Research Article

A New Conic Approach to Semisupervised Support Vector Machines

Ye Tian,¹ Jian Luo,² and Xin Yan³

¹*School of Business Administration and Collaborative Innovation Center of Financial Security, Southwestern University of Finance and Economics, Chengdu 611130, China*

²*School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian 116025, China*

³*Department of Mathematics, Shanghai University, 99 Shangda Road, Shanghai 200444, China*

Correspondence should be addressed to Jian Luo; luojian546@hotmail.com

Received 1 January 2016; Accepted 23 February 2016

Academic Editor: Jean-Christophe Ponsart

Copyright © 2016 Ye Tian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a completely positive programming reformulation of the 2-norm soft margin S^3VM model. Then, we construct a sequence of computable cones of nonnegative quadratic forms over a union of second-order cones to approximate the underlying completely positive cone. An ϵ -optimal solution can be found in finite iterations using semidefinite programming techniques by our method. Moreover, in order to obtain a good lower bound efficiently, an adaptive scheme is adopted in our approximation algorithm. The numerical results show that the proposed algorithm can achieve more accurate classifications than other well-known conic relaxations of semisupervised support vector machine models in the literature.

1. Introduction

Support vector machine (SVM) is a novel and important machine learning method for classification and pattern recognition. Ever since the first appearance of SVM models around 1995 [1], they have attracted a great deal of attention from numerous researchers due to their attractive theoretical properties and a wide range of applications in the recent two decades [2–5].

Notice that traditional SVM models use only labeled data points to get the separating hyperplane. However, labeled instances are often difficult, expensive, or time-consuming to be obtained, since they require much effort of experienced human annotators [6]. Meanwhile, unlabeled data points (i.e., the points only with feature information) are usually much easier to be collected but seldom used. Therefore, as the natural extensions of SVM models, semisupervised support vector machines (S^3VM) models address the classification problem by using the large amount of unlabeled data together with the labeled data to build better classifications. The main idea of S^3VM is to maximize the margin between two classes in the presence of unlabeled data, by keeping the boundary

traversing through low density regions while respecting labels in the input space [7].

To the best of our knowledge, most of the traditional SVM models are polynomial-time solvable problems. However, the S^3VM models are formulated as mixed integer quadratic programming (MIQP) problem, which cause computational difficulty in general [8]. Therefore, researchers have proposed several optimization methods for solving the nonconvex quadratic programming problems associated with S^3VM . Joachims [9] developed a local combinatorial search. Blum and Chawla [10] proposed a graph-based method. Lee et al. [11] applied the entropy minimization principle to the semisupervised learning for image pixel classification. Besides, some classical techniques for solving MIQP problem are used for S^3VM models, such as branch-and-bound method [12], cutting plane method [13], gradient descent method [14], convex-concave procedures [15], surrogate functions [16], deterministic methods [17], and semidefinite relaxation [18]. For a comprehensive survey of the methods, we refer to Zhu and Goldberg [19]. It is worth pointing out that the linear conic approaches (semidefinite and doubly nonnegative

relaxations) are generally quite efficient among these methods [20, 21].

Recently, a new but important linear conic tool called completely positive programming (CPP) has been used to study the nonconvex quadratic program with linear and binary constraints. Burer [22] has pointed out that this type of quadratic program can be equivalently modeled as a linear conic program over the cone of completely positive matrices. Here, “equivalent” means these two programs have the same optimal value. This result shows a new angle to analyze the structure of the quadratic and combinatorial problems and provides a new way to approach them. But, unfortunately, the cone of completely positive matrices is not computable; that is, detecting whether a matrix belongs to the cone is NP-hard [23]. Thus, a natural way for deriving a polynomial-time solvable approximation is to replace the cone of completely positive matrices by some computable cones. Note that two commonly used computable cones are the cone of positive semidefinite matrices and cone of doubly nonnegative matrices which lead to the semidefinite and doubly nonnegative relaxations, respectively [21].

However, these relaxations cannot further improve the lower bounds. Hence, these methods are not suitable for some situations with high accuracy requirement. Therefore, in this paper, we propose a new approximation to the 2-norm soft margin S^3VM model. This method is based on a sequence of computable cones of nonnegative quadratic forms over a union of second-order cones. It is worth pointing out that this approximation can get an ϵ -optimal solution in finite iterations. This method also provides a novel angle to approach the S^3VM model. Moreover, we design an adaptive scheme to improve the efficiency of the algorithm. The numerical results show that our method can achieve better classification rates than other benchmark conic relaxations.

The paper is arranged as follows. In Section 2, we briefly review the basic models of S^3VM and show how to reformulate the corresponding MIQP problem as a completely positive programming problem. In Section 3, we use the computable cones of nonnegative quadratic forms over a union of second-order cones to approximate the underlying cone of completely positive matrices in the reformulation. Then, we prove that this approximation algorithm can get an ϵ -optimal solution in finite iterations. An adaptive scheme is proposed to reduce the computational burden and improve the efficiency in Section 4. In Section 5, we investigate the effectiveness of this proposed algorithm using some artificial and real-world benchmark datasets. At last, we summarize the paper in the final section.

2. Semisupervised Support Vector Machines

Before we start the paper, we introduce some notation used later. $\mathbb{N}_+$ denotes the set of positive integers. e^n and $\mathbf{0}^n$ denote the n -dimensional vectors with all elements being 1 and 0, respectively. I_n is the identity matrix of order n . For a vector $x \in \mathbb{R}^n$, let x_i denote the i_{th} component of x . For two vectors $a, b \in \mathbb{R}^n$, $a \circ b$ denotes the elementwise product of vectors a and b . Moreover, $\mathcal{S}^n$ and $\mathcal{S}_+^n$ denote the set of $n \times n$ real

symmetric matrices and the set of $n \times n$ positive semidefinite matrices, respectively. Besides, $\mathcal{N}_+^n$ denotes the set of $n \times n$ real symmetric matrices with all elements being nonnegative. For two matrices $A, B \in \mathcal{S}^n$, $A \circ B$ denotes the element-wise product of matrices A and B , $A \cdot B = \text{trace}(A^T B) = \sum_{i=1}^n \sum_{j=1}^n A_{i,j} B_{i,j}$, where $A_{i,j}$ and $B_{i,j}$ denote the elements of A and B in the i_{th} row and j_{th} column, respectively. For a nonempty set $F \in \mathbb{R}^n$, $\text{int}(F)$ denotes the interior of F , $\text{cl}(F)$ and $\text{Cone}(F) = \{x \in \mathbb{R}^n \mid \exists m, \lambda_i \geq 0, y_i \in F, i = 1, \dots, m, \text{ s.t. } x = \sum_{i=1}^m \lambda_i y_i\}$ stands for the closure and the conic hull of F .

Now we briefly recall the basic model of 2-norm soft margin S^3VM . Given a dataset of n data points $\{x^i\}_{i=1}^n$, where $x^i = (x_1^i, \dots, x_m^i)^T \in \mathbb{R}^m$. Let $y = ((y^a)^T, (y^b)^T)^T$ be the indicator vector, where $y^a = (y_1, \dots, y_l)^T \in \{-1, 1\}^l$ is known while $y^b = (y_{l+1}, \dots, y_n)^T \in \{-1, 1\}^{n-l}$ is unknown. In order to handle nonlinearity in the data structure, researchers propose a method by projecting these nonlinear problems into some linear problems in a high-dimensional feature space via a feature function $\phi(x) : \mathbb{R}^m \rightarrow \mathbb{R}^d$, where d is the dimension of the feature space [24]. Then the points are separated by hyperplane $w^T \phi(x) + b = 0$ in the new space, where $w \in \mathbb{R}^d$, $b \in \mathbb{R}$. For the linearly inseparable data points, the slack variables $\eta = (\eta_1, \dots, \eta_n) \in \mathbb{R}^n$ are used to measure the misclassification errors if the (labeled or unlabeled) points do not fall in certain side of the hyperplane [12]. The error η is penalized in the objective function of S^3VM models by multiplying a positive penalty parameter $C > 0$. Moreover, in order to avoid the nonconvexity in the reformulated problem, a tradition trick is to drop the bias term b [8]. It is worth pointing out that this negative effect can be mitigated by centering the data at the origin [18]. Like the traditional SVM model, the main idea of S^3VM models is to classify labeled and unlabeled data points into two classes with a maximum separation between them.

Above all, the 2-norm soft margin S^3VM model with kernel function $\phi(x) : \mathbb{R}^m \rightarrow \mathbb{R}^d$ can be written as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|_2^2 + C \|\eta\|_2^2 \\ \text{s.t.} \quad & y_i w^T \phi(x^i) \geq 1 - \eta_i, \quad i = 1, \dots, n, \\ & y^b \in \{-1, 1\}^{n-l}. \end{aligned} \quad (1)$$

To handle the kernel function ϕ , a kernel matrix is introduced as $K_{i,j}^* = \phi(x^i)^T \phi(x^j)$. Cristianini and Shawe-Taylor [25] have pointed out that K^* is positive semidefinite for the kernel functions such as linear kernel and Gaussian kernel. It is worth pointing out that problem (1) can be reformulated as the following problem [21]:

$$\begin{aligned} \min \quad & \frac{1}{2} (e^n + \mu)^T (K \circ yy^T)^{-1} (e^n + \mu) \\ \text{s.t.} \quad & \mu \geq \mathbf{0}^n, \\ & y^b \in \{-1, 1\}^{n-l}, \end{aligned} \quad (2)$$

where $K = K^* + (1/2C)I_n$ and $\mu = (\mu_1, \dots, \mu_n)^T$ is the dual variables vector. Note that this problem is a MIQP problem which is generally difficult to solve.

In order to handle the nonconvex objective function of problem (2), we reformulate it into a completely positive programming problem. Note that Bai and Yan [21] have proved that the objective function of problem (2) can be equivalently written as $(1/2)((e^n + \mu) \circ y)^T K^{-1} ((e^n + \mu) \circ y)$. Moreover, let $\delta = e^n + \mu$ and replace $y = 2z - e^n$, where $z \in \{0, 1\}^n$. Then problem (2) can be equivalently reformulated as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} (\delta \circ (2z - e^n))^T K^{-1} (\delta \circ (2z - e^n)) \\ \text{s.t.} \quad & \delta \geq e^n, \\ & z = \left((z^l)^T, (z^u)^T \right)^T, \\ & z^u \in \{0, 1\}^{n-l}. \end{aligned} \quad (3)$$

Moreover, let $u = \begin{pmatrix} \delta \circ z \\ \delta \circ e \end{pmatrix}$, $\tilde{K} = \begin{pmatrix} 4K^{-1} & -2K^{-1} \\ -2K^{-1} & K^{-1} \end{pmatrix}$. It is worth pointing out that $\tilde{K}$ is positive semidefinite [8]. Let $A = \{i \mid y_i = -1, 1 \leq i \leq l\}$ and $B = \{i \mid y_i = 1, 1 \leq i \leq l\}$ denote the two index sets, respectively. It is easy to verify that problem (3) can be equivalently written as

$$\begin{aligned} \min \quad & \frac{1}{2} u^T \tilde{K} u \\ \text{s.t.} \quad & u_i = 0, \quad i \in A, \\ & u_i \geq 1, \quad i \in B, \\ & u_j \geq 1, \quad j = n+1, \dots, 2n, \\ & u_k u_{n+k} - u_k^2 = 0, \quad k = l+1, \dots, n. \end{aligned} \quad (4)$$

Now, problem (4) is a nonconvex quadratic programming problem with mixed binary and continuous variables. Since $y = 2z - e^n$, we have $\delta \circ y = 2\delta \circ z - \delta \circ e^n$. Thus, for a feasible solution u of problem (4), we can get a feasible solution y of problem (2) by the following equation:

$$y_i = \text{sgn}(2\delta \circ z - \delta \circ e^n)_i = \text{sgn}(2u_i - u_{n+i}), \quad i = 1, \dots, n. \quad (5)$$

Following the standard relaxation techniques in [22], we get the equivalent completely positive programming problem as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \tilde{K} \cdot U \\ \text{s.t.} \quad & u_i = 0, \quad i \in A, \\ & u_i \geq 1, \quad i \in B, \\ & u_i \geq 1, \quad i = n+1, \dots, 2n, \\ & U_{i,i} - u_i \geq 0, \quad i = 1, \dots, 2n, \end{aligned}$$

$$U_{j,n+j} - U_{j,j} = 0, \quad j = l+1, \dots, n,$$

$$\begin{pmatrix} 1 & u^T \\ u & U \end{pmatrix} \in \mathcal{C}_{2n+1}^*, \quad (6)$$

where $\mathcal{C}_{2n+1}^*$ is the cone of completely positive matrices; that is, $\mathcal{C}_{2n+1}^* = \text{cl Cone}\{xx^T \in \mathcal{S}_+^{2n+1} \mid x \in \mathbb{R}_+^{2n+1}\}$.

From Theorem 2.6 of [22], we know that the optimal value of problem (6) is equal to the optimal value of problem (2). However, detecting whether a matrix belongs to $\mathcal{C}_{2n+1}^*$ is NP-hard [23]. Thus, $\mathcal{C}_{2n+1}^*$ is not computable. A natural way for deriving a polynomial-time solvable approximation is to replace $\mathcal{C}_{2n+1}^*$ by some commonly used computable cones, such as $\mathcal{S}_+^{2n+1}$ or $\mathcal{S}_+^{2n+1} \cap \mathcal{N}_+^{2n+1}$ [21]. However, $\mathcal{S}_+^{2n+1}$ and $\mathcal{S}_+^{2n+1} \cap \mathcal{N}_+^{2n+1}$ are simple relaxations which lead to some loose lower bounds. Moreover, these conic relaxations cannot get improved to a desired accuracy. Thus, they are not proper for some situations with high accuracy requirements. Therefore, we propose a new approximating way to get an ϵ -optimal solution of the original mixed integer constrained quadratic programming (MIQP) problem in the next section.

3. An ϵ -Optimal Approximation Method

We first introduce the definitions of the cone of nonnegative quadratic forms and its dual cone. Given a nonempty set $\mathcal{F} \subseteq \mathbb{R}^{2n+1}$, Sturm and Zhang [26] defined the cone of nonnegative quadratic forms over $\mathcal{F}$ as $\mathcal{C}_{\mathcal{F}} = \{M \in \mathcal{S}^{2n+1} \mid x^T M x \geq 0 \text{ for all } x \in \mathcal{F}\}$. Its dual cone is $\mathcal{C}_{\mathcal{F}}^* = \text{cl Cone}\{xx^T \in \mathcal{S}^{2n+1} \mid x \in \mathcal{F}\}$. From the definition, it is easy to see that if $\mathbb{R}_+^{2n+1} \subset \mathcal{F}$ then $\mathcal{C}_{2n+1}^* \subset \mathcal{C}_{\mathcal{F}}^* \subset \mathcal{S}_+^{2n+1} \subset \mathcal{C}_{\mathcal{F}} \subset \mathcal{C}_{2n+1}$. Therefore, if $\mathcal{F}$ is a tight approximation of $\mathbb{R}_+^{2n+1}$, then $\mathcal{C}_{\mathcal{F}}^*$ is a good approximation of $\mathcal{C}_{2n+1}^*$. In particular $\mathcal{F}$ can be a union of several computable cones. Let $\mathcal{F}_1, \dots, \mathcal{F}_s \subseteq \mathbb{R}^{2n+1}$ be s nonempty cones; the summation of these sets is defined by

$$\sum_{j=1}^s \mathcal{F}_j = \left\{ \sum_{j=1}^s x_j \in \mathbb{R}^{2n+1} \mid x_j \in \mathcal{F}_j, i = 1, \dots, s \right\}. \quad (7)$$

Note that Corollary 16.4.2 in [27] and Lemma 2.4 in [28] indicate that the summation of cone of nonnegative forms over each cone is an approximation of $\mathcal{C}^*$ when $\mathbb{R}_+^{2n+1} \subset (\mathcal{F} = \cup_{i=1}^s \mathcal{F}_j)$.

In the rest of this paper, we will set each $\mathcal{F}_j$ to be a nontrivial second-order cone $\mathcal{F}_{\text{SOC}} = \{x \in \mathbb{R}^{2n+1} \mid \sqrt{x^T M x} \leq f^T x\}$, where $M \in \mathcal{S}_{++}^{2n+1}$ and $f \in \mathbb{R}^{2n+1}$. Here, “nontriviality” means the second-order cone contains at least one point other than the origin. The author and his coauthors have proved that $\mathcal{C}_{\mathcal{F}_{\text{SOC}}}^*$ has a linear matrix representation (LMI), thus computable. This result will be shown in the next theorem.

Theorem 1 (Theorem 2.13 in [28]). *Let $\mathcal{F}_{\text{SOC}} = \{x \in \mathbb{R}^{2n+1} \mid \sqrt{x^T M x} \leq f^T x\}$ be a nontrivial second-order cone with*

$M \in \mathcal{S}_{++}^{2n+1}$, $f \in \mathbb{R}^{2n+1}$, and $B = ff^T$. Then, $X \in \mathcal{C}_{\mathcal{F}_{\text{SOC}}}^*$ if and only if X satisfies that $(M - B) \cdot X \leq 0$ and $X \in \mathcal{S}_{+}^{2n+1}$.

Then the next issue is to cover $\mathbb{R}_+^{2n+1}$ by a union of second-order cones. Notice that $\mathbb{R}_+^{2n+1} = \text{Cone}(\Delta)$, where $\Delta = \{x \in \mathbb{R}_+^{2n+1} \mid (e^{2n+1})^T x = 1\}$ is the standard simplex. Let $\mathcal{P} = \{P_1, \dots, P_s\}$ be a set of polyhedrons in $\mathbb{R}^{2n+1}$, where $\text{int}(P_i) \cap \text{int}(P_j) = \emptyset$ for $i \neq j$. Assume $\mathcal{P}$ is a partition of Δ ; then we have $\Delta = P_1 \cup P_2 \cdots \cup P_s$. If we can find a corresponding second-order cone $\mathcal{F}_{\text{SOC}}^j = \{x \in \mathbb{R}^{2n+1} \mid \sqrt{x^T M_j x} \leq f_j^T x\}$ such that $P_j \subseteq \mathcal{F}_{\text{SOC}}^j$ for $j = 1, \dots, s$, then $\mathbb{R}_+^{2n+1} \subseteq \bigcup_{j=1}^s \mathcal{F}_{\text{SOC}}^j$ leads to $\mathcal{C}_{2n+1}^* \subseteq \sum_{j=1}^s \mathcal{C}_{\mathcal{F}_{\text{SOC}}^j}^*$. Thus, we can replace the uncomputable cone $\mathcal{C}_{2n+1}^*$ by the computable cone $\sum_{j=1}^s \mathcal{C}_{\mathcal{F}_{\text{SOC}}^j}^* \subset \mathcal{S}_{+}^{2n+1}$ in problem (6) to generate a better lower bound.

Now, we show how to generate a proper second-order cone to cover a polyhedron. Suppose $v_1^j, \dots, v_q^j$ are the vertices of P_j . Since $\text{int}(P_j) \neq \emptyset$, $\text{rank}([v_1^j, \dots, v_q^j]) = 2n + 1$, where $[v_1^j, \dots, v_q^j]$ is the matrix formed by vectors $v_1^j, \dots, v_q^j$. Note that $y = e^{2n+1}$ is the unique solution of the system of equations: $(v_i^j)^T y = 1$, $i = 1, \dots, q$. Let $f = e^{2n+1}$; then we have $P_j \subseteq \mathcal{F}_{\text{SOC}}^j = \{x \in \mathbb{R}^{2n+1} \mid \sqrt{x^T M_j x} \leq (e^{2n+1})^T x\}$ if and only if $v_i^j \in \mathcal{C}_{\text{SOC}} = \{x \in \mathbb{R}^{2n+1} \mid \sqrt{x^T M_j x} \leq 1, (e^{n+1})^T x = 1\}$ for $i = 1, \dots, q$. Therefore, we only need to find proper $M_j \in \mathcal{S}_{++}^{n+1}$ for $\mathcal{F}_{\text{SOC}}^j$ by solving the following convex programming problem [28]:

$$\begin{aligned} \min \quad & \log \det(M_j^{-1}) \\ \text{s.t.} \quad & M_j \cdot \left[v_i^j (v_i^j)^T \right] \leq 1, \\ & i = 1, \dots, q, \quad M_j \in \mathcal{S}_{++}^{2n+1}. \end{aligned} \quad (8)$$

It is worth pointing out that if M_j^* is the optimal solution of problem (8), then $\{x \in \mathbb{R}^{2n+1} \mid x^T M_j^* x \leq 1\}$ is the smallest ellipsoid that centers at the origin and covers P_j . Hence, $\mathcal{F}_{\text{SOC}}^j$ is a tight cover of P_j .

Now suppose there is a fixed polyhedron partition $\mathcal{P} = \{P_1, \dots, P_s\}$ of Δ and the corresponding second-order cone $\mathcal{F}_{\text{SOC}}^j = \{x \in \mathbb{R}^{2n+1} \mid \sqrt{x^T M_j x} \leq (e^{2n+1})^T x\}$ covers P_j for $j = 1, \dots, s$. Then, $\mathbb{R}_+^{2n+1} \subseteq \bigcup_{j=1}^s \mathcal{F}_{\text{SOC}}^j$ and $\mathcal{C}_{2n+1}^* \subseteq \sum_{j=1}^s \mathcal{C}_{\mathcal{F}_{\text{SOC}}^j}^*$. Let $B = e^{2n+1}(e^{2n+1})^T$; then we have the following computable approximation:

$$\begin{aligned} \min \quad & \frac{1}{2} \bar{K} \cdot U \\ \text{s.t.} \quad & u_i = 0, \quad i \in A, \\ & u_i \geq 1, \quad i \in B, \\ & u_i \geq 1, \quad i = n + 1, \dots, 2n, \end{aligned}$$

$$\begin{aligned} u_i &= U_{i,i}, \quad i = 1, \dots, 2n, \\ U_{k,n+k} - U_{k,k} &= 0, \quad k = l + 1, \dots, n, \\ \begin{pmatrix} 1 & u^T \\ u & U \end{pmatrix} &= \mathcal{T}_1 + \mathcal{T}_2 + \cdots + \mathcal{T}_s, \\ \mathcal{T}_j &\in \mathcal{S}_{+}^{2n+1}, \quad j = 1, \dots, s, \\ (M_j - B) \cdot \mathcal{T}_j &\leq 0, \quad j = 1, \dots, s. \end{aligned} \quad (9)$$

In order to measure the accuracy of the approximation, we define the maximum diameter of a polyhedron partition $\mathcal{P}$ and a σ -neighborhood of a domain $\mathcal{D}$.

Definition 2. For a polyhedron partition $\mathcal{P} = \{P_1, \dots, P_s\}$, the maximum diameter of $\mathcal{P}$ is defined as

$$\delta(\mathcal{P}) = \max_{j \in \{1, \dots, s\}} \max_{x_j, y_j \in P_j} \|x_j - y_j\|. \quad (10)$$

Definition 3 (Definition 3 of [29]). For a set $\mathcal{D} \subseteq \mathbb{R}^n$ and $\sigma > 0$, the σ -neighborhood of $\mathcal{D}$ is defined as

$$\mathcal{N}_{\sigma, \mathcal{D}} = \{x \in \mathbb{R}^n \mid \exists y \in \mathcal{D}, \text{ s.t. } \|x - y\|_{\infty} \leq \sigma\}, \quad (11)$$

where $\|\cdot\|_{\infty}$ denotes the infinity norm.

Then the next theorem shows that the lower bounds obtained from this approximation can converge to the optimal value as the number of polyhedrons increases.

Theorem 4. Let V^* be the optimal objective value of problem (2) and let L_s , $s = 1, 2, \dots$, be the lower bounds sequentially returned by solving corresponding problem (9) with s polyhedrons in the partition of Δ . For any $\epsilon > 0$, if $\delta(\mathcal{P})$ converges to 0 when the number of polyhedrons in $\mathcal{P}$ increases, then there exists $N_{\epsilon} \in \mathbb{N}_+$ such that $|V^* - \max_{1 \leq j \leq N} L_s| < \epsilon$ for any $N \geq N_{\epsilon}$.

Proof. Since $\bar{K}$ is positive semidefinite, problem (4) is bounded below. Thus, the optimal value of problem (2) and (6) is finite. Let $\text{opt} : \mathbb{R}_+ \rightarrow \mathbb{R}$ be a function, where $\text{opt}(\sigma)$ is the optimal value of problem (6) by replacing $\mathcal{C}_{2n+1}^*$ with $\mathcal{C}_{\mathcal{N}_{\sigma, \Delta}}^*$. By definition, for $0 < \sigma_1 < \sigma_2$, we have $\mathcal{C}_{2n+1}^* \subset \mathcal{C}_{\mathcal{N}_{\sigma_1, \Delta}}^* \subset \mathcal{C}_{\mathcal{N}_{\sigma_2, \Delta}}^*$. Thus opt is monotonically increasing as σ tends to zero. Note that when $\sigma = 0$, $\mathcal{C}_{\mathcal{N}_{\sigma, \Delta}}^* = \mathcal{C}_{2n+1}^*$, thus $V^* = \text{opt}(0)$ is an upper bound of function opt . It is easy to verify that opt is also a continuous function. Thus, for any $\epsilon > 0$; there exists $\sigma_{\epsilon} > 0$, when $0 \leq \sigma \leq \sigma_{\epsilon}$, we have $|V^* - \text{opt}(\sigma)| < \epsilon$. Because $\delta(\mathcal{P})$ converges to 0, there exists N_{ϵ} such that $\delta(\mathcal{P}) = \sigma < \sigma_{\epsilon}$ when $N \geq N_{\epsilon}$. Then, for each simplex $P_j \in \mathcal{P}$ and its corresponding second-order cone $\mathcal{F}_{\text{SOC}}^j$, $\mathcal{F}_{\text{SOC}}^j \subseteq \text{Cone}(\mathcal{N}_{\sigma, P_j})$. Thus, for each simplex P_j , the corresponding second-order cone $\mathcal{F}_{\text{SOC}}^j \subseteq \text{Cone}(\mathcal{N}_{\sigma, \Delta})$ and we have $\mathcal{C}_{\mathcal{F}_{\text{SOC}}^j}^* \subseteq \mathcal{C}_{\mathcal{N}_{\sigma, \Delta}}^*$. Therefore, $\sum_{j=1}^{N_{\epsilon}} \mathcal{C}_{\mathcal{F}_{\text{SOC}}^j}^* \subseteq \mathcal{C}_{\mathcal{N}_{\sigma, \Delta}}^*$. Let $D_{N_{\epsilon}}$ and $L_{N_{\epsilon}}$ denote the feasible domain and optimal value

of problem (9), where N_e simplices are used in the partition. Then, we have $\mathcal{C}_{2n+1}^* \subset D_{N_e} \subset F_o$. Thus, $\text{opt}(\sigma) \leq L_{N_e} \leq V^*$, $|V^* - \max_{1 \leq s \leq N} L_s| < \epsilon$ for any $N \geq N_e$. $\square$

In summary, we have shown that our new approximation method indeed can get an ϵ -optimal solution of problem (6) in finite iterations. However, the exact number required to achieve an ϵ -optimal solution depends on the instance and the strategy for the partition of Δ .

4. An Adaptive Approximation Algorithm

Theorem 4 guarantees that an ϵ -optimal solution can be obtained in finite iterations. However, it may take an enormous amount of time to partite the standard simplex Δ finely enough to meet an extremely high accuracy requirement. Therefore, consider the balance between computing burden and accuracy; we design an adaptive approximation algorithm by partitioning the underlying Δ into s (> 1) special polyhedrons at one time to get a good approximated solution in this section.

First, we solve problem (9) for $s = 1$ to find an optimal solution (u^*, U^*) . Let $G = I_{2n+1} - B$ and $\mathcal{T}^* = \begin{pmatrix} 1 & (u^*)^T \\ u^* & U^* \end{pmatrix}$; then $G \cdot \mathcal{T}^* \leq 0$. According to the decomposition scheme in Lemma 2.2 in [30], there exists a rank-one decomposition for $\mathcal{T}^*$ such that $\mathcal{T}^* = \sum_{i=1}^r t_i t_i^T$ and $t_i^T G t_i \leq 0$, $i = 1, 2, \dots, r$. Let $\mathcal{J} = \{1, \dots, r\}$ be the index set for the decomposition. If $t_i \in \mathbb{R}_+^{2n+1}$ for each $i \in \mathcal{J}$, then $\mathcal{T}^*$ is completely positive and feasible to problem (6). Hence the optimal value of problem (9) is also the optimal value of problem (6). Otherwise, there exists at least one $i \in \mathcal{J}$ such that $t_i \notin \mathbb{R}_+^{2n+1}$. Denote $\mathcal{J}_p = \{i \mid i \in \mathcal{J}, t_i \notin \mathbb{R}_+^{2n+1}\} \subseteq \mathcal{J}$ as an index set of the decomposed vector t_i 's which violate the nonnegative constraints. Let t_{ij} denote the j th element in the vector t_i . For any $i \in \mathcal{J}_p$, let $\bar{t}_i \in \mathbb{R}^{2n+1}$ be a new vector such that $\bar{t}_{ij} = 0$ for $t_{ij} \geq 0$ and $\bar{t}_{ij} = t_{ij}$ for $t_{ij} < 0$, $j = 1, \dots, 2n + 1$. Then we can define the sensitive index as follows.

Definition 5. Let $\mathcal{T}^* = \sum_{i=1}^r t_i t_i^T$ be a decomposition of an optimal solution of problem (9) for $s = 1$. Define any $\bar{t}_{i^*}$ to be a sensitive solution, where $i^* = \arg\max_{i \in \mathcal{J}_p} \|\bar{t}_i\|_\infty$. Pick the smallest number of indexes i^* among all sensitive solutions and suppose $k^* \in \{1, \dots, 2n + 1\}$ is the smallest index such that $t_{i^*k^*}$ is the smallest among all the components t_{ij} in that sensitive solution t_{i^*} . then define k^* to be the sensitive index.

Actually, the sensitive index is the one in which the decomposed vector most “violates” the nonnegative constraint. Based on this information, we can shrink the feasible domain of problem (9) and cut off the corresponding optimal solution $\mathcal{T}^*$ which is infeasible to problem (6).

For $i = 1, \dots, 2n + 1$, let $e_i \in \mathbb{R}^{2n+1}$ denote the vector with all elements being 0 except the i th element being 1. Obviously, e_i is a vertex of the standard simplex Δ . Now suppose k^* is the sensitive index; the simplex Δ can be partitioned into s small polyhedrons based on this index as follows. Let $V_0 = \{e_i \mid 1 \leq i \leq 2n + 1, i \neq k^*\}$ be the set of all vertices of Δ except

the vertex e_{k^*} and let $\{\beta_1, \beta_2, \dots, \beta_s\}$ be a series of real values such that $0 < \beta_1 < \beta_2 < \dots < \beta_{s-1} < \beta_s = 1$. Then we can generate some new sets of points in Δ by convexly combining the vertices in set V_0 and the vertex e_{k^*} according to different weight values:

$$V_j = \{(1 - \beta_j)e_i + \beta_j e_{k^*} \mid e_i \in V_0\}, \quad j = 1, \dots, s. \quad (12)$$

Note that, for $j = 0, \dots, s - 1$, V_j has $2n$ linearly independent points while V_s contains only one point e_{k^*} . Let P_j be the polyhedron generated by the convex hull of the points in the union set $V_{j-1} \cup V_j$ for $j = 1, \dots, s$. Let $[V_{j-1} \cup V_j]$ be the matrix formed by the points in V_{j-1} and V_j as the column vectors. It is easy to verify that $\text{rank}([V_{j-1} \cup V_j]) = 2n + 1$. Hence, there is only one unique solution $y = e^{2n+1}$ of the system of equations: $v_i^T y = 1$ for $v_i \in V_{j-1} \cup V_j$. Thus, for each polyhedron P_j , we only need to solve problem (P_e) to find the corresponding second-order cone $\mathcal{F}_{\text{SOC}}^j = \{x \in \mathbb{R}^{2n+1} \mid \sqrt{x^T M_j x} \leq (e^{2n+1})^T x\}$ such that $P_j \subseteq \mathcal{F}_{\text{SOC}}^j$.

Remark 6. Since P_j is highly symmetric with respect to the sensitive index and vertices in V_j are rotated, the positive definite matrix M_j in the second-order cone $\mathcal{F}_{\text{SOC}}^j$ which covers P_j has a simple structure. Thus, problem (P_e) can be simplified and quickly solved.

Note that practical users can adjust the number of polyhedrons in the partition to get solutions with different accuracies as they wish. Besides, redundant constraints can be added to improve the performance of a relaxed problem [31]. Note that $\mathcal{C}_{2n+1}^* \subset \mathcal{N}_+^{2n+1}$ and $\mathcal{C}_{\mathcal{P}_j}^* \subset \mathcal{N}_+^{2n+1}$ for $j = 1, \dots, k$. Therefore, we can add the redundant constraints $\mathcal{T}_i \in \mathcal{N}_+^{2n+1}$, $i = 1, \dots, s$ to further improve problem (9).

5. Numerical Tests

In this section, we compare our algorithm (AS³VMP) to some existing solvable conic relaxations on the 2-norm soft margin S³VM model, such as TSDP in [18] and SSDP and TDNNP in [21]. Besides, we also add the transductive support vector machine (TSVM) [9] which is a classical local search algorithm to the comparison. Several artificial and real-world benchmark datasets are used to test the performances of these methods.

To obtain the artificial datasets for the computational experiment, we first generate different quadratic surfaces $(1/2)x^T W x + b^T x = 0$ with various matrices W and vectors b in different dimensions. Here, the eigenvalues of each matrix W are randomly selected from the interval $[-10, 10]$. Then, for each case, we randomly generate some points on one side of the surface (labeled as Class C_1) and some points on the other side (labeled as Class C_2). Moreover, in order to prevent some points too far away from the separating surface, all the points are generated in the ball $x^T x \leq R$, where R is a positive value.

The real-world datasets come from the semisupervised learning (SSL) benchmark datasets [32] and UC Irvine

Machine Learning Repository (UCI) datasets [33]. We use four datasets in our computational experiment, two from SSL (Digit 1 and USPS) and two from UCI (Iono and Sonar). Particularly, since our main aim is to compare our method with the state-of-the-art methods in [21], we follow the same way to restrict the total number of points as 70 in all real-world datasets.

For each dataset, we conduct 100 independent trials by randomly picking the labeled points. The information of these datasets is provided in Table 1.

The kernel matrices K^* are constructed by the Gaussian kernel $k(x^i, x^j) = \exp(-\|x^i - x^j\|^2 / 2\sigma^2)$ throughout the test, where the parameter σ is chosen as the median of the pairwise distances. Besides, the optimal penalty parameter C is tuned by the grid method: $\log_2 C \in \{-10, -9, \dots, 9, 10\}$. We set $\bar{s} = 10$ and the value series $\{\beta_1, \dots, \beta_{\bar{s}}\} = \{0.001, 0.002, 0.004, 0.008, 0.016, 0.032, 0.064, 0.12, 0.24, 0.48\}$. All the tests are carried out by Matlab 7.9.0 on a computer equipped with Intel Core i5 CPU 3.3 Ghz and 4 G memory. Moreover, the cvx [34] and SeDuMi 1.3 [35] solvers are incorporated to solve those problems.

The performance of these methods is measured by their classification error rates, standard deviation of the error rates, and computational times. Here the classification error rate is the ratio of the number of misclassified points to the total number of unlabeled points, which is a very important index to reflect the classification accuracy of the method. Moreover, the standard deviation implies the stability of the method while the computational time indicates the efficiency of the method.

Table 2 summarizes the comparison results of four methods on both artificial and real-world datasets. In this table, “rate” denotes the classification error rate as a percentage and “time” denotes the corresponding CPU time in seconds. The number outside the bracket denotes the average value while the number inside the bracket denotes the standard deviation. Note that all of these results are derived from 100 independent trials.

The results from Table 2 clearly show that our algorithm (AS³VMP) achieves promising classification error rates in all instances. AS³VMP provides much better classification accuracy than any other methods in any case. Therefore, our method can be a very effective and powerful tool in solving the 2-norm soft margin S³VM model, especially for the situation with high accuracy requirement. On the other hand, our method takes much longer time than other methods. This long computational time is due to the slow speed of current SDP solver. And it is a kind of sacrifice for the improvement on the classification accuracy.

Moreover, we analyze the impact of the number of second-order cones on the classification accuracy for our proposed method. As we mentioned in Section 3, a finer cover of the completely positive cone would lead to a better classification accuracy. In this test, we take 9 different numbers as 1, 4, 7, 10, 13, 16, 19, 22, and 25. For simplicity, we averagely assign the values for the series $\{\beta_1, \dots, \beta_{\bar{s}}\}$ in each case. For example, if $\bar{s} = 4$, then the series is $\{0.2, 0.4, 0.6, 0.8\}$. Besides,

TABLE 1: Information of tested datasets.

Dataset	Number of features	Number of total points	Number of labeled points
Art 1	3	60	10
Art 2	10	60	10
Art 3	20	60	10
Art 4	30	60	10
Digit 1	241	70	10
USPS	241	70	10
Sonar	60	70	20
Iono	34	70	20

we also check the effect of the level of data uncertainty on the classification accuracy for different methods. Note that we take 6 different levels of data uncertainty (5%, 10%, 15%, 20%, 25%, and 30% of data points are randomly picked as the labeled ones). The numerical results for these two tests on the artificial dataset (Artificial 3) are summarized in Figure 1.

As we expect, we achieve a better classification accuracy by using more second-order cones. Moreover, as the number of second-order cones increases, the classification error rate decreases rapidly at the beginning and slowly at the end. Thus, the marginal contribution of the second-order cones decreases as the total number increases. Similarly, the classification accuracy increases as the level of data uncertainty decreases. And our proposed method beats other methods in all levels of data uncertainty. Therefore, our method has a very good and robust performance with data uncertainty.

6. Conclusion

In this paper, we have provided a new conic approach to the 2-norm soft margin S³VM model. This new method achieves a better approximation and leads to a more accurate classification than the classical local search method and other known conic relaxations in the literature. Moreover, in order to improve the efficiency of the method, an adaptive scheme is adopted. Eight datasets, including both artificial and real-world ones, have been used in the numerical experiments to test the performances of four different conic relaxations. The results show that our proposed approach produces a much smaller classification error rate than other methods in all instances. This verifies that our method is quite effective and indicates a big potential for some real-life applications. Besides, our approach provides a novel angle to study the conic relaxation and sheds some light on the future research of approximation for S³VM.

Achilles’ hell of this method is the efficiency of solving the SDP relaxations. Thus, it is not proper for solving some large-sized problems. However, the computational time can be significantly shortened as the efficiency of the SDP solver gets improved. Note that some new techniques, such as the alternating direction method of multipliers (ADMM), have been proved to be very efficient in solving the SDP problems. Therefore, our future research can consider incorporating these techniques in our scheme.

TABLE 2: Performance of different methods on artificial and real-world datasets.

Dataset	TSVM		TSDP		SSDP		TDNNP		AS ³ VMP	
	Rate	Time	Rate	Time	Rate	Time	Rate	Time	Rate	Time
Art 1	4.01 (1.65)	6.98 (1.29)	6.31 (1.25)	3.30 (1.04)	1.04 (0.62)	15.4 (3.86)	9.41 (1.03)	5.56 (0.73)	22.7 (2.04)	89.6 (5.42)
Art 2	6.39 (3.21)	10.16 (3.47)	8.99 (3.53)	6.12 (2.71)	3.03 (1.43)	171 (4.28)	9.93 (1.21)	5.84 (0.81)	23.6 (1.85)	86.1 (5.21)
Art 3	7.83 (3.67)	12.72 (3.56)	11.12 (3.41)	7.61 (2.72)	3.55 (1.51)	19.3 (4.41)	9.77 (1.16)	6.12 (0.74)	24.8 (2.33)	87.5 (6.43)
Art 4	11.62 (4.39)	15.17 (4.10)	13.64 (3.85)	9.94 (2.74)	5.60 (1.48)	23.6 (4.91)	9.15 (1.13)	6.07 (0.78)	27.3 (2.40)	90.5 (6.87)
Digit 1	23.42 (8.82)	28.85 (7.21)	27.84 (7.45)	25.27 (7.21)	16.93 (5.52)	25.4 (5.37)	12.8 (1.40)	8.01 (0.95)	33.6 (2.87)	119.7 (7.01)
USPS	16.27 (6.83)	20.04 (6.33)	18.73 (6.20)	16.11 (5.48)	10.71 (3.91)	271 (5.31)	13.2 (1.37)	7.30 (1.13)	34.2 (3.04)	122.5 (7.27)
Sonar	19.34 (6.07)	27.44 (5.67)	25.78 (5.51)	18.45 (4.73)	12.42 (4.03)	25.8 (5.49)	12.4 (1.53)	7.62 (1.07)	33.1 (2.77)	116.1 (7.48)
Ionos	12.35 (5.94)	15.73 (6.62)	15.34 (5.31)	13.03 (4.17)	7.22 (3.12)	26.3 (5.91)	12.7 (1.27)	7.28 (1.18)	34.7 (2.94)	113.2 (7.36)

FIGURE 1: Classification error rate versus number of second-order cones and level of data uncertainty on Artificial 3.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

Tian's research has been supported by the National Natural Science Foundation of China Grants 11401485 and 71331004.

References

- [1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [2] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang, "Accurate multisteps traffic flow prediction based on SVM," *Mathematical Problems in Engineering*, vol. 2013, Article ID 418303, 8 pages, 2013.
- [3] X. Wang, J. Wen, S. Alam, X. Gao, Z. Jiang, and J. Zeng, "Sales growth rate forecasting using improved PSO and SVM," *Mathematical Problems in Engineering*, vol. 2014, Article ID 437898, 13 pages, 2014.
- [4] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, 2001.
- [5] H. Fu and Q. Xu, "Locating impact on structural plate using principal component analysis and support vector machines," *Mathematical Problems in Engineering*, vol. 2013, Article ID 352149, 8 pages, 2013.
- [6] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences Technical Report 1530, University of Wisconsin, Madison, Wis, USA, 2006.
- [7] O. Chapelle, V. Sindhwani, and S. S. Keerthi, "Optimization techniques for semi-supervised support vector machines," *Journal of Machine Learning Research*, vol. 9, no. 1, pp. 203–233, 2008.
- [8] Y. Q. Bai, Y. Chen, and B. L. Niu, "SDP relaxation for semi-supervised support vector machine," *Pacific Journal of Optimization*, vol. 8, no. 1, pp. 3–14, 2012.
- [9] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of the 16th International Conference on Machine Learning (ICML '99)*, pp. 200–209, Bled, Slovenia, June 1999.
- [10] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*, pp. 19–26, San Francisco, Calif, USA, 2001.
- [11] C. Lee, S. Wang, F. Jiao, D. Schuurmans, and R. Greiner, "Learning to model spatial dependency: semi-supervised discriminative random fields," in *Advance of Neural Information Processing System 19: Proceedings of the 2006 Conference*, The MIT Press, Cambridge, Mass, USA, 2006.
- [12] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proceedings of the Conference on Advances in Neural Information Processing Systems II*, pp. 368–374, MIT Press, 1999.
- [13] B. Zhao, F. Wang, and C. Zhang, "CutS3VM: a fast semi-supervised SVM algorithm," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 830–838, Las Vegas, Nev, USA, August 2008.
- [14] F. Gieseke, A. Airola, T. Pahikkala, and O. Kramer, "Fast and simple gradient-based optimization for semi-supervised support vector machines," *Neurocomputing*, vol. 123, no. 1, pp. 23–32, 2014.
- [15] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1687–1712, 2006.
- [16] I. S. Reddy, S. Shevade, and M. N. Murty, "A fast quasi-Newton method for semi-supervised SVM," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2305–2313, 2011.

- [17] V. Sindhwani, S. Keerthi, and O. Chapelle, "Deterministic annealing for semi-supervised kernel machines," in *Proceedings of the ACM 23rd International Conference of Machine Learning (ICML '06)*, pp. 841–848, Pittsburgh, Pa, USA, 2006.
- [18] T. de Bie and N. Cristianini, "Semi-supervised learning using semi-definite programming," in *Semi-Supervised Learning*, O. Chapelle, B. Schoikopf, and A. Zien, Eds., MIT Press, Cambridge, Mass, USA, 2006.
- [19] X. Zhu and A. Goldberg, *Introduction to Semi-Supervised Learning*, Morgan & Claypool Publishers, New York, NY, USA, 2009.
- [20] Y. Q. Bai, B. L. Niu, and Y. Chen, "New SDP models for protein homology detection with semi-supervised SVM," *Optimization*, vol. 62, no. 4, pp. 561–572, 2013.
- [21] Y. Bai and X. Yan, "Conic relaxations for semi-supervised support vector machines," *Journal of Optimization Theory and Applications*, pp. 1–15, 2015.
- [22] S. Burer, "On the copositive representation of binary and continuous nonconvex quadratic programs," *Mathematical Programming*, vol. 120, no. 2, pp. 479–495, 2009.
- [23] K. G. Murty and S. N. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Mathematical Programming*, vol. 39, no. 2, pp. 117–129, 1987.
- [24] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, Cambridge University Press, Cambridge, UK, 2002.
- [25] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.
- [26] J. F. Sturm and S. Zhang, "On cones of nonnegative quadratic functions," *Mathematics of Operations Research*, vol. 28, no. 2, pp. 246–267, 2003.
- [27] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, USA, 1970.
- [28] Y. Tian, S.-C. Fang, Z. Deng, and W. Xing, "Computable representation of the cone of nonnegative quadratic forms over a general second-order cone and its application to completely positive programming," *Journal of Industrial and Management Optimization*, vol. 9, no. 3, pp. 703–721, 2013.
- [29] Z. Deng, S.-C. Fang, Q. Jin, and W. Xing, "Detecting copositivity of a symmetric matrix by an adaptive ellipsoid-based approximation scheme," *European Journal of Operational Research*, vol. 229, no. 1, pp. 21–28, 2013.
- [30] Y. Ye and S. Zhang, "New results on quadratic minimization," *SIAM Journal on Optimization*, vol. 14, no. 1, pp. 245–267, 2003.
- [31] K. M. Anstreicher, "Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming," *Journal of Global Optimization*, vol. 43, no. 2-3, pp. 471–484, 2009.
- [32] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, Cambridge, UK, 2006.
- [33] K. Bache and M. Lichman, *UCI Machine Learning Repository*, School of Information and Computer Science, University of California, 2013, <http://archive.ics.uci.edu/ml>.
- [34] M. Grant and S. Boyd, "CVX: matlab Software for Disciplined Programming," 2010, version 1.2, <http://cvxr.com/cvx/>.
- [35] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods & Software*, vol. 11, no. 1, pp. 625–653, 1999.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>