Journal of Data Analysis and Information Processing, 2013, 1, 1-7
http://dx.doi.org/10.4236/jdaip.2013.11001 Published Online February 2013 (http://www.scirp.org/journal/jdaip)

An Inexact Implementation of Smoothing Homotopy Method for Semi-Supervised Support Vector Machines

Huijuan Xiong, Feng Shi
College of Science, Huazhong Agricultural University, Wuhan, China
Email: shifeng@mail.hzau.edu.cn

Received January 12, 2013; revised February 16, 2013; accepted February 25, 2013

ABSTRACT

Semi-supervised Support Vector Machines is an appealing method for using unlabeled data in classification. Smoothing homotopy method is one of feasible method for solving semi-supervised support vector machines. In this paper, an inexact implementation of the smoothing homotopy method is considered. The numerical implementation is based on a truncated smoothing technique. By the new technique, many "non-active" data can be filtered during the computation of every iteration so that the computation cost is reduced greatly. Besides this, the global convergence can make better local minima and then result in lower test errors. Final numerical results verify the efficiency of the method.

Keywords: Semi-Supervised Classification; Support Vector Machines; Truncated Smoothing Technique; Global Convergence

1. Introduction

In the field of machine learning, it's essential to collect a large amounts of labeled data for the purpose of training learning algorithms. However, for many applications, huge number of data can be cheaply collected, but manual labeling of them is often a slow, expensive and error-prone process. It's desirable to utilize the unlabeled data points for the implementation of the learning task. The goal of semi-supervised classification is to employ the large collection of unlabeled data jointly with a few labeled data to finish the task of classification and prediction [11,18].

Semi-supervised support vector machines (S^3VMs) is an appealing method for the semi-supervised classification. In [7], K.P. Bennett *et al*. first formulated it as a mixed integer programming such that some state-of-the-art softwares can handle the formulation. Since that, a large number of methods have been applied to solve the non-convex optimization problem associated with S^3VMs, such as convex-concave procedures [5], non-differntiable methods [1], gradient descent method [13], continuation technique [12], branch-and-bound algorithms [7,14], and semi-definite programming [17] etc. A survey of these methods can be seen from [11,18].

As pointed out in [12], one reason for the large number of proposed algorithms for S^3VMs is that the resulting optimization problem is non-convex that generates local minima. Hence, it's necessary to find better local minima because better local minima tend to lead to higher prediction accuracy. In [12], a global continuation technique is presented. In [21], a similar global smoothing homotopy method is given. However, both the method is experiential and the calculations are lengthy.

The focus of this paper is giving a new efficient implementation of the smoothing homotopy method for the S^3VMs. In Section 2, we first introduce the new S^3VMs model used in [21] and list two smoothing functions called aggregate function and twice aggregate function respectively. The two smoothing functions are given to approximate the nonsmooth objective function (the detailed discussion of these two smoothing functions can be seen from [4]). And then the smoothing homtopy method solving S^3VMs is recalled. In Section 3, the new truncated smoothing technique is established to give a more efficient pathfollowing implementation of the smoothing homotopy method. The new technique is based on a fact that, some "non-active" data do little effect on the value of the smooth approximation functions with their gradients and Hessian during the computation, as a result, these "non-active" data can be filtered by the new truncated technique to save the computation cost. With the inexact computation technique, only a part of original data is necessary during the computation of every iteration. In the last section, Two artificial data sets with six standard test data from [10] are given to show the efficiency of our method.

A word about the notations in this paper. All vectors will be column vectors unless transposed to a row vector by a prime superscript T. The scalar (inner) product of

two vectors x and y in the n-dimensional real space R^n will be denoted by $x^{\mathrm{T}}y$. For a matrix $A \in R^{m\times n}$, A_i will denote the i th row of A. For a real number a, $|a|$ denotes its absolute value. For a vector $x \in R^n$, $\|x\|_1$ denotes its 1-norm, *i.e.*, $\|x\|_1 = \sum_{i=1}^{n}|x_i|$, $\|x\|_\infty$ denotes its infty-norm, *i.e.*, $\|x\|_\infty = \max_{1\le i\le n}|x_i|$. For an index set I, $|I|$ denotes the element number of it. For a given function $f: R^n \to R$, if f is smooth, its gradient is denoted by $\nabla f(x)$, if f is nondifferential, denote its subdifferential as $\partial f(x)$.

2. Semi-Supervised Support Vector Machines

There lies several formulations for S^{3}VMs such as the mixed integer programming model by K.P. Bennett *et al.* [7], the nonsmooth constrained optimization model by O.L. Mangasarian [5], and the smooth nonconvex programming formulation by O. Chapelle [13] and etc. Here we mention the contributions by O. Chapelle *et al.* in [11-14]).

Given a dataset consisting of m labeled points and p unlabeled points all in R^n, where the m labeled points are represented by the matrix $A \in R^{m\times n}$, p unlabeled points are represented by the matrix $B \in R^{p\times n}$ and the labels for A are given by $m\times m$ diagonal matrix D of ± 1. The linear S^{3}VMs to find a hyperplane $\omega^{\mathrm{T}}x = b$ far away from both the labelled and unlabeled points can be formulated as follows:

$$\begin{aligned} \min_{\omega,b} \quad & \frac{1}{2}\left(\omega^{\mathrm{T}}\omega + b^2\right) + C_1\overline{f}_1(\omega,b) + C_2\overline{f}_2(\omega,b), \\ \text{s.t.} \quad & \frac{1}{p}\left(\sum_{i=1}^{p}\omega^{\mathrm{T}}B_i + b\right) = \frac{1}{m}\sum_{i=1}^{m}D_{ii} \end{aligned} \tag{1}$$

where

$$\overline{f}_1(\omega,b) = \sum_{i=1}^{m}\max\left\{0,\, f_1^i(\omega,b)\right\}$$

$$\overline{f}_2(\omega,b) = \sum_{i=1}^{p}\max\left\{0,\, f_2^i(\omega,b)\right\}$$

$f_1^i(\omega,b)$ and $f_2^i(\omega,b)$ are loss functions corresponding to the labeled data and unlabeled data respectively and defined as follows,

$$f_1^i(\omega,b) = 1 - D_{ii}\left(\omega^{\mathrm{T}}A_i + b\right)$$

$$f_2^i(\omega,b) = 1 - \left|\omega^{\mathrm{T}}B_i + b\right|$$

where $\left|\omega^{\mathrm{T}}B_i + b\right|$ denotes the absolute value of $\omega^{\mathrm{T}}B_i + b$. The constraint is called balanced constraint that is used to avoid unbalanced solutions which classify all the unlabeled points in the same class.

For arbitrary vector $\xi \in R^n$, there lies an equivalent relation between its 1-norm and inf-norm in the sense that $\frac{1}{n}\|\xi\|_1 \le \|\xi\|_\infty \le \|\xi\|_1$, then the sum term of model (1) can be substituted by the inf-norm form and model (1) can be reformulated as follows,

$$\begin{aligned} \min_{\omega,b} \quad & \frac{1}{2}\left(\omega^{\mathrm{T}}\omega + b^2\right) + C_1 f_1^*(\omega,b) + C_2^* f_2^*(\omega,b) \\ \text{s.t.} \quad & \frac{1}{p}\left(\sum_{i=1}^{p}\omega^{T}B_i + b\right) = \frac{1}{m}\sum_{i=1}^{m}D_{ii} \end{aligned} \tag{2}$$

where

$$f_1^*(\omega,b) = \max_{1\le i\le m}\left\{0,\, f_1^i(\omega,b)\right\}$$

$$f_2(\omega,b) = \max_{1\le i\le p}\left\{0,\, f_2^i(\omega,b)\right\}.$$

We rewrite the constraint into the objective as a barrier term and reformulate $f_2^i(\omega,b)$ into its equivalent formulation $f_2^i(\omega,b) = \min\left\{1-\omega^{\mathrm{T}}B_i - b, 1+\omega^{\mathrm{T}}B_i + b\right\}$, and then obtain the following formulation that is our goal in the paper.

$$\min_{\omega,b} \quad \frac{1}{2}\left(\omega^{T}\omega + b^2\right) + C_1 f_1(\omega,b) + C_2 f_2(\omega,b) + f_3(\omega,b)$$

where

$$f_1(\omega,b) = \max_{1\le i\le m}\left\{0,\, f_1^i(\omega,b)\right\}$$

$$f_2(\omega,b) = \max_{1\le i\le p}\left\{0, \min\left\{f_2^{i1}(\omega,b),\, f_2^{i2}(\omega,b)\right\}\right\}$$

$$f_2^{i1}(\omega,b) = 1 - \omega^{\mathrm{T}}B_i - b$$

$$f_2^{i2}(\omega,b) = 1 + \omega^{\mathrm{T}}B_i + b$$

$$f_3(\omega,b) = M\left|\frac{1}{p}\left(\sum_{i=1}^{p}\omega^{\mathrm{T}}B_i + b\right) - \frac{1}{m}\sum_{i=1}^{m}D_{ii}\right|^2$$

M is a barrier parameter.

If the dataset is nonlinear separable, we need construct a surface separation based on some kernel trick (detailed discussion of it can be seen from [2] and etc.). Denote $\overline{A} = \left[A; B\right]^{\mathrm{T}}$, assume that the surface we want to find is $K\left(x, \overline{A}\right)u + b = 0$, where $K(\cdot,\cdot)$ is usually taken as Gaussian kernel function with the form of

$$K(x,y) = \exp\left(-\mu\|x-y\|^2\right)$$

μ is the kernel parameter, $u \in R^{m+p}$. To find the nonlinear decision surface, we need to solve the following problem:

$$\begin{aligned} \min_{u,b} \quad f(u,b) = {} & \frac{1}{2}\left(u^{\mathrm{T}}u + b^2\right) + C_1 f_1(u,b) \\ & + C_2 f_2(u,b) + f_3(u,b) \end{aligned} \tag{4}$$

where

$$f_1(u,b)=\max_{1\le i\le m}\left\{0,1-D_{ii}\left(K\left(A_i^{\mathrm{T}},\bar{A}\right)u+b\right)\right\}$$

$$f_2(u,b)=\max_{1\le i\le p}\left\{0,1+\min\left\{K\left(B_i^{\mathrm{T}},\bar{A}\right)u+b,-K\left(B_i^{\mathrm{T}},\bar{A}\right)u-b\right\}\right\}$$

$$f_3(u,b)=M\left|\frac{1}{p}\left(\sum_{i=1}^{p}K\left(B_i^{\mathrm{T}},\bar{A}\right)u+b\right)-\frac{1}{m}\sum_{i=1}^{m}D_{ii}\right|^2$$

2.1. Aggregate Function and Aggregate Homotopy Method

Aggregate function is an attractive smooth approximate function. It has been used extensively for the the non-smooth min-max problem [4], variational inequalities [6], mathematical programm with equilibrium constraints (MPEC) [16], non-smooth min-max-min problem [6] and etc. In the following, we will utilize the approximate function with its modification to establish an globally convergent method, called as aggregate homotopy method, for solving model (3) or (4).

In short, let $x=(\omega,b), s=n+1$ (or, $x=(u,b), s=m+p+1$) and denote model (3) or (4) as the following unified formulation:

$$\min_x f(x)=f_1(x)+f_2(x) \tag{5}$$

where

$$\begin{aligned}
&f_1(x)=\max_{1\le i\le m}\left\{\bar{f}_1^{\,i}(x)\right\},\\
&f_2(x)=\max_{1\le i\le p}\left\{\min\left\{\bar{f}_2^{\,i1}(x),\bar{f}_2^{\,i2}(x)\right\}\right\},\\
&\bar{f}_1^{\,i}(x)=\frac{1}{2}x^{\mathrm{T}}x+f_1^{\,i}(x),\\
&\bar{f}_2^{\,i1}(x)=f_2^{\,i1}(x)+f_3(x),\\
&\bar{f}_2^{\,i2}(x)=f_2^{\,i2}(x)+f_3(x),
\end{aligned} \tag{6}$$

and $f_1^{\,i}(x), f_2^{\,i1}(x), f_2^{\,i2}(x), f_3(x)$ are defined as (3) or (4).

Denote $f_2^{\,i}(x)=\min\left\{f_2^{\,i1}(x),f_2^{\,i2}(x)\right\}$, based on the optimality condition of non-smooth optimization theory in [9], we know that the subdifferential of $f_1(x)$ and $f_2(x)$ can be computed as follows,

$$\begin{aligned}
&\partial f_1(x)=x+\sum_{i\in I_1(x)}\lambda_i\left[A_i^{\mathrm{T}};1\right],\\
&\partial f_2(x)=\sum_{i\in I_2(x)}\eta_i\sum_{j\in J_2^i(x)}\delta_{ij}\nabla f_2^{ij}(x)+\nabla f_3(x)
\end{aligned} \tag{7}$$

where

$$I_1(x)=\left\{i\in\{1,\cdots,m\}:f_1^{\,i}(x)=f_1(x)\right\}$$

$$\lambda_i\in[0,1],\ \sum_{i\in I_1(x)}\lambda_i=1$$

$$I_2(x)=\left\{i\in\{1,\cdots,p\}:f_2^{\,i}(x)=f_2(x)\right\}$$

$$J_2^i(x)=\left\{j\in\{1,2\}:f_2^{ij}(x)=f_2^{\,i}(x)\right\}$$

$$\eta_i\in[0,1],\ \sum_{i\in I_2(x)}\eta_i=1$$

$$\delta_{ij}\in[0,1],\ \sum_{j\in J_2^i(x)}\delta_{ij}=1$$

$$\nabla f_2^{i1}(x)=\left[B_i^{\mathrm{T}};1\right]$$

$$\nabla f_2^{i2}(x)=-\nabla f_2^{i1}(x)$$

moreover, a point x^* can be called a stationary point or a solution point of (5) if satisfying $0\in\partial f(x^*)$.

To solve model (5) by an aggregate homotopy method, we first introduce the following two smoothing functions,

$$\begin{aligned}
f_1(x,t)&=t\ln\left(\sum_{i=1}^{m}\exp\left(\frac{\bar{f}_1^{\,i}(x)}{t}\right)\right)\\
f_2(x,t)&=t\ln\left(\sum_{i=1}^{p}\exp\left(\frac{\bar{f}_2^{\,i}(x,t)}{t}\right)\right)
\end{aligned} \tag{8}$$

where $\bar{f}_i^{\,i}(x)$ is defined as (6) and

$$\bar{f}_2^{\,i}(x,t)=-t\ln\left(\exp\left(-\frac{\bar{f}_2^{\,i1}(x)}{t}\right)+\exp\left(-\frac{\bar{f}_2^{\,i2}(x)}{t}\right)\right).$$

We call $f_1(x,t)$ and $f_2(x,t)$ as aggregate function and twice aggregate function respectively. The two smoothing functions have many good properties such as uniform approximation and etc. More details can be seen from [19].

Using above two uniformly approximations functions in (8), we define the following homotopy mapping:

$$H_{x^0}(x,t)=(1-t)\nabla_x f(x,t)+t\left(x-x^0\right)=0 \tag{9}$$

where $x^0\in R^n$ is an arbitrarily initial point and

$$\nabla_x f(x,t)=\nabla_x f_1(x,t)+\nabla_x f_2(x,t).$$

We call Equation (9) as an aggregate homotopy equation. It contains two limiting problems. On the one hand as $t=1$, it has a unique solution $x=x^0$. On the other hand, as $t\to 0$, the solution of it approaches to a stationary point of (5), *i.e.*, a solution x satisfying $0\in\partial f(x)$.

For a given initial point $x^0\in R^s$, we denote the zeros point set of the aggregate homotopy mapping $H_{x^0}(\cdot,\cdot):R^s\times[0,1]\to R$ as

$$H_{x^0}^{-1}(0)=\left\{(x,t)\in R^s\times[0,1]:H_{x^0}(x,t)=0\right\} \tag{10}$$

It can be proved that $H_{x^0}^{-1}(0)$ includes a smooth path Γ with no bifurcation points, starting from $(x^0,1)$ and

approaching to the hyperplane $t=0$ that leads to a solution of the original problem [21].

3. Inexact Predictor-Corrector Implementation of the Aggregate Homotopy Method

The path-following of the homotopy path Γ can be implemented by some predictor-corrector procedure. Some detailed discussion on the predictor-corrector algorithm with the convergence can be seen from [3,8] and etc. In the following, we first give the framework of the predictor-corrector procedure in this paper, and then discuss how to make an inexact predictor-corrector implementation.

3.1. Predictor-Corrector Path-Following Algorithm

Parameters. initial stepsize h_0, maximum stepsize h_m, minimum stepsize h_ε, stop criteria $\varepsilon_1>0$ for procedure terminated, stop criteria $\varepsilon_N>0$ for Newton corrector stopped, criteria $\varepsilon_c>0$ for judging corrector plane, counter $N_i=0$.

Data. $\left(x^{(0)},t_0\right)\in R^s\times\{1\}$,

Step 0. $h=h_0$, $d_0=[0;-1]$, $k=0$.

Step 1. Compute a predictor point $\left(\overline{x}^{(k)},\overline{t}_k\right)$

1) Solve the following linear equation

$$\begin{bmatrix} DH\left(x^{(k)},t_k\right) \\ d_0^T \end{bmatrix} d_1=\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

to obtain a unit tangent vector $d_1=\dfrac{d_1}{\|d_1\|}$;

2) $h=\min\left(h_m,h\right)$; $\left(\overline{x}^{(k)},\overline{t}_k\right)=\left(x^{(k)},t_k\right)+hd_1$;

3) If $\overline{t}_k<0$ or $\overline{t}_k>1$, $h=\dfrac{h}{2}$, return to 2); else, go to Step 2;

Step 2. Compute a corrector point $\left(x^{(k+1)},t_{k+1}\right)$

1) If $\overline{x}_k<\varepsilon_c$, take $d=[0;1]$ and $h=\min\{h,h_\varepsilon\}$; else, take $d=d_1$. Go to 2);

2) Solve equation

$$\begin{cases} H\left(x,t\right)=0, \\ d^{\mathrm{T}}\left[x-\overline{x}^k;t-\overline{t}_k\right]=0 \end{cases}$$

by Newton method with the stopping criteria ε_N and go to 3);

3) If Newton corrector fail, $h=0.7h$, go to 2); else, denote the solution as $\left(x^{k+1},t_{k+1}\right)$, go to 4);

4) If $t_{k+1}>1$ or $t_{k+1}<0$, $h=\dfrac{h}{2}$, go to 2); else, go to 5);

5) If $t_k<\varepsilon_1$, stop ; else $d_0=d$, $k=k+1$, go to 6);

6) $N_i=N_i+1$. If $N_i>3$ and the iteration number of Newton corrector is less than 3, go to 7); else , go to 1);

7) $N_i=0$, $h=1.5h$; go to 1).

Notice that, the main computation cost of **Algorithm 3.1** lies in the equations solver in Step 1) and 2), some inexact computation technique can be introduce to save the computation cost of them. we take the following approximate homotopy equation $\tilde{H}\left(x,t\right)$ with its Jacobian $D\tilde{H}\left(x,t\right)$ in place of the original $H\left(x,t\right)$ with $DH\left(x,t\right)$ during the computation of step 1) and 2).

Given parameters $\varepsilon_1\left(x,t\right)>0$, $\varepsilon_2\left(x,t\right)>0$, denote $M=\{1,\cdots,m\}$, $P=\{1,\cdots,p\}$,

$$I_1\left(x,t\right)=\left\{i\in M:f_1^i\left(x\right)-f_1\left(x\right)>-\varepsilon_1\left(x,t\right)\right\},$$

$$f_2^i\left(x,t\right)=-t\ln\left(\exp\left(-\frac{f_2^{i1}\left(x\right)}{t}\right)+\exp\left(-\frac{f_2^{i2}\left(x\right)}{t}\right)\right),$$

$$I_2\left(x,t\right)=\left\{i\in P:f_2^i\left(x,t\right)-\max_{i\in P}\left\{f_2^i\left(x,t\right)\right\}>-\varepsilon_2\left(x,t\right)\right\},$$

$$\tilde{f}_1\left(x,t\right)=\frac{1}{2}x^Tx+t\ln\left(\sum_{i\in I_1(x,t)}\exp\left(\frac{f_1^i\left(x\right)}{t}\right)\right),$$

$$\tilde{f}_2\left(x,t\right)=f_3\left(x\right)+t\ln\left(\sum_{i\in I_2(x,t)}\exp\left(\frac{f_2^i\left(x,t\right)}{t}\right)\right),$$

$$\begin{aligned}\tilde{H}\left(x,t\right)&=\left(1-t\right)\nabla_x\tilde{f}\left(x,t\right)+t\left(x-x^0\right)\\&=\left(1-t\right)\left(\nabla_x\tilde{f}_1\left(x,t\right)+\nabla_x\tilde{f}_2\left(x,t\right)\right)+t\left(x-x^0\right),\end{aligned}$$

$$D\tilde{H}\left(x,t\right)=\left(\frac{\partial H}{\partial x},\frac{\partial H}{\partial t}\right)$$

where

$$\frac{\partial H}{\partial x}=\left(1-t\right)\left(\nabla_x^2\tilde{f}_1\left(x,t\right)+\nabla_x^2\tilde{f}_2\left(x,t\right)\right)+tI_s$$

$$\begin{aligned}\frac{\partial H}{\partial t}&=x-x^0-\left(\nabla_x\tilde{f}_1\left(x,t\right)+\nabla_x\tilde{f}_2\left(x,t\right)\right)\\&\quad+\left(1-t\right)\left(\nabla_{xt}^2\tilde{f}_1\left(x,t\right)+\nabla_{xt}^2\tilde{f}_2\left(x,t\right)\right)\end{aligned}$$

It's proven in [20] that, only if the error $\left\|\tilde{H}\left(x,t\right)-H\left(x,t\right)\right\|$ and $\left\|D\tilde{H}\left(x,t\right)-DH\left(x,t\right)\right\|$ are small enough, or equivalently, $\varepsilon_1\left(x,t\right)$ and $\varepsilon_2\left(x,t\right)$ are chosen appropriately, the approximate Euler-Newton predictor-corrector also approaches to a solution of original problem. Here we only list the main results and omit the proofs.

Denote

$$E_1\left(x,t\right)=\left\|\tilde{H}\left(x,t\right)-H\left(x,t\right)\right\|$$

and $E_2\left(x,t\right)=\left\|D\tilde{H}\left(x,t\right)-DH\left(x,t\right)\right\|$, $\tilde{d}\left(x,t\right)$ is the unit tangent vector obtained by the approximate computation, $d\left(x,t\right)$ is the tangent vector by exact Euler

predictor, we have the following lemma to guarantee the efficiency of the approximate tangent vector.

3.2. Lemma

For a given $(x,t)\in\Gamma$, if $E_2(x,t)$ is small enough and satisfies

$$cond(DH)\frac{E_2(x,t)}{\|DH\|}+O\left(E_2(x,t)^2\right)<\sqrt{2}$$

The approximate unit predictor tangent vector $\tilde{d}(x,t)$ is effective, *i.e.*, $\tilde{d}(x,t)$ still makes a direction with arclength increased.

During the correction process, the following equation must be solved

$$F(x,t)=\begin{pmatrix} H(x,t) \\ d^{\mathrm{T}}\left((x,t)-(x^0,t^0)\right)\end{pmatrix}=0 \qquad (11)$$

where (x^0,t^0) is an appropriate predictor point obtained from the former predictor step. We adopt the following approximate Newton method to solve (11),

$$(x^{k+1},t_{k+1})=(x^k,t_k)-\left(\tilde{F}(x^k,t_k)\right)^{-1}\tilde{F}(x^k,t_k) \qquad (12)$$

where

$$\left(\tilde{F}(x^k,t_k)\right)^{-1}=\begin{pmatrix} D\tilde{H}(x^k,t_k)\\ d^{\mathrm{T}}\end{pmatrix}^{-1},$$

$$\tilde{F}(x^k,t_k)=\begin{pmatrix} \tilde{H}(x^k,t_k)\\ d^{\mathrm{T}}\left((x^k,t_k)-(x^0,t_0)\right)\end{pmatrix}$$

From 0 is a regular value of $H(x,t)$ and d is a unit tangent vector induced by $\tilde{D}H(x,t)$, we know, if the step h is chosen appropriately, the equation (11) has a solution and the approximate Newton iteration (12) is effective.

3.3. Approximate Newton Corrector Convergence Theorem

Suppose that $F(x,t)=0$ have solution (x^*,t_*) with nonsingular $F'(x^*,t_*)$, there exists a neighborhood $S=\overline{S}\left((x^*,t_*),\delta\right)$ and $\gamma_1>0,\gamma_2>0$, for any $(x^0,t^0)\in S$, if for each $k=1,2,\cdots$, $E_1(x^k,t_k)$ and $E_2(x^k,t_k)$ satisfying

$$E_1(x^k,t_k)\le\gamma_1, E_2(x^k,t_k)\le\gamma_2\left\|F(x^k,t_k)\right\|.$$

Then the approximate Newton iteration point sequence $\{(x^k,t_k)\}$, from (12) is well defined and converges to (x^*,t_*).

4. Numerical Results

In this section, some numerical examples and comparisons are given to illustrate the efficiency of our method. Two artificial datasets are generated first. The first one consists of 34 points generated by "rand" function, 14 of them are labeled and the remaining 30 are seen as unlabeled data. The second one consists of 242 points taken from two nonlinear bihelix curves that are generated by $\rho=a\theta+b$, where one is obtained by taking $a=b=0.2$, the other is by taking $a=0.2,b=0.3$, $\theta\in[0:\pi/40:3\pi]$. We take randomly 30% of them as labeled and the remaining 70% as unlabeled. The comparisons of our method with the LSVM method from [15] without the consideration of unlabeled data are given. Final results are illustrated in **Figures 1** and **2**.

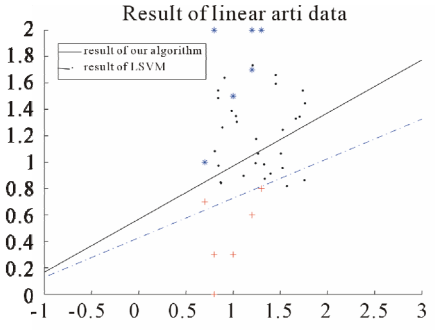

Figure 1. Result on linear artificial data of LSVM and the new algorithm.

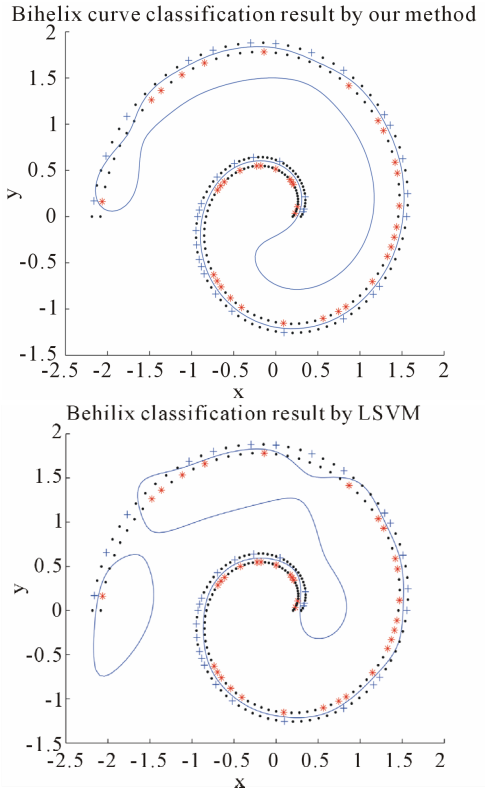

Figure 2. Result on nonlinear bihelix curve artificial data of LSVM and the new algorithm.

To reveal the efficiency of our algorithm for S^3VMs, comparisons of our algorithm (InSH) with some other existing algorithms for S^3VMs such as the convex-concave procedure in [5] (vS^3VM), the continuation method in [12] (cS^3VM) and the gradient descent method in [13] (∇ S^3VM), are given for some standard test data from [10]. For the linear programming subproblem arising in [5], we solve it with the Matlab function *linprog* in optimization toolbox. The comparison results are listed in the following **Table 1**.

Table 1. Comparison Results on test data (test error %).

Dataset $(m+p)\times n$	InSH (C_1, C_2, M)	vS3VM (C_1, C_2)	cS3VM (C_1, C_2)	∇S3VM (C_1, C_2)
Ionosphere (351 × 34)	11.03 (1, 0.25, 1e - 3)	11.66 (8, 2)	11.34 (100, 10)	13.37 (1, 0.1)
MUSK1 (476 × 166)	12.26 (1, 1, 0.1)	15.93 (2, 2)	15.75 (1, 1)	15.34 (100, 10)
NDC (1000 × 32)	13.60 (2, 2, 0.05)	13.90 (4, 2)	14.60 (100, 1)	24.00 (0.01, 0.1)
Pima (769 × 8)	22.66 (8, 2, 0.05)	22.65 (10, 1)	23.42 (8, 0.5)	39.45 (10, 1)
Sonar (208 × 60)	15.45 (16, 0.5, 0.01)	23.02 (100, 1)	23.10 (100, 10)	23.50 (100, 10)
Votes (435 × 16)	3.44 (8, 2, 2)	4.59 (1, 0.1)	-	4.38 (10, 1)
Aver. Time (sec.)	5.7367	9.2271	7.3925	6.0676

-: denotes the method fails for the dataset.

All the computations are performed on a computer running the software Matlab 7.0 on Microsoft Windows Vista with Intel(R) Core(TM)2 Duo CPU 1.83 GHz processor and 1789 megabytes of memory. During the computation, we take $h_0 = 0.1$, $h_m = 1$, $h_\varepsilon = 1e-3$, $\varepsilon_1 = 1e-3$, $\varepsilon_N = 1e-3$, $\varepsilon_c = 1e-3$, $C, C^* \in \left[2^{-3}, 2^3\right] \cup [1e-3, 1e+2]$, $M \in [1e-3, 2]$ are taken as the one for the least test error. If necessary, the kernel parameter μ is taken the best leads to the least test error among $[1e-1, 3]$. The parameters for determining the inexact index set are taken as

$$\varepsilon_1(x,t) = \begin{cases} t\ln\left(\max\left(\dfrac{4(m-1)}{\delta_2}, 1\right)\right), & t = 1; \\ t\ln\left(2(m-1)\max\left\{\dfrac{2p}{\delta_1}, \dfrac{2q}{\delta_2}, 1\right\}\right), & t \neq 1. \end{cases}$$

where $p = 2(1-t)B_f$, $q = 6(1-t)B_g\left(B_f/t + B_g\right)/t + 2B_g$, $B_f = \max\left(\left|f_1^i(x)\right|\right)$ and $B_g = \max\left(\max\left(|A|\right)\right)$. $\varepsilon_2(x,t)$ has the same expression as $\varepsilon_1(x,t)$ where $B_f = \max\left(\left|f_2^i(x,t)\right|\right)$ and $B_g = \max\left(\max\left(|B|\right)\right)$. δ_1 and δ_2 are taken as $\delta_1 = \delta_2 = 1e-3$ that are given to bound the error of $\left\|H - \tilde{H}\right\|$ and $\left\|DH - D\tilde{H}\right\|$.

5. Acknowledgements

The research was supported by the National Nature Science Foundation of China (No. 11001092) and the Tian-Yuan Special Funds of the National Natural Science Foundation of China (Grant No. 11226304).

REFERENCES

[1] A. Astorino and A. Fuduli, "Nonsmooth Optimization Techniques for Semi-Supervised Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 12, 2007, pp. 2135-2142. doi:10.1109/TPAMI.2007.1102

[2] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition, "*Data Mining and Knowledge Discovery*, Vol. 2, No. 2, 1998, pp. 121-167. doi:10.1023/A:1009715923555

[3] E. L. Allgower and K. Georg, "Numerical Continuation Methods: An Introduction," Springer-Vergal, Berlin, 1990. doi:10.1007/978-3-642-61257-2

[4] E. Polak, J. O. Royset and R. S. Womersley, "Algorithms with Adaptive Smoothing for Finite Minimax Problems," *Journal of Optimization Theory and Application*, Vol. 119, No. 3, 2003, pp. 459-484. doi:10.1023/B:JOTA.0000006685.60019.3e

[5] G. Fung and O. Mangasarian, "Semi-Supervised Support Vector Machines for Unlabeled Data Classification," *Optimization Methods and Software*, Vol. 15, No. 1, 2001, pp. 29-44. doi:10.1080/10556780108805809

[6] G. X., Liu, "Aggregate Homotopy Methods for Solving Sequential Max-Min Problems, Complementarity Problems and Variational Inequalities," PhD Thesis, Jilin University, Jilin, 2003.

[7] K. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines," In: M. S. Kearns, S. A. Solla and D. A. Cohn, Eds, *Advances in Neural Information Processing Systems*, MIT Press, Vol. 10, 1998, pp. 368-374.

[8] L. T. Watson, S. C. Billups and A. P. Morgan, "Algorithm 652 Hompack: A Suite of Codes for Globally Convergent Homotopy Algorithms," *ACM Transactions on Mathematical Software*, Vol. 13, No. 3, 1987, pp. 281-310. doi:10.1145/29380.214343

[9] M. M. Mkela and P. Neittaanmki, "Nonsmooth Optimizatin: Analysis and Algorithms with Application to Optimal Control," Utopia Press, Singapore, 1992.

[10] P. M. Murphy and D. W. Aha, "UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[11] O. Chapelle, V. Sindhwani and S. S. Keerthi, "Optimization Techniques for Semi-Supervised Support Vector Machines," *Journal of Machine Learning Research*, Vol. 9, 2008, pp. 203-233.

[12] O. Chapelle, M. Chi and A. Zien, "A Continuation Method for Semi-Supervised SVMs," *ACM International Conference Proceeding Series*, *Proceedings of the 23rd international conference on Machine learning*, Vol. 148, 2006, pp. 185-192.

[13] O. Chapelle and A. Zien, "Semi-Supervised Classification by Low Density Separation," *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Vol. 1, 2005, pp. 57-64.

[14] O. Chapelle, V. Sindwani and S. Keerthi, "Branch and Bound for Semi-Supervised Support Vector Machines," *Advances in Neural Information Processing Systems* 19, *Proceedings of the* 2006 *Conference*, MIT Press, Cambridge, 2007, pp. 217-224.

[15] O. L. Mangasarian and D. R. Musicant, "Lagrangian Support Vector Machines," *Journal of Machine Learning Research*, Vol. 1, 2001, pp. 161-177.

[16] S. Birbil, S. C. Fang and J. Han, "Entropic Regularization Approach for Mathematical Programs with Equilibrium Constraints," *Technical Report*, *Industrial Engineering and Operations Research*, Carolina, 2002.

[17] T. D. Bie, N. Cristianini, "Semi-Supervised Learning Using Semi-Definite Programming," In: O. Chapelle, B. Scholkopf and A. Zien, Eds., *Semi-Supervised Learning*, MIT Press, Cambridge, 2006.

[18] X. J. Zhu, "Semi-Supervised Learning Literature Survey," Technical Report 1530, Computer Science, University of Wisconsin-Madison, 2005.

[19] X. S. Li and S. C. Fang, "On the Entropic Regularization Method for Solving Min-Max Problems with Applications," *Mathematical Methods and Operations Research*, Vol. 46, No. 1, 1997, pp. 119-130. doi:10.1007/BF01199466

[20] Y. Xiao, H. J. Xiong and B. Yu, "Truncated Aggregate Homotopy Method for Nonconvex Nonlinear Programming," *Optimization Methods and Software*, 2012, pp. 1-18.

[21] H. J. Xiong and B. Yu, "Aggregate Homotopy Method for Semi-Supervised SVMs," 2011 *International Conference on Electric Information and Control Engineering*, pp. 1147-1150.

