Copyright by the American Institute of Mathematical Sciences. Yuan, Y. B.; Fan, W. G.; Pu, D. M., "Spline function smooth support vector machine for classification," J. Industrial and Management Optimization 3(3), 529-542, (2007); DOI: 10.3934/jimo.2007.3.529

JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **3**, Number **3**, August **2007**

Website: http://AIMsciences.org

pp. **529–542**

SPLINE FUNCTION SMOOTH SUPPORT VECTOR MACHINE FOR CLASSIFICATION

Yubo Yuan

School of Applied Mathematics
University of Electronic Science and Technology of China
Chengdu, 610054,China
and
School of Economy and Finance
Xi'an Jiao Tong University
Xi'an, 710049, China

Weiguo Fan

Accounting & Information Systems
Virginia Polytechnic Institute and State University
VA, 24061, USA

Dongmei Pu

School of Applied Mathematics
University of Electronic Science and Technology of China
Chengdu, 610054, China

(Communicated by David Yang Gao)

Abstract. Support vector machine (SVM) is a very popular method for binary data classification in data mining (machine learning). Since the objective function of the unconstrained SVM model is a non-smooth function, a lot of good optimal algorithms can't be used to find the solution. In order to overcome this model's non-smooth property, Lee and Mangasarian proposed smooth support vector machine (SSVM) in 2001. Later, Yuan et al. proposed the polynomial smooth support vector machine (PSSVM) in 2005. In this paper, a three-order spline function is used to smooth the objective function and a three-order spline smooth support vector machine model (TSSVM) is obtained. By analyzing the performance of the smooth function, the smooth precision has been improved obviously. Moreover, BFGS and Newton-Armijo algorithms are used to solve the TSSVM model. Our experimental results prove that the TSSVM model has better classification performance than other competitive baselines.

1. Introduction.

Among all the methods that are commonly used for data classification, support vector machine (SVM), a well-known method based on statistic learning theory([1-8]) has become one of the most popular methods nowadays. In fact, support vector machine has surpassed neural network and becomes the most popular method among all the statistically learning methods. Now, SVM is often

2000 *Mathematics Subject Classification.* Primary: 90C20; Secondary: 65L07; 65N12.

Key words and phrases. Quadratic Programming, Data Mining,Support Vector Machine.

This work is supported by the Applied Basic Research Foundations of Sichuan (05JY029-068-2) and the Youth Key Foundations of Univ. of Electronic Science and Technology of China (JX04042).

used in pattern recognition, regression analysis, probability density estimate and so on. For many of these tasks, SVM is as good as or better than traditional machine learning methods.

In this paper, we focus on the SVM approach for data classification. The SVM model for classification can be formulated as a non-smooth unconstrained optimization problem ([9-12]). However, the objective function is non-differentiable at zero. In our approach, we change the model slightly and apply the smooth techniques that have been extensively used for solving important mathematical programming problems ([13-23]). In 2001, Lee et al.([20]) have employed a smooth method to solve the resulting optimization problem. They proposed to use signal function integral

$$p(x,k) = x + \frac{1}{k} log(1+\varepsilon^{kx}), k > 0. \tag{1}$$

They got a smooth SSVM model. Here, ε is the fondue of natural logarithm and called as smooth parameter. Later, they used the same smooth function to smooth support vector machine regression in 2005 (see in [21]).

In 2005, we presented two polynomial functions ([22]) as following

$$q(x,k) = \begin{cases} x, & \text{if } x > \frac{1}{k}, \\ \frac{k}{4}x^2 + \frac{1}{2}x + \frac{1}{4k}, & \text{if } -\frac{1}{k} \le x \le \frac{1}{k}, \\ 0, & \text{if } x < -\frac{1}{k}, \end{cases} \tag{2}$$

$$h(x,k) = \begin{cases} x, & \text{if } x > \frac{1}{k}, \\ -\frac{k^3}{16}(x+\frac{1}{k})^3(x-\frac{3}{k}), & \text{if } -\frac{1}{k} \le x \le \frac{1}{k}, \\ 0, & \text{if } x < -\frac{1}{k}. \end{cases} \tag{3}$$

Using the above smooth functions to the non-smooth unconstrained optimization problem, we can get smooth SVM model (we termed it PSSVM.). It can be proved that PSSVM is more effective than SSVM through theory analysis.

The primal goal of this paper is to use the three-order spline function to smooth the objective function of the original model. After this, the three-order spline smooth support vector machine model (we call it TSSVM) is obtained. Also, we give the conclusion through theoretical analysis and experimental results in this paper that the TSSVM model is better than given models in smooth precision and classification capability.

The rest of the paper is organized as follows. In Section 2 we will briefly introduce how to obtain the SSVM model and state the three-order spline function to form the three-order spline smooth support vector machine (TSSVM) model. In Section 3 we study the smooth property of the three-order spline function such as smooth capability, the approach degree to original function. In Section 4 we prove the convergence of the TSSVM model. In Section 5 we discuss the optimization algorithms, e.g. BFGS and Newton method, for the TSSVM optimal model to obtain the smooth parameter. In Section 6 the experimental results are given. We conclude the paper in Section 7.

For ease of understanding, we use the following notations throughout the paper. All vectors will be column vectors unless transposed to a row vector by a prime superscript. The transpose of the vector or matrix is denoted by $(\cdot)^T$. For a vector x in the n-dimensional real space R^n, the plus function x_+ is defined as $(x_+)_i = max(0, x_i), i = 1, 2, 3, \cdots, n$. The scalar (inner) product of two vectors x, y in the n-dimensional real space will be denoted by $x^T y$ and the p-norm of x will be denoted by $\|x\|_p$. For a matrix $A \in R^{m\times n}$, A_i is the $i - th$ row of which is a row vector in R^n. A column vector of ones of arbitrary dimension will be denoted by e. If f is a real valued function defined in the n-dimensional real space R^n, the gradient of f is denoted by $\bigtriangledown f(x)$ which is a row vector in R^n and the $n \times n$ Hessian matrix of f at x is denoted by $\bigtriangledown^2 f(x)$. The level set of f is defined as $L_\mu(f) = \{x | f(x) \le \mu, x \in Domain(f)\}$ for a given real number .

2. Three-order spline smooth support vector machine model.

In this section, the unconstrained optimal model of SVM is obtained.

A pattern classification problem is to classify m points in $n-$dimensional real space R^n, represented by an $m \times n$ matrix A, according to membership of each point A_i in the classes 1 or -1 as specified by a given $m \times m$ diagonal matrix D with 1 or -1 diagonals. The standard support vector machine (see in [20-25]) for this problem is given by the following

$$\begin{aligned} \min_{(\omega,\gamma,\mathrm{y})\in \mathrm{R}^{\mathrm{n+1+m}}} \quad & \nu e^T y + \frac{1}{2}\omega^T \omega, \\ \text{s.t.} \quad & D(A\omega - e\gamma) + y \ge e, \\ & y \ge 0. \end{aligned} \tag{4}$$

for some $\nu > 0$, ω is the normal to the bounding plane and γ is the distance of the bounding plane to the origin. Let us define the linear separating plane

$$P = \{x | x \in R^n, x^T \omega = \gamma\}, \tag{5}$$

with normal $\omega \in R^n$ and distance $\frac{|\gamma|}{\|\omega\|_2}$ to the origin.

In order to understand these concepts, a simple illustration diagram is given below. In Figure 1, the circle points on the left corner are in class with index -1 and the star points are in class with index 1. The dashed line denotes the hyper-surface used to classify data points. The two solid lines denote support hyper-surfaces

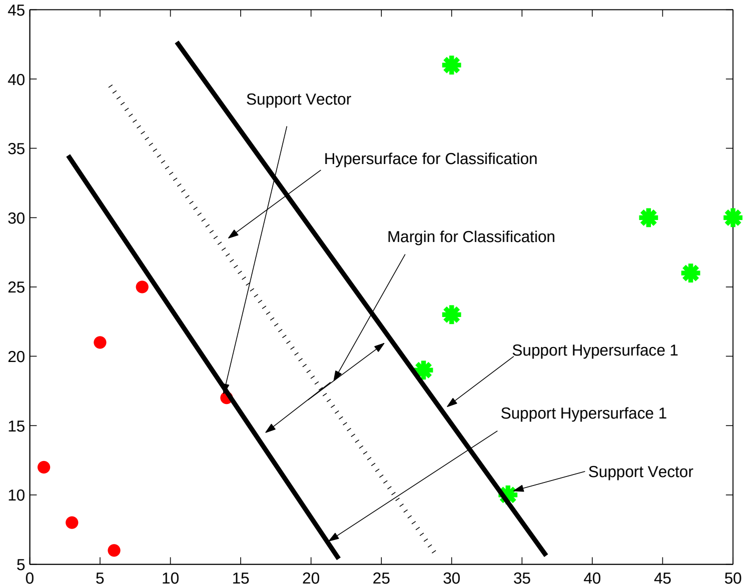

Figure 1 Meanful Figure for SVM

determined by support vectors. The distance between these two solid lines is margin for classification (can be computed by $\frac{1}{\|\omega\|_2}$).

The first term in the objective function of (3) is the 1-norm of the slack variable y with weight ν. The second term $\omega^T\omega$ is the square of the 2-norm of the vector ω with half introduced for simplicity. Replacing the first term with the 2-norm vector y, the SVM problem can be modified into the following form

$$\begin{aligned} \min_{(\omega,\gamma,\mathrm{y})\in\mathrm{R}^{\mathrm{n+1+m}}} \quad & \frac{\nu}{2}\|y\|_2^2+\frac{1}{2}(\|\omega\|_2^2+\gamma^2), \\ \text{s.t.} \quad & D(A\omega-e\gamma)+y\geq e, \\ & y\geq 0. \end{aligned} \tag{6}$$

As a solution of problem (6), y is given by

$$y=(e-(D(A\omega-e\gamma)))_+, \tag{7}$$

where the element of the vector $(a)_+$ is defined by

$$(a_i)_+=\begin{cases} a_i, & \text{if } a_i>0, \\ 0, & \text{if } a_i\leq 0. \end{cases} \tag{8}$$

Substituting y into the objective function of (6) converts problem (6) into an equivalent unconstrained optimization problem

$$\min_{(\omega,\gamma)\in R^{n+1}} \frac{\nu}{2}\|(e-(D(A\omega-e\gamma)))_+\|_2^2+\frac{1}{2}(\|\omega\|_2^2+\gamma^2). \tag{9}$$

This is a strongly convex minimization problem without any constraints and it has a unique solution. However, the objective function in (9) is not differentiable at zero which precludes the use of existing optimization methods using derivatives. This property means that a lot of good algorithms can't be used to solve it. Since many optimal algorithms require that the objective function is first or twice differentiable, it is necessary to smooth the objective function.

In this paper, a new smooth function is introduced. It is three-order spline function as following

$$T(x,k)=\begin{cases} 0, & \text{if } x<-\frac{1}{k}, \\ \frac{k^2}{6}x^3+\frac{k}{2}x^2+\frac{1}{2}x+\frac{1}{6k}, & \text{if } -\frac{1}{k}\leq x<0, \\ -\frac{k^2}{6}x^3+\frac{k}{2}x^2+\frac{1}{2}x+\frac{1}{6k}, & \text{if } 0\leq x\leq\frac{1}{k}, \\ x, & \text{if } \frac{1}{k}<x. \end{cases} \tag{10}$$

If we replace the plus function in (9) by this function, a new smooth SVM model is obtained as following

$$\min_{(\omega,\gamma)\in R^{n+1}} F(\omega,\gamma,k)=\frac{\nu}{2}\|T(e-(D(A\omega-e\gamma)),k)\|_2^2+\frac{1}{2}(\|\omega\|_2^2+\gamma^2). \tag{11}$$

In the next section, we will prove that the smooth performance of the three-order spline function is better than the previous smooth functions with the same smooth parameter k.

3. **Performance analysis of smooth functions.**

Before we do performance analysis, we need to introduce the following lemmas.

Lemma 1. *Let $\Omega\subset R$,$p(x,k)$ is defined as (1) and x_+ is plus function. The following results are easily obtained.*
(i) $p(x,k)\in C^\infty(\Omega),\forall x\in\Omega$;

(ii) $p(x,k) \geq x_+$;
(iii) $\forall \rho > 0, p(x,k)^2 - x_+^2 \leq (\frac{log2}{k})^2 + \frac{2\rho}{k} log2$.

The proof can be seen in [21].

Lemma 2. *Let* $\Omega \subset R$, $q(x,k)$ *and* $f(x,k)$ *are defined as (2) and (3) and* x_+ *is plus function. The following results are easily obtained.*
(i) $q(x,k) \in C^1(\Omega), f(x,k) \in C^2(\Omega), \forall x \in \Omega$;
(ii) $q(x,k) \geq x_+, f(x,k) \geq x_+, \forall x \in \Omega$;
(iii) $\forall x \in \Omega, k \in R^+$,

$$q(x,k)^2 - x_+^2 \leq \frac{1}{11k^2}, f(x,k)^2 - x_+^2 \leq \frac{1}{19k^2}.$$

The proof can be seen in [22,23].

Remark 1. These results in lemma 1 and lemma 2 are easy to verify when x is a real value in R.

Theorem 1. *Let* $\Omega \subset R$, $T(x,k)$ *be defined as (10) and* x_+ *is plus function. The following results are easily obtained.*
(i) $T(x,k) \in C^2(\Omega), \forall x \in \Omega$, *in another word,* $T(x,k)$ *satisfies the following equalities at the points* $x = \pm\frac{1}{k}, x = 0$,

$$\begin{cases} T(-\frac{1}{k},k) = 0, & \lim\limits_{x\to 0^-} T(x,k) = \lim\limits_{x\to 0^+} T(x,k), & T(\frac{1}{k},k) = \frac{1}{k}, \\ T^{'}(-\frac{1}{k},k) = 0, & \lim\limits_{x\to 0^-} T^{'}(x,k) = \lim\limits_{x\to 0^+} T^{'}(x,k), & T^{'}(\frac{1}{k},k) = 1, \\ T^{''}(-\frac{1}{k},k) = 0, & \lim\limits_{x\to 0^-} T^{''}(x,k) = \lim\limits_{x\to 0^+} T^{''}(x,k), & T^{''}(\frac{1}{k},k) = 0. \end{cases}$$

(ii) $T(x,k) \geq x_+, \forall x \in \Omega$;
(iii) $\forall x \in \Omega, k \in R^+$,

$$T(x,k)^2 - x_+^2 \leq \frac{1}{24k^2}.$$

Proof of Theorem. (i) Let us observe the definition of $T(x,k)$ in (10). If we substitute points $x = \pm\frac{1}{k}, x = 0$ into it directly, the results in (i) are obtained easily.

(ii) If $x < 0$ or $x > \frac{1}{k}$, the values of $T(x,k)$ and x_+ are same, that is, $T(x,k) = x_+$. Else, let $g(x) = T(x,k) - x_+$, then

$$g(x) = \begin{cases} \frac{k^2}{6}x^3 + \frac{k}{2}x^2 + \frac{1}{2}x + \frac{1}{6k}, & \text{if } -\frac{1}{k} \leq x \leq 0, \\ -\frac{k^2}{6}x^3 + \frac{k}{2}x^2 - \frac{1}{2}x + \frac{1}{6k}, & \text{if } 0 \leq x \leq \frac{1}{k}. \end{cases}$$

If $-\frac{1}{k} \leq x \leq 0$, since $g(-\frac{1}{k}) = 0$ and $g^{'}(x) = \frac{1}{2}(kx+1)^2 \geq 0$, $g(x)$ is an increasing function. The result is $g(x) \geq g(-\frac{1}{k}) = 0$. Therefore, $T(x,k) \geq x_+$, $\forall x \in [-\frac{1}{k}, 0]$.

If $0 \leq x \leq \frac{1}{k}$, since $g(\frac{1}{k}) = 0$ and $g^{'}(x) = -\frac{1}{2}(kx-1)^2 \geq 0$, $g(x)$ is an decreasing function. The result is $g(x) \geq g(\frac{1}{k}) = 0$. Therefore, $T(x,k) \geq x_+$, $\forall x \in [0, \frac{1}{k}]$.

(iii)If $x < 0$ or $x > \frac{1}{k}$, the values of $T(x,k)$ and x_+ are same, so $T(x,k)^2 - x_+^2 = 0$, the inequality in result (iii) is satisfied naturally.

If $-\frac{1}{k} \leq x \leq 0$, since $x_+ = 0$, $T(x,k)^2 - x_+^2 = T(x,k)^2$. Because T(x,k) is positive-value, continuous and increasing function for $\forall x \in [-\frac{1}{k}, 0]$, the result is $T(x,k)^2 \leq T(0,k)^2 = \frac{1}{36k^2}$. It is obvious that $T(x,k)^2 - x_+^2 \leq \frac{1}{24k^2}$.

If $0 \leq x \leq \frac{1}{k}$, in order to obtain the result, we introduce a transformation $a = kx$ (obviously $a \in [0,1]$). Let $s(x) = T(x,k)^2 - x_+^2 = -\frac{k^2}{6}x^3 + \frac{k}{2}x^2 - \frac{1}{2}x + \frac{1}{6k} - x^2$, after taking $a = kx$ in it,

$$s(x) = s(a) = \frac{1}{36k^2}((-a^3 + 3a^2 + 3a + 1)^2 - 36a^2).$$

For $\forall a \in [0, 1]$, the maximum point of $s(a)$ is $a = 0.1814$. So, $s(x) = T(x, k)^2 - x_+^2 \leq s(0.1814) < \frac{1}{24k^2}$. □

According to results of Lemma 1, Lemma 2 and Theorem 1, the following performance comparison results of smooth functions are obtained.

Theorem 2. *Let* $\rho = \frac{1}{k}$.
(i) If the smooth function is defined as (1), by Lemma 1,

$$p(x, k)^2 - x_+^2 \leq (\frac{log2}{k})^2 + \frac{2\rho}{k} log2 = ((log2)^2 + 2log2)\frac{1}{k^2} \approx 0.6927\frac{1}{k^2}. \tag{12}$$

(ii) If the smooth function is defined as (2) or (3), by Lemma 2,

$$q(x, k)^2 - x_+^2 \leq \frac{1}{11k^2} \approx 0.0909\frac{1}{k^2}, \tag{13}$$

$$f(x, k)^2 - x_+^2 \leq \frac{1}{19k^2} \approx 0.0526\frac{1}{k^2}. \tag{14}$$

(iii) If the smooth function is defined as (10), by Theorem 1,

$$T(x, k)^2 - x_+^2 \leq \frac{1}{24k^2} \approx 0.0415\frac{1}{k^2}. \tag{15}$$

The results can be obtained directly form Lemma 1, Lemma 2 and Theorem 1.

From the above theorem, it is obvious that $T(x, k)$ is the best smooth function among all of them. In order to show the difference more clearly, we present the following smooth performance comparison diagram. The smooth parameter is set at $k = 10$.

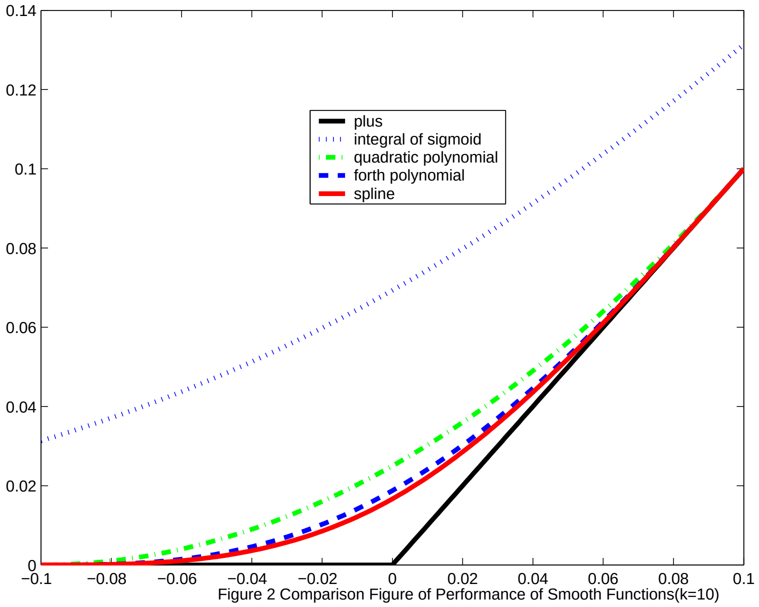

Figure 2 Comparison Figure of Performance of Smooth Functions(k=10)

As can be seen from Figure 2, our proposed spline smooth function is the closest to the original non-smooth function, which indicates the superiority of our proposed smooth function.

4. **Performance analysis of smooth functions.** In this section, the convergence of TSSVM model (11) is presented. We will prove that the solution of TSSVM can closely approximate the optimal solution of the original model (9) when k moves towards positive infinity. The smooth function can be applied to every element of a multi-dimensional vector.

Theorem 3. *Let $A \in R^{m\times n}$, $b \in R^{m\times 1}$, define real function $h(x) : R^n \to R$, and $g(x,k) : R^n \times N \to R$ as follows,*

$$h(x) = \frac{1}{2}\|(Ax-b)_+\|_2^2 + \frac{1}{2}\|x\|_2^2,$$

$$g(x,k) = \frac{1}{2}\|T(Ax-b,k)\|_2^2 + \frac{1}{2}\|x\|_2^2,$$

where $T(x,k)$ is defined in (10). The following conclusions can be obtained.
(i) $h(x)$ and $g(x,k)$ are strong convex functions;
(ii) There is a unique solution x^* to $\min\limits_{x\in R^n} h(x)$ and there is also a unique solution $(x^*)^k$ to $\min\limits_{x\in R^n} g(x,k)$;
(iii) For $\forall k \geq 1$, x^* and $(x^*)^k$ both satisfy the following condition

$$\|(x^*)^k - x^*\|^2 \leq \frac{m}{48k^2}; \tag{16}$$

(iv) x^* and $(x^*)^k$ satisfy

$$\lim_{k\to\infty} (x^*)^k = x^*. \tag{17}$$

Proof of Theorem. (i) $h(x)$ and $g(x,k)$ are strong convex functions because $\|\cdot\|_2^2$ is strong convex function;
(ii) If $L_v(h(x))$ is the level set of $h(x)$ and $L_v(g(x,k))$ is the level set of $g(x,k)$, then, according to result (ii) of Theorem 1, we have

$$L_v(g(x,k)) \subseteq L_v(h(x)) \subseteq \{x | \|x\|_2^2 \leq 2v\}.$$

Therefore, $L_v(g(x,k))$ and $L_v(h(x))$ are strict convex sets. Because of this, there is a unique solution to both $\min\limits_{x\in R^n} h(x)$ and $\min\limits_{x\in R^n} g(x,k)$.
(iii) If x^* is the optimal solution to $\min\limits_{x\in R^n} h(x)$ and $(x^*)^k$ is the optimal solution to $\min\limits_{x\in R^n} g(x,k)$, because of the optimal condition and convex property of $h(x)$ and $g(x,k)$, the following inequalities are hold,

$$h((x^*)^k) - h(x^*) \geq \nabla h(x^*)((x^*)^k - x^*) + \frac{1}{2}\|(x^*)^k - x^*\|_2^2 = \frac{1}{2}\|(x^*)^k - x^*\|_2^2,$$

$$g((x^*),k) - g((x^*)^k,k) \geq \nabla g((x^*)^k,k)(x^* - (x^*)^k) + \frac{1}{2}\|(x^*)^k - x^*\|_2^2 = \frac{1}{2}\|(x^*)^k - x^*\|_2^2.$$

If we add the two formulas above and note $T(x,k) \geq x_+$, we can obtain

$$\begin{aligned}\|(x^*)^k - x^*\|_2^2 &\leq h((x^*)^k) - h(x^*) + g((x^*),k) - g((x^*)^k,k) \\ &= (g((x^*),k) - h(x^*)) - (g((x^*)^k,k) - h((x^*)^k)) \\ &\leq g((x^*),k) - h(x^*) \\ &= \tfrac{1}{2}\|T(Ax^* - b,k)\|_2^2 - \tfrac{1}{2}\|(Ax^* - b)_+\|_2^2.\end{aligned}.$$

According to result (iii) of Theorem 1, $\|(x^*)^k - x^*\|_2^2 \leq \frac{m}{48k^2}$, so the conclusion (16) is correct.
(iv) When k goes towards infinity in (16), we can obtain

$$\lim_{k\to+\infty} \|(x^*)^k - x^*\|_2^2 \leq \lim_{k\to+\infty} \frac{m}{48k^2} = 0,$$

so $\lim\limits_{k\to+\infty} (x^*)^k = x^*$. □

The conclusion (iv) of Theorem 3 explains that the optimal solution of the smooth model TSSVM is close to the original support vector machine model when k moves towards positive infinity. With the above in mind, we design two algorithms to solve TSSVM in the next section.

5. Design of algorithms.

Two things need to be considered when designing algorithms for solving the SVM model. One is the choice of an optimal smooth parameter, the other is the choice of solution method. We have presented above that TSSVM's optimization solution converges to the real solution of the original model when k goes towards positive infinite. However, there is a tradeoff between the size of k and computational time. Thus k can not be too large in practice. It is very important to choose a proper optimization smooth parameter k before an algorithm design.

Definition 1. For m dimensional training data set $A \in R^{m\times n}$, ϵ is a given parameter, $(x^*)^k$ is an optimization solution of TSSVM and x^* is optimization solution of the original model. When the algorithm stops, k must satisfy $\|(x^*)^k - x^*\|_2^2 \leq \epsilon$. We call the minimal k as a minimal smooth parameter and denote it as $kopt(m,\epsilon)$.

From the Definition 1, if the smooth parameter k in the TSSVM model satisfies $k \geq kopt(m,\epsilon)$, the solution of TSSVM can satisfy the precision requirement $\|(x^*)^k - x^*\|_2^2 \leq \epsilon$. We next present a theorem on how to estimate the minimal smooth parameter.

Theorem 4. *$A \in R^{m\times n}$ is a sample data set, $(x^*)^k$ is the optimal solution of TSSVM and x^* is the optimal solution of SVM. m is the number of sample points and ϵ is the algorithm terminating control parameter.*

(i) The minimum smooth parameter $kopt(m,\epsilon)$ of the sigmoid function integral ([21]) is

$$kopt(m,\epsilon) = int[\sqrt{\frac{0.6929m}{2\epsilon}}] + 1; \tag{18}$$

(ii) The minimum smooth parameter $kopt(m,\epsilon)$ of the quadratic polynomial smooth function ([22,23]) is

$$kopt(m,\epsilon) = int[\sqrt{\frac{m}{22\epsilon}}] + 1; \tag{19}$$

(iii) The minimum smooth parameter $kopt(m,\epsilon)$ of the fourth-order polynomial smooth function ([22,23]) is

$$kopt(m,\epsilon) = int[\sqrt{\frac{m}{38\epsilon}}] + 1; \tag{20}$$

(iv) The minimum smooth parameter $kopt(m, \epsilon)$ of the three-order spline smooth function is

$$kopt(m, \epsilon) = int[\sqrt{\frac{m}{48\epsilon}}] + 1. \tag{21}$$

where $int[\cdot]$ is a integer function, i.e., int[0.6547]=0, int[10.1216]=10 and so on.

The results can be easily obtained from results of Theorem 2. The proof is omitted.

We next present two algorithms to solve SVM models.

BFGS method is suitable for unconstrained optimization problems when the objective function and its gradient value can easily be obtained. BFGS method is the most widely used one among various quasi-Newton methods [27][28].

The BFGS method for Problem (11) is as follows:

Algorithm 1 (The BFGS([27][28]) algorithm for TSSVM (11)).

step 1: (Initialization) $H^0 = I, ((\omega^0)^T, \gamma^0) = p^0 \in R^{n+1}$, ϵ,$\alpha^0 = 1$ and set $i := 0$;

step 2: Compute $F^i = F(p^i, kopt)$ and $g^i = \nabla F(p^i, kopt)$;

step 3: If $\|g^i\|_2^2 \leq \epsilon$ or $\alpha^0 < 10^{-12}$, then stop, and accept $p^i = ((\omega^i)^T, \gamma^i)$ as the optimal solution of (11), else calculate $d^i = -H^i g^i$;

step 4: Perform linear search along direction d^i to get a step length $\alpha^i > 0$; Let

$$p^{i+1} = p^i + \alpha^i d^i,\ s^i = p^{i+1} - p^i = -\alpha^i H^i g^i,$$

and compute $F^{i+1} = F(p^{i+1}, kopt)$, $g^{i+1} = \nabla F(p^{i+1}, kopt)$ and $y^i = g^{i+1} - g^i$;

step 5: Update H^i to get H^{i+1}:

$$H^{i+1} = (I - \frac{s^i(y^i)^T}{(s^i)^T y^i}) H^i (I - \frac{y^i(s^i)^T}{(s^i)^T y^i}) + \frac{s^i(s^i)^T}{(s^i)^T y^i}; \tag{22}$$

step 6: Set $i := i + 1$, go to step 2.

Alternatively, we can use the Newton-Armijo method as the optimization algorithm. The Newton-Armijo method is a fast optimization algorithm for optimization problems.

The Newton-Armijo method for problem (11) is as follows:

Algorithm 2 (The Newton-Armijo algorithm for TSSVM(11)).

step 1: (Initialization) $H^0 = I, ((\omega^0)^T, \gamma^0) = p^0 \in R^{n+1}$, ϵ ,$\alpha^0 = 1$ and set $i := 0$;

step 2: Compute $F^i = F(p^i, kopt)$ and $g^i = \nabla F(p^i, kopt)$;

step 3: If $\|g^i\|_2^2 \leq \epsilon$ or $\alpha^0 < 10^{-12}$, then stop, and accept $p^i = ((\omega^i)^T, \gamma^i)$ as the optimal solution of (11), else calculate Newton direction d^i from the array of equations

$$\bigtriangledown^2 F(p^i, kopt) d^i = -g^i;$$

step 4: Perform a linear search along direction d^i with the Armijo step to get a step length $\alpha^i > 0$; Let

$$p^{i+1} = p^i + \alpha^i d^i;$$

step 5: Set $i := i + 1$, go to step 2.

Overall, the BFGS algorithm has a less restricted requirement on the objective functions than the Newton method. It does not require the objective functions to have second-order derivatives. Thus it needs less storage and memory.

6. Numerical experiments and results.

6.1. Experimental Design and Results. We synthetically create the data set for our experiments. The first thing is to generate (randomly) the experimental data $D(D \in R^{(m+m_1)\times n})$, where m is the number of the training data set records, m_1 is the number of the testing data set records, and n is the number of attributes of both training and test data sets. $e \in R^n$ is a vector, of which each element is equal to 1. Next, we classify every data point in D. We classify each data record into three classes according to the following rules: (i) If $e^T x \geq 1$, it is classified into the first class; (ii) If $e^T x \leq -1$, it is classified into the second class; (iii) If $-1 < e^T x < 1$, it is classified as noise data. We probabilistically assign each noisy data record into the existing two classes. For example, if $-1 < e^T x < 1$, we can keep this data point according to the probability of p_1(such as $p_1 = 0.7$). In this case, if $e^T x < 0$, it is classified into the second class according to the probability p_2 (such as $p_2 = 0.8$), and into the first class according to the probability $1 - p_2$. Conversely, if $e^T x \geq 0$, it is classified into the first class according to the probability p_2 (such as $p_2 = 0.8$), and into the second class according to the probability $1 - p_2$.

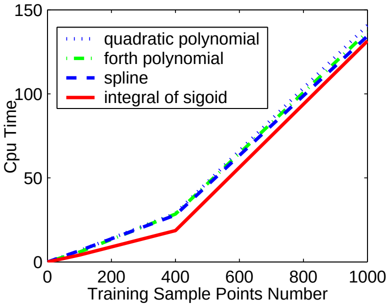

Figure3 BFGS Method(m=10)

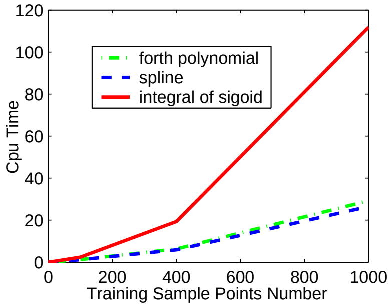

Figure 4 Newton Method(m=10)

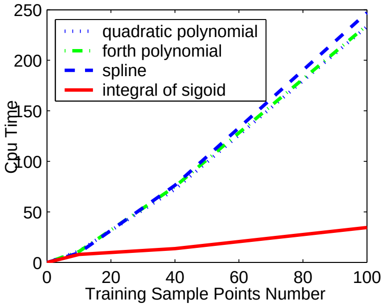

Figure 5 BFGS Method(m=200)

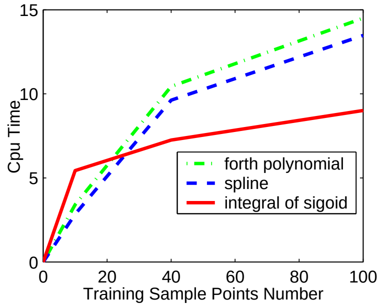

Figure 6 Newton Method(m=200)

We conduct numerical experiments to test the following three aspects of the proposed algorithm:

(i) **Algorithm validity**. The algorithm validity can be tested with three indexes: the optimization target value (namely *optv*), the norm of the gradient when the algorithm ends (namely *ng*) and the step length of the gradient when the algorithm ends (namely *setpl*). The algorithm often terminates when $ng < \epsilon$ or $setpl < \epsilon_1$. When the algorithm terminates because $setpl < \epsilon_1$, the condition $ng < \epsilon$ usually could not be satisfied, which will result in low classification accuracy. On the other hand, if the algorithm ends because

$ng < \epsilon$, the algorithm usually has a high accuracy irrespective whether the condition *setpl* $< \epsilon_1$. Thus smaller *ng* often means better algorithm validity.

(ii) **Algorithm efficiency**. The algorithm efficiency is measured using computing time (namely *cput*). The algorithm efficiency is higher when *cput* is smaller.

(iii) **Classification performance**. The performance of linear classification tools obtained with different smooth functions are measured by two indexes: one is the training correct rate (namely $TrCR$) and the other is the test correct rate (namely $TeCR$). Higher $TrCR$ and $TeCR$ mean higher classification performance.

This numerical experiment was carried out on a computer with 1.8Ghz CPU and 256MB memory. The coding was done using MATLAB 6.1. The parameters of the experiment were set as follows: $\epsilon = 10^{-3}$, $\epsilon_1 = 10^{-3}$, and the initialization iteration point was set $p^0 = 0$. The experimental results are shown in Tables 1 and 2. In both tables, the first column shows the scale of training data set, in which m means the number of the training data set points and n means the dimension of every training data point; the rest of the columns are numerical experimental results for different algorithms with different smooth functions. The number of testing data points m_1 is 1000. The values in each cell of the tables are $TrCR$, $TeCR$, *cupt*, *optv*, *ng* and *stepl*, respectively.

6.2. **Analysis of the Experimental Results.** In terms of the optimization algorithm, both BFGS and Newton algorithms can help find optimal solutions for our TSSVM model because they both end with small *ng* values. Among all the smooth functions, our proposed spline function achieved the lowest *optv* and *ng*. This demonstrates the advantage of our proposed TSSVM mdoel.

In terms of the algorithm efficiency, Newton-Armijo method is faster than BFGS. However, Newton-Armijo can be be applied to QPSSVM model due to lack of the second order derivative. In order to show clearly the relationship between the sample size and the CPU time, we show graphically the results in Figures 3, 4, 5 and 6.

Among the four smooth functions, our proposed TSSVM model is the best in terms of the classification performance as measured by $TeCR$, followed by the PSSVM. SSVM has some troubles in finding the optimal solutions.

7. Conclusions and future work.

From the analysis of the above experimental results, we can see that the TSSVM model is more effective than the previous models. Our proposed TSSVM model in this paper has very good classification performance and computational stability. We can use the Newton method as the preferred optimization algorithm for the TSSVM model. In other words, for a given spline-based smooth function, the Newton method can find the optimal solution in a shorter time than BFGS.

For future work, we are going to do the following:

(i) The investigation of other smooth functions. In this paper we presented a three-order spline function. We believe there are better smooth functions yet to be discovered and evaluated.

(ii) Research about the optimization algorithms for solving smooth function-based SVM models. There are many optimization algorithms available. However, these algorithms are often suitable for different objective functions. We need to look at different smooth functions and identify the best optimization algorithm for each model.

m,n	BFGS-Armijo				NEWTON-Armijo		
	QPSSVM	FPSSVM	SSVM	TSSVM	FPSSVM	SSVM	TSSVM
100,10	100 94 6.6 0.0097 8.7615e-5 1	100 94 5.82 0.0097 1.6859e-4 1	99 93.2 4.07 3.4644 180.8036 9.0949e-13	100 96.5 6.71 0.0094 3.79714e-4 1	100 94 1.26 0.0097 1.4001e-5 1	99 93.2 2.36 3.4296 174.5281 9.0949e-13	100 96.5 1.15 0.0094 9.2557e-5 1
400,10	100 99 28.84 0.1077 3.7456e-4 1	100 99 28.45 0.1077 6.4271e-4 1	98.5 98.6 18.62 39.9770 1.9726e3 9.0949e-13	100 99 27.92 0.1045 7.1124e-4 1	100 99 6.26 0.1077 5.3075e-6 1	98.75 97.8 19.34 39.3661 1.9463e3 2.2737e-13	100 99 5.92 0.1045 5.7728e-6 1
1000,10	99.70 99 140.77 6.2841 6.6255e-4 1	99.70 99 136.93 6.2841 3.3982e-4 1	95.8 94.2 131.43 178.9223 1.4156e4 1.1369e-13	99.70 99 134.37 6.2157 4.4042e-4 1	99.70 99 29.22 6.2841 3.1827e-12 1	98.6 98.8 111.83 169.5077 1.02371e4 9.0949e-13	99.70 99 26.47 6.2157 4.0019e-12 1

TABLE 1. Results for different smooth functions using different algorithms with fixed dimension.

m,n	BFGS-Armijo				NEWTON-Armijo		
	QPSSVM	FPSSVM	SSVM	TSSVM	FPSSVM	SSVM	TSSVM
200,10	100 98.5 11.10 0.0458 0.04319 1	100 98.5 10.66 0.0458 2.0532e-4 1	97.5 96.1 7.97 14.2473 663.7311 5.6843e-14	100 99 9.52 0.04319 2.2769e-4 1	100 98.5 3.41 0.0458 3.9904e-4 1	97 96.9 5.44 14.1161 534.6567 1.1369e-13	100 99 2.85 0.04319 4.5312e-4 1
200,40	100 90.4 72.39 0.0045 5.4458e-4 1	100 90.4 75.03 0.0045 7.6189e-4 0.5	98.5 90.6 13.62 15.9653 1.2016e3 9.0949e-13	100 90.4 76.37 0.0043 8.4131e-4 0.5	100 90.4 10.44 0.0045 4.4918e-4 1	98.5 90.3 7.25 15.9046 1.1094e3 9.0949e-13	100 90.4 9.62 0.0043 4.2873e-4 1
200,100	100 80.4 233.43 5.8131e-4 9.3240e-4 1	100 80.5 234.59 5.8224e-4 9.2980e-4 1	100 79.9 34.61 2.2265 832.0167 9.0949e-13	100 81 247.16 5.6149e-4 9.1794e-4 1	100 80.5 14.5 5.8216e-4 4.7904e-4 1	100 78.1 9.01 1.2798 624.5682 9.0949e-13	100 81 9.01 5.6149e-4 4.9869e-4 1

TABLE 2. Results for different smooth functions using different algorithms with fixed training size.

(iii) Research about duality theory of TSSVM. In recent years, some duality theories were developed [31, 32, 33, 34] – strong duality theory with no duality gap, canonical duality theory proposed by Gao and so on [31, 32, 33, 34]. The authors believe that some more interesting results maybe be obtained by applying the duality theory.

Acknowledgements. The authors would like to thank David Yang Gao for his valuable suggestions. Comments from referees are acknowledged.

REFERENCES

[1] P. J. Bickel and K. A. Doksum, "Mathematical Statistics - Basic Ideas and Selected Topics" (Second Edition), Prentice -Hall, Inc. 2001.

[2] J. O. Berger, "Statistical Decision Theory and Bayesian Analysis," Springer Verlag, New York,1985.

[3] C. J. C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition.*, Data Mining and Knowledge Discovery, **2** (1998), 121-167.

[4] K. Fukunaga, "Introduction to Statistical Pattern Recognition," Academic Press,Inc.,1990.

[5] T. M. Mitchell, "Machine Learning," McGraw-Hill Companies Inc.,1997.

[6] T. Mitchell, "Statistical Approaches to Learning and Discovery," The course of Machine Learning at CMU, 2003.

[7] D. Montgomery, "Design and Analysis of Experiments," John Wiley & sons, Inc., 1991.

[8] B. Schölkoft, "Support Vector Learning," R. Oldenbourg Verlag, Munich,1997.

[9] V.Vapnik, "The Nature of Statistical Learning Theory," Springer,1995.

[10] V.Vapnik, *The support vector method of function estimation NATO ASI Series*, in "Neural Network and Machine Learning" (ed. C. Bishop),Springer,1998.

[11] V.Vapnik, *An overview of statistical learning theory*, in "Advanced in Kernel methods: Support Vector Learning" (eds. B. Scholkopf, B. Burges and A. Smola), The MIT Press, Cambridge, Massachusetts, 1999.

[12] V.Vapnik, *Three remarks on support vector function estimation*, IEEE transactions on Neural Networks, **10** (1999), 988-1000.

[13] Q. HE, Z. Z. SHI, L. A. REN and E. S. LEE, *A Novel Classification Method Based on Hyper-surface*, Mathematical and Computer Modelling, **38**(2003), 395-407.

[14] Ping-Feng Pai, *System reliability forecasting by support vector machines with genetic algorithms*, Mathematical and Computer Modelling, **43** (2006), 262-274.

[15] B. Chen and P.T. Harker, *Smooth Approximations to Nonlinear Complementarity Problems*, SIAM J. Optimization, **7** (1997), 403-420.

[16] C. Chen and O.L. Mangasarian, *A Class of Smoothing Functions for Nonlinear and Mixed Complementarity Problems*, Computational Optimization and Applications, **5** (1996), 97-138.

[17] C. Chen and O.L. Mangasarian, *Smoothing Methods for Convex Inequalities and Linear Complementarity Problems*, Math.Programming, **71** (1995), 51-69.

[18] X. Chen, L. Qi and D. Sun, *Global and Superlinear Convergence of the Smoothing Newton Method and Its Application to Parameterral Box Constrained Variational Inequalities*, Math. of Computation, **67** (1998), 519-540.

[19] X. Chen and Y. Ye, *On Homotopy-Smoothing Methods for Variational Inequalities*, SIAM J. Control and Optimization, **37** (1999), 589-616.

[20] Lee Yuh-Jye, Wen-Feng Hsieh, and Chien-Ming Huang, *ϵ-SSVR: A Smooth Support Vector Machine for ϵ-Insensitive Regression*, IEEE Transaction on Knowledge and Data Engineering, **17** (2005), 678-685.

[21] Lee Yuh-Jye and O. L. Mangarasian, *SSVM: A smooth support vector machine for classification*, Computational Optimization and Applications, **22** (2001), 5-21.

[22] Y. Yuan, J. Yan and C. Xu, *Polynomial Smooth Support Vector Machine(PSSVM)*, Chinese Journal Of computers, **28** (2005), 9-17.

[23] Y. Yuan and T. Huang, *A Polynomial Smooth Support Vector Machine for Classification*, Lecture Note on Artificial Intelligence, **3584** (2005), 157-164.

[24] O.L. Mangasarian and David R.Musicant, *Successive overrelaxation for support vector machines*, IEEE Transactions on Neural Networks, **10** (1999), 1032-1037.

[25] J. Platt, *Sequential minimal optimization: A fast algorithm for training support vector machines*, Advances in Kernel Methods-Support Vector Learning[R], 1999, 185-208.

[26] T. Joachims, *Making large-scale support vector machine learning practical*, in "Advanced in Kernel methods: Support Vector Learning" (eds. B. Scholkopf, B. Burges and A. Smola), The MIT Press, Cambridge, Massachusetts, 1999.

[27] Y. Yuan and R. Byrd, *Non-quasi-Newton updates for unconstrained optimization*, J. Comput. Math., **13** (1995), 95-107.

[28] Y. Yuan, *A modified BFGS algorithm for unconstrained optimization*, IMA J. Numer. Anal., **11** (1991), 325-332.

[29] Navneet Panda and Edward Y. Chang, *KDX: An Indexer for Support Vector Machines*, IEEE Transaction on Knowledge and Data Engineering, **18** (2006), 748-763.

[30] K. Schittkowski, *Optimal parameter selection in support vector machines*, Journal of Industrial and Management Optimization, **1** (2005), 465-476.

[31] Gao, D.Y., *Perfect duality theory and complete set of solutions to a class of global optimization*, Optimization, **52** (2003), 467-493.

[32] Gao, D.Y. , *Complete solutions to constrained quadratic optimization problems*, Journal of Global Optimisation, **29** (2004), 377-399.

[33] Gao, D.Y. , *Sufficient Conditions and Canonical Duality in Nonconvex Minimization with Inequality Constraints*, Journal of Industrial and Management Optimisation, **1** (2005), 53-63.

[34] Gao, D.Y. , *Complete Solutions and Extremality Criteria to Polynomial Optimization Problems*, Journal of Global Optimisation, **35** (2006), 131-143.

[35] K.F.C. Yiu, K.L. Mak, K.L. Teo, *Airfoil design via optimal control theory*, Journal of Industrial and Management Optimisation, **1** (2005), 133-148.

[36] A. Ghaffari Hadigheh and T. Terlaky,*Generalized support set invariancy sensitivity analysis in linear optimization*, Journal of Industrial and Management Optimisation, **2** (2006), 1-18.

[37] Z.Y. Wu, H.W.J. Lee, F.S. Bai and L.S. Zhang, *Quadratic smoothing approximation to* l_1 *exact penalty function in global optimization*, Journal of Industrial and Management Optimisation, **1** (2005), 533-547.

[38] Giovanni P. Crespi, Ivan Ginchev and Matteo Rocca, *Two approaches toward constrained vector optimization and identity of the solutions*, Journal of Industrial and Management Optimisation, **1** (2005), 549-563.

Received September 2006; 1st revision December 2006; final revision May 2007.

E-mail address: ybyuan@vt.edu