Journal of the Operational Research Society (2016) 67, 1001–1011

www.palgrave-journals.com/jors/

A kernel-free quadratic surface support vector machine for semi-supervised learning

Xin Yan[1], Yanqin Bai[1], Shu-Cherng Fang[2] and Jian Luo[3]*

[1]Shanghai University, Shanghai, China; [2]North Carolina State University, Raleigh, USA; and [3]Dongbei University of Finance and Economics, Dalian, China

In this paper, we propose a kernel-free semi-supervised quadratic surface support vector machine model for binary classification. The model is formulated as a mixed-integer programming problem, which is equivalent to a non-convex optimization problem with absolute-value constraints. Using the relaxation techniques, we derive a semi-definite programming problem for semi-supervised learning. By solving this problem, the proposed model is tested on some artificial and public benchmark data sets. Preliminary computational results indicate that the proposed method outperforms some existing well-known methods for solving semi-supervised support vector machine with a Gaussian kernel in terms of classification accuracy.

Journal of the Operational Research Society (2016) **67**(7), 1001–1011. doi:10.1057/jors.2015.89

Published online 2 March 2016

Keywords: semi-supervised support vector machine; quadratic surface support vector machine; semi-supervised learning; kernel-free; semi-definite relaxation

1. Introduction

Classification is an important task for extracting information from data with real-life applications (Li and Hand, 2002; Schebesch and Stecking, 2005; Boylan *et al*, 2008). Support vector machine (SVM), using the maximal margin approach to deal with binary classification, was provided in details by Vapnik in 1995 (Cortes and Vapnik, 1995). SVM aims to find an optimal hyperplane that separates the labelled data points into two classes. It was the first touch for many optimization researchers on machine learning. SVMs have been successfully applied in many fields such as face detection (Osuna *et al*, 1997), time series forecasting (Hansen *et al*, 2006), financial distress prediction (Sun *et al*, 2014) and credit scoring (Baesens *et al*, 2003; Schebesch and Stecking, 2005; Lessmann *et al*, 2015). In these applications, we often face the issue of lacking labelled data since it is labour-intensive and time-consuming to label all data points, while a large number of unlabelled data points can be obtained relatively easily. Moreover, the unlabelled points may reduce the bias of the classifier when compared with the case with only labelled points. Take one application of semi-supervised learning in credit scoring as an example, instead of building a model on only the accepted applicants (labelled data), reject inference is used to infer the status of applicants who have been rejected (unlabelled data) (Maldonado and Paredes, 2010; Kennedy *et al*, 2012). Consequently, transductive SVM (Vapnik and Sterin, 1977; Joachims, 1999), was first proposed by treating the unknown data labels as additional optimization variables in SVM so that the labels of unknown data can be derived directly, and implemented in text classification. For semi-supervised learning, semi-supervised support vector machine (S^3VM) (Bennett and Demiriz, 1999; Chapelle *et al*, 2008) was further proposed to learn an inductive rule from the partially labelled data set, which is to output a prediction function that is defined over the entire input space. Since transductive SVM can also learn an inductive rule, we refer to this approach as S^3VM in this paper.

The main idea of S^3VM is to maximize the margin between two classes in the presence of unlabelled data, by keeping the boundary traversing through low-density regions while respecting labels in the input space (Chapelle *et al*, 2008). This leads to a non-convex optimization problem, which causes computational difficulty. Recently, several optimization methods have been developed for solving non-convex optimization problems associated with S^3VM, such as S^3VM^{light} (Joachims, 1999), branch-and-bound algorithm (Bennett and Demiriz, 1999), semi-definite relaxation (De Bie and Cristianini, 2004; Xu and Schuurmans, 2005; Valizadegan and Jin, 2006; Xu *et al*, 2008; Bai *et al*, 2012, 2013; Bai and Yan, 2015), deterministic annealing (Sindhwani *et al*, 2006), convex–concave procedures (CCCP) (Collobert *et al*, 2006), low-density separation (LDS) (Chapelle and Zien, 2005) and cutting plane S^3VM (CutS3VM) (Zhao *et al*, 2008). Among these methods, semi-definite relaxation is in general

**Correspondence: Jian Luo, School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian, 116025, China.*
E-mail: luojian546@hotmail.com

efficient for solving S^3VM. It may produce some approximate global solutions from any initial solution.

For the non-linearly separable data sets, a kernel function (Deng *et al*, 2012) is commonly used to map each training data point from the original space to a higher dimensional space, where a hyperplane is then sought to separate all mapped data points into two classes. There is no universal rule to automatically choose a suitable kernel function for a given data set and the parameters in the kernel function influence greatly the performance of S^3VM. Moreover, for the effective usage of a kernel function, people often solve the dual problem of S^3VM, which requires the computation of the inverse of a kernel matrix (Valizadegan and Jin, 2006; Xu *et al*, 2008; Bai *et al*, 2013). Since the kernel matrix in general is only positive semi-definite (Cristianini and Shawe-Taylor, 2000), a small positive scalar needs be added to each diagonal element to deal with the singular case. This leads to an approximate solution to the S^3VM model. When the primal problem of S^3VM is handled, one needs to approximate the mapping function by decomposing its kernel matrix. This again leads to an approximate solution of the S^3VM model.

To overcome potential difficulties caused by using kernel functions, we propose a kernel-free non-linear S^3VM model for semi-supervised classification, based on the quadratic surface support vector machine (QSSVM) suggested in Luo *et al* (2014) for binary classification directly using a quadratic surface for separation. The QSSVM model was originally proposed and tested to provide more accurate classification for labelled data sets than using other SVM models (Issam, 2008; Deng *et al*, 2012). Luo *et al* (2016) further proposed a fuzzy QSSVM model based on Fisher discriminant analysis to deal with data sets containing a large amount of outliers and noise.

The proposed semi-supervised quadratic surface support vector machine (SSQSSVM) model is formulated as a mixed-integer programming problem, which can be converted to a non-convex optimization problem with absolute-value constraints. Since the underlying problem is NP-hard, we relax each of the above two problems into a semi-definite programming problem using some relaxation techniques. By analysing the relationship between the two relaxation problems, we found that they are equivalent if the rank-1 condition is satisfied in both relaxed problems. Compared with the first relaxation, the second one has fewer variables; hence, we adopt the second relaxation for the proposed SSQSSVM model. The SSQSSVM model is tested for performance based on some artificial and public benchmark data sets. Numerical results show that the proposed approach outperforms some existing methods in terms of classification accuracy. This indicates that the proposed SSQSSVM model could be more effective than the kernel-based S^3VM model.

The rest of the paper is arranged as follows. Section 2 provides a review of the QSSVM model. In Section 3 we propose the SSQSSVM model and its equivalent optimization problem. In Section 4, two semi-definite relaxations are proposed, and the relationship between these two problems is analysed. Computational experiments are conducted on some artificial data sets and public benchmark data sets to investigate the performance of the proposed SSQSSVM model in Section 5. Some concluding remarks are provided in Section 6.

In this paper, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ the n-dimensional Euclidean space, $\mathbb{R}^{n\times n}$ the space of $n\times n$-dimensional matrices and $\mathbb{S}^n$ the space of $n\times n$-dimensional symmetric matrices. For a given matrix $\boldsymbol{A}$, $\boldsymbol{A}\succeq\boldsymbol{0}$ means that $\boldsymbol{A}$ is positive semi-definite. For $A,B\in\mathbb{S}^n$, $\boldsymbol{A}\bullet\boldsymbol{B}$ $=\mathrm{trace}(\boldsymbol{A}^T\boldsymbol{B})$ denotes the inner product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Moreover, $\boldsymbol{e}_n$ denotes an n-dimensional vector in which all elements are ones, $\boldsymbol{0}$ the vector in which all elements are zeros, $\boldsymbol{0}_{n\times n}$ the $n\times n$-dimensional matrix in which all elements are zeros, $\boldsymbol{I}_{m\times m}$ the $m\times m$-dimensional identity matrix and opt($\cdot$) the optimal value of the problem. For a non-zero constant a, the sign function is defined by the formula $\mathrm{sign}(a):=a/|a|$, where $|a|$ is the absolute value of a. For a vector $\boldsymbol{b}$, $\mathrm{sign}(\boldsymbol{b}):=(\mathrm{sign}(b_i))_i$.

2. Quadratic surface support vector machine (QSSVM)

In this section, we review the QSSVM model briefly. More details can be found in Luo *et al* (2016, 2014).

Given a training data set of l labelled points $\{(\boldsymbol{x}^i,y_i)\}_{i=1}^l$, where $\boldsymbol{x}^i=(x_1^i,x_2^i,\ \ldots\ ,x_m^i)^T\in\mathbb{R}^m$, $y_i\in\{1,-1\}, i=1,\ldots,l$. The QSSVM model is to find the parameter set $(\boldsymbol{W},\boldsymbol{b},c)$ of a quadratic surface

$$g(x)\triangleq\frac{1}{2}x^T\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}^Tx+c=0,$$

where $\boldsymbol{W}=(w_{ij})_{m\times m}\in\mathbb{S}^m$, $\boldsymbol{b}=(b_i)_m\in\mathbb{R}^m$ and $c\in\mathbb{R}$, which separates the l training points $\{\boldsymbol{x}^i\}_{i=1}^l$ into two classes according to their labels.

In Luo *et al* (2014), by maximizing the sum of relative geometrical margin of each training point with respect to $g(\boldsymbol{x})=0$ and minimizing the classification errors of all training points, the following QSSVM model is proposed:

$$\text{(QSSVM)}\quad \min_{W\in\mathbb{S}^m,b\in\mathbb{R}^m,c\in\mathbb{R},\xi\in\mathbb{R}^l}\quad \sum_{i=1}^{l}\|\boldsymbol{W}\boldsymbol{x}^i+\boldsymbol{b}\|_2^2+C_l\sum_{i=1}^{l}\xi_i$$

$$\text{s.t.}\quad y_i\left(\frac{1}{2}\left(\boldsymbol{x}^i\right)^T\boldsymbol{W}\boldsymbol{x}^i+\boldsymbol{b}^T\boldsymbol{x}^i+c\right)\geqslant 1-\xi_i, i=1,\ \ldots\ ,l,$$

$$\boldsymbol{\xi}=(\xi_1,\xi_2,\ \ldots\ ,\xi_l)^T\geqslant\boldsymbol{0}, \tag{1}$$

where the slack variable ξ_i is the margin of classification error for $\boldsymbol{x}_i$ and $C_l>0$ is a penalty parameter.

The QSSVM model can be further simplified (Luo *et al*, 2014). First, let $\boldsymbol{w}$ be the vector formulated by taking the $(m^2+m)/2$ elements of the upper triangle part of the matrix $\boldsymbol{W}$, that is,

$$\boldsymbol{w}\triangleq(w_{11},w_{12},\ \ldots\ ,w_{1m},w_{22},w_{23},\ \ldots\ ,w_{2m},\ \ldots\ ,w_{mm})^T\in\mathbb{R}^{\frac{m^2+m}{2}}.$$

Then, for $i=1,\ldots,l$, we can construct an $m\times(m^2+m)/2$ matrix $\boldsymbol{M}_i$ for the training point $\boldsymbol{x}^i\in\mathbb{R}^m$ as follows. For the j-th row of $\boldsymbol{M}_i$ in $\mathbb{R}^{(m^2+m)/2}, j = 1, 2, \ldots, m$, if the p-th element of $\boldsymbol{w}$ is w_{jk} or w_{kj} for some $k = 1, 2, \ldots, m$, we assign the p-th element of the j-th row of $\boldsymbol{M}_i$ to be x_k^i. Otherwise, we assign it to be 0. Afterwards, let $\boldsymbol{H}_i \triangleq (\boldsymbol{M}_i, \boldsymbol{I}_{m\times m}) \in \mathbb{R}^{m\times((m^2+m)/2+m)}$, $i = 1, \ldots, l$, $\boldsymbol{G} \triangleq \sum_{i=1}^{l} \boldsymbol{H}_i^T \boldsymbol{H}_i \in \mathbb{S}^{(m^2+3m)/2}$, $\boldsymbol{v} \triangleq (\boldsymbol{w}^T, \boldsymbol{b}^T)^T \in \mathbb{R}^{(m^2+3m)/2}$ and $\mathbf{s}_i \triangleq ((1/2)x_1^i x_1^i, \ldots, x_1^i x_m^i, (1/2) x_2^i x_2^i, \ldots, x_2^i x_m^i, \ldots, (1/2)x_{m-1}^i x_{m-1}^i, x_{m-1}^i x_m^i, (1/2) x_m^i x_m^i, x_1^i, x_2^i, \ldots, x_m^i) \in \mathbb{R}^{(m^2+3m)/2}$.

Then, Problem (1) can be reformulated as

$$\min_{v\in\mathbb{R}^{\frac{m^2+3m}{2}}, c\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^l} \boldsymbol{v}^T \boldsymbol{G} \boldsymbol{v} + C_l \sum_{i=1}^{l} \xi_i$$

$$\textbf{s.t.} \quad y_i\left(\boldsymbol{s}_i^T \boldsymbol{v} + \boldsymbol{c}\right) \geqslant 1 - \xi_i,\ i = 1, \ldots, l,$$

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_l)^T \geqslant \mathbf{0}. \tag{2}$$

Notice that the matrix $\boldsymbol{G}$ is positive semi-definite, thus, Problem (2) is a convex quadratic programming problem with linear constraints.

3. Semi-supervised quadratic surface support vector machine (SSQSSVM)

In this section, we propose the SSQSSVM model for semi-supervised binary classification and convert it to an equivalent non-convex optimization problem.

Given a data set of n data points $\{\boldsymbol{x}^i\}_{i=1}^n$, in which the first l points are labelled and the rest $n-l$ points are unlabelled. Semi-supervised binary classification is to classify unknown data points into two groups based on both of the labelled and unlabelled data points. Let $\boldsymbol{y} = ((\boldsymbol{y}^l)^T, (\boldsymbol{y}^{n-l})^T)^T$ be the label vector, where $\boldsymbol{y}^l = (y_1, y_2, \ldots, y_l)^T \in \{-1, 1\}^l$ is known, and $\boldsymbol{y}^{n-l} = (y_{l+1}, y_{l+2}, \ldots, y_n)^T \in \{-1, 1\}^{n-l}$ is unknown.

Following the logic of the S^3VM model in Chapelle *et al* (2008), an SSQSSVM model is proposed as below, by adding continuous variables $\xi_{l+1}, \ldots, \xi_n$ and binary integer variables $y_{l+1}, y_{l+2}, \ldots, y_n$ to the QSSVM model:

$$(\text{SSQSSVM}) \quad \min_{v\in\mathbb{R}^{\frac{m^2+3m}{2}}, c\in\mathbb{R}, \xi\in\mathbb{R}^n, y^{n-l}\in\mathbb{R}^{n-l}} \boldsymbol{v}^T \boldsymbol{G} \boldsymbol{v} + C_l \sum_{i=1}^{l} \xi_i + C_{n-l} \sum_{j=l+1}^{n} \xi_j$$

$$\boldsymbol{s.t.} \quad y_i\left(\boldsymbol{s}_i^T \boldsymbol{v} + c\right) \geqslant 1 - \xi_i, i = 1, \ldots, l,$$

$$y_j\left(\boldsymbol{s}_j^T \boldsymbol{v} + c\right) \geqslant 1 - \xi_j, j = l+1, \ldots, n,$$

$$\boldsymbol{y}^{n-l} = (y_{l+1}, y_{l+2}, \ldots, y_n)^T \in \{-1, 1\}^{n-l},$$

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)^T \geqslant \mathbf{0}, \tag{3}$$

where C_l and C_{n-l} are the penalty parameters for labelled and unlabelled data sets, respectively, which are used to balance the three expressions in the objective function. By tuning C_l and C_{n-l}, we may derive the best separating surface. Problem (3) is a mixed-integer programming problem, which is NP-hard in general.

One thing we need to pay attention to is that the constraints $\xi_i \geqslant 0, j = l+1, \ldots, n$ can be further restricted. If $\boldsymbol{\xi}$ is feasible to Problem (3), it is easy to see that the inequalities $\xi_j \leqslant 1, j = l+1, \ldots, n$ always hold since the objective of Problem (3) is to minimize the sum of ξ_j for $j = l+1, \ldots, n$. Hence, we can replace the constraints $\xi_j \geqslant 0$, $j = l+1, \ldots, n$ by the constraints $0 \leqslant \xi_j \leqslant 1, j = l+1, \ldots, n$ in Problem (3).

Following the fact that constraints $0 \leqslant \xi_j \leqslant 1$ hold for $j = l+1, \ldots, n$, we may derive an equivalent problem of Problem (3) as the following, similar to that in Chapelle *et al* (2008):

$$(\text{SSQSSVM}') \quad \min_{v\in\mathbb{R}^{\frac{m^2+3m}{2}}, c\in\mathbb{R}, \xi\in\mathbb{R}^n} \boldsymbol{v}^T \boldsymbol{G} \boldsymbol{v} + C_l \sum_{i=1}^{l} \xi_i + C_{n-l} \sum_{j=l+1}^{n} \xi_j$$

$$\textbf{s.t.} \quad y_i\left(\boldsymbol{s}_i^T \boldsymbol{v} + c\right) \geqslant 1 - \xi_i, i = 1, \ldots, l,$$

$$\mid \boldsymbol{s}_j^T \boldsymbol{v} + c \mid \ \geqslant 1 - \xi_j, j = l+1, \ldots, n,$$

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)^T \geqslant \mathbf{0}. \tag{4}$$

This problem is an optimization problem with a convex objective function and linear and non-convex absolute-value constraints.

The proof of the equivalence between Problems (3) and (4) can be easily verified. If $(\boldsymbol{v}^*, c^*, \boldsymbol{\xi}^*, (\boldsymbol{y}^{n-l})^*)$ is optimal to Problem (3), it is easy to verify that $(\boldsymbol{v}^*, c^*, \boldsymbol{\xi}^*)$ is feasible to Problem (4). Hence the optimal value of Problem (3) is not lower than that of Problem (4). On the contrary, if we have an optimal solution $(\bar{\boldsymbol{v}}, \bar{c}, \bar{\boldsymbol{\xi}})$ of Problem (4), $(\bar{\boldsymbol{v}}, \bar{c}, \bar{\boldsymbol{\xi}}, \bar{\boldsymbol{y}}^{n-l})$ is feasible to Problem (3), where $\bar{\boldsymbol{y}}^{n-l} = (\text{sign}(\boldsymbol{s}_{l+1}^T \bar{\boldsymbol{v}} + \bar{c}), \text{sign}(\boldsymbol{s}_{l+2}^T \bar{\boldsymbol{v}} + \bar{c}), \ldots, \text{sign}(\boldsymbol{s}_n^T \bar{\boldsymbol{v}} + \bar{c}))$. Correspondingly, the optimal value of Problem (4) is not lower than that of Problem (3). Therefore, Problems (3) and (4) are equivalent.

4. Semi-definite relaxations of SSQSSVM

Since the semi-definite relaxation is a computationally efficient technique for solving non-convex quadratic programming problems approximately, various relaxation-based methods for S3VM have been proposed (De Bie and Cristianini, 2004; Xu and Schuurmans, 2005; Valizadegan and Jin, 2006; Xu *et al*, 2008; Bai *et al*, 2012, 2013; Bai and Yan, 2015). In this section, we derive a semi-definite relaxation for each of the two equivalent formulations of the SSQSSVM model.

4.1. Semi-definite relaxation of SSQSSVM

Define $n_1 \triangleq (m^2+3m)/2 + 2n - l + 1$, $\boldsymbol{u} \triangleq (\boldsymbol{v}^T c\, \boldsymbol{\xi}^T (\boldsymbol{y}^{n-l})^T)^T \triangleq (u_1, u_2, \ldots, u_{n_1}) \in \mathbb{R}^{n_1}$. For $i = 1, \ldots, l$, let $\boldsymbol{e}_l^i$ be an l-dimensional vector in which all elements are 0 except that its i-th element is 1. For $j = 1, \ldots, n-l$, let e_{n-l}^j be an $(n-l)$-dimensional vector in which all elements are 0 except that its j-th element is 1. Then, Problem (3) can be reformulated as:

$$\min_{u\in\mathbb{R}^{n_1}} \quad \boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u}+\boldsymbol{q}^T\boldsymbol{u}$$
$$\text{s.t.} \quad \boldsymbol{a}_i^T\boldsymbol{u}\geqslant 1, i=1,\ \ldots\ ,l,$$
$$\left(\boldsymbol{f}_j^T\boldsymbol{u}\right)\left(\boldsymbol{d}_j^T\boldsymbol{u}\right)\geqslant 1-\boldsymbol{g}_j^T\boldsymbol{u}, j=1,\ \ldots\ ,n-l,$$
$$u_k\geqslant 0, k=\frac{m^2+3m}{2}+2,\ \ldots\ ,\frac{m^2+3m}{2}+n+1,$$
$$u_t\in\{-1,1\}, t=\frac{m^2+3m}{2}+n+2,\ \ldots\ ,n_1, \quad (5)$$

where $\boldsymbol{q}=\begin{pmatrix}\boldsymbol{0}_{1\times\left(\frac{m^2+3m}{2}+1\right)} & C_l\boldsymbol{e}_l^T & C_{n-l}\boldsymbol{e}_{n-l}^T & \boldsymbol{0}_{1\times(n-l)}\end{pmatrix}^T\in\mathbb{R}^{n_1}$,

$$\boldsymbol{Q}=\begin{pmatrix}\boldsymbol{G} & \boldsymbol{0}_{\frac{m^2+3m}{2}\times(2n-l+1)}\\ \boldsymbol{0}_{(2n-l+1)\times\frac{m^2+3m}{2}} & \boldsymbol{0}_{(2n-l+1)\times(2n-l+1)}\end{pmatrix}\in\mathbb{S}^{n_1},$$
$$\boldsymbol{d}_j=\begin{pmatrix}\boldsymbol{s}_j & 1 & \boldsymbol{0}_{1\times(2n-l)}\end{pmatrix}^T\in\mathbb{R}^{n_1},$$
$$\boldsymbol{a}_i=\begin{pmatrix}y_i\boldsymbol{s}_i^T & y_i & \left(\boldsymbol{e}_l^i\right)^T & \boldsymbol{0}_{1\times(2n-2l)}\end{pmatrix}^T\in\mathbb{R}^{n_1},$$
$$\boldsymbol{f}_j=\begin{pmatrix}\boldsymbol{0}_{1\times\left(\frac{m^2+3m}{2}+n+1\right)} & \left(\boldsymbol{e}_{n-l}^j\right)^T\end{pmatrix}^T\in\mathbb{R}^{n_1},$$
$$\boldsymbol{g}_j=\begin{pmatrix}\boldsymbol{0}_{1\times\left(\frac{m^2+3m}{2}+l+1\right)} & \left(\boldsymbol{e}_{n-l}^j\right)^T & \boldsymbol{0}_{1\times(n-l)}\end{pmatrix}^T\in\mathbb{R}^{n_1},$$
$$\text{for } i=1,\ \ldots\ ,l, j=1,\ \ldots\ ,n-l.$$

This problem is a quadratic programming problem with a convex objective function and a non-convex feasible domain. An effective approach to approximate Problem (5) is to use the semi-definite relaxation technique. The idea is to reformulate the problem by introducing a rank-1 matrix variable $\boldsymbol{U}=\boldsymbol{u}\boldsymbol{u}^T$. After dropping the constraint $\text{rank}(\boldsymbol{U})=1$, we derive

$$(\text{SDP1}) \min_{u\in\mathbb{R}^{n_1},U\in\mathbb{S}^{n_1}} \boldsymbol{Q}\bullet\boldsymbol{U}+\boldsymbol{q}^T\boldsymbol{u}$$
$$\text{s.t.} \quad \boldsymbol{a}_i^T\boldsymbol{u}\geqslant 1, \left(\boldsymbol{a}_i\boldsymbol{a}_i^T\right)\bullet\boldsymbol{U}\geqslant 1, i=1,\ \ldots\ ,l,$$
$$\boldsymbol{f}_j\boldsymbol{d}_j^T\bullet\boldsymbol{U}\geqslant 1-\boldsymbol{g}_j^T\boldsymbol{u}, j=1,\ \ldots\ ,n-l,$$
$$U_{tt}=1, t=\frac{m^2+3m}{2}+n+2,\ \ldots\ ,n_1,$$
$$u_k\geqslant 0, U_{kk}\geqslant 0, k=\frac{m^2+3m}{2}+2,\ \ldots\ ,\frac{m^2+3m}{2}+n+1,$$
$$\boldsymbol{U}-\boldsymbol{u}\boldsymbol{u}^T\succcurlyeq\boldsymbol{0}. \quad (6)$$

Problem (6) is a semi-definite programming problem, which is polynomial-time solvable. Suppose that $(\boldsymbol{u}^*,\boldsymbol{U}^*)$ is optimal to Problem (6), we can obtain an approximated optimal solution for Problem (3) by letting $\boldsymbol{y}^{n-l}=(\text{sign}(u^*_{(m^2+3m)/2+n+2}), \text{sign}(u^*_{(m^2+3m)/2+n+3}),\ \ldots\ ,\text{sign}(u^*_{n_1}))^T$. Besides, to avoid unbalanced solutions, the following class-balancing constraint in Chapelle *et al* (2008) can be added: $(1/(n-l))\sum_{i=l+1}^n y_i=2r-1$, where r is estimated by the ratio of the number of data points with a positive label in the total number of data points in the labelled data set.

4.2. Semi-definite relaxation of SSQSSVM′

Define $n_2\triangleq(m^2+3m)/2+n+1$, $z\triangleq(v^Tc\ \boldsymbol{\xi}^T)^T\in\mathbb{R}^{n_2}$, then Problem (4) can be reformulated as

$$\min_{z\in\mathbb{R}^{n_2}} \quad z^T\tilde{\boldsymbol{Q}}z+\tilde{\boldsymbol{q}}^Tz$$
$$\text{s.t.} \quad \tilde{\boldsymbol{a}}_i^Tz\geqslant 1, i=1,\ \ldots\ ,l,$$
$$|\ \tilde{\boldsymbol{d}}_j^Tz\ |\ \geqslant 1-\tilde{\boldsymbol{g}}_j^Tz, j=1,\ \ldots\ ,n-l,$$
$$z_k\geqslant 0, k=\frac{m^2+3m}{2}+2,\ \ldots\ ,n_2, \quad (7)$$

where $\tilde{\boldsymbol{q}}=\begin{pmatrix}\boldsymbol{0}_{1\times\left(\frac{m^2+3m}{2}+1\right)} & 1 & C_l\mathbf{e}_l^T & C_{n-l}\mathbf{e}_{n-l}^T\end{pmatrix}^T\in\mathbb{R}^{n_2}$,

$$\tilde{\boldsymbol{d}}_j=\begin{pmatrix}\boldsymbol{s}_j^T & 1 & \boldsymbol{0}_{1\times n}\end{pmatrix}^T\in\mathbb{R}^{n_2},$$
$$\tilde{\boldsymbol{Q}}=\begin{pmatrix}G & \boldsymbol{0}_{\frac{m^2+3m}{2}\times(n+1)}\\ 0_{(n+1)\times\frac{m^2+3m}{2}} & \boldsymbol{0}_{(n+1)\times(n+1)}\end{pmatrix}\in\mathbb{S}^{n_2},$$
$$\tilde{\boldsymbol{a}}_i=\begin{pmatrix}y_i\boldsymbol{s}_i^T & y_i & \left(\boldsymbol{e}_l^i\right)^T & \boldsymbol{0}_{1\times(n-l)}\end{pmatrix}^T\in\mathbb{R}^{n_2},$$
$$\tilde{\boldsymbol{g}}_j=\begin{pmatrix}\boldsymbol{0}_{1\times\left(\frac{m^2+3m}{2}+l+1\right)} & \left(\boldsymbol{e}_{n-l}^j\right)^T\end{pmatrix}^T\in\mathbb{R}^{n_2},$$
$$\text{for } i=1,\ \ldots\ ,l, j=1,\ \ldots\ ,n-l.$$

This problem is also a non-convex optimization problem with a convex objective function and a non-convex feasible domain. We see that the constraints $|\ \tilde{\boldsymbol{d}}_j^Tz\ |\ \geqslant 1-\tilde{\boldsymbol{g}}_j^Tz$, $j=1,\ \ldots\ ,n-l$, in Problem (7) can be reformulated as $(\tilde{\boldsymbol{d}}_j^Tz)^2\geqslant(1-\tilde{\boldsymbol{g}}_j^Tz)^2$, $j=1,\ \ldots\ ,n-l$, equivalently. By introducing a matrix variable $\boldsymbol{Z}=zz^T$, then dropping the non-convex constraint of $\text{rank}(\boldsymbol{Z})=1$, we have the following semi-definite relaxation problem:

$$(\text{SDP2}) \min_{z\in\mathbb{R}^{n_2},Z\in\mathbb{S}^{n_2}} \tilde{Q}\bullet\boldsymbol{Z}+\tilde{\boldsymbol{q}}^Tz$$
$$\text{s.t.} \quad \tilde{\boldsymbol{a}}_i^Tz\geqslant 1, \tilde{\boldsymbol{a}}_i\tilde{\boldsymbol{a}}_i^T\bullet\boldsymbol{Z}\geqslant 1, i=1,\ \ldots\ ,l,$$
$$\left(\tilde{\boldsymbol{d}}_j\tilde{\boldsymbol{d}}_j^T-\tilde{\boldsymbol{g}}_j\tilde{\boldsymbol{g}}_j^T\right)\bullet\boldsymbol{Z}+2\tilde{\boldsymbol{g}}_j^Tz\geqslant 1, j=1,\ \ldots\ ,n-l,$$
$$z_k\geqslant 0, Z_{kk}\geqslant 0, k=\frac{m^2+3m}{2}+2,\ \ldots\ ,\frac{m^2+3m}{2}+n+1,$$
$$\boldsymbol{Z}-zz^T\succcurlyeq\boldsymbol{0}. \quad (8)$$

If we have an optimal solution $(z^*,\boldsymbol{Z}^*)$ of Problem (8), the labels of unlabelled data points can be approximately derived

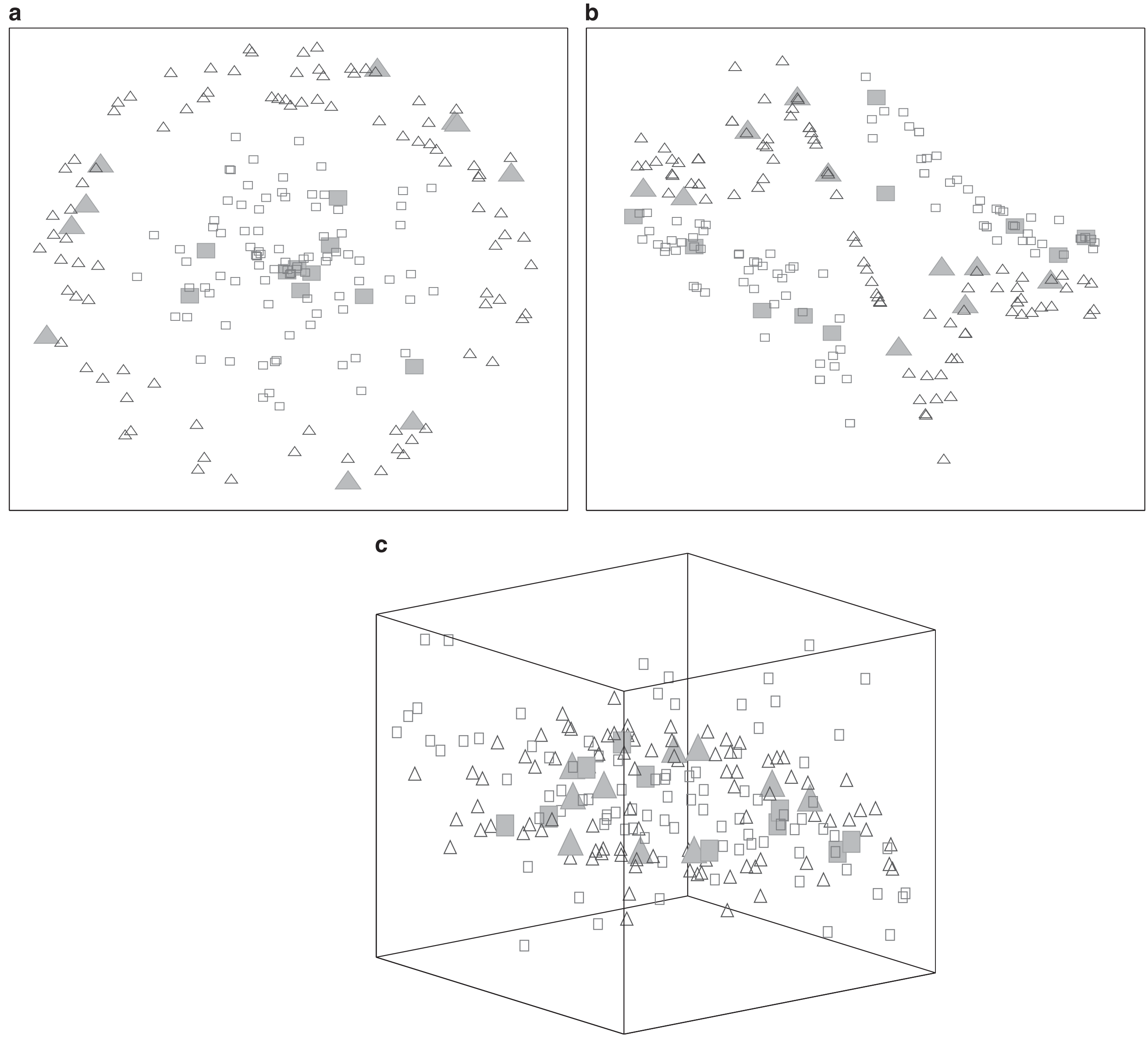

Figure 1 The distribution of artificial data sets: (a) 2circles; (b) 2d-non-convex; (c) 3d-non-convex. Triangle and square represent two different classes of data points, respectively. The filled points are labelled, and the hollow ones are unlabelled.

by letting $y_{j+l} = \text{sign}(\tilde{\boldsymbol{d}}_j^T \boldsymbol{z}^*)$, $j = 1, \ldots, n-l$. Similar to SDP1, the following balancing constraints similar to that in Chapelle *et al* (2008) can also be added: $(1/(n-l)) \sum_{i=l+1}^{n} \boldsymbol{v}^T \boldsymbol{s}_i + c = 2r - 1$, where r is estimated by the class ratio of the labelled data set. Compared with problem SDP1, problem SDP2 has fewer variables for more efficient computation.

4.3. Relationship between SDP1 and SDP2

Theorem 1 *Let ($\boldsymbol{u}^*$, $\boldsymbol{U}^*$) and (z^*, $\boldsymbol{Z}^*$) be the optimal solutions to the problems SDP1 and SDP2, respectively. If rank ($\boldsymbol{U}^*$) = rank($\boldsymbol{Z}^*$) = 1, then problems SDP1 and SDP2 are equivalent.*

This theorem tells us that these two semi-definite relaxations are equivalent, if the rank-1 condition holds for both problems (for more details of Theorem 1, please refer to the Appendix). Since problem SDP2 has fewer variables than problem SDP1, we adopt problem SDP2 for testing the performance of the SSQSSVM model for more efficient computation.

5. Numerical experiments

In this section, we investigate the performance of the SSQSSVM model on some artificial and public benchmark

Table 1 Information of tested data sets

Data sets	*Number of features*	*Positive class*		*Negative class*		*Number of labelled points*
		Name	*Number of points*	*Name*	*Number of points*	
2circles	2	Class 1	100	Class 2	100	20
2d-hyperbola	2	Class 1	100	Class 2	100	20
3d-hyperbola	3	Class 1	100	Class 2	100	20
Iris	4	Versicolour	50	Virginica	50	10
Seeds	7	Kama	70	Canadian	70	14
Iono	34	Bad	225	Good	126	20
Sonar	60	Rock	97	Mine	111	20
Uspst	256	Class 4	196	Class 9	180	15

Table 2 Mean and standard deviation of misclassification rates (%) for tested data sets

Data sets	*SSQSSVM*		*CTSVM*		*CutS3VM*		*LDS*		*SVM*	
	Mean	*std*	*Mean*	*std*	*Mean*	*std*	*Mean*	*std*	*Mean*	*std*
2circles	**1.67**	2.23	6.33	4.36	2.44	0.95	2.17	3.60	7.06	3.65
2d-hyperbola	**10.11**	8.85	16.61	9.71	11.28	7.40	49.17	4.30	29.67	5.63
3d-hyperbola	**22.94**	12.06	49.07	3.33	47.94	4.00	48.96	3.98	44.44	0
Iris	5.37	2.22	8.33	3.42	6.00	3.31	**4.59**	0.96	8.78	4.69
Seeds	**8.44**	1.58	8.70	1.80	8.62	1.90	10.03	1.51	12.30	3.11
Iono	**13.49**	3.60	17.07	5.08	14.15	6.15	14.76	5.59	20.86	8.33
Sonar	**25.05**	4.89	33.48	4.47	32.36	4.03	33.72	5.24	32.70	4.68
Uspst	8.85	1.36	15.87	4.05	**8.08**	6.02	11.28	5.06	35.84	7.25

Note: The bold values in this table are the best results among that of all the methods.

data sets. For comparisons, the CCCP (Collobert *et al*, 2006), LDS (Chapelle and Zien, 2005), convex relaxation (CTSVM) (Xu *et al*, 2008) and cutting plane S^3VM (CutS3VM) (Zhao *et al*, 2008) methods are also tested on the same data sets for semi-supervised classification. Moreover, to see whether the unlabelled points make contributions to a suitable classifier, we test the standard SVM model on the same data sets using only the labelled data points for comparisons.

Notice that the tested artificial data sets include the 2circles, 2d-hyperbola and 3d-hyperbola data sets. The distribution of data points in these three data sets are shown in Figure 1. Five public benchmark data sets (Iris, Seeds, Iono, Sonar and Uspst) are also used for computational experiments. The Iris, Seeds, Iono and Sonar data sets are obtained from the UC Irvine Machine Learning Repository (UCI) (downloaded from http://archive.ics.uci.edu/ml/datasets.html), while the Uspst data set is chosen from Zhao *et al* (2008). For binary classification, if one data set contains more than two classes, we only choose two of them for test. The information of tested data sets (with chosen classes) is shown in Table 1.

For each tested data set, like most practices in the literature of S^3VM models (Chapelle and Zien, 2005; Collobert *et al*, 2006; Xu *et al*, 2008; Zhao *et al*, 2008), we randomly pick a fixed number of data points (shown in Table 1) as the labelled data points, and the remaining points are treated as the unlabelled ones. SSQSSVM is first trained using the labelled and unlabelled data points to generate the parameters of the classifying quadratic surface. Then the quadratic surface is used to classify the unlabelled data points, and then the misclassification rate is calculated. Here, the misclassification rate is the ratio of the number of mislabelled data points in the total number of unlabelled data points. To be statistically meaningful, we repeat the test with randomly selected labelled points for 30 times. In the kernel-based methods, the Gaussian kernel $k(\boldsymbol{x}^i, \boldsymbol{x}^j) = exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2/2\sigma^2)$ is used, where the parameter σ is chosen as the median of the pairwise distances. For all methods, we use the grid method to find the optimal parameters

Table 3 Mean of CPU time(s) for tested data sets

Data sets	*SSQSSVM*	*CTSVM*	*CutS3VM*	*LDS*
2circles	184.19	8.19	13.82	0.24
2d-hyperbola	188.58	8.79	13.48	0.25
3d-hyperbola	210.06	8.44	7.71	0.27
Iris	26.41	3.14	3.58	0.20
Seeds	118.18	4.60	0.78	0.21
Iono	6.90	34.67	20.19	1.16
Sonar	1.52	8.35	10.90	0.53
Uspst	32.65	27.90	267.35	1.22

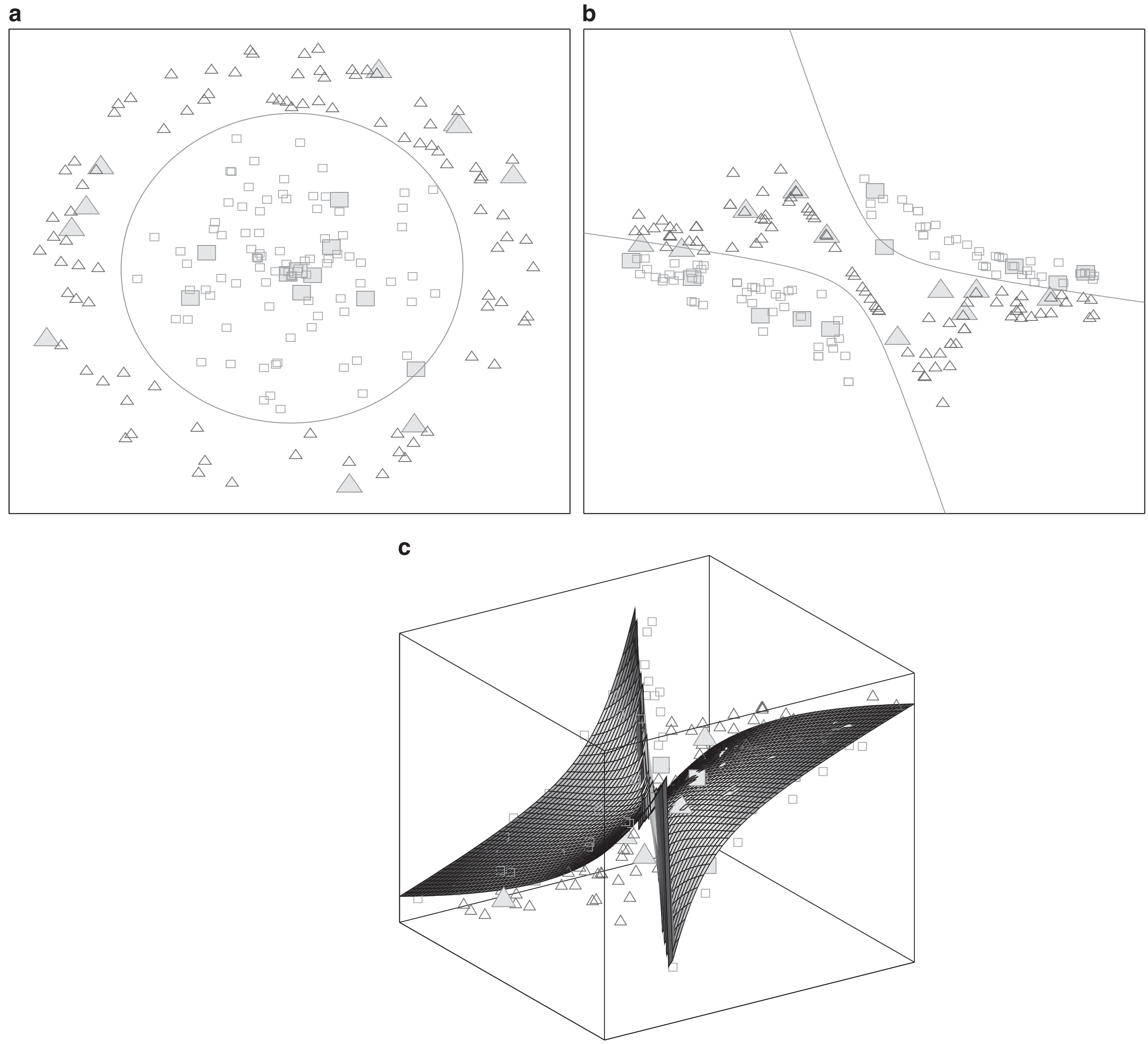

Figure 2 Separating quadratic surfaces produced by SSQSSVM on artificial data sets: (a) 2circles; (b) 2d-non-convex; (c) 3d-non-convex.

C_l and C_{n-l}: $\log_2 C_l, \log_2 C_{n-l} \in \{-4, -5, \ldots, 9, 10\}$. For other parameters in the compared methods, the default values are used. All computational experiments in this paper are carried out using Matlab 7.12 (R2011a) on a PC equipped with 2.60 GHz CPU and 4 GB usable RAM.

First, we test the SSQSSVM model on artificial, Iris and Seeds data sets by solving problem SDP2 via the optimization package CVX (Grant *et al*, 2015). The mean and standard deviation of the calculated misclassification rates and average CPU time of the 30 experiments are reported in Tables 2 and 3. Then, for fair comparisons, we carry out similar experiments for CTSVM, CutS3VM, LDS and SVM using the same labelled and unlabelled data points, and record the results in Tables 2 and 3. Notice that a smaller misclassification rate indicates that the corresponding model performs better in terms of the classification accuracy. Furthermore, we depict the separating quadratic surface of each artificial data set for one single test, generated by the SSQSSVM model, in Figure 2.

Besides, we test the SSQSSVM model on Iono, Sonar and Uspst data sets by directly using the decomposition algorithm (Le *et al*, 2013) instead of solving problem SDP2. That is because, for these data sets with relative larger number of features, solving problem SDP2 would lead to memory overflows, which is a common phenomenon for solving the SDP relaxations of S^3VM model (De Bie and Cristianini, 2004; Xu and Schuurmans, 2005; Bai and Yan, 2015). Using the similar experimental procedure as that implemented on the artificial data set, all methods (including SSQSSVM, CCCP, LDS, Cut3SVM and SVM) are implemented on the Iono, Sonar or Uspst data set, respectively,

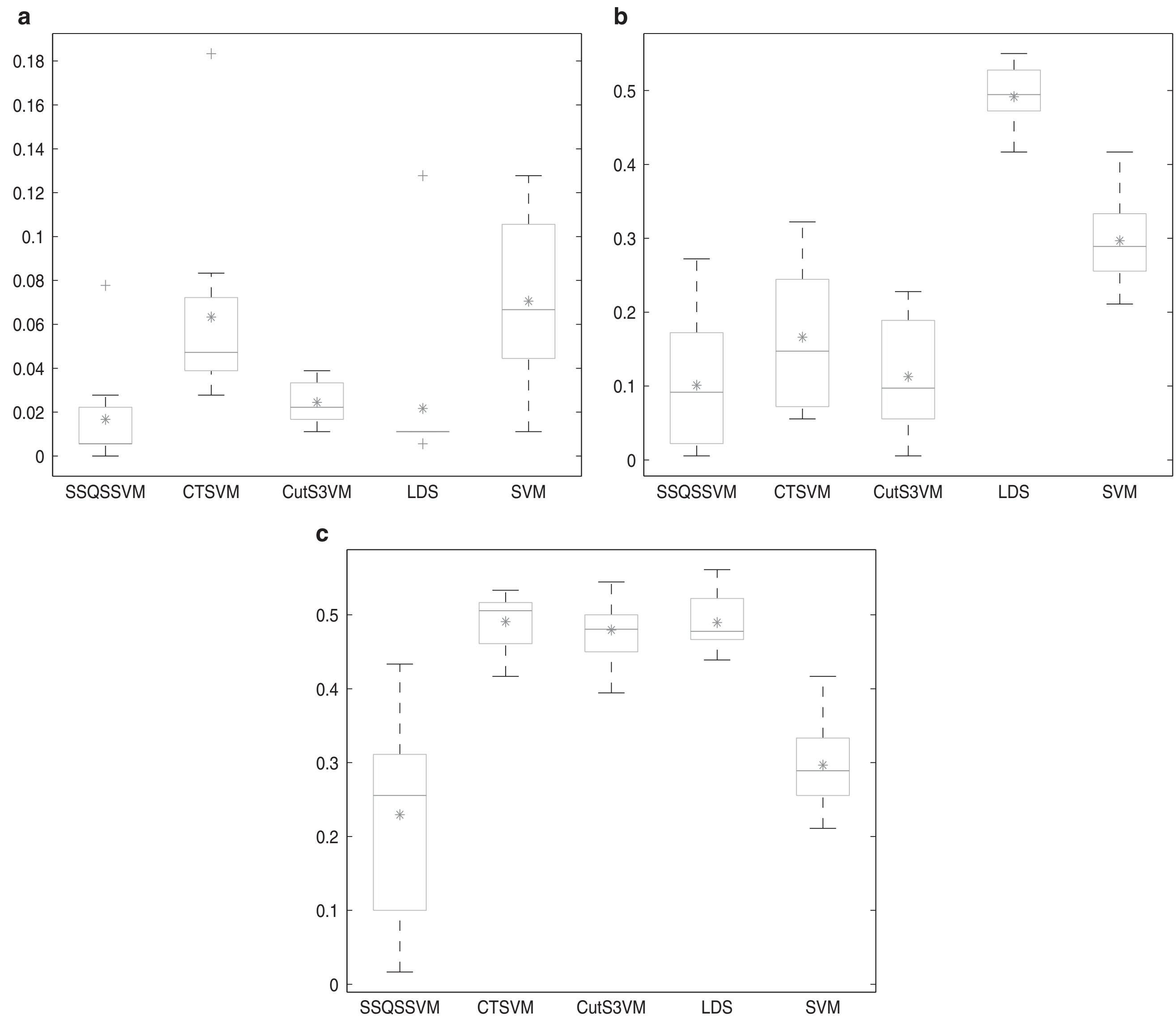

Figure 3 Boxplots for artificial data sets: **(a)** 2circles data set; **(b)** 2d-hyperbola data set; **(c)** 3d-hyperbola data set. The star and the middle line denote the mean and median, respectively.

and all the computational results are included in Tables 2 and 3. Finally, we draw the boxplots in Figures 3 and 4 for the misclassification rates on all tested data sets to further show the performance of our model.

From Tables 2–3 and Figures 2–4, we have the following observations:

- Our methods for solving SSQSSVM outperform other existing well-known methods for most experiments on tested data sets in terms of misclassification rate. Especially for the artificial, Seeds and Sonar data sets, the mean, median and minimum values of our methods are all the smallest
- For experiments on tested data sets, the tested semi-supervised SVM model yields more accurate classification than the supervised SVM model, which indicates that the unlabelled data points always help make the classifiers more suitable
- The average CPU time of the SSQSSVM model on the artificial, Iris or Seeds data set is comparably large because of solving its SDP relaxation problem SDP2, while the average CPU time of the SSQSSVM model on the Iono, Sonar and Uspst data set is comparably small by using the decomposition algorithm

6. Concluding remarks

In this paper, we have proposed a kernel-free SSQSSVM model directly using a quadratic surface for semi-supervised binary

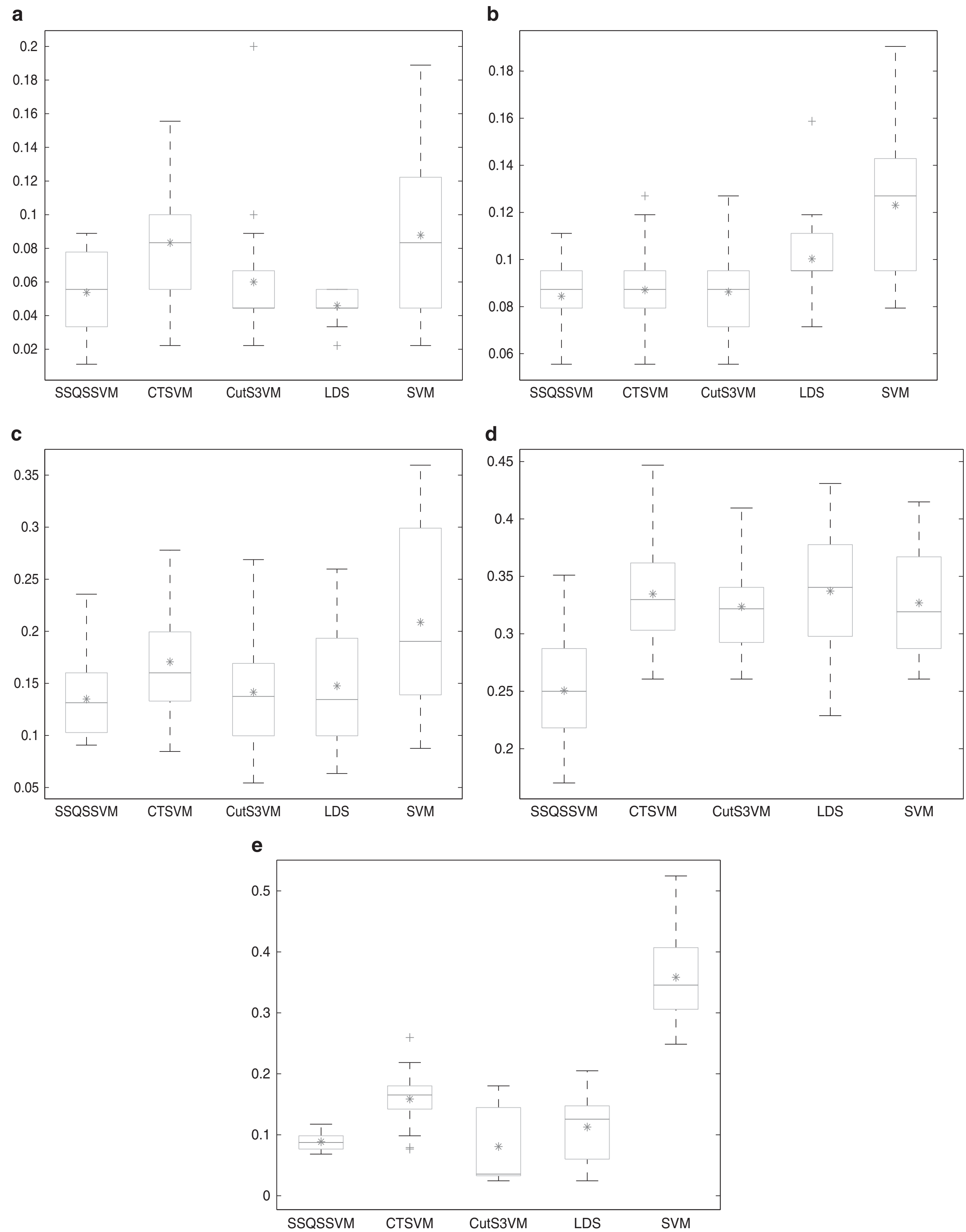

Figure 4 Boxplots for benchmark data sets: **(a)** Iris data set; **(b)** Seeds data set; **(c)** Iono data set; **(d)** Sonar data set; **(e)** Uspst data set. The star and the middle line denote the mean and median, respectively.

classification. The proposed model corresponds to a mixed-integer programming problem, which is NP-hard. We derived the equivalent non-convex problem with absolute-value constraints, and presented the polynomial-time solvable semi-definite relaxation of this problem for finding approximate solutions. Moreover, we implemented the proposed SSQSSVM model partially by solving the relaxed problem on both artificial and public benchmark data sets. The numerical results indicated that our methods for solving SSQSSVM produced more accurate classification than some existing well-known methods for solving S^3VM such as CTSVM, LDS and Cut3SVM methods. And the computational results also indicated that our proposed SSQSSVM model outperforms the S^3VM model in terms of classification accuracy.

In Wen *et al* (2010), an alternating direction method of multipliers (ADMM) for semi-definite programming problem is presented, which leads to a computationally efficient algorithm for solving high-dimensional problems. For future research, we plan to reformulate the SDP relaxations for SSQSSVM and then develop an ADMM algorithm for solving such problems.

Acknowledgements—This research was partially supported by the National Natural Science Foundation of China Grant (No.11371242) and US National Science Foundation Grant (No. DMI-0553310).

References

Baesens B, Van Gestel T, Viaene S, Stepanova M, Suykens J and Vanthienen J (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society* **54**(6): 627–635.

Bai Y and Yan X (2015). Conic relaxation for semi-supervised support vector machines. *Journal of Optimization Theory and Applications*. DOI: 10.1007/s10957-015-0843-4.

Bai Y, Chen Y and Niu B (2012). SDP relaxation for semi-supervised support vector machine. *Pacific Journal of Optimization* **8**(1): 3–14.

Bai Y, Niu B and Chen Y (2013). New SDP models for protein homology detection with semi-supervised SVM. *Optimization* **62**(4): 561–572.

Bennett K and Demiriz A (1999). Semi-supervised support vector machines. In: Kearns MJ, Solla SA and Cohn DA (eds). *Advances in Neural Information Processing Systems 11*. MIT Press: Cambridge, MA, pp 368–374.

Boylan J, Syntetos AA and Karakostas G (2008). Classification for forecasting and stock control: A case study. *Journal of the Operational Research Society* **59**(4): 473–481.

Chapelle O and Zien A (2005). Semi-supervised classification by low density separation. In: Cowell RG and Ghahramani Z (eds). *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, Society for Artificial Intelligence and Statistics, http://www.gatsby.ucl.ac.uk/aistats/, pp 57–64.

Chapelle O, Sindhwani V and Keerthi SS (2008). Optimization techniques for semi-supervised support vector machines. *The Journal of Machine Learning Research* **9**(Feb): 203–233.

Collobert R, Sinz F, Weston J and Bottou L (2006). Large scale transductive SVMs. *The Journal of Machine Learning Research* **7**(Aug): 1687–1712.

Cortes C and Vapnik V (1995). Support-vector networks. *Machine learning* **20**(3): 273–297.

Cristianini N and Shawe-Taylor J (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press: Cambridge.

De Bie T and Cristianini N (2004). Convex methods for transduction. In: Thrun S, Saul LK and Schölkopf B (eds). *Advances in Neural Information Processing Systems 16*, MIT Press: Cambridge, pp 73–80.

Deng NY, Tian YJ and Zhang CH (2012). *Support Vector Machines-Optimization Based Theory, Algorithms and Extensions*. CRC Press: Boca Raton.

Grant M, Boyd S and Ye Y (2015). *CVX*: *Matlab software for disciplined convex programming*, version 2.1. Available from http://cvxr.com/cvx/.

Hansen J, McDonald J and Nelson R (2006). Some evidence on forecasting time-series with support vector machines. *Journal of the Operational Research Society* **57**(9): 1053–1063.

Issam D (2008). Quadratic kernel-free non-linear support vector machine. *Journal of Global Optimization* **41**(1): 15–30.

Joachims T (1999). Transductive inference for text classification using support vector machines. In: Bratko I and Dzeroski S (eds). *Proceedings of 16th International Conference on Machine Learning*, Morgan Kaufmann: San Francisco, CA, pp 200–209.

Kennedy K, Mac Namee B and Delany SJ (2012). Using semi-supervised classifiers for credit scoring. *Journal of the Operational Research Society* **64**(4): 513–529.

Le HM, Le Thi HA and Nguyen MC (2013). DCA based algorithms for feature selection in semi-supervised support vector machines. In: Perner P (ed). *Machine Learning and Data Mining in Pattern Recognition*, LNAI 7988, pp 528–542.

Lessmann S, Baesens B, Seow HV and Thomas LC (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research* **247**(1): 124–136.

Li H and Hand D (2002). Direct versus indirect credit scoring classifications. *Journal of the Operational Research Society* **53**(6): 647–654.

Luo J, Fang SC, Bai Y and Deng Z (2016). Fuzzy quadratic surface support vector machine based on Fisher discriminant analysis. *Journal of Industrial and Management Optimization* **12**(1): 357–373.

Luo J (2014). Quadratic surface support vector machines with applications. Ph.D. dissertation, North Carolina State University.

Maldonado S and Paredes G (2010). A semi-supervised approach for reject inference in credit scoring using SVMs. In: Perner P (ed). *Advances in Data Mining, Applications and Theoretical Aspects*. Springer: Berlin, pp 558–571.

Osuna E, Freund R and Girosi F (1997). Training support vector machines: An application to face detection. In: Medioni G and Nevatia R (eds). *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society: Washington DC, pp 130–136.

Schebesch KB and Stecking R (2005). Support vector machines for classifying and describing credit applicants: Detecting typical and critical regions. *Journal of the Operational Research Society* **56**(9): 1082–1088.

Sindhwani V, Keerthi SS and Chapelle O (2006). Deterministic annealing for semi-supervised kernel machines. In: Cohen W and Moore A (eds). *Proceedings of the 23rd International Conference on Machine Learning*, Omni Press: Madison, WI, pp 841–848.

Sun J, Shang Z and Li H (2014). Imbalance-oriented SVM methods for financial distress prediction: A comparative study among the new SB-SVM-ensemble method and traditional methods. *Journal of the Operational Research Society* **65**(12): 1905–1919.

Valizadegan H and Jin R (2006). Generalized maximum margin clustering and unsupervised kernel learning. In: Schölkopf B, Platt J and Hofmann T (eds). *Advances in Neural Information Processing Systems 19,* MIT Press: Cambridge, pp 1417–1424.

Vapnik NK and Sterin A (1977). On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control* **10**(3): 1495–1503.

Wen ZW, Goldfarb D and Yin WT (2010). Alternating direction augmented Lagrangian methods for semidefinite programming. *Mathematical Programming Computation* **2**(3–4): 203–230.

Xu L and Schuurmans D (2005). Unsupervised and semi-supervised multi-class support vector machines. In: Veloso M and Kambhampati S (eds). *Proceedings of the 20th National Conference on Artificial Intelligence*, Vol. 20, AAAI Press: Menlo Park, CA, pp 904–910.

Xu ZL, Jin R, Zhu JK, King I and Lyu M (2008). Efficient convex relaxation for transductive support vector machine. In: Platt JC, Koller D, Singer Y and Roweis S (eds). *Advances in Neural Information Processing Systems 20*. MIT Press: Cambridge: pp 1641–1648.

Zhao B, Wang F and Zhang CS (2008). CutS3VM: A fast semi-supervised SVM algorithm. In: Li Y, Liu B and Sarawagi S (eds). *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM: New York, pp 830–838.

Appendix

Proof of Theorem 1

The Proof of Theorem 1 can be derived as follows:

Proof First, we show that opt(SDP1) $\geqslant$ opt(SDP2). Given an optimal solution $(\boldsymbol{u}^*, \boldsymbol{U}^*)$ of problem SDP1, let

$$z_i \triangleq u_i^*, Z_{ij} \triangleq U_{ij}^*, i,j = 1, \ldots, \frac{m^2+3m}{2}+n+1.$$

Then,

$$\tilde{\boldsymbol{a}}_i^T \boldsymbol{z} = \boldsymbol{a}_i^T \boldsymbol{u}^* \geqslant 1, \tilde{\boldsymbol{a}}_i(\tilde{\boldsymbol{a}}_i)^T \bullet \boldsymbol{Z} = \boldsymbol{a}_i \boldsymbol{a}_i^T \bullet \boldsymbol{U}^* \geqslant 1,\ i = 1, \ldots, l.$$

Since

$$U_{tt}^* = 1, t = \frac{m^2+3m}{2}+n+2, \ldots, n_1,$$

$$\text{and } \boldsymbol{g}_j^T \boldsymbol{u}^* \leqslant 1,$$

$$j = 1, \ldots, n-l,$$

we have

$$\tilde{\boldsymbol{d}}_j \tilde{\boldsymbol{d}}_j^T \bullet \boldsymbol{Z} \geqslant \left(\boldsymbol{f}_j \boldsymbol{d}_j^T \bullet \boldsymbol{U}^*\right)^2.$$

Since rank$(\boldsymbol{U}^*) = 1$, we have

$$\tilde{\boldsymbol{d}}_j \tilde{\boldsymbol{d}}_j^T \bullet \boldsymbol{Z} \geqslant \left(1 - \boldsymbol{g}_j^T \boldsymbol{u}^*\right)^2 = 1 - 2\tilde{\boldsymbol{g}}_j^T \boldsymbol{z} + \tilde{\boldsymbol{g}} \tilde{\boldsymbol{g}}_j^T \bullet Z,$$

$$j = 1, \ldots, n-l.$$

Moreover, we have

$$z_k = u_k^* \geqslant 0,\ Z_{kk} = U_{kk}^* \geqslant 0,$$

$$k = \frac{m^2+3m}{2}+2, \ldots, \frac{m^2+3m}{2}+n+1.$$

Thus $(\boldsymbol{z}, \boldsymbol{Z})$ is a feasible solution to problem SDP2.

Finally, we have

$$\tilde{\boldsymbol{Q}} \bullet \boldsymbol{Z} + \tilde{\boldsymbol{q}}^T \boldsymbol{z} = \boldsymbol{Q} \bullet \boldsymbol{U}^* + \boldsymbol{q}^T \boldsymbol{u}^*,$$

which implies that opt(SDP1) $\geqslant$ opt(SDP2).

Next, we show that opt(SDP2) $\geqslant$ opt(SDP1). Suppose that $(\boldsymbol{z}^*, \boldsymbol{Z}^*)$ is an optimal solution of problem SDP2. Let $\boldsymbol{D} \triangleq (\tilde{\boldsymbol{d}}_1, \tilde{\boldsymbol{d}}_2, \ldots, \tilde{\boldsymbol{d}}_{n-l})$, and

$$\boldsymbol{u} \triangleq \begin{pmatrix} \boldsymbol{z}^* \\ \text{sign}\left(\boldsymbol{D}^T \boldsymbol{z}^*\right) \end{pmatrix},$$

$$\boldsymbol{U} \triangleq \begin{pmatrix} \boldsymbol{Z}^* & \boldsymbol{z}^* \text{sign}\left(\boldsymbol{D}^T \boldsymbol{z}^*\right)^T \\ \text{sign}\left(\boldsymbol{D}^T \boldsymbol{z}^*\right)(\boldsymbol{z}^*)^T & \text{sign}\left(\boldsymbol{D}^T \boldsymbol{z}^*\right) \text{sign}\left(\boldsymbol{D}^T \boldsymbol{z}^*\right)^T \end{pmatrix}.$$

We have

$$\boldsymbol{a}_i^T \boldsymbol{u} = \tilde{\boldsymbol{a}}_i^T \boldsymbol{z}^* \geqslant 1,\ \left(\boldsymbol{a}_i \boldsymbol{a}_i^T\right) \bullet \boldsymbol{U} = \tilde{\boldsymbol{a}}_i \tilde{\boldsymbol{a}}_i^T \bullet \boldsymbol{Z}^* \geqslant 1,\ i = 1, \ldots, l.$$

Since

$$\tilde{\boldsymbol{d}}_j \tilde{\boldsymbol{d}}_j^T \bullet \boldsymbol{Z}^* \geqslant 1 - 2\tilde{\boldsymbol{g}}_j^T \boldsymbol{z}^* + \tilde{\boldsymbol{g}}_j \tilde{\boldsymbol{g}}_j^T \bullet \boldsymbol{Z}^*$$

$$\text{and } \tilde{\boldsymbol{g}}_j^T \boldsymbol{z}^* \leq 1,$$

$$j = 1, \ldots, n-l,$$

using the fact that rank$(\boldsymbol{Z}^*) = 1$, we have

$$\left|\ \tilde{\boldsymbol{d}}_j^T \boldsymbol{z}^*\ \right| \geqslant 1 - \tilde{\boldsymbol{g}}_j^T \boldsymbol{z}^*, j = 1, \ldots, n-l.$$

Consequently,

$$\boldsymbol{f}_j \boldsymbol{d}_j^T \bullet \boldsymbol{U} = \text{sign}\left(\tilde{\boldsymbol{d}}_j^T \boldsymbol{z}^*\right) \tilde{\boldsymbol{d}}_j^T \boldsymbol{z}^* \geqslant 1 - \boldsymbol{g}_j^T \boldsymbol{u},\ j = 1, \ldots, n-l.$$

Moreover,

$$U_{tt} = \left(\text{sign}\left(\tilde{\boldsymbol{d}}_t^T \boldsymbol{z}^*\right)\right)^2 = 1,\ t = \frac{m^2+3m}{2}+n+2, \ldots, n_1,$$

and

$$u_k = z_k^* \geqslant 0, U_{kk} = Z_{kk}^* \geqslant 0,$$

$$k = \frac{m^2+3m}{2}+2, \ldots, \frac{m^2+3m}{2}+n+1.$$

Therefore, $(\boldsymbol{u}, \boldsymbol{U})$ is a feasible solution to problem SDP1. Also, we have

$$\boldsymbol{Q} \bullet \boldsymbol{U} + \boldsymbol{q}^T \boldsymbol{u} = \tilde{\boldsymbol{Q}} \bullet \boldsymbol{Z}^* + \tilde{\boldsymbol{q}}^T \boldsymbol{z}^*,$$

which implies that opt(SDP2) $\geqslant$ opt(SDP1). □

Received 5 January 2015;
accepted 25 September 2015 after one revision