Robust Enhancement for Proximal Support Vector Machines

Meng Zhang[1], Gaofeng Wang[1], Lihua Fu[2]

[1] C.J.Huang Institute of Information Technology, Wuhan University, Wuhan, China.

[2] Department of Mathematics and physics, China University of Geosciences, Wuhan, China.

Abstract. Proximal support vector machines (PSVM) is a new version of SVM, which involves equality instead of inequality constraints, and works with a square error function. In this way, the solution follows from a linear Karush-Kuhn-Tucker system instead of a quadratic programming problem. The linear PSVM can easily solve the classification problems of extremely large datasets. However, according to the experiments below, PSVM is sensitive to noise. To overcome the drawback, this note proposes a weighted version of PSVM. The distance between each point and the center of corresponding class is used to calculate the weight value associated with the related point. In this way, the effect of noise is reduced greatly. The experiments indicate that the new SVM, weighted proximal support vector machine (WPSVM), is much more robust to noise than PSVM without loss of computationally attractive feature of PSVM.

Keywords: Data Classification, support vector machines, linear equation

1 Introduction

Support vector machines (SVMs), which have been introduced as a powerful tool for solving classification problems, classify points by assigning them to one of two disjoint halfspaces. These halfspaces, separated by a hyperplane, are either in the original input space of the problem for linear classifiers, or in a higher dimensional feature space for nonlinear classifiers [2]. In the framework of standard SVMs, one assigns the hyperplane to make the margin between two classes maximized in order to improve the generality of SVMs. Since it has strong theoretical foundations and good generalization capability, the standard SVMs have been gained wide acceptance.

A limitation of the SVMs design algorithms, particularly for large datasets, is the need to solve a quadratic programming (QP) problem involving a dense $m \times m$ matrix, where m is the number of points in the dataset. Since QP routines have high complexity, SVMs design requires huge memory and computational time for large data applications.

In contrast to standard SVM, both [1] and [8] have constructed much simpler

classifiers, namely PSVM and least squares support vector machine (LS-SVM) respectively, which obtain the separating hyperplane by solving a system of linear equations instead of a complex QP problem. Since it has stronger convex property of object function of QP, PSVM classify points much faster than LS-SVM [1]. More in detail, PSVM is based on replacing an inequality constraint by equality in the defining structure of SVM framework, and also on replacing absolute error by squared error in defining optimized problem. In this way, classifying given dataset including millions of data by PSVM costs less than half a minute [1]. However, we think, it is the replacement that makes PSVM sensitive to noise and outlier. (Figure 1 and 2).

The aim of the present paper is to show that one can improve PSVM's robustness in its original framework without the loss of the computationally attractive feature.

This note proposes weighted proximal support vector machine (WPSVM), which modifies the original optimization problem of PSVM by applying a given weight value to the error variable. We have shown that the technique of WPSVM is equivalent to the weighted ridge regression, which is a well-known topic in statistics community. For PSVM, the procedure can be a cheap and efficient way to make the solution robust.

The idea presented here is partially motivated by [4]. However fuzzy support vector machine proposed in [4] is based on the framework of standard SVM, which need to solve a more complex QP problem than SVM does. While the computationally attractive feature of PSVM remains in WPSVM formulation.

We summarize the contents of the paper now. In Section 2, PSVM is presented briefly. The WPSVM will be derived in Section 3 after some analysis to PSVM. Finally, there are some experiments and conclusions in Sections 4 and 5.

A word about the notation used in this note. All vectors will be column vectors unless transposed to a row vector by a prime superscript '. All matrix and vectors are in bold. The scalar (inner) product of two vectors $\mathbf{x}$ and $\mathbf{y}$ in the n-dimensional real space $\mathbf{R}^n$ will denoted by $\mathbf{x}'\mathbf{y}$ and the 2-norm of $\mathbf{x}$ will be denote by $\|\mathbf{x}\|$. For a matrix $\mathbf{A} \in \mathbf{R}^{m \times n}$, $\mathbf{A}_i$ is the ith row of $\mathbf{A}$ which is a row vector in $\mathbf{R}^n$, while $\mathbf{A}_{\cdot j}$ is the jth column of $\mathbf{A}$. A column vector of ones of arbitrary dimension will be denoted by $\mathbf{e}$. For $\mathbf{A} \in \mathbf{R}^{m \times n}$ and $\mathbf{B} \in \mathbf{R}^{n \times k}$, the kernel $K(\mathbf{A}, \mathbf{B})$ maps $\mathbf{R}^{m \times n} \times \mathbf{R}^{n \times k}$ into $\mathbf{R}^{m \times k}$. In this note, we will make use of the following Gaussian kernel that is frequently used in SVM literature:

$$(K(\mathbf{A}, \mathbf{B}))_{ij} = \varepsilon^{-\mu \|\mathbf{A}_i' - \mathbf{B}_{\cdot j}\|^2}, \quad i = 1, \cdots, m, \quad j = 1, \cdots k,$$

where μ is a positive constant and ε is natural logarithm. The identity matrix of arbitrary dimension will be denoted by $\mathbf{I}$. The vector $\mathbf{y}$ always refers to the error variable if there is no special declaration.

2 PSVM

This section will briefly introduce PSVM proposed by Glenn Fung and Olvi

L.Mangasrian in [1].

First we present the mathematical model of standard SVM. Consider the problem of classifying m points

$$(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)$$

where $\mathbf{x}_i \in \mathbf{R}^n$ is given a label $y_i \in \{-1, 1\}$. The standard SVM classifier is constructed by maximizing the separating planes margin, that is, the distance between the parallel planes. This data classification problem can be regarded as the following QP problem with a given parameter ν

$$\begin{aligned} &\min_{(\mathbf{w},r,\mathbf{y})\in R^{n+1+m}} \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{w}'\mathbf{w} + r^2) \\ &\text{s.t.} \quad \mathbf{D}(\mathbf{A}\mathbf{w} - \mathbf{e}r) + \mathbf{y} = \mathbf{e} \qquad \mathbf{y} \geq 0 \end{aligned} \tag{1}$$

where $m \times n$ matrix $\mathbf{A}$ represents the points, and a given diagonal matrix $\mathbf{D}$ is specified with plus ones and minus ones along its diagonal by the membership of each point $\mathbf{A}_i$ in the class A+ or A-. That is

$$\begin{aligned} \mathbf{A}_i\mathbf{w} \geq r + 1 \quad &\text{for} \quad D_{ii} = 1 \\ \mathbf{A}_i\mathbf{w} \leq r - 1 \quad &\text{for} \quad D_{ii} = -1 \end{aligned} \tag{2}$$

PSVM modifies this formulation based on maximizes the separating margin, which is the distance $1/(\|\mathbf{w}\|^2 + r^2)$, and the formulation (1) is replaced by the following problem:

$$\begin{aligned} &\min_{(\mathbf{w},r,\mathbf{y})\in R^{n+1+m}} \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{w}'\mathbf{w} + r^2) \\ &\text{s.t.} \quad \mathbf{D}(\mathbf{A}\mathbf{w} - \mathbf{e}r) + \mathbf{y} = \mathbf{e} \end{aligned} \tag{3}$$

To solve the optimization problem (3) with equality constraint, we construct the Lagrangian function:

$$L(\mathbf{w}, r, \mathbf{y}, \mathbf{u}) = \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{w}'\mathbf{w} + r^2) - \mathbf{u}'[\mathbf{D}(\mathbf{A}\mathbf{w} - \mathbf{e}r) + \mathbf{y} - \mathbf{e}] \tag{4}$$

where $\mathbf{u} \in \mathbf{R}^m$ is the Lagrangian multiplier associated with the equality constraint of (3). Based on the Karush-Kuhn-Tucker (KKT) conditions, we set the gradients of L equal to zero and obtain the following KKT optimality conditions:

$$\begin{aligned} \mathbf{w} - \mathbf{A}'\mathbf{D}\mathbf{u} &= 0 \\ r + \mathbf{e}'\mathbf{D}\mathbf{u} &= 0 \\ \nu\mathbf{y} - \mathbf{u} &= 0 \\ \mathbf{D}(\mathbf{A}\mathbf{w} - \mathbf{e}r) + \mathbf{y} - \mathbf{e} &= 0 \end{aligned} \tag{5}$$

The first three optimality conditions of (5) give the following expressions for the variables $(\mathbf{w}, r, \mathbf{y})$ in the optimization problem (3) in terms of the Lagrange multiplier $\mathbf{u}$

$$\mathbf{w} = \mathbf{A}'\mathbf{D}\mathbf{u}, \quad r = -\mathbf{e}'\mathbf{D}\mathbf{u}, \quad \mathbf{y} = \frac{\mathbf{u}}{\nu} \tag{6}$$

Substituting these expressions in the last equality of (5) allows us to obtain an explicit

expression for $\mathbf{u}$ in terms of $\mathbf{A}$ and $\mathbf{D}$ as follows:

$$\begin{aligned}\mathbf{u} &= (\frac{\mathbf{I}}{\nu} + \mathbf{D}(\mathbf{A}\mathbf{A}'+\mathbf{e}\mathbf{e}')\mathbf{D})^{-1}\mathbf{e} \\ &= (\frac{\mathbf{I}}{\nu} + \mathbf{H}\mathbf{H}')^{-1}\mathbf{e}\end{aligned} \tag{7}$$

where $\mathbf{H}$ is defined as:

$$\mathbf{H} = \mathbf{D}[\mathbf{A} \quad -\mathbf{e}]. \tag{8}$$

Because the solution (7) for $\mathbf{u}$ includes an inversion of a possibly massive $m \times m$ matrix, we make use of the Sherman-Morrison-Woodbury formula [10] for matrix inversion. And (7) is replaced by the following expression which just computes the inversion of a $(n+1)\times(n+1)$ matrix

$$\mathbf{u} = \nu(\mathbf{I} - \mathbf{H}(\frac{\mathbf{I}}{\nu} + \mathbf{H}'\mathbf{H})^{-1}\mathbf{H}') \tag{9}$$

We know, in most cases, $n << m$ is valid. So the computational complexity of linear PSVM is reduced greatly.

The decision function for linear case is

$$\mathbf{x}'\mathbf{w} - r \begin{cases} > 0 & \text{then} \quad \mathbf{x} \in A+ \\ < 0 & \text{then} \quad \mathbf{x} \in A- \\ = 0 & \text{then } \mathbf{x} \in A- \text{ or } \mathbf{x} \in A- \end{cases} \tag{10}$$

The following parts discuss the nonlinear case for PSVM. Replacing the primal variables $\mathbf{w}$ of the equality constrained optimization problem (3) by its dual equivalent $\mathbf{w} = \mathbf{A}'\mathbf{D}\mathbf{u}$ from (6) to obtain:

$$\min_{(w,r,y)\in R^{n+1+m}} \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{u}'\mathbf{u} + r^2)$$

$$\text{s.t. } \mathbf{D}(\mathbf{A}\mathbf{A}'\mathbf{D}\mathbf{u} - \mathbf{e}r) + \mathbf{y} = \mathbf{e} \tag{11}$$

We replace the linear kernel $\mathbf{A}\mathbf{A}'$ by the nonlinear Gaussian kernel $K(\mathbf{A},\mathbf{A}')$, which has been introduced in Section 1, and obtain the optimization problem for the nonlinear PSVM:

$$\min_{(w,r,y)\in R^{n+1+m}} \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{u}'\mathbf{u} + r^2)$$

$$\text{s.t. } \mathbf{D}(K(\mathbf{A},\mathbf{A}')\mathbf{D}\mathbf{u} - \mathbf{e}r) + r = \mathbf{e} \tag{12}$$

For simplicity, we represent the matrix $K(\mathbf{A},\mathbf{A}')$ by $\mathbf{K}$. Similar to linear PSVM, the Lagrangian multiplier $\mathbf{z}$ according to the equality constraint (12) is given as follows:

$$\mathbf{z} = (\frac{\mathbf{I}}{v} + \mathbf{D}(\mathbf{KK'} + \mathbf{ee'})\mathbf{D})^{-1}\mathbf{e} = (\frac{\mathbf{I}}{v} + \mathbf{GG'})^{-1} \qquad (13)$$

where **G** is defined as:

$$\mathbf{G} = \mathbf{D}[\mathbf{K} \quad -\mathbf{e}] \qquad (14)$$

According to (6), the separating hyperplane can be converted to as follows:

$$\mathbf{x'w} - r = \mathbf{x'A'Du} - r = 0 \qquad (15)$$

We replace $\mathbf{x'A'}$ by the kernel expression $K(\mathbf{x'}, \mathbf{A'})$. According to the KKT optimality condition $\mathbf{u} = \mathbf{DK'Dv}$ and $r = -\mathbf{e'Dv}$, we obtain the nonlinear classifier:

$$(K(\mathbf{x'}, \mathbf{A'})K(\mathbf{A}, \mathbf{A'}) + \mathbf{e'})\mathbf{Dz} \begin{cases} > 0 & \text{then} \quad \mathbf{x} \in A+ \\ < 0 & \text{then} \quad \mathbf{x} \in A- \\ = 0 & \text{then} \quad \mathbf{x} \in A+ \text{ or } \mathbf{x} \in A- \end{cases} \qquad (16)$$

The extensions of PSVM also can be found in [7], [11] and [12].

3 Weighted Proximal Support Vector Machines

3.1 Analysis of PSVM

Obviously, PSVM is much more computationally efficient than standard SVM. But there are still some limitations in the theory. From the formulation discussed above, each point belongs to either of the classes. And in each class, we can easily check that all training samples are treated uniformly in the theory of PSVM. In real life, the effects of training samples sometimes are different. For example, when some training samples are polluted by noise, the clean training samples are more important than those polluted points in the classification problem. The meaningful training points must be classified correctly while the misclassification of the others like noise should be ignored. The theory of PSVM is not suitable to the case, which results in bad performance of PSVM in the case of noisy training set.

Base on the idea presented above, we think, the training point polluted by noise should not be regarded as a full point belonging to one of two class. That is, it may stand 80% possibility to belong to one class and 20% is meaningless. Namely, there should be a weight value $0 < s \leq 1$ applied to the training point, which describe the attitude of the point belonging to one of the two classes.

3.2 Reformulation to WPSVM

Suppose we are given m linear separable points $(y_i, \mathbf{x}_i)$, each of the

points $\mathbf{x}_i \in R^n$ belongs to either of two classes and is given a label $y_i \in \{-1, 1\}$ for $i = 1 \cdots m$. Let s_i be the attitude for the point $\mathbf{x}_i$ toward one class and we weight the error variable vector $\mathbf{y}$ by weight value matrix $\mathbf{S}$, which is a diagonal matrix given by

$$\mathbf{S} = \text{diag}\{s_1, \cdots, s_m\}.$$

This leads to the optimization problem:

$$\begin{aligned} &\min_{(w,r,y)\in R^{n+1+m}} \frac{\nu}{2}\|\mathbf{Sy}\|^2 + \frac{1}{2}(\mathbf{w}'\mathbf{w} + r^2) \\ &\text{s.t.} \quad \mathbf{D}(\mathbf{Aw} - \mathbf{e}r) + \mathbf{y} = \mathbf{e} \end{aligned} \tag{17}$$

The Lagrangian becomes

$$L(\mathbf{w}, r, \mathbf{y}, \mathbf{u}) = \frac{\nu}{2}\|\mathbf{Sy}\|^2 + \frac{1}{2}(\mathbf{w}'\mathbf{w} + r^2) - \mathbf{u}'[\mathbf{D}(\mathbf{Aw} - \mathbf{e}r) + \mathbf{y} - \mathbf{e}]$$

Here, $\mathbf{u} \in \mathbf{R}^m$ is the Lagrange multiplier associated with the equality constraint of (17). Setting the gradients of L equal to zero gives the following KKT optimality conditions:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \mathbf{A}'\mathbf{Du} &= 0 \\ \frac{\partial L}{\partial r} &= r + \mathbf{e}'\mathbf{Du} &= 0 \\ \frac{\partial L}{\partial \mathbf{y}} &= \nu\mathbf{S}^2\mathbf{y} - \mathbf{u} &= 0 \\ \frac{\partial L}{\partial \mathbf{u}} &= \mathbf{D}(\mathbf{Aw} - \mathbf{e}r) + \mathbf{y} - \mathbf{e} = 0 \end{aligned}$$

From the system of linear equations, we can get explicit formulation of **u**, **w** and **y**

$$\mathbf{u} = [\mathbf{HH}' + \mathbf{S}^{-2}/\nu]\mathbf{e} \tag{18}$$

$$r = -\mathbf{e}'\mathbf{Du}, \quad \mathbf{w} = \mathbf{A}'\mathbf{Du} \quad \text{and} \quad \mathbf{y} = \frac{\mathbf{S}^{-2}\mathbf{u}}{\nu} \tag{19}$$

The decision function for linear case is

$$\mathbf{x}'\mathbf{w} - r \begin{cases} > 0 & \text{then} \quad \mathbf{x} \in A+ \\ < 0 & \text{then} \quad \mathbf{x} \in A- \\ = 0 & \text{then} \quad \mathbf{x} \in A- \text{ or } \mathbf{x} \in A- \end{cases} \tag{20}$$

By letting $\mathbf{H} = \mathbf{D}[\mathbf{A} \quad -\mathbf{e}]$, we implement the Sherman -Morrison-Woodbury formula to (18) and obtain:

$$\mathbf{u} = \{\nu\mathbf{S}^2 - [\nu\mathbf{S}^2\mathbf{H}(\mathbf{I} + \mathbf{H}'\mathbf{S}^2\mathbf{H})^{-1}\mathbf{H}'\nu\mathbf{S}^2]\mathbf{e} \tag{21}$$

This expression includes an inversion of $(n+1)\times(n+1)$ matrix, which should be much simpler than the inversion of $m \times m$ matrix in (17) in the case of $n << m$.

3.3 Generating Weight values

The weight values can be generated in a simple way as below. Since each class of data should satisfy an unknown statistical distribution, there should be an center in each class. According to Chebyshev inequality [11], the training point close to the center is supposed to stand more chance to be an unpolluted point and make greater effect on the separating hyperplane formulation. The training samples far away from the center should have less effect. Following this idea, for a given point x, we let weight value as below:

$$s = 1 - d/(R+q) \qquad (22)$$

Here s represents weight value associated with $\mathbf{x}$, d is the Euclidian distance between the center and the training point $\mathbf{x}$ while R is the radius of the class, and q is a given tune positive real number which prevents s from being zero.

3.4 Algorithms for Linear WPSVM

Given m data points in $\mathbf{R}^n$ represented by the $m \times n$ matrix $\mathbf{A}$ and a diagonal matrix $\mathbf{D}$ of ± 1 labels denoting the class of each row of $\mathbf{A}$, we generate the linear classifier (20) as follows:

1. *In the class 1 and −1 of training dataset, we implement the following procedure respectively: let the center equal to the mean vector of the m data points, and then compute every Euclidian distance d between center and every training sample. The radius R is defined as the biggest d.*

2. *According to (22), we obtain every weight value applied to each sample for a given positive q, that is, the weight value matrix* $\mathbf{S}$ *is defined.*

3. *According to (8),* $\mathbf{H}$ *is defined where* $\mathbf{e}$ *is an* $m \times 1$ *vector of ones and compute* $\mathbf{u}$ *by (21) for some positive v. Typically v is chosen by means of a tuning (validating) set.*

4. *Determine (w, r) from (19).*

5. *Classify a new* $\mathbf{x}$ *by using (20).*

3.5 Analysis of WPSVM

In this subsection, we prove that the optimal problem of linear WPSVM is identical to a weighted ridge regression, which is known in the statistical community.

In fact, the optimization problem (17) can be converted as follows:

$$\min_{(w,r,y)\in R^{n+1+m}} \nu \mathbf{y}'\mathbf{S}\mathbf{S}\mathbf{y}/2 + (\mathbf{w}\mathbf{w}'+r^2)/2 \qquad \mathbf{D}(\mathbf{A}\mathbf{w}-\mathbf{e}r)+\mathbf{y}=\mathbf{e}$$

$$\Leftrightarrow \min_{(w,r)\in R^{n+1}} \nu[(\mathbf{e}-\mathbf{D}(\mathbf{A}\mathbf{w})-\mathbf{e}r)'\mathbf{S}\mathbf{S}(\mathbf{e}-\mathbf{D}(\mathbf{A}\mathbf{w})-\mathbf{e}r)] + (\mathbf{w}'\mathbf{w}+r^2)$$

$$\Leftrightarrow \min_{(w,r)\in R^{n+1}} \nu|\mathbf{e}-\mathbf{S}\mathbf{D}[\mathbf{A},-\mathbf{e}][\mathbf{w}',r]'|^2 + (\mathbf{w}'\mathbf{w}+r^2)$$

$$\Leftrightarrow \min_{(w,r)\in R^{n+1}} |\mathbf{S}[\mathbf{A},-\mathbf{e}][\mathbf{w}',r]'-\mathbf{D}\mathbf{e}|^2 + \frac{1}{\nu}|[\mathbf{w},r]|^2$$

which is weighted ridge regression actually.

In this ridge regression model, an additional dimension is added for input vector $\mathbf{x}$ with corresponding weight such that $\hat{\mathbf{x}}_i = s_i\begin{pmatrix}\mathbf{x}_i\\1\end{pmatrix}$ and the bias term is incorporated in the weight vector $\hat{\mathbf{w}}=\begin{pmatrix}\mathbf{w}\\r\end{pmatrix}$. The problem is converted to fit the points $(\hat{\mathbf{x}}, y_i)$ by $y=\hat{\mathbf{w}}'\mathbf{x}$, where y_i is the corresponding label.

We believe the equivalence of ridge regression and WPSVM with linear kernels enable us to take advantage of the rich statistics literature to estimate the tuning parameter more effectively, an issue that was ignored in earlier work.

3.6 Nonlinear Case

In most cases, the searching of suitable hyperplane in an input space is too restrictive to be of practical use. A solution of this situation is mapping the input space into a higher dimension feature space and searching the optimal hyperplane in this feature space. Here we introduce a nonlinear function φ, by which the dataset is mapped to a higher dimensional space where the samples is linear separable. According to (19), the equality constraint (17) can be replaced by

$$\mathbf{D}(\mathbf{A}\mathbf{A}'\mathbf{D}\mathbf{u}-\mathbf{e}r)+\mathbf{y}=\mathbf{e}\ .$$

After mapping the input space to a feature space by φ, the constraint is converted to

$$\mathbf{D}(\varphi(\mathbf{A})\varphi(\mathbf{A}')\mathbf{D}\mathbf{u}-\mathbf{e}r)+\mathbf{y}=\mathbf{e}\ .$$

Sometime it is difficult to obtain the explicit expression of φ. We just only need to know a function K called *kernel* that can compute the dot product of data points in the feature space, that is

$$\varphi(\mathbf{A})\varphi(\mathbf{A}')=K(\mathbf{A},\mathbf{A}')\ .$$

The optimization problem for WPSVM in the nonlinear case is represented as follows:

$$\min_{(w,r,y)\in R^{n+1+m}} \frac{\nu}{2}\|\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{u}'\mathbf{u}+r^2) \qquad \text{s.t. } \mathbf{D}(K(\mathbf{A},\mathbf{A}')\mathbf{D}\mathbf{u}-\mathbf{e}r)+\mathbf{y}=\mathbf{e} \qquad (23)$$

For simplicity, we represent the matrix $K(\mathbf{A},\mathbf{A}')$ with $\mathbf{K}$. The corresponding Lagrangain can be written as:

$$L(\mathbf{w},r,\mathbf{y},\mathbf{z})=\frac{\nu}{2}\|\mathbf{S}\mathbf{y}\|^2 + \frac{1}{2}(\mathbf{u}'\mathbf{u}+r^2) - \mathbf{z}'[\mathbf{D}(\mathbf{K}\mathbf{D}\mathbf{u}-\mathbf{e}r)+\mathbf{y}-\mathbf{e}]$$

where $\mathbf{z} \in \mathbf{R}^m$ is the Lagrange multiplier associated with the equality constraint of (23).

The KKT condition comes to

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{u}} = \mathbf{u} - \mathbf{DK'Dz} = 0 \\ \dfrac{\partial l}{\partial r} = r + \mathbf{e'Dz} = 0 \\ \dfrac{\partial L}{\partial \mathbf{y}} = v\mathbf{S}^2\mathbf{y} - \mathbf{z} = 0 \\ \dfrac{\partial L}{\partial \mathbf{z}} = \mathbf{D}(\mathbf{KDu} - \mathbf{e}r) + \mathbf{y} - \mathbf{e} = 0 \end{cases} \tag{24}$$

and we get

$$r = -\mathbf{e'Dz}, \quad \mathbf{u} = \mathbf{DK'Dz} \quad \text{and} \quad \mathbf{y} = \frac{\mathbf{S}^{-2}\mathbf{z}}{v} \tag{25}$$

Substituting these expressions in the last equality of (24) gives an explicit expression for $\mathbf{z}$

$$\mathbf{z} = [(\mathbf{S}^{-1})^2 / v + \mathbf{GG'}]^{-1}\mathbf{Se} \tag{26}$$

where

$$\mathbf{G} = \mathbf{D}[\mathbf{K} - \mathbf{e}]. \tag{27}$$

The nonlinear separating hyperplane corresponding to the kernel K can be deduced by the linear separating rule (20) and $\mathbf{w} = \mathbf{A'Du}$ from (19) as follows:

$$\mathbf{x'w} - r = \mathbf{x'A'Du} - r = 0$$

Replace $\mathbf{x'A'}$ by the kernel expression $K(\mathbf{x'}, \mathbf{A'})$, and substitute from (25) for $\mathbf{u}$ and r, we obtain the separating surface:

$$\begin{aligned} K(\mathbf{x'}, \mathbf{A})\mathbf{Du} - r &= K(\mathbf{x'A'})\mathbf{DDK}(\mathbf{A}, \mathbf{A'})'\mathbf{Dz} + \mathbf{e'Dz} \\ &= (K(\mathbf{x'}, \mathbf{A'})K(\mathbf{A}, \mathbf{A'})'+\mathbf{e'})\mathbf{Dz} = 0 \end{aligned} \tag{28}$$

Below we give the nonlinear classifier as bellows:

$$(K(\mathbf{x'}, \mathbf{A'})K(\mathbf{A}, \mathbf{A'}) + \mathbf{e'})\mathbf{Dz} \begin{cases} > 0 & \text{then} \quad \mathbf{x} \in A+ \\ < 0 & \text{then} \quad \mathbf{x} \in A- \\ = 0 & \text{then} \quad \mathbf{x} \in A+ \text{ or } \mathbf{x} \in A- \end{cases} \tag{29}$$

Unlike the situation with linear kernels, the Sherman-Morrison-Woodbury formula is useless here because the kernel matrix $\mathbf{K}$ is a square $m \times m$ matrix, so the inversion in (25) cannot be converted to an inversion of $n \times n$ matrix as the linear case.

The reduced kernel techniques of [10] can be used here to reduce the $m \times m$ matrix $\mathbf{K} = K(\mathbf{A}, \mathbf{A'})$ to a much smaller $m \times \bar{m}$ dimensionality of a rectangular kernel $\mathbf{K} = K(\mathbf{A}, \bar{\mathbf{A}}')$, where $\bar{m}$ is as small as 1% of m. In this way, the computational complexity of nonlinear WPSVM is reduced greatly.

We now give an explicit statement of our nonlinear classifier algorithm.

Given m data points in R^n represented by the $m \times n$ matrix $\mathbf{A}$ and a diagonal matrix $\mathbf{D}$

of ± 1 labels denoting the class of each row of **A**, we generate the linear classifier (29) as follows:

Step 1 and 2 is the same as Step 1 and 2 of Algorithm in linear case mentioned above.

3 Choose a kernel function $K(\mathbf{A},\mathbf{A}')$, *typically the Gaussian kernel.*

4 According to (27), **G** *is defined where* **e** *is an* $m\times 1$ *vector of ones and compute* **z** *by (26) for some positive v. Typically v is chosen by means of a tuning (validating) set.*

5 The nonlinear surface (28) with the computed v constitutes the nonlinear classifier (29) for classifying a new point x.

4 Experiments

4.1 Simulation results

This experiment is conducted in the Matlab environment. We generate two classes of points in $\mathbf{R}^2$ randomly. The Figures below show the performance of linear PSVM and WPSVM in an unpolluted setting (without noise) and noisy setting respectively. The circles and crosses represent the training samples in different classes. The dotted line shows classification rule obtained by PSVM while the solid line is obtain by WPSVM. Figure 1 indicates that PSVM can classify data into two classes well before adding noise data. After adding noise, PSVM cannot work any more while WPSVM classifier still do well in the classification problem (Figure 2), which indicate that WPSVM is more robust than PSVM. Points in (-20, -20), (-18, -10), (-19, -20) are the noise we add in the original data.

4.2 Real data result

Heart dataset is obtained from UCI Machine Learning Repository, from which we randomly select 270 samples with 13 attributes. Using different scale of training set, we compare performance of WPSVM with that of PSVM. The experiments indicate that WPSVM is much more robust than PSVM (Table 1). Although a little more time consuming happens in generating the diagonal matrix **S** in executing WPSVM, we believe it still workable in some real life applications for WPSVM's higher accuracy.

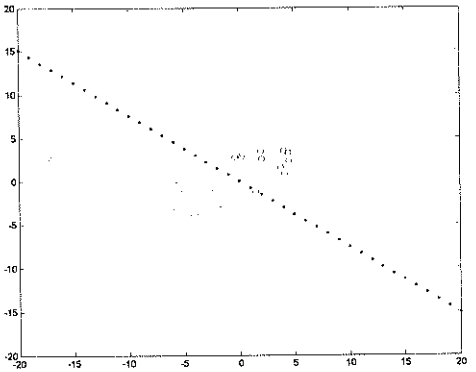

Fig. 1. Performance of PSVM before noise adding

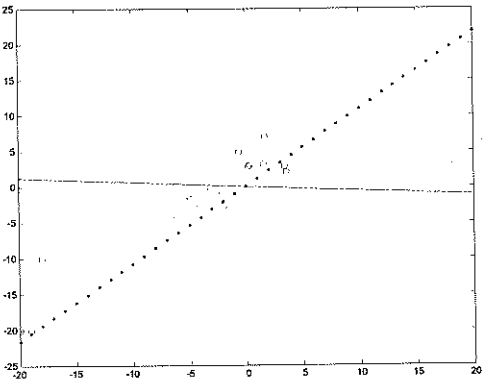

Fig. 2. Performance of PSVM and WPSVM after noise adding

Table 1. Test accuracy comparison between WPSVM and PSVM

The size Of training dataset and noise dataset	WPSVM		PSVM	
	Number of misclassification Test accuracy		Number of misclassification Test Accuracy	
	Before adding noise data	After noise data added	Before adding noise data	After noise data added
40(3)	**47** **79.6%**	**59** **74.3%**	56 75.7%	69 70.0%
80(10)	**32** **83.2%**	**41** **78.4%**	34 82.6%	57 70.0%
160(20)	**14** **87.3%**	**27** **75.6%**	**14** **87.3%**	34 69.1%

5 Conclusion

In this note, we propose a new version of SVM, that is, WPSVM which improve the robustness of PSVM without loss the computationally attractive feature of PSVM. The future work will try to solve robust and sparse regression problem in the framework of PSVM.

References

[1] G. Fung and O. L. Mangasarian, "Proximal support vector machine classifiers", *Proc. 7th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 77-86, 2001.

[2] V. N. Vapnik, *The nature of statistical learning Theory*, Springer, New York, Second edition, 2000.

[3] X. Zhang, "Using class-center vectors to build support vector machines", in *Proc. IEEE NNSP'99*, pp.3-11, 1999.

[4] C. F. Lin and S.D, Wang, "Fuzzy support vector machine," *IEEE Trans. Neural Networks*, vol. 13. March. 2002, pp. 1067-1077

[5] J.C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines", Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[6] J. A. K. Suykens et al. "Least squares support vector machine classifiers", *Neural Process Lett.* 9 (3), 1999 pp.293-300.

[7] D. K. Agarwal and William DuMouchel, "Shrinkage Estimator Generalizations of Proximal Support Vector Machines" *Proc. 8th ACM SIGKDD Intl. Conf. on* Knowledge.Discovery and Data Mining (pp. 173-182), 2002.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.

[9] Papoulis, A Probability, Random Variables, and Stochastic Processes, Second edition. New York: McGraw-Hill, pp. 149-151, 1984.

[10] Y. -J. Lee and O. L. Mangasarian, "RSVM: Reduced support vector machines," *Proc. Of the 1st SIAM Intl Conf. on Data Mining*, Chicago, April 5-7, 2001, CD-ROM.

[11] Glenn Fung, O. L. Mangasarian "Multicategory Proximal Support Vector Classifiers," *Machine Learning Journal* submitted

[12] Li kai, Huang Hong-kuan, "Incremental learning proximal support vector machine classifiers" *Proceedings of 2002 International Conference on Machine Learning and Cybernetics*, v3, pp:1635-1637, 2002