Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 135149, 7 pages
http://dx.doi.org/10.1155/2013/135149

Research Article

Piecewise-Smooth Support Vector Machine for Classification

Qing Wu and Wenqing Wang

School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Correspondence should be addressed to Qing Wu; xidianwq@yahoo.com.cn

Received 18 November 2012; Revised 12 March 2013; Accepted 13 March 2013

Academic Editor: Jun Zhao

Support vector machine (SVM) has been applied very successfully in a variety of classification systems. We attempt to solve the primal programming problems of SVM by converting them into smooth unconstrained minimization problems. In this paper, a new twice continuously differentiable piecewise-smooth function is proposed to approximate the plus function, and it issues a piecewise-smooth support vector machine (PWSSVM). The novel method can efficiently handle large-scale and high dimensional problems. The theoretical analysis demonstrates its advantages in efficiency and precision over other smooth functions. PWSSVM is solved using the fast Newton-Armijo algorithm. Experimental results are given to show the training speed and classification performance of our approach.

1. Introduction

In the last several years, support vector machine (SVM) has become one of the most promising learning machines because of its high generalization performance and wide applicability for classification, forecasting, and estimation in small-sample cases [1–6]. In addition, SVMs have surpassed the performance of artificial neural networks in many areas such as text categorization, speech recognition, and bioinformatics [7–11].

Basically, the main idea behind SVM is the construction of an optimal hyper plane, which has been used widely in classification [5, 8–16]. It can be formulated into an unconstrained optimization problem [17–21], but the objective function is nonsmooth. To overcome this disadvantage, Lee and Mangasarian used the integral of the sigmoid function to get a smooth SVM(SSVM) model in 2001 [17]. It is a very important and significant result to SVM since many famous algorithms can be used to solve it. In 2005, Yuan et al. proposed two polynomial functions, namely, the smooth quadratic polynomial function and the smooth forth polynomial function, and got QPSSVM and FPSSVM models [20, 21]. Xiong et al. derived an important recursive formula and a new class of smoothing functions using the technique of interpolation functions in [20]. In 2007, Yuan et al. used a three-order spline function to smooth the objective function of unconstrained optimization problem of SVM and obtained TSSVM model [21]. However, the efficiency or the precision of the algorithms was limited.

A natural problem is whether there is another smooth function to get a more efficient smooth SVM than existing works. In this paper, we introduce a piecewise function to smooth SVM and obtain a novel piecewise smooth support vector machine (PWSSVM). Theoretical analyses show that approximation accuracy of the piecewise smooth function to the plus function is higher than the available. Rough set theory is used to prove the global convergence of PWSSVM and the upper bound of convergence is proposed. The fast Newton-Armijo algorithm [22, 23] is employed to train the PWSSVM. Our new method is implemented in batches and can efficiently handle large-scale and high dimensional problems. Numerical experiments confirm the theoretical results and demonstrate that PWSSVM is more effective than the previous smooth support vector machine models.

The paper is organized as follows. In Section 2, we state the pattern classification and describe the PWSSVM. The approximation performance of smooth functions to the plus function is compared in Section 3. The convergence performance of PWSSVM is given in Section 4. The Newton-Armijo algorithm is applied to train PWSSVM in Section 5. Section 6 shows numerical comparisons. Finally, a brief conclusion of this paper is made.

In this paper, unless otherwise stated, all vectors are column vectors. For a vector x in the n-dimensional real space R^n, the plus function x_+ is defined as $(x_+)_i = \max(0, x_i)$, $i = 1, \ldots, n$. The scalar (inner) product of two vectors x, y in the n-dimensional real space will be denoted by $x \cdot y$ and the p-norm will be denoted by $\| \cdot \|_p$. For a matrix $A \in R^{m \times n}$, A_i is the ith row of A which is a row vector in R^n. A column vector of ones of n dimension will be denoted by e. If φ is a real valued function defined in the n-dimensional real space R^n, the gradient of φ is denoted by $\nabla\varphi(x)$ which is a row vector in R^n and $n \times n$ the Hessian matrix of φ at x is denoted by $\nabla^2\varphi(x)$.

2. Piecewise-Smooth Support Vector Machine

In this paper, let us consider a binary classification problem with m training samples in the n-dimensional real space R^n. It is represented by the $m \times n$ matrix A, according to membership of each point A_i in the class 1 or −1 as specified by a given $m \times m$ diagonal matrix D with 1 or −1 along its diagonal. For this problem, the standard SVM with a linear kernel is given by the following quadratic program with parameter $v > 0$

$$\begin{aligned} \min_{(w,\gamma,y)\in R^{n+1+m}} \quad & v e^{\mathrm{T}} y + \frac{1}{2} w^{\mathrm{T}} w \\ \text{s.t.} \quad & D(Aw - e\gamma) + y \geq e \quad y \geq 0, \end{aligned} \tag{1}$$

where e is a vector of ones, w is the normal to the bounding plane, and b is the distance of the bounding plane to the origin. The linear separating plane is defined as follows:

$$P = \left\{ x_i \mid x_i \in R^n, w^{\mathrm{T}} x_i = b \right\}. \tag{2}$$

The first term in the objective function of (1) is the 1-norm of the slack variable y with weight v. Replace the first term with the 2-norm vector y. Add $(1/2)\gamma^{\mathrm{T}}\gamma$ to the objective function which induces strong convexity but has little or no effect on the problem. SVM model is replaced by the following problem:

$$\begin{aligned} \min_{(w,\gamma,y)\in R^{n+1+m}} \quad & \frac{v}{2} y^{\mathrm{T}} y + \frac{1}{2}\left(w^{\mathrm{T}} w + \gamma^{\mathrm{T}}\gamma\right) \\ \text{s.t.} \quad & D(Aw - e\gamma) + y \geq e \quad y \geq 0. \end{aligned} \tag{3}$$

Let $y = (e - D(Aw - e\gamma))_+$, where $(\cdot)_+$ replaces negative components of a vector by zeros, then we can convert the SVM problem (3) into the following unconstrained optimization problem:

$$\min_{(w,\gamma)\in R^{n+1}} \frac{1}{2} v \left\| \left(e - D(Aw - e\gamma)\right)_+ \right\|^2 + \frac{1}{2}\left(w^{\mathrm{T}} w + \gamma^{\mathrm{T}}\gamma\right). \tag{4}$$

This is a strongly convex minimization problem and it has a unique solution. Let $\max\{0, x\} = x_+$. The function $(x)_+$ is a continuous but nonsmooth function. Therefore, many optimization algorithms based on derivative and gradient cannot solve the problem (4) directly.

In 2001, Lee and Mangasarian [17] employed the integral of the sigmoid function $p(x, k)$ to approximate the nondifferentiable function x_+ as follows:

$$p(x, k) = x + \frac{1}{k}\log\left(1 + \varepsilon^{-kx}\right), \quad k > 0, \tag{5}$$

where ε is the base of natural logarithm and k is a smoothing parameter. They got the SSVM model.

In 2005, Yuan et al. [18] presented two polynomial functions as follows:

$$q(x, k) = \begin{cases} x, & x \geq \dfrac{1}{k}, \\ \dfrac{k}{4}x^2 + \dfrac{1}{2}x + \dfrac{1}{4k}, & -\dfrac{1}{k} < x < \dfrac{1}{k}, \quad k > 0, \\ 0, & x \leq -\dfrac{1}{k}, \end{cases}$$

$$h(x, k) = \begin{cases} x, & x \geq \dfrac{1}{k}, \\ -\dfrac{1}{16k}(kx + 1)^3 (kx - 3), & -\dfrac{1}{k} < x < \dfrac{1}{k}, \quad k > 0, \\ 0, & x \leq -\dfrac{1}{k}. \end{cases} \tag{6}$$

Using the above smooth functions to proximate plus function x_+, they got two smooth polynomial support vector machine models (FPSSVM and QPSSVM). The authors also showed that FPSSVM and QPSSVM were more effective than SSVM in [18].

In 2007, Yuan et al. [21] presented a three-order spline function as follows:

$$T(x, k) = \begin{cases} 0, & x < -\dfrac{1}{k}, \\ \dfrac{k^2}{6}x^3 + \dfrac{k}{2}x^2 + \dfrac{1}{2}x + \dfrac{1}{6k}, & -\dfrac{1}{k} \leq x < 0, \\ -\dfrac{k^2}{6}x^3 + \dfrac{k}{2}x^2 + \dfrac{1}{2}x + \dfrac{1}{6k}, & 0 \leq x \leq \dfrac{1}{k}, \\ x, & x > \dfrac{1}{k}. \end{cases} \tag{7}$$

They used the smooth function $T(x, k)$ to approach the plus function and got a new smooth SVM model TSSVM. However, the efficiency or the precision of these algorithms above was limited.

In this paper, we propose a novel smooth function $\varphi(x, k)$ with smoothing parameter $k > 0$ to approximate to the function x_+ as follows:

$$\varphi(x, k) = \begin{cases} 0, & x < -\dfrac{1}{3k}, \\ \dfrac{3}{2}k^2\left(x + \dfrac{1}{3k}\right)^3, & -\dfrac{1}{3k} \leq x < 0, \\ x + \dfrac{3}{2}k^2\left(\dfrac{1}{3k} - x\right)^3, & 0 \leq x \leq \dfrac{1}{3k}, \\ x, & x > \dfrac{1}{3k}. \end{cases} \tag{8}$$

The first- and second-order derivatives of $\varphi(x,k)$ are

$$\nabla\varphi(x,k)=\begin{cases}0, & x<-\dfrac{1}{3k},\\ \dfrac{9}{2}k^2\left(x+\dfrac{1}{3k}\right)^2, & -\dfrac{1}{3k}\le x<0,\\ 1-\dfrac{9}{2}k^2\left(\dfrac{1}{3k}-x\right)^2, & 0\le x\le\dfrac{1}{3k},\\ 1, & x>\dfrac{1}{3k},\end{cases}$$

$$\nabla^2\varphi(x,k)=\begin{cases}0, & x<-\dfrac{1}{3k},\\ 9k^2\left(x+\dfrac{1}{3k}\right), & -\dfrac{1}{3k}\le x<0,\\ 9k^2\left(\dfrac{1}{3k}-x\right), & 0\le x\le\dfrac{1}{3k},\\ 0, & x>\dfrac{1}{3k}.\end{cases} \tag{9}$$

The solution of the problem (3) can be obtained by solving the following smooth unconstrained optimization problem with the smoothing parameter k approaching infinity as

$$\min_{(w,\gamma)\in R^{n+1}}\Phi_k(w,\gamma)=\frac{1}{2}\upsilon\|\varphi(e-D(Aw-e\gamma),k)\|^2+\frac{1}{2}\left(w^{\mathrm{T}}w+\gamma^{\mathrm{T}}\gamma\right). \tag{10}$$

Thus, we develop a new smooth approximation for problem (3).

3. Approximation Performance Analysis of Smooth Functions

In this section, we will compare the approximation performance of smooth functions to plus function.

Lemma 1 (see [17]). *$p(x,k)$ is defined as the integral of the sigmoid function in [17], and x_+ is the plus function. The following conclusions can be obtained:*

(1) *$p(x,k)$ is arbitrary rank smooth about x;*

(2) *$p(x,k)\ge x_+$;*

(3) *for $\rho>0$, $|x|<\rho$, $p(x,k)^2-x_+^2\le(\log 2/k)^2+(2\rho/k)\log 2$.*

Lemma 2 (see [18]). *$q(x,k)$ and $h(x,k)$ are defined as (6), and x_+ is plus function. We can obtain the following conclusions:*

(1) *$q(x,k)$ is one rank smooth about x, and $h(x,k)$ is twice rank smooth about x;*

(2) *$q(x,k)\ge x_+$ and $h(x,k)\ge x_+$;*

(3) *for any $x\in R$, $q(x,k)^2-x_+^2\le 1/11k^2$ and $h(x,k)^2-x_+^2\le 1/19k^2$.*

Lemma 3 (see [19]). *$T(x,k)$ is defined as (7), and x_+ is plus function. The following results are easily obtained:*

(1) *$T(x,k)$ is twice rank smooth about x;*

(2) *$T(x,k)\ge x_+$;*

(3) *for any $x\in R$, $T(x,k)^2-x_+^2\le 1/24k^2$.*

Theorem 4. *The piecewise approximation function $\varphi(x,k)$ defined in (8) has the following properties:*

(1) *$\varphi(x,k)$ is twice rank smooth about x;*

(2) *for any $x\in R$, $\varphi(x,k)\ge x_+$,*

(3) *for any $x\in R$, then $\varphi(x,k)^2-x_+^2\le 1/216k^2$.*

Proof. (1) According to the formulas (8) and (9), one can easily obtain the results in (1).

(2) In the following, we verify the fact $\varphi(x,k)\ge(x)_+\ge 0$. (i) The equation $\varphi(x,k)=(x)_+$ holds while $x\in(-\infty,-(1/3k))\cup(1/3k,\infty)$. (ii) Since $\varphi(x,k)$ is a monotone increasing function, we have the following result $\varphi(x,k)-(x)_+=\varphi(x,k)\ge\varphi(-(1/3k),k)=0$ while $x\in[-(1/3k),0)$. (iii) For $x\in[0,1/3k]$, we have $\varphi(x,k)-(x)_+=(3/2)k^2(1/3k-x)^3\ge 0$. Hence, we have the conclusion $\varphi(x,k)\ge(x)_+\ge 0$.

(3) For $x\in(-\infty,-(1/3k))\cup(1/3k,\infty)$, $\varphi(x,k)^2-x_+^2=0$, the inequality in conclusion (3) is satisfied naturally.

For $-(1/3k)\le x\le 0$, since $x_+=0$, $\varphi(x,k)^2-x_+^2=\varphi(x,k)^2$. Because $\varphi(x,k)$ is positive value, continuous, and increasing function for $-(1/3k)\le x\le 0$, we have $\varphi(x,k)^2\le\varphi(0,k)^2=1/324k^2\le 1/216k^2$.

For $0<x\le 1/3k$, let

$$\begin{aligned}s(x)=\varphi(x,k)^2-x_+^2&=\left(x+\frac{3}{2}k^2\left(\frac{1}{3k}-x\right)^3\right)^2-x^2\\&=\frac{9}{4}k^4\left(x-\frac{1}{3k}\right)^6-3k^2x\left(x-\frac{1}{3k}\right)^3.\end{aligned} \tag{11}$$

In order to obtain the result, making the variable substitution $t=kx$ (obviously $t\in(0,1/3)$), then we have $s(t)=(3/k^2)[(3/4)(t-(1/3))^6-t(t-(1/3))^3]$. For $t\in(0,1/3)$, the maximum point of $s(t)$ is $t=0.0605$ and $s(t)=\varphi^2(x,k)-(x)_+^2\le g(0.0605)\approx 0.0046/k^2\le 1/216k^2$.

In conclusion, we have $\varphi(x,k)^2-x_+^2\le 1/216k^2$. □

Theorem 5. *Let $\rho=1/k$, and $k>0$. Consider the following.*

(1) *If the smooth function is defined as (5), then by Lemma 1, we have*

$$\begin{aligned}p(x,k)^2-x_+^2&\le\left(\frac{\log 2}{k}\right)^2+\frac{2\rho}{k}\log 2\\&=\left(\log^2 2+2\log 2\right)\frac{1}{k^2}\approx 0.6927\frac{1}{k^2}.\end{aligned} \tag{12}$$

(2) *If the smooth functions are defined as (6), by Lemma 2,*

$$\begin{aligned}q(x,k)^2-x_+^2&\le\frac{1}{11k^2}\approx 0.0909\frac{1}{k^2},\\ h(x,k)^2-x_+^2&\le\frac{1}{19k^2}\approx 0.0526\frac{1}{k^2}.\end{aligned} \tag{13}$$

(3) *If the smooth function is defined as* (7), *by Lemma 3,*

$$T(x,k)^2 - x_+^2 \le \frac{1}{24k^2} \approx 0.0417\frac{1}{k^2}. \tag{14}$$

(4) *If the smooth function is defined as* (8), *by Theorem 4,*

$$\varphi(x,k)^2 - x_+^2 \le \frac{1}{216k^2} \approx 0.0046\frac{1}{k^2}. \tag{15}$$

Theorem 5 shows that the proposed piecewise smooth function $\varphi(x,k)$ achieves the best degree of approximation to the plus function x_+. When k is fixed, it is easy to obtain the different smooth capability of the above smooth functions. The smooth performance comparison is given in Figure 1, where we set the smooth parameter $k = 10$ and $\rho = 1/k$.

4. Convergence Performance of PWSSVM

In this section, the convergence of PWSSVM will be presented. By using rough set theory, we prove that the solution of PWSSVM can closely approximate the optimal solution of the original model (4) when k goes to infinity. Furthermore, a formula for computing the upper bound of convergence is deduced.

Theorem 6. *Let $A \in R^{m\times n}$ and $b \in R^{m\times 1}$. Define the real-valued functions in the n-dimensional real space R^n as follows:*

$$\begin{aligned} f(x) &= \frac{1}{2}\left\|(Ax-b)_+\right\|_2^2 + \frac{1}{2}\|x\|_2^2, \\ g(x,k) &= \frac{1}{2}\left\|\varphi(Ax-b,k)\right\|_2^2 + \frac{1}{2}\|x\|_2^2, \end{aligned} \tag{16}$$

where $\varphi(\cdot)$ is defined in (8), *$k > 0$. Then we have the following results:*

(1) *$f(x)$ and $g(x,k)$ are strongly convex functions;*

(2) *there exists a unique solution x^* to $\min_{x\in R^n} f(x)$, and a unique solution x_k^* to $\min_{x\in R^n} g(x,k)$;*

(3) *for any $k > 0$, x_k^* and x^* satisfy the following condition:*

$$\left\|x_k^* - x^*\right\|^2 \le \frac{m}{432k^2}; \tag{17}$$

(4) $\lim_{k\to\infty}\|x_k^* - x^*\| = 0$.

Proof. (1) For any $k > 0$, $f(x)$ and $g(x,k)$ are strongly convex functions because $\|\cdot\|^2$ is strong convex function.

(2) Let $L_\nu(f(x))$ be the level set of $f(x)$ and let $L_\nu(g(x,k))$ be the level set of $g(x,k)$. Since $x_+ \le \varphi(x,k)$, it is easy to obtain $L_\nu(g(x,k)) \subset L_\nu(f(x)) \subset \{x \mid \|x\|_2^2 \le 2\nu\}$. Therefore, $L_\nu(f(x))$ and $L_\nu(g(x,k))$ are compact subsets in R^n. Using the strong convexity property of $f(x)$ and $g(x,k)$ for $k > 0$, there is a unique solution to $\min_{x\in R^n} f(x)$ and $\min_{x\in R^n} g(x,k)$, respectively.

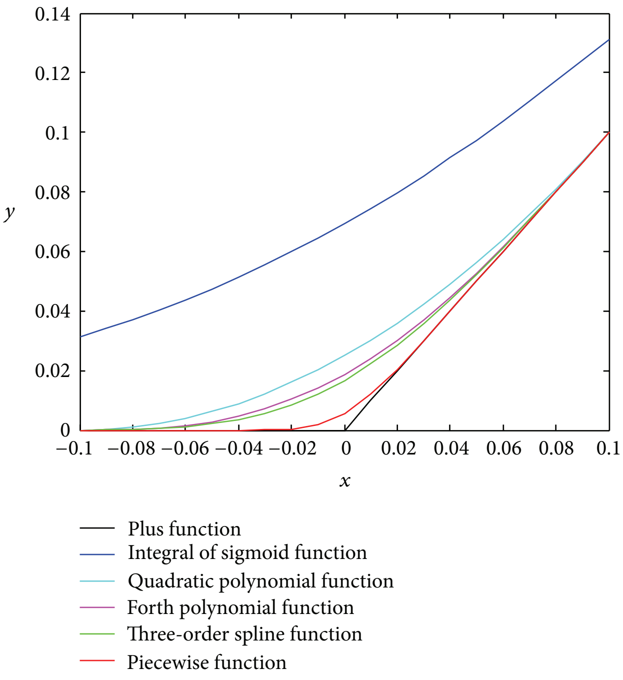


FIGURE 1: Comparison of approximation performance of smooth functions ($k = 10$).

(3) By using the first order optimization condition and considering convex property of $f(x)$ and $g(x,k)$, we have

$$\begin{aligned} f(x_k^*) - f(x^*) &\ge \nabla f(x^*)(x_k^* - x^*) + \frac{1}{2}\left\|x_k^* - x^*\right\|_2^2 \\ &= \frac{1}{2}\left\|x_k^* - x^*\right\|_2^2, \\ g(x^*,k) - g(x_k^*,k) &\ge \nabla g(x_k^*,k)(x^* - x_k^*) + \frac{1}{2}\left\|x_k^* - x^*\right\|_2^2 \\ &= \frac{1}{2}\left\|x_k^* - x^*\right\|_2^2. \end{aligned} \tag{18}$$

Add the two formulas above and notice that $\varphi(x,k) \ge x_+$, and then we have

$$\begin{aligned} \left\|x_k^* - x^*\right\|_2^2 &\le f(x_k^*) - f(x^*) + g(x^*,k) - g(x_k^*,k) \\ &= (g(x^*,k) - f(x^*)) - (g(x_k^*,k) - f(x_k^*)) \\ &\le g(x^*,k) - f(x^*) \\ &= \frac{1}{2}\left\|\varphi(Ax^* - b,k)\right\|_2^2 - \frac{1}{2}\left\|(Ax^* - b)_+\right\|_2^2. \end{aligned} \tag{19}$$

According to the third result of Theorem 4, we obtain the conclusion $\|x_k^* - x^*\|^2 \le m/432k^2$.

(4) According to $\|x_k^* - x^*\|^2 \le m/432k^2$, we have $\lim_{k\to\infty} x_k^* = x^*$. □

5. The Newton-Armijo Algorithm for PWSSVM

Following the results of the previous section, one can obtain the twice continuous differentiability of the objective function of problem (10). In order to take advantage of this feature, we use the Newton-Armijo method to train PWSSVM since it is a faster method than the BFGS algorithm [18, 19, 21]. The Newton-Armijo algorithm for problem (10) works as follows.

5.1. Newton-Armijo Algorithm

Step 1 (initialization). Start with any $(w^0, \gamma^0) \in R^{n+1}$, τ and set $i := 0$.

Step 2. Compute $\Phi^i = \Phi_k(w^i, \gamma^i)$ and $g^i = \nabla\Phi_k(w^i, \gamma^i)$.

Step 3. If $\|g^i\|_2 \le \tau$, then stop and accept (w^i, γ^i). Otherwise, compute Newton direction $d^i \in R^{n+1}$ from the following linear system:

$$\nabla^2\Phi\left(w^i, \gamma^i; k^i\right) d^i = -\left(g^i\right)^{\mathrm{T}}, \tag{20}$$

where "T" denotes transpose symbol.

Step 4 (Armijo stepsize). Choose a stepsize $\lambda_i = \max\{1, 1/2, 1/4, \ldots\}$ such that

$$\Phi_k\left(w^i, \gamma^i\right) - \Phi_k\left(\left(w^i, \gamma^i\right) + \lambda_i d^i\right) \ge -\rho\lambda_i g^i d^i, \tag{21}$$

where $\rho \in (0, 1/2)$ and set

$$\left(w^{i+1}, \gamma^{i+1}\right) = \left(w^i, \gamma^i\right) + \lambda_i d^i. \tag{22}$$

Step 5. Replace i by $i + 1$ and go to Step 2.

We need to only solve a linear system of (20) instead of a quadratic program in our smooth approach. Because the objective function is strong convex, it is not difficult to obtain that our Newton-Armijo algorithm for training PWSSVM converges globally to the unique solution [17, 23]. Hence, the start point is not important. In this paper, we always set $(w^0, \gamma^0) = e$, where e denotes a column vector of ones of n dimension.

PWSSVM described above can solve the linear classification problems. In fact, we can extend some of the results in Section 2 to nonlinear PWSSVM with kernel technique as [17]. Furthermore, The Newton-Armijo algorithm can also solve nonlinear PWSSVM successfully.

6. Numerical Experiments

Newton-Armijo cannot be applied to QPSSVM model due to lack of the second-order derivative. In fact, the classification capacity of FPSSVM is slightly better than QPSSVM [18–21]. In our experiment, we do not compare QPSSVM with the other smooth SVM method. To demonstrate the effectiveness and speed of PWSSVM, we compare the performance numerically among SSVM, FPSSVM, TSSVM, and PWSSVM. The four smooth SVMs are all trained by the fast Newton-Armijo algorithm. All experiments are run on Personal Computer with 3.0 GHz and a maximum of 1.99 GB of the memory available. The programs of PWSVM, FPSSVM, and TSSVM are written in the MATLAB language. This computer runs Win7 with MATLAB 7.0.1. The source code of SSVM, "ssvm.m," is obtained from the author's website for the linear problem [24], and "lsvmk.m" for the nonlinear problem. In our experiments, all of the input data and the variables needed in programs are kept in the memory. For SSVM, TSSVM, FPSSVM, and PWSSVM, an optimality tolerance of 10^{-8} is used to determine when to terminate. Gaussian kernel is used in all our experiments.

Table 1: PWSSVM compared with SSVM, FPSSVM, and TSSVM on NDC generated dataset with difference sizes ($C = 10$).

Trains/dimension	Algorithm	Train correctness (%)	Test correctness (%)	Time (s)
2,000,000/10	SSVM	90.86	91.25	**278.97**
	FPSSVM	90.86	91.25	367.46
	TSSVM	90.98	91.33	342.45
	PWSSVM	**91.34**	**91.75**	339.64
2,000,000/20	SSVM	87.64	87.08	**417.64**
	FPSSVM	87.88	88.05	449.28
	TSSVM	87.89	88.05	446.45
	PWSSVM	**88.14**	**88.36**	449.20
10,000/100	SSVM	94.26	93.60	11.17
	FPSSVM	94.78	93.60	11.33
	TSSVM	94.77	93.77	6.24
	PWSSVM	**94.85**	**95.52**	**4.26**
10,000/1000	SSVM	96.67	86.20	56.22
	FPSSVM	96.69	86.14	66.52
	TSSVM	96.69	86.16	26.69
	PWSSVM	**97.94**	**86.23**	**18.73**

The first experiment is used to demonstrate the capability of PWSSVM in solving larger problems. The results in Table 1 are designed to compare the training correctness, the testing correctness, and the training time among the four smooth SVMs on a massively sized dataset. The datasets are created using Musicants NDC Data Generator [25] with different sizes. The test samples are 5% of the training samples. The experiment results show that PWSSVM has the highest training accuracy and testing accuracy. Furthermore, PWSSVM can be used to solve problems more quickly than the other three smooth SVMs when the number of the sample data is relative small.

The second experiment is designed to demonstrate the effectiveness of PWSSVM through the "tried and true" checkerboard dataset [26]. One highly nonlinearly separable but simple example is the checkerboard dataset which has often been used to show the effectiveness of nonlinear kernel methods [17]. The checkerboard dataset is generated by uniformly discretizing the regions $[0, 1] \times [0, 1]$ to $200^2 = 40000$ points and labeling two classes "White" and "Black" spaced by 3×3 grid as Figure 2 shows.

Table 2: PWSSVM compared with SSVM, FPSSVM, and TSSVM on checkerboard dataset with difference sizes ($C = 100$).

Training size	Algorithm	Train correctness (%)	Test correctness (%)	Time (s)
1000	SSVM	99.60	98.28	4.61
	FPSSVM	99.60	98.28	4.35
	TSSVM	99.62	98.28	4.23
	PWSSVM	**99.80**	**98.76**	**4.01**
2000	SSVM	98.84	98.54	11.97
	FPSSVM	99.22	98.59	9.63
	TSSVM	99.22	98.59	**9.35**
	PWSSVM	99.22	**98.68**	9.55
3000	SSVM	98.47	98.88	25.21
	FPSSVM	98.64	98.88	21.58
	TSSVM	98.68	98.90	**17.21**
	PWSSVM	**98.85**	**98.94**	17.36
5000	SSVM	99.34	99.51	47.09
	FPSSVM	99.48	99.51	39.29
	TSSVM	99.52	99.51	38.76
	PWSSVM	**99.79**	**99.65**	**38.23**

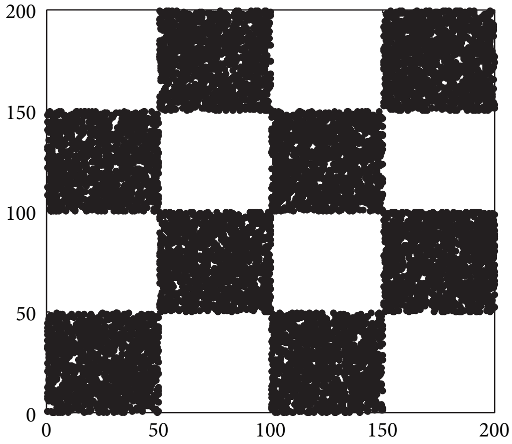


Figure 2: The figure of the checkerboard dataset.

In the first trial of this experiment, the training set contains 1000 points randomly sampled from the checkerboard (for comparison, they are obtained from [17]) which contain 514 "white" samples and 486 "black" samples and the rest 39,000 points are in the testing set. Gaussian kernel function $K(x, y) = \exp(-0.5\|x-y\|^2)$ is used and $C = 100$. Total time for the 1000-point training set using PWSSVM with a Gaussian kernel is 4.01 s. The train accuracy of PWSSVM is 99.80%. The test accuracy of PWSSVM is 98.76% on a 39,000-point test set. TSSVM solves the same problem within 4.23 s, and the train accuracy and the test accuracy are 99.62% and 98.28%. FPSSVM and SSVM obtain the train accuracy of 99.60% within 4.35 s and 4.61 s, respectively. The test accuracy of them are 98.28%.

The rest results are presented in Table 2. The training set is randomly selected from the checkerboard with different sizes. The remaining samples are used as test samples. We compare the classification results of PWSSVM, TSSVM, FPSSVM, and SSVM with the same Gaussian kernel function. The results in Table 2 demonstrate that PWSSVM can solve massive problems quickly, followed by TSSVM, FPSVM, and SSVM in turn. The experimental results show that PWSSVM can obtain the highest train precision and test precision.

7. Conclusions

A novel PWSSVM is proposed in this paper. It only needs to find the unique minima of the unconstrained differentiable convex quadratic function. The proposed method has many advantages over those available, such as good classification performance and less training time cost. The numerical results show that PWSSVM has excellent generalization ability.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported in part by the National Natural Science Foundation of China under Grants 61100165, 61100231, and 51205309 and the Natural Science Foundation of Shaanxi Province (2010JQ8004, 2012JQ8044).

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[2] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[3] K. R. Mller, A. J. Smola, G. Rtsch, B. Schlkopf, J. Kohlmorgen, and V. Vapnik, "Using support vector machines for time series prediction," in *Advances in Kernel Methods: Support Vector Machine*, B. Scholkopf, J. Burges, and A. Smola, Eds., MIT Press, Cambridge, Mass, USA, 1999.

[4] T. Farooq, A. Guergachi, and S. Krishnan, "Knowledge-based Green's Kernel for support vector regression," *Mathematical Problems in Engineering*, vol. 2010, Article ID 378652, 16 pages, 2010.

[5] J. Zheng and B. L. Lu Bao-Liang, "A support vector machine classifier with automatic confidence and its application to gender classification," *Neurocomputing*, vol. 74, no. 11, pp. 1926–1935, 2011.

[6] J. Y. Zhu, B. Ren, H. X. Zhang, and Z. T. Deng, "Time series prediction via new support vector machines," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '02)*, pp. 364–366, November 2002.

[7] J. Ramana and D. Gupta, "LipocalinPred: a SVM-based method for prediction of lipocalins," *BMC Bioinformatics*, vol. 10, p. 445, 2009.

[8] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning (ECML '98)*, pp. 137–142, Springer, Heidelberg, Germany, 1998.

[9] P. S. Jaume, M. I. Darío, and D. M. Fernando, "Support vector machines for continuous speech recognition," in *Proceedings of*

the 14th European Signal Processing Conference (EUSIPCO '06), Florence, Italy, September 2006.

[10] V. Bevilacqua, P. Pannarale, M. Abbrescia, and C. Cava, "Comparison of data-merging methods with SVM attribute selection and classification in breast cancer gene expression," *BMC Bioinformatics*, vol. 13, supplement 7, p. S9, 2012.

[11] E. J. Spinosa and A. C. Carvalho, "Support vector machines for novel class detection in bioinformatics," *Genetics and Molecular Research*, vol. 4, no. 3, pp. 608–615, 2005.

[12] A. Nurettin and G. Cüneyt, *An Application of Support Vector Machine in Bioinformatics: Automated Recognition of Epileptiform Patterns in EEG Using SVM Classifier Designed by a Perturbation Method*, vol. 3261 of *Advances in Information Systems Lecture Notes in Computer Science*, 2005.

[13] Y. F. Sun, X. D. Fan, and Y. D. Li, "Identifying splicing sites in eukaryotic RNA: support vector machine approach," *Computers in Biology and Medicine*, vol. 33, no. 1, pp. 17–29, 2003.

[14] H. J. Lin and J. P. Yeh, "Optimal reduction of solutions for support vector machines," *Applied Mathematics and Computation*, vol. 214, no. 2, pp. 329–335, 2009.

[15] A. Christmann and R. Hable, "Consistency of support vector machines using additive kernels for additive models," *Computational Statistics and Data Analysis*, vol. 56, no. 4, pp. 854–873, 2012.

[16] Y. H. Shao and N. Y. Deng, "A coordinate descent margin based-twin support vector machine for classification," *Neural Networks*, vol. 25, pp. 114–121, 2012.

[17] Y.-J. Lee and O. L. Mangasarian, "SSVM: a smooth support vector machine for classification," *Computational Optimization and Applications*, vol. 20, no. 1, pp. 5–22, 2001.

[18] Y. B. Yuan, J. Yan, and C. X. Xu, "Polynomial smooth support vector machine (PSSVM)," *Chinese Journal of Computers*, vol. 28, no. 1, pp. 9–17, 2005.

[19] Y. B. Yuan and T. Z. Huang, *A Polynomial Smooth Support Vector Machine for Classification*, vol. 3584 of *Lecture Note on Artificial Intelligence*, 2005.

[20] J. Z. Xiong, J. L. Hu, H. Q. Yuan, T. M. Hu, and G. M. Li, "Research on a new class of functions for smoothing support vector machines," *Acta Electronica Sinica*, vol. 35, no. 2, pp. 366–370, 2007.

[21] Y. Yuan, W. Fan, and D. Pu, "Spline function smooth support vector machine for classification," *Journal of Industrial and Management Optimization*, vol. 3, no. 3, pp. 529–542, 2007.

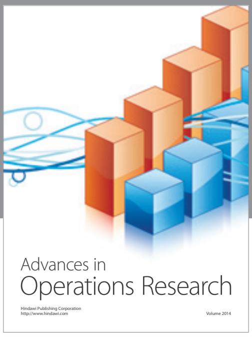
[22] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Ma, USA, 2nd edition, 1999.

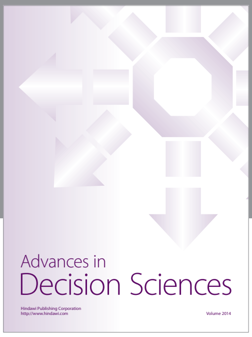
[23] C. Xu and J. Zhang, "A survey of Quasi-Newton equations and Quasi-Newton methods for optimization," *Annals of Operations Research*, vol. 103, no. 1–4, pp. 213–234, 2001.

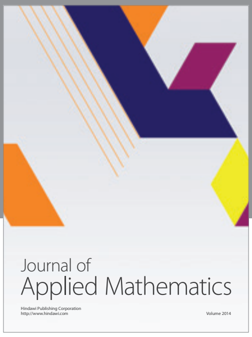
[24] D. R. Musicant and O. L. Managsarian, "LSVM: Lagrangian support vector machine," 2000, http://www.cs.wisc.edu/dmi/svm/ .

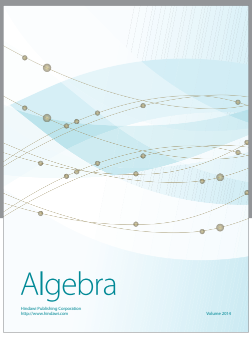
[25] D. R. Musicant, "NDC: normally distributed clustered datasets," 1998, http://www.cs.wisc.edu/~musicant/data/ndc/ .

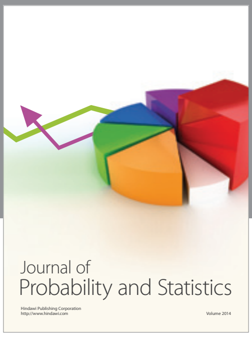
[26] T. K. Ho and E. M. Kleinberg, "Checkerboard dataset," 1996, http://www.cs.wisc.edu/~musicant/data/ndc/ .


Advances in
Operations Research

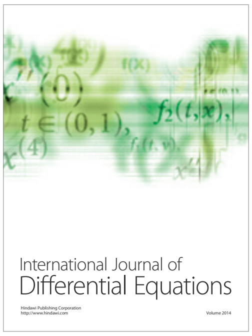

Advances in
Decision Sciences


Journal of
Applied Mathematics

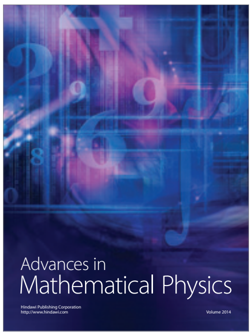

Algebra

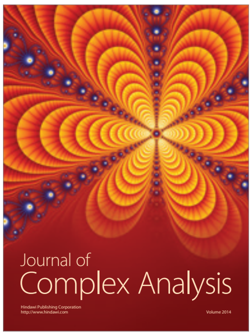

Journal of
Probability and Statistics

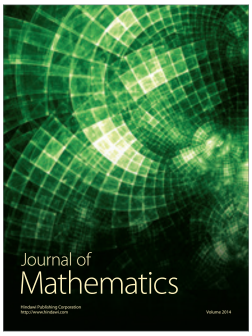
The Scientific
World Journal

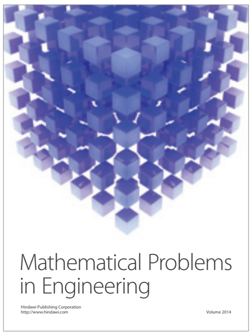

International Journal of
Differential Equations

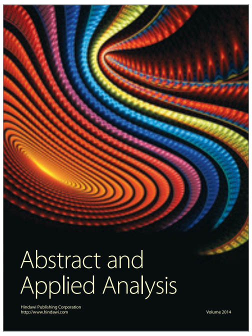
International Journal of
Combinatorics

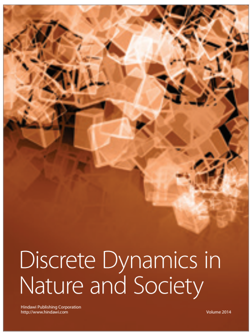

Advances in
Mathematical Physics

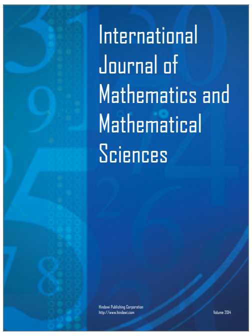

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

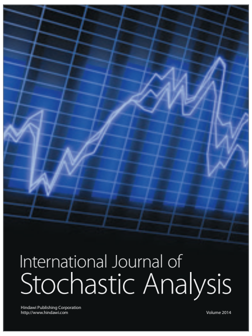

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization