] It is a 20-dimensional, 2 class classification example. Each class is drawn from a multivariate normal distribution with unit variance. Class 1 has mean $(a, a, \ldots, a)$ while Class 2 has mean $(-a, -a, \ldots, -a)$, where $a = 2/\text{sqrt}(20)$.

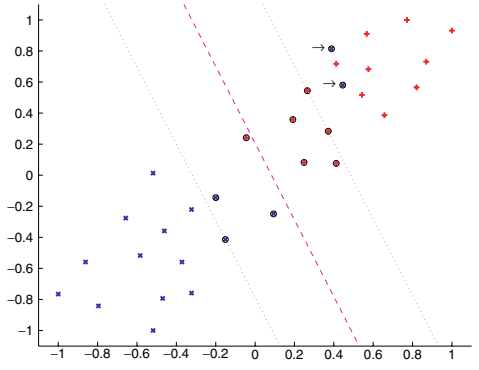
[c]The Matlab code of the KPCM algorithm and the modified SVM programming is available for the reader who wants to test the method. Just contact the authors by e-mail at yangxulei@pmail.ntu.edu.sg or visit the web-page http://www.ntu.edu.sg/home5/pg04092292/supplements.htm.



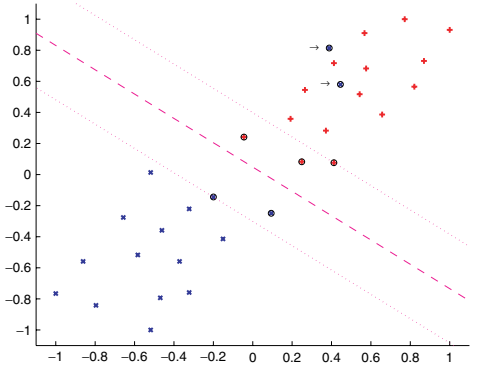
(a1) SVM, Linear Kernel, $C = 1$



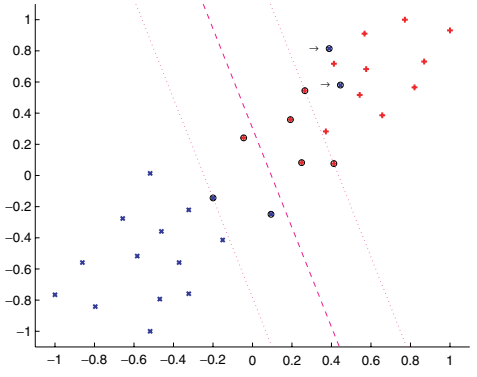
(b1) WSVM, Linear Kernel, $C = 1$



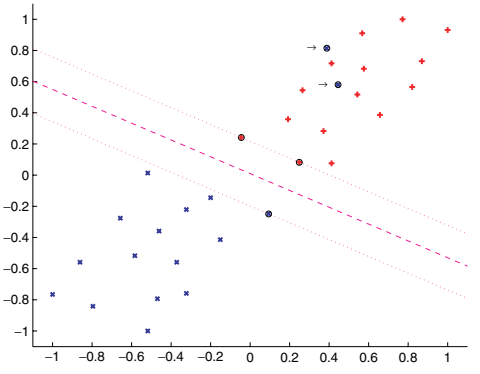
(a2) SVM, Linear Kernel, $C = 10$



(b2) WSVM, Linear Kernel, $C = 10$



(a3) SVM, Linear Kernel, $C = 100$



(b3) WSVM, Linear Kernel, $C = 100$

Fig. 3. The performance comparison between SVM and WSVM while outliers exist in the training data set.

Table 1. Comparison of test error between WSVM and standard SVM for "Twonorm" benchmark data set according to the different number of mislabeled data points (the number is denoted by N_o) using linear kernel function.

	$N_o = 0$	$N_o = 10$	$N_o = 20$	$N_o = 30$	$N_o = 40$	$N_o = 50$	$N_o \geq 90$
SVM	2.50%	2.69%	3.06%	3.51%	4.14%	4.90%	$\geq$ 28.7%
WSVM	2.52%	2.54%	2.67%	2.83%	3.01%	3.30%	$\geq$ 15.1%

The number of data points is 7,400 (400 training data points and 7,000 testing data points). Here, we use linear kernel to classify this benchmark data set. The value of $C = 20$ is determined through cross-validation. The classification error of standard SVM and the proposed WSVM are 2.50% and 2.52% (see the first column of numerical results in Table 1), respectively. We have to point out that WSVM (and other outlier reduction methods mentioned in the introduction section) is originally proposed to solve the outlier sensitivity problem of standard SVM when outliers exist in the training data. However, the benchmark data may not contain outliers, in this case, the performance of WSVM will be similar to that of standard SVM.

To show the effectiveness of WSVM algorithm to reduce the effect of outliers, we gradually *corrupt* the "Twonorm" benchmark data set by mislabeling some data points (the number is denoted as N_o): we select N_o data points, for example, from the positive $(+1)$ class, then artificially set the label of these points to -1. These mislabeled data points can be regarded as *outliers*. Table 1 compares the percentage test error of WSVM to standard SVM according to the different number of mislabeled data points. From the table, we can see the performance of standard SVM is severely distorted by the mislabeled data points: the test error of standard SVM increases obviously from 2.50% to 4.90% as the number of mislabeled data points N_o increases from 0 to 50. While the proposed WSVM appears to be more robust and has better performance: the test error increases only from 2.52% to 3.30%. This indicates that the proposed method successfully reduces the effect of outliers (mislabeled data points in this case) and yields higher classification rate than standard SVM does when the training data set is contaminated by outliers (mislabeled data points in this case). Note when N_o is beyond a large value (such as 90), the test error of both SVM and WSVM classifiers become undesirable (see the last column of numerical results in Table 1).

5. Remarks and Discussions

5.1. *On algorithms*

1. For KPCM algorithm, we define convergence to be reached when the maximum change in the possibility values p_{kj} over all points between consecutive iterations is less than a given threshold value. In practice, we used a threshold value of 0.001. Please note we cannot use the cluster center $\phi_k(x)$ to tell whether or not the convergence is reached. This is because we do not know the explicit expression of the nonlinear mapping ϕ, such that we cannot directly compute the cluster

center by using (16). In practice, the cluster center is implicitly updated by updating the distance in kernel space, i.e. using (13).

2. KPCM algorithm requires an initial estimate of possibility values p_{kj}. Proper selection will generally improve accuracy and convergence of the algorithm. We use standard FCM to generate initial possibility values for KPCM as recommended in Ref. 15. This method seems to be suitable and we find the KPCM algorithm is always converged in several iterations. We also find that KPCM algorithm generally converges faster than the standard PCM algorithm does. This observation may be due to the relative simpler data structure in kernel space than that in input space provided that suitable kernel function and kernel parameter(s) are used, as mentioned in Ref. 10.
3. We use KPCM to partition positive class and negative class respectively by setting the cluster number $c = 1$ for each class. The partition time is quite acceptable and can be negligible compared to the training time of SVM algorithm such that the total computation time (including partitioning and training) of WSVM is similar to that of standard SVM. The computation complexity of WSVM may be further reduced through a simple pruning method when the training data set is contaminated by outliers, as discussed later in this section.

5.2. *On experiments*

1. In our experiments, we only compare the performances between SVM and WSVM by using linear kernel function for "Twonorm" benchmark data set. In fact, the proposed method is universal and the similar observations can be obtained by using any kernel function for most benchmark data sets if we *corrupt* (artificially adding or mislabeling some data points) the original data sets. For instance, Table 2 shows the percentage test error between WSVM and standard SVM by using RBF kernel function ($C = 10$ and $\sigma = 3$ taken from Ref. 14) for "Thyroid" data set from IDA benchmark repository[14] (originally taken from UCI benchmark repository[30]) according to the different number of mislabeled data points. Note that the advantage of the proposed WSVM over standard SVM shown in Table 2 is much more obvious: the test error of WSVM retains unchanged (2.67%) if the number of mislabeled data points keeps small ($N_o \leq 10$).

Table 2. Comparison of test error between WSVM and standard SVM for "Thyroid" benchmark data set according to the different number of mislabeled data points (the number is denoted by N_o) using RBF kernel function.

	$N_o = 0$	$N_o = 5$	$N_o = 10$	$N_o = 15$	$N_o = 20$	$N_o = 25$
SVM	2.67%	6.67%	10.7%	14.7%	20.0%	24.1%
WSVM	2.67%	2.67%	2.67%	4.00%	5.33%	14.7%

2. According to Eq. (6), the decision surface is determined by all the support vectors. If the number of support vectors increases but the number of outliers (which become support vectors after training process) remain unchanged, then the contribution of the outliers to the decision surface will accordingly decrease. In Figs. 3(a1) and 3(b1), the proportion of the number of outliers to total number of support vectors is quite small such that the reduction of outliers cannot significantly affect the learning of the decision surface. This just explains why the decision surface of WSVM [shown in Fig. 3(b1)] is only a bit better than that of SVM [shown in Fig. 3(a1)]. Please note that the sparse property is sacrificed in this case.
3. In our experiments, we set K (the parameter of KPCM algorithm, see (12)) a constant with the value 1 for benchmark data set and 2 for artificial data set. This indicates tuning of K could give even better performance for WSVM though the improvement is not significant. As a general rule, the value of K should be decreased when the number of outliers in training data increases.

5.3. *On further investigations*

1. As mentioned above, the WSVM (or other outlier reduction techniques mentioned in the introduction section) is originally proposed to improve the outlier sensitivity problem of standard SVM when outliers exist in the training data, our future work will focus on the contaminated data set to further investigate the ability of WSVM in reducing the effect of outliers.
2. Choosing C effectively is still an opening problem. Our further research will investigate how the WSVM formulation affects the ability to choose the penalty parameter C. Unlike the penalty term in standard SVM, where the value of C is fixed and all training data points are equally treated during the training process, WSVM weighs the penalty term in order to reduce the effect of outliers if they exist in the training set. Though the preliminary experiments reveal that the proposed WSVM is less sensitive to the value of C than the standard SVM for the contaminated training data set, further fundamental research is needed in this interesting direction.
3. The training and testing complexities of WSVM can be greatly reduced through a simple pruning method when the training data set is contaminated by outliers. While in pruning methods for classical MLP the procedure requires the computation of a Hessian matrix or its inverse, the pruning in WSVM can be done based upon the weighted data points themselves. Less meaningful data points as indicated by their (smaller) weights are removed and the WSVM is trained on the basis of the remaining points, this procedure may significantly reduce both the training time and the effect of outliers. The testing time can similarly be reduced by removing the less meaningful support vectors with smaller weights. Though the pruning procedure may induce the loss of training and testing accuracy, this topic is desirable to be further investigated.

4. It is well known that the kernel-based learning algorithms suffer from the expensive computation time and storage memory for large data samples. In the past years, several efficient methods, especially the decomposition algorithms,[33,34,35] have been proposed to tackle this problem for the support vector machines (SVMs). The key idea of decomposition is to freeze all but a small number of optimization variables, and to solve a sequence of constant-size problems. This method breaks large quadratic programming (QP) problem into a series of smaller QP subproblems such that significantly reduces the storage and computation. But so far, seldom attempts have been proposed for the unsupervised learning in feature space. The development of a speed-up method for data clustering in feature space, especially for the proposed KPCM algorithm, is desirable for further investigation.

6. Conclusion

In this paper, a weighted support vector machine (WSVM) is proposed to deal with the outlier sensitivity problem in traditional support vector machine (SVM) for two-class data classification. The robust fuzzy clustering algorithm, i.e. possibilistic c-means (PCM), is extended into kernel space to generate different weight values for main training data points and outliers according to their relative importance in the training set. Experimental results have shown that the proposed WSVM can reduce the effect of outliers and yield higher classification rate than standard SVM does when the training data set is contaminated by outliers.

Acknowledgment

The authors sincerely thank the anonymous reviewers for their insightful comments and valuable suggestions on an earlier version of this paper.

References

1. B. E. Boser, I. M. Guyon and V. N. Vapnik, A training algorithm for optimal margin classifiers, *The Fifth Annual Workshop on Computational Learning Theory* (1992), pp. 144–152.
2. L. J. Cao, H. P. Lee and W. K. Chong, Modified support vector novelty detector using training data with outliers, *Patt. Recogn. Lett.* **24**(14) (2003) 2479–2487.
3. C.-C. Chuang and J.-T. Jeng, A soft computing technique for noise data with outliers, *IEEE Int. Conf. Networking, Sensing Control* **2** (2004) 1171–1176.
4. C.-C. Chuang, Y.-C. Lee and J.-T. Jeng, A new annealing robust fuzzy basis function for modeling with outliers, *IEEE Int. Conf. Syst. Man Cybern.* **5** (2003) 4451–4456.
5. C.-C. Chuang, S.-F. Su, J.-T. Jeng and C.-C. Hsiao, Robust support vector regression networks for function approximation with outliers, *IEEE Trans. Neural Networks* **13**(6) (2002) 1322–1330.
6. C. Cortes and V. N. Vapnik, Support vector networks, *Mach. Learn.* **20**(3) (1995) 273–297.

7. N. Cristianin and J. Shawe-Taylar, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods* (Cambridge University Press, Cambridge, 2000).
8. R. N. Dave and R. Krishnapuram, Robust clustering methods: a unified view, *IEEE Trans. Fuzzy Syst.* **5**(2) (1997) 270–293.
9. DELVE Benchmark repository l-A collection of artificial and real world data sets, available at http://www.cs.utoronto.ca/delve/data/datasets.html.
10. M. Girolami, Mercer kernel-based clustering in feature space, *IEEE Trans. Neural Networks* **13**(3) (2002) 780–784.
11. S. Gunn, Support vector machines for classification and regression, Technical Report, Image, Speech and Intelligent Systems Research Group ISIS, University of Southampton (1998).
12. R. Herbrich and J. Weston, Adaptive margin support vector machines for classification, *The Ninth Int. Conf. Artificial Neural Network (ICANN 99)* **2** (1999) 880–885.
13. W. J. Hu and Q. Song, An accelerated decomposition algorithm for robust support vector machines, *IEEE Trans. Circuits Syst.* **51**(5) (2004) 234–240.
14. IDA Benchmark repository used in several boosting, KFD and SVM papers, available at http://ida.first.gmd.de/ratsch/data/benchmarks.htm.
15. R. Krishnapuram and J. M. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Syst.* **1**(2) (1993) 98–110.
16. R. Krishnapuram and J. M. Keller, The possibilistic c-means algorithm: insights and recommendations, *IEEE Trans. Fuzzy Syst.* **4**(3) (1996) 385–393.
17. R. Kumar, V. K. Jayaraman and B. D. Kulkarni, An SVM classifier incorporating simultaneous noise reduction and feature seletion: illustrative case examples, *Patt. Recogn.* **38** (2005) 41–49.
18. C. F. Lin and S. D. Wang, Fuzzy support vector machines, *IEEE Trans. Neural Networks* **13**(2) (2002) 464–471.
19. C. F. Lin and S. D. Wang, Training algorithms for fuzzy support vector machines with noisy data, *2003 IEEE 8th Workshop on Neural Networks for Signal Processing* (2003) 517–526.
20. K. R. Muller, S. Mike, G. Ratsch, K. Tsuda and B. Scholkopf, An introduction to kernel based learning algorithms, *IEEE Trans. Neural Networks* **12**(2) (2001) 181–201.
21. G. Ratsch, T. Onoda and K. R. Muller, Soft margins for AdaBoost, *Mach. Learn.* **42**(3) (2001) 287–320.
22. B. Scholkopf, J. Platt, J. Shawe-Taylor and A. J. Smola, Estimating the support of a high-dimensional distribution, *Neural Comput.* **13**(7) (2001) 1443–1470.
23. B. Scholkopf and A. J. Smola, *Learning with Kernels* (MIT Press, Cambridge, MA, 2002).
24. B. Scholkopf, A. J. Smola and K. R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Technical Report, Max Planck Institute for Biological Cybernetics, Tubingen, Germany (1996).
25. Q. Song, W. J. Hu and W. F. Xie, Robust support vector machine with bullet hole image classification, *IEEE Trans. Syst. Man Cybern.* **32**(4) (2002) 440–448.
26. Z. Sun and Y. Sun, Fuzzy support vector machine for regression estimation, *IEEE Int. Conf. Syst. Man Cybern.* **4** (2003) 3336–3341.
27. J. A. K. Suykens, J. De Brabanter, L. Lukas and J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, *Neurocomputing* **48**(1–4) (2002) 85–105.
28. J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, *Least Squares Support Vector Machines* (World Scientific, Singapore, 2002).

29. D. M. J. Tax and R. P. W. Duin, Support vector data description, *Mach. Learn.* **54**(1) (2004) 45–66.
30. UCI Benchmark repository 1-A huge collection of artificial and real world data sets, University of California Irvine, http://www.ics.uci.edu/ mlearn.
31. V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer, New York, 1995).
32. X. G. Zhang, Using class-center vectors to build support vector machines, *Neural Networks for Signal Processing IX, 1999, Proc. 1999 IEEE Signal Processing Society Workshop* (1999), pp. 3–11.

Added in Proof

33. C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, the software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.
34. T. Joachims, Making large-scale support vector machine learning practical, in *Advances in Kernel Methods-Support Vector Learning*, eds. Scholkopf *et al.* (Cambridge, MIT Press, MA, 1999), pp. 169–184.
35. J. C. Platt, Fast training of support vector machines using sequential minimal optimization, in *Advances in Kernel Methods-Support Vector Learning*, eds. Scholkopf *et al.* (Cambridge, MIT Press, MA, 1999), pp. 185–208.

Xulei Yang received the B.E. degree and M.E. degree from EE School, Xi'an Jiaotong University in 1999 and 2002, respectively. He obtained the Ph.D. degree from EEE School, NTU in 2005.

He has published more than 20 papers in scientific book chapters, journals and conference proceedings. His current research interests include pattern recognition, image processing, and machine vision.

Qing Song received the B.S. and the M.S. degrees from Harbin Shipbuilding Engineering Institute and Dalian Maritime University, China in 1982 and 1986, respectively. He obtained the Ph.D. degree from the Industrial Control Center at Strathclyde University, UK in 1992. He is currently an associate professor and an active industrial consultant at the school of EEE, NTU.

His research interests is focused on a few computational intelligence related research programs targeted for practical applications.

Yue Wang received his Bachelor degree from Wuhan University, China, and the Master and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University, Singapore. He has published more than 13 papers in scientific journals and conference proceedings.

His research interests include computer vision and pattern recognition, object segmentation and matching, biomedical image processing, spline approximation and deformable model.