Engineering Applications of Artificial Intelligence 67 (2018) 309–316

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A safe screening rule for Laplacian support vector machine

Zhiji Yang, Yitian Xu *

College of Science, China Agricultural University, Beijing 100083, China

ARTICLE INFO

Keywords:
Semi-supervised learning
Support vector machine
Safe screening
Variational inequality

ABSTRACT

Laplacian support vector machine (LapSVM) has received much concern in semi-supervised learning (SSL) field. To further improve its computational speed, many efficient algorithms have been developed. However, they just focus on the method of solving optimization problem not the scale of the problem itself. Inspired by the sparsity of LapSVM, in this paper, an efficient safe screening rule for LapSVM (SSR-LapSVM) is proposed to address this issue. The proposed method could significantly accelerate the original LapSVM. Through the rule, most of the training samples can be eliminated before solving optimization problem without sacrificing the optimal solution. An important advantage is the safety, in the sense that the solution is exactly the same as the original LapSVM. Different from most existing methods, our approach can effectively deal with the multiple parameter problems. Experiments on both 3 artificial datasets and 24 real world benchmark datasets demonstrate its feasibility and efficiency.

1. Introduction

Since support vector machine (SVM) was proposed by Vapnik (1998), it has been widely applied and well studied (Cortes and Vapnik, 1995; Platt and John, 1999; Steinwart and Christmann, 2008). It has many advantages. Firstly, SVM solves a quadratic programming problem (QPP), which guarantees to obtain the unique solution. Secondly, SVM implements the structural risk minimization principle rather than the empirical risk minimization principle, which can minimize the upper bound of the generalization error.

The traditional SVM is a supervised method which needs the label of each instance for establishing the model (Hu et al., 2013). However, in many real situations, the acquisition of class labels is costly and difficult. Semi-supervised learning (SSL) is a category of learning tasks that also make use of unlabeled data for training (Altınel et al., 2017). In the last decades, SSL has attracted a notable attention (Chapelle et al., 2006; Li et al., 2010) and has been introduced into the SVM to improve its prediction performance. Specifically, Laplacian support vector machine (LapSVM) for SSL has been proposed by Chova et al. (2008), which introduces an additional regularization term on the geometry of both labeled and unlabeled samples (Belkin et al., 2006; Zhu, 2005, 2008). This method has been demonstrated to yield an excellent performance. The model is built on manifold assumption which is suitable for most cases. Its solution characterized by a convex quadratic optimization problem is guaranteed to be globally optimal. By contrast, in most other semi-supervised SVMs (Chapelle et al., 2008; Yang et al., 2014), it needs to solve complex non-convex problems, and the solutions are locally optimal.

Many improvements for LapSVM have been presented to enhance its computational speed and prediction accuracy. In the literature (Melacci and Belkin, 2011), two strategies were presented to solve the primal problem of LapSVM in order to reduce the training time. Qi et al. (2014) proposed a fast LapSVM (FLapSVM) in which authors modify the model of LapSVM to avoid extra matrix and reduce the computation. However, these algorithms just focus on the methods of solving optimization problem not the scale of the problem itself. In addition, the Laplacian graph approaches have been successfully introduced into twin support vector machines (TSVMs) (Jayadeva et al., 2007; Xu et al., 2016, 2017; Xu, 2017; Qi et al., 2012; Yang and Xu, 2016). However, it is still challenging to deal with large datasets on account of the burden of computational cost and memory.

In general, the methods mentioned above make use of both labeled and unlabeled samples to construct more reasonable classifiers. However, compared with the traditional SVM, it takes more computational costs to construct the model. The memory consumption becomes heavy as the scale of data increases. This puts a high demand on the configuration of computers. How to improve the computational speed is a valuable topic.

Recently, a promising sample selection approach called "screening" (Ogawa et al., 2013; Wang et al., 2014; Liu et al., 2014; Wang

* Corresponding author.
E-mail address: xytshuxue@126.com (Y. Xu).

https://doi.org/10.1016/j.engappai.2017.10.011
Received 25 June 2017; Received in revised form 19 August 2017; Accepted 12 October 2017
0952-1976/

et al., 2015; Yang et al., 2015; Pan et al., 2017) has been proposed to reduce the scale of SVM. In the SVM, the classifier is determined only by the so called "support vectors". The screening technique could identify most of the non-support vectors, and remove them before solving QPP. That saves the computational cost. One of the important advantages is that the solution obtained via screening method is exactly the same as the original problem. So this technique is regarded as a safe approach. In the literature (Wang et al., 2014), the method DVI (dual problem via variational inequalities) has shown its outstanding performance in the traditional SVM. However, the DVI rule cannot be simply applied to LapSVM. Specifically, the DVI-SVM only deals with one parameter C (Hastie et al., 2004), while there are multiple parameters in LapSVM.

Motivated by the studies above, in this paper a safe screening rule (SSR) is presented to accelerate LapSVM without sacrificing the optimal solution. At first, the Karush–Kuhn–Tucker (KKT) conditions are analyzed to motivate the proposed method. Then, via variational inequalities, the feasible set of the dual solution is constructed with the information obtained at the previous parameters. Combined the KKT conditions and the feasible set, the SSR is finally derived in an analytical form. By this rule, some components of the optimal solution could be achieved directly before solving QPP. Thus, the number of samples to be entered into the optimization problem can be substantially reduced. An important advantage of SSR-LapSVM is the safety, in the sense that the SSR is guaranteed to have the exactly same solution as solving the original LapSVM. In addition, by using Cholesky decomposition (Golub and Loan, 1996), the computational speed of inverse operation is improved. The effectiveness of our proposed method is verified by the experiments on both 3 artificial datasets and 24 real world benchmark datasets.

The main contributions of our method are summarized as follows:

(i) By our screening rule, the scale of the optimization problem could be significantly reduced. Moreover, the solution is guaranteed to be exactly same as the original problem.
(ii) Most existing screening rules only deal with one parameter problems. However, our screening rule can effectively address multiple parameter problems.
(iii) Our screening rule is independent from the solvers, then some efficient methods can be embedded into it. In this paper, the dual coordinate descent method (DCDM) is chosen as the solving algorithm.

The rest of this paper is organized as follows: Section 2 gives related works. The SSR-LapSVM is proposed in Section 3. Experiments on artificial datasets and real world benchmark datasets are conducted to verify the excellence of our SSR-LapSVM in Section 4. The last section gives the conclusion.

2. Related works

For the supervised learning, consider a binary classification problem with a training set $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, where $x_i \in R^n$ is the input and $y_i \in \{1, -1\}$ is the corresponding output. The traditional SVM searches for a separating hyper-plane $f(x) = w^T\phi(x)+b = 0$, where $w \in R^n$, $b \in R$, and $\phi(\cdot)$ is a nonlinear mapping to a higher dimensional Hilbert space. On one hand, maximizing the margin between two classes is equivalent to minimization of the regularization term $\frac{1}{2}\|w\|^2$. On the other hand, training error $L_{hinge}(x, y, f(x)) = max\{1 - yf(x), 0\}$ is minimized. Then, the following formulation is achieved:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, 2, \dots, l, \end{aligned}$$

where $C > 0$ is a trade-off parameter.

In SSL, a set of l labeled samples $\{x_i, y_i\}_{i=1}^{l}$ and a set of u unlabeled samples $\{x_i\}_{i=l+1}^{l+u}$ are given to train the model. LapSVM introduces an additional regularization term by using a graph Laplacian matrix L (Chova et al., 2008). The geometry of the data is modeled with a graph in which nodes represent both labeled and unlabeled samples connected by weights W_{ij} (Chapelle et al., 2006). Regularizing the graph follows from manifold assumption.

The decision function of LapSVM is given by $f(x^*) = \langle w, \phi(x^*)\rangle + b = \sum_{i=1}^{l+u}\alpha_i K(x_i, x^*) + b$. K is the kernel matrix formed by kernel function $K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$. The regularization term can be expressed in terms of the kernel matrix and the expansion coefficients: $\|w\|^2 = \alpha^T K\alpha$.

Finally, the formulation of LapSVM is written as:

$$\begin{aligned} \min_{\xi\in R^l,\ \alpha\in R^{l+u}} \quad & \gamma_1\alpha^T K\alpha + \gamma_2\alpha^T KLK\alpha + e^T\xi \\ \text{s.t.} \quad & y_i K(x_i, x)\alpha \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, 2, \dots, l, \end{aligned} \tag{1}$$

where ξ_i are slack variables to deal with errors in the labeled samples. Note that we let $w = [w;\ b] \in R^{n+1}$, $x_i = [x_i,\ 1] \in R^{n+1}$.

Although the LapSVM has many admirable advantages, there is a difficulty to achieve the optimal parameters. The optimal values of γ_1 and γ_2 are unknown and it needs to estimate them. Commonly used model selection strategies such as cross validation and stability selection need to solve the optimization problems over a grid of turning parameters $\{(\gamma_1^{(k)}, \gamma_2^{(k)})|, k = 1, 2, \dots, T\}$ to determine appropriate values. This procedure is usually very time consuming.

To address this issue, in this paper a SSR is proposed for accelerating LapSVM. A block diagram for the whole scheme of the proposed method is given in Fig. 1. The repetitive process of parameter selection is denoted in red. The proposed screening process is denoted in green. Our goal is to safely reduce the optimization problem during the repetitive process of parameter selection.

3. A safe screening rule for Laplacian support vector machine

In this section, the motivations for our method are given via KKT conditions. Then the feasible set is constructed via variational inequalities. The SSR is presented in the last subsection.

3.1. Motivations

KKT conditions of LapSVM are studied to find out the instances which have no effect on the solution of the optimization problem.

The following Lagrangian function is introduced to solve QPP (1):

$$\begin{aligned} L(\alpha, \xi, \beta, \eta) = {} & \gamma_1\alpha^T K\alpha + \gamma_2\alpha^T KLK\alpha + e^T\xi \\ & - \sum_{i=1}^{l}\beta_i[y_i K(x_i, x)\alpha - 1 + \xi_i] - \sum_{i=1}^{l}\eta_i\xi_i, \end{aligned} \tag{2}$$

where $\beta \geq 0$ and $\eta \geq 0$ are the Lagrangian multipliers.

In the following, α, β, η and ξ represent the feasible solutions. α^*, β^*, η^* and ξ^* denote the corresponding optimal values.

Differentiating the Lagrangian function with respect to α and ξ yields the following KKT conditions:

$$\frac{\partial L}{\partial\alpha} = 2\gamma_1 K\alpha^* + 2\gamma_2 KLK\alpha^* - \sum_{i=1}^{l}\beta_i^* y_i K(x_i, x) = \mathbf{0}, \tag{3}$$

$$\frac{\partial L}{\partial\xi_i} = 1 - \beta_i^* - \eta_i^* = 0, \tag{4}$$

$$\beta_i^*(y_i K(x_i, x)\alpha^* - 1 + \xi_i^*) = 0, \tag{5}$$

$$\eta_i^*\xi_i^* = 0, \tag{6}$$

$$\beta_i^* \geq 0,\ \eta_i^* \geq 0,\ i = 1, 2, \dots, l. \tag{7}$$

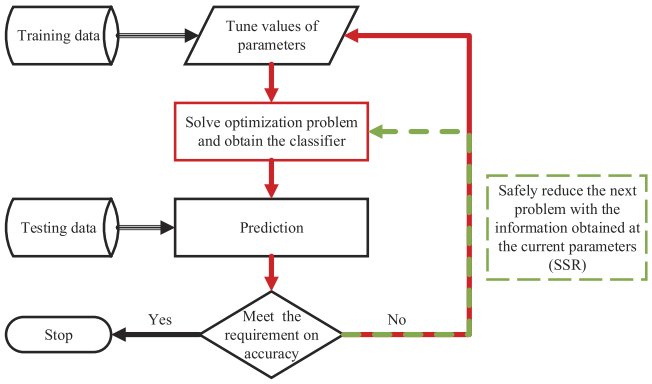

Fig. 1. The block diagram for the whole scheme of the proposed SSR-LapSVM. Our goal is to safely reduce the optimization problem in the process of parameter selection. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

From Eq. (3), α^* can be represented as:

$$\alpha^* = (2\gamma_1 I + 2\gamma_2 LK)^{-1} J^T Y \beta^*, \tag{8}$$

where $J = [I\ 0]$ is a $l \times (l+u)$ matrix with I as the $l \times l$ identity matrix and $Y = \mathrm{diag}(y_1, \ldots, y_l)$.

Let

$$\mathcal{R} = \{i : y_i K(x_i, x)\alpha^* > 1\},$$
$$\mathcal{E} = \{i : y_i K(x_i, x)\alpha^* = 1\},$$
$$\mathcal{L} = \{i : y_i K(x_i, x)\alpha^* < 1\}.$$

From Eqs. (5), (6), (7) and the constraints of QPP (1), the following condition is obtained:

$$\beta_i^* = \begin{cases} 0, & \text{if } i \in \mathcal{R}; \\ [0,\ 1], & \text{if } i \in \mathcal{E}; \\ 1, & \text{if } i \in \mathcal{L}. \end{cases} \tag{9}$$

Substituting Eqs. (4) and (8) into (2), the following QPP is derived,

$$\begin{aligned} \min_{\beta \in R^l} \quad & g(\beta) = \frac{1}{2}\beta^T Q \beta - \sum_{i=1}^{l} \beta_i \\ \text{s.t.} \quad & 0 \le \beta_i \le 1,\ i = 1, 2, \ldots, l, \end{aligned} \tag{10}$$

where

$$Q = YJK(2\gamma_1 I + 2\gamma_2 LK)^{-1} J^T Y. \tag{11}$$

From (9), if some samples are known to be members of $\mathcal{R}$ and $\mathcal{L}$, then the corresponding components of β^* can be set accordingly. So it only needs to determine the rest components of the variables. More precisely, the following Lemma 1 can be demonstrated easily.

Lemma 1. *For the problem (10), given index sets $\hat{\mathcal{R}} \subseteq \mathcal{R}$ and $\hat{\mathcal{L}} \subseteq \mathcal{L}$, we have*

1. $\beta^*_{\hat{\mathcal{R}}} = 0$ and $\beta^*_{\hat{\mathcal{R}}} = 1$.

2. Let $\hat{\mathcal{S}} = \hat{\mathcal{R}} \bigcup \hat{\mathcal{L}}$, $\mathcal{S} = \mathcal{R} \bigcup \mathcal{L}$, $\hat{\mathcal{S}}^C = \mathcal{S} \setminus \hat{\mathcal{S}}$. $l' = |\hat{\mathcal{S}}^C|$ denotes the cardinality of the set $\hat{\mathcal{S}}^C$, $Q_{11} = Q_{\hat{\mathcal{S}}^C, \hat{\mathcal{S}}^C}$, $Q_{12} = Q_{\hat{\mathcal{S}}^C, \hat{\mathcal{S}}}$ and $\hat{f} = Q_{12}\beta_{\hat{\mathcal{S}}} - e$. e is the vector of ones of appropriate dimensions.

The reduced optimization problem is as follows,

$$\begin{aligned} \min_{\hat{\beta} \in R^{l'}} \quad & \frac{1}{2}\hat{\beta}^T Q_{11}^T \hat{\beta} + \hat{f}^T \hat{\beta} \\ \text{s.t.} \quad & 0 \le \hat{\beta} \le 1. \end{aligned} \tag{12}$$

To implement LapSVM, it needs to solve the QPP (10). From Lemma 1, when $\hat{\mathcal{S}}$ is large, the computational cost for solving problem (12) can be much cheaper than solving the full problem (10). That motivates us to build the screening rule to reduce the computational cost.

3.2. Feasible set construction

To reduce the computational cost, it needs to determine the membership of $\hat{\mathcal{S}}$. The condition (9) seems to be useful. However, it is not applicable since β^* is unknown. To overcome this difficulty, a feasible region Γ such that $\beta^* \in \Gamma$ is estimated. As a result, the relaxed version of (9) is derived,

$$\begin{aligned} &\min_{\beta \in \Gamma} \ y_i K(x_i, x)\alpha(\beta) > 1 \Rightarrow \beta_i^* = 0 \Leftrightarrow i \in \mathcal{R}; \\ &\max_{\beta \in \Gamma} \ y_i K(x_i, x)\alpha(\beta) < 1 \Rightarrow \beta_i^* = 1 \Leftrightarrow i \in \mathcal{L}; \end{aligned} \tag{13}$$

where $\alpha(\beta)$ changes with β. It can be obtained from (8).

Note that (13) serves as the foundation of the proposed SSR. In the subsequent sections, the feasible region Γ including β^* is estimated. Then the SSR is proposed based on this estimation and condition (13).

The main technique for estimating the region Γ is the so called *variational inequalities*. For self-completeness, the definition is cited as follows.

Lemma 2 (*Güler, 2010*). *Let $A \subseteq R^n$ be a convex set, and let g be a Gâteaux differentiable function on an open set containing A. If x^* is a local minimizer of g on A, then*

$$\langle \nabla g(x^*),\ x - x^* \rangle \ge 0,\ \forall x \in A.$$

Via the variational inequalities, the following theorem shows that $\beta^*(\gamma_1, \gamma_2)$ can be estimated in terms of $\beta^*(\gamma_1^{(0)}, \gamma_2^{(0)})$. In the following, $\beta^*(\gamma_1^{(0)}, \gamma_2^{(0)})$ and $Q(\gamma_1^{(0)}\gamma_2^{(0)})$ are abbreviated as $\beta^{*(0)}$ and $Q^{(0)}$, respectively.

Theorem 3. *For problem (10), let $\gamma_1 \ge \gamma_1^{(0)} > 0$, $\gamma_2 \ge \gamma_2^{(0)} > 0$. Assume $GG^T = (2\gamma_1 K + 2\gamma_2 KLK)^{-1}$, $G \in R^{(l+u)\times(l+u)}$ and $Q = ZZ^T$ where $Z = YJKG$, $Z \in R^{l\times(l+u)}$. We have*

$$\|Z^T\beta^* - \frac{Z^{-1}H}{2}\beta^{*(0)}\|^2 \le \beta^{*(0)T}(\frac{1}{4}HQ^{-1}H - Q^{(0)})\beta^{*(0)}, \tag{14}$$

where $H = Q^{(0)} + Q$.

Proof. The variational inequality implies that

$$\langle \nabla g(\beta^{*(0)}),\ \beta - \beta^{*(0)} \rangle \ge 0,\ \forall \beta \in [0,1]^l; \langle \nabla g(\beta^*),\ \beta - \beta^* \rangle \ge 0,\ \forall \beta \in [0,1]^l.$$

Notice that $\beta^{*(0)}, \beta^* \in [0,1]^l$. It is easy to obtain the following result:

$$\langle \nabla g(\beta^{*(0)}),\ \beta^* - \beta^{*(0)} \rangle \ge 0; \langle \nabla g(\beta^*),\ \beta^{*(0)} - \beta^* \rangle \ge 0. \tag{15}$$

From (15), the following result is obtained:

$$\langle \nabla g(\beta^{*(0)}) - \nabla g(\beta^*),\ \beta^* - \beta^{*(0)} \rangle \ge 0. \tag{16}$$

Notice that $\nabla g(\beta) = Q\beta - e$. Plugging $\nabla g(\beta^*)$ and $\nabla g(\beta^{*(0)})$ into (16) leads to

$$\langle Q^{(0)}\beta^{*(0)} - Q\beta^*,\ \beta^* - \beta^{*(0)} \rangle \ge 0.$$

Then the following formulation is derived:

$$\beta^{*T}Q^{(0)}\beta^{*(0)} - \beta^{*T}Q\beta^* - \beta^{*(0)T}Q^{(0)}\beta^{*(0)} + \beta^{*(0)T}Q\beta^* \ge 0.$$

Assume $GG^T = (2\gamma_1 K + 2\gamma_2 KLK)^{-1}$ and $Q = ZZ^T$ where $Z = YJKG$, we can get

$$\begin{aligned} &\|Z^T\beta^* - \frac{Z^{-1}(Q^{(0)} + Q)}{2}\beta^{*(0)}\|^2 \\ &\le \frac{1}{4}\beta^{*(0)T}(Q^{(0)} + Q)^T Q^{-1}(Q^{(0)} + Q)\beta^{*(0)} - \beta^{*(0)T}Q^{(0)}\beta^{*(0)}. \end{aligned} \tag{17}$$

Let $H = Q^{(0)} + Q$, then (14) is demonstrated to be true.

3.3. A safe screening rule for LapSVM

Via Theorem 3, given $\gamma_1 \ge \gamma_1^{(0)} > 0$ and $\gamma_2 \ge \gamma_2^{(0)} > 0$, the optimal solution β^* at (γ_1, γ_2) can be estimated in terms of the solution $\beta^{*(0)}$ at $(\gamma_1^{(0)}, \gamma_2^{(0)})$. Based on this estimation, the basic screening rule for problem (10) is summarized in the following theorem:

Theorem 4 (*SSR*). *For problem (10), suppose we are given $\beta^{*(0)}$. Then, for any $\gamma_1 > \gamma_1^{(0)} > 0$, $\gamma_2 > \gamma_2^{(0)} > 0$, we have $\beta_i^* = 0$, i.e. $i \in \mathcal{R}$, if the following holds*

$$\frac{1}{2}H_{[i]}\beta^{*(0)} - Q_{ii}\sqrt{\beta^{*(0)T}(\frac{1}{4}HQ^{-1}H - Q^{(0)})\beta^{*(0)}} > 1.$$

Similarly, we have $\beta_i^* = 1$, i.e. $i \in \mathcal{L}$, if

$$\frac{1}{2}H_{[i]}\beta^{*(0)} + Q_{ii}\sqrt{\beta^{*(0)T}(\frac{1}{4}HQ^{-1}H - Q^{(0)})\beta^{*(0)}} < 1,$$

where $H_{[i]}$ represents the i-th row of matrix H, and Q_{ii} is the i-th row i-th column element of matrix Q.

Proof. From (9) and Theorem 3, when $y_i K(x_i, x)\alpha > 1$, the following result can be derived:

$$\begin{aligned}
&y_i K(x_i, x)\alpha \\
&= y_i K(x_i, x)(2\gamma_1 I + 2\gamma_2 LK)^{-1} J^T Y \beta^* \\
&= y_i K(x_i, x)(2\gamma_1 K + 2\gamma_2 KLK)^{-1} K J^T Y \beta^* \\
&= y_i K(x_i, x) G G^T K J^T Y \beta^* \\
&= y_i K(x_i, x) G Z^T \beta^* \\
&= y_i K(x_i, x) G (Z^T \beta^* - \frac{1}{2} Z^{-1} H \beta^{*(0)}) + \frac{1}{2} y_i K(x_i, x) G Z^{-1} H \beta^{*(0)} \\
&\geq -\|y_i K(x_i, x) G\| \cdot \|Z^T \beta^* - \frac{1}{2} Z^{-1} H \beta^{*(0)}\| + \frac{1}{2} y_i K(x_i, x) G Z^{-1} H \beta^{*(0)} \\
&\geq -\|y_i K(x_i, x) G\| \cdot \sqrt{\beta^{*(0)T}(\frac{1}{4}HQ^{-1}H - Q^{(0)})\beta^{*(0)}} \\
&\quad + \frac{1}{2} y_i K(x_i, x) G Z^{-1} H \beta^{*(0)} \\
&> 1.
\end{aligned}$$

Note that, the second inequality is due to Theorem 3, and the last line is due to the statement. Finally, the inequality is obtained

$$\frac{1}{2}H_{[i]}\beta^{*(0)} - Q_{ii}\sqrt{\beta^{*(0)T}(\frac{1}{4}HQ^{-1}H - Q^{(0)})\beta^{*(0)}} > 1.$$

When $y_i K(x_i, x)\alpha < 1$, the inequality (19) can be acquired similarly.

This completes the proof.

The work flow of the proposed SSR is given in Fig. 2. At first, the primal and dual problems are analyzed by Karush–Kuhn–Tucker (KKT) conditions. Then, via variational inequalities, the feasible set of the dual solution is constructed with the information obtained at the previous parameters. Combined the KKT conditions and the feasible set, the SSR is finally derived in an analytical form. Through this rule, the scale of optimization problem at (γ_1, γ_2) is substantially reduced with the information obtained at $(\gamma_1^{(0)}, \gamma_2^{(0)})$.

In applications, it needs to estimate the optimal γ_1 and γ_2 and solve the optimization problem over a grid of turning parameters $\{(\gamma_1^{(0)}, \gamma_2^{(0)}), (\gamma_1^{(1)}, \gamma_2^{(1)}), \ldots, (\gamma_1^{(T)}, \gamma_2^{(T)})\}$ to determine appropriate values. Based on this procedure, a sequential version of the proposed SSR is presented below.

Corollary 5 (*Sequential SSR*). *For problem (10), suppose given a sequence of parameters $\{(\gamma_1^{(k)}, \gamma_2^{(k)}) | \gamma_1^{(k+1)} \geq \gamma_1^{(k)} > 0$ and $\gamma_2^{(k+1)} \geq \gamma_2^{(k)} > 0$, $k = 0, \ldots, T-1\}$. Assume $\beta^{*(k)}$ is known for an arbitrary integer $1 < k < T$. Then, for $\beta^{*(k+1)}$, we have $\beta_i^{*(k+1)} = 0$, i.e. $i \in \mathcal{R}$, if the following holds*

$$-Q_{ii}^{(k+1)}\sqrt{\beta^{*(k)T}(\frac{1}{4}H^{(k+1)}(Q^{(k)})^{-1}H^{(k+1)} - Q^{(k)})\beta^{*(k)}} + \frac{1}{2}H_{[i]}^{(k+1)}\beta^{*(k)} > 1. \tag{18}$$

Similarly, we have $\beta_i^{*(k+1)} = 1$, i.e. $i \in \mathcal{L}$, if

$$Q_{ii}^{(k+1)}\sqrt{\beta^{*(k)T}(\frac{1}{4}H^{(k+1)}(Q^{(k)})^{-1}H^{(k+1)} - Q^{(k)})\beta^{*(k)}} + \frac{1}{2}H_{[i]}^{(k+1)}\beta^{*(k)} < 1. \tag{19}$$

The proposed SSR-LapSVM aims at accelerating original LapSVM. In our method, it needs to solve QPP (10) at $(\gamma_1^{(0)}, \gamma_2^{(0)})$ and the reduced problems (12) at $(\gamma_1^{(k)}, \gamma_2^{(k)})$, $k = 1, 2, \ldots, T$. Since these problems are convex quadratic programming, their corresponding solutions are guaranteed to be globally optimal. The DCDM (Hsieh et al., 2008) is chosen to solve these optimization problems. Its convergence has been fully studied by Hsieh et al. (2008). It has been proved to reach an ϵ-accurate solution in $O(\log(1/\epsilon))$ iterations. Meanwhile, via the proposed screening rule, the redundant instances in QPP (10) could be directly eliminated. Therefore, the effectiveness of the proposed methodology is guaranteed theoretically.

Note that $(2\gamma_1 I + 2\gamma_2 LK)^{-1}$ in (8) and (11) can be written as $(2\gamma_1 K + 2\gamma_2 KLK)^{-1}K^{-1}$. Thus the ordinary matrix inverse operation is transformed to a symmetric matrix inverse operation. To further improve computational efficiency, it is computed by using Cholesky decomposition (Golub and Loan, 1996). To take care of the problem due to possible ill-condition, a term ϵI is added before the inverse operation, where ϵ is a very small positive number and I is an identity matrix of appropriate dimensions. K^{-1} is calculated in initial step.

The pseudo code for SSR-LapSVM is presented in Algorithm 1, where $P^{(k)}$ is a combination of $(\gamma_1^{(k)}, \gamma_2^{(k)})$. $\mathrm{ID}^{(k)}$ denotes index of detected samples in each parameter selection step, and each element of $\beta^*_{\mathrm{ID}^{(k)}}$ is determined to be 0 or 1 before solving QPP (10).

Algorithm 1 SSR-LapSVM.

Input: $X_{\text{train}}, Y_{\text{train}}, X_{\text{test}}, Y_{\text{test}}, \gamma_1 \in R^{r_1}, \gamma_2 \in R^{r_2}$;
Output: γ_1^*, γ_2^*, ID, Accuracy;
Initialization: $step = 0$; $\mathrm{ID}^{(0)} = \text{zeros}(l, 1)$; calculate K, K^{-1}, L, $P^{(k)}, k = 0, \ldots, r_1 \times r_2 - 1$;
$\beta^{*(0)}, \alpha^{*(0)} \leftarrow$ LapSVM with $P^{(0)}$;
$f(x) \leftarrow \alpha^{*(0)}$;
$\text{Accuracy}^{(0)} \leftarrow \text{predict}(f(x), X_{\text{test}}, Y_{\text{test}})$;
Screening process:
for $k = 0$ *to* $(r_1 \times r_2 - 1)$ **do**
  $step \leftarrow step + 1$;
  $\mathrm{ID}^{(k+1)} \leftarrow$ put $\beta^{*(k)}, P^{(k)}, P^{(k+1)}$ into (18) (19);
  $\hat{\beta}^{*(k+1)} \leftarrow$ solve (12) with $P^{(k+1)}, \beta^*_{\mathrm{ID}^{(k+1)}}$;
  $\beta^{*(k+1)} \leftarrow$ combine $\hat{\beta}^{*(k+1)}$ and $\beta^*_{\mathrm{ID}^{(k+1)}}$;
  $\alpha^{*(k+1)} \leftarrow$ put $\beta^{*(k+1)}$ into (8);
  $f(x) \leftarrow \alpha^{*(k+1)}$;
  $\text{Accuracy}^{(k+1)} \leftarrow \text{predict}(f(x), X_{\text{test}}, Y_{\text{test}})$;
end for
$\gamma_1^*, \gamma_2^* \leftarrow P^*$ corresponding to max(Accuracy);
return γ_1^*, γ_2^*, ID, Accuracy

4. Numerical experiments

In this section, the numerical experiments are conducted on artificial datasets and real world datasets to demonstrate the validity of our proposed SSR-LapSVM. Artificial datasets are used to compare our proposed algorithm with traditional SVM and LapSVM. They include popular 2-circles dataset, 2-moons dataset[1] and a Gaussian distribution dataset. We further compare our proposed method with the state-of-the-art learning algorithms LapSVM, LapSVM-Newton, and LapSVM-PCG (Melacci and Belkin, 2011) on 24 benchmark datasets, such as USPS,[2] PCMAC,[3] Spambase,[4] Musk[5] and so on (Lichman, 2013; Chang and Lin, 2011). All of them are 2-class datasets except USPS. For USPS, some categories are taken out to be in 2-class version, including

[1] http://www.dii.unisi.it/~melacci/lapsvmp/.
[2] http://www.cs.nyu.edu/~roweis/data.html.
[3] http://qwone.com/~jason/20Newsgroups/.
[4] http://archive.ics.uci.edu/ml/datasets/Spambase.
[5] http://archive.ics.uci.edu/ml/datasets/Musk+%28Version+2%29.

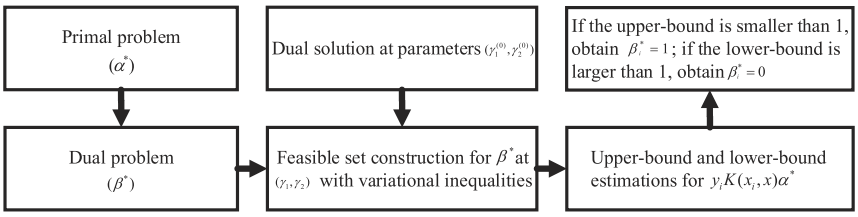

Fig. 2. The work flow of the proposed SSR. The purpose is to determine the value of β^* with the information obtained at $(\gamma_1^{(0)},\gamma_2^{(0)})$.

Fig. 3. Examples of 0, 3, 5, 6, 8 from USPS dataset.

USPS-3VS5, USPS-3VS8 and USPS-0VS6. Fig. 3 gives examples of USPS dataset.

All methods are implemented by using Matlab R2014a on Windows 7 running on a PC with system configuration Intel(R) Core(TM) 2 Duo CPU E7500 (2.93 GHz) with 8.00 GB of RAM. The package of LapSVM-Newton and LapSVM-PCG is downloaded from website.[6] The LapSVM and SSR-LapSVM are solved by DCDM.

The parameters related to build the graph Laplacian matrix (the number of neighbors and the degree) are fixed as specified by Melacci and Belkin (2011). The Laplacian matrix is computed using its normalized expression.

Gaussian kernel function $k(x_i,x_j)=exp(-\|x_i-x_j\|^2/2\sigma^2)$ is considered as it is often applied and yields great generalization performance, where σ is a kernel parameter (Taylor and Cristianini, 2004; Cristianini and Shawe-Taylor, 2000). The parameter σ in SVM is selected over the range $\{2^i|i=-5,-3\ldots,5\}$. For the SSL methods, the kernel parameter is chosen according to the priori information (Melacci and Belkin, 2011).

4.1. Artificial datasets

First, the popular two 2-D artificial datasets: 2-moons and 2-circles data are considered. There are only 2 labeled points for each class in two datasets. Parameters c, γ_1 and γ_2 in SVM, LapSVM and SSR-LapSVM are searched from set $\{2^i|i=-5,-4,\ldots,5\}$. For the LapSVM and SSR-LapSVM, the kernel parameter σ is fixed as 2^{-3} on 2-moons dataset and 2^3 on 2-circles dataset, respectively.

Illustrations of the algorithms with the optimal parameters are shown in Fig. 4. It can be observed that the supervised methods LapSVM and SSR-LapSVM have better prediction accuracy than the traditional SVM. More specifically, the SSR-LapSVM obtains the same solution as solving the LapSVM. Since the labeled samples are few, SSR-LapSVM works faster than LapSVM.

We further show that SSR-LapSVM is very effective in screening labeled samples even for largely overlapping classes. Our SSR-LapSVM is compared with the LapSVM on Gaussian distribution dataset, plotted in Fig. 5. We generate two classes of samples and each class has 500 labeled and 500 unlabeled data points. The dataset is generated from $N(0.75,-0.75,0.75^2,0.75^2)$. The blue dots represent positive class and the red dots represent the negative class.

Parameters γ_1 and γ_2 in LapSVM and SSR-LapSVM are searched from set $\{2^i|i=-5,-4,\ldots,10\}$ and $\{2^i|i=-5,-4,\ldots,5\}$, respectively, and σ is fixed as 0.125. The optimal classifier of LapSVM is presented on the Toy dataset in Fig. 5(a). Figs. 5(b), 5(c) and 5(d) are illustrations of SSR-LapSVM in different parameter selection steps. Green points represent the detected points in advance of solving QPP. The total computational time of SSR-LapSVM is 257.0591 s. By contrast, the running time of original LapSVM is 371.1248 s which is slower over 114 s than our methods.

Fig. 6(a) indicates that our SSR-LapSVM always has the same accuracy as LapSVM in each step, which also demonstrates that the process of data screening is guaranteed to have the exactly same solution as the original LapSVM.

Fig. 6(b) gives detailed ratio of detected labeled points along with the selecting parameter steps. With the increase of steps, there is a growing number of detected labeled points. Fig. 6(b) indicates that the proposed methods can identify almost all of the labeled points.

Table 1
The statistics of 24 benchmark datasets.

Dataset	#Sample	#Feature	#Training		#Testing
			Labeled	Unlabeled	
DBWorld	64	4702	26	15	13
Fertility	100	9	40	40	20
LSVT	126	310	51	50	25
PlanningRelax	146	12	73	73	36
Sonar	208	60	84	84	40
SpectHeart	267	44	107	107	53
BCIa	268	5376	107	107	54
Haberman	306	4	123	122	61
LiverDisorder	345	6	138	138	69
Ionosphere	351	34	141	140	70
Monks	432	6	173	172	87
BreastCancer569	569	30	227	227	115
BreastCancer683	683	274	273	9	136
Australian	690	14	276	275	139
Pima	768	8	308	307	153
Banknote	1372	4	549	548	275
USPS-3VS8	1532	256	50	1175	307
USPS-3VS5	1540	256	50	1182	308
PCMAC	1943	3289	200	1354	389
USPS-0VS6	2387	256	50	1859	478
Adult	3185	122	1274	1274	637
Spambase	4601	57	500	3182	919
Musk	6598	166	2640	2639	1319
Mushroom	8124	112	3250	3249	1625

4.2. Benchmark datasets

In this experiment, we compare our SSR-LapSVM with semi-supervised algorithms LapSVM, LapSVM-Newton and LapSVM-PCG in terms of prediction accuracy and computational time. These algorithms are implemented along a sequence of 40 × 40 parameter combinations of $\gamma_1,\gamma_2\in[2^{-5},2^{10}]$ equally spaced in the logarithmic scale.

The statistics of benchmark datasets are given in Table 1. Each dataset is divided into two parts: training data and testing data. The training set is composed of labeled and unlabeled data. In our experiments, 4/5 of positive samples and 4/5 of negative samples are randomly selected as the training set, in which the labels of the training data are randomly deleted with a certain proportion. The rest samples are reserved as a testing set.

[6] http://www.dii.unisi.it/~melacci/lapsvmp/.

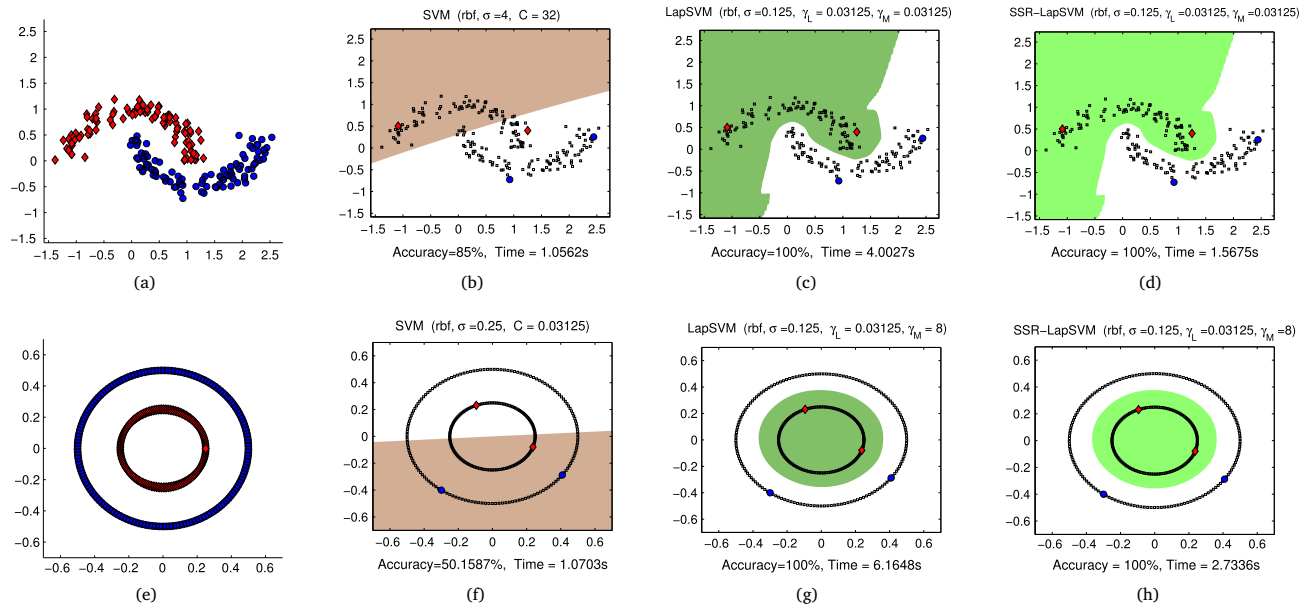

Fig. 4. Performance comparisons of SVM, LapSVM, and SSR-LapSVM on 2-moons and 2-circles datasets.

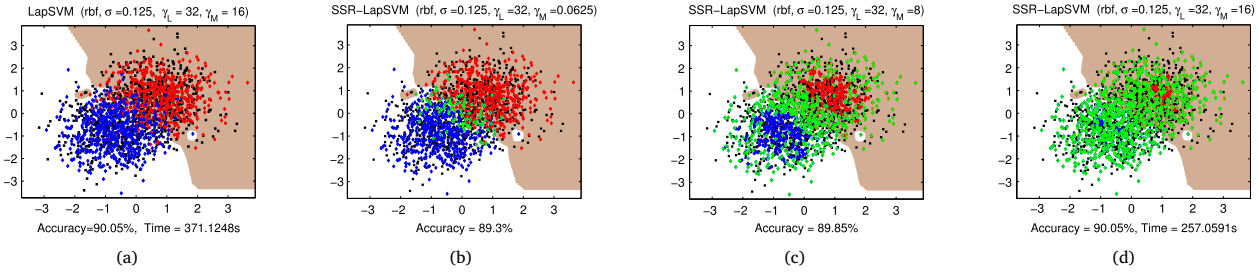

Fig. 5. Classification maps of LapSVM and SSR-LapSVM on Toy dataset. (a) LapSVM with the optimal parameters, (b) SSR-LapSVM in parameter selection step 2, (c) SSR-LapSVM in parameter selection step 9, (d) SSR-LapSVM in parameter selection step 10. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

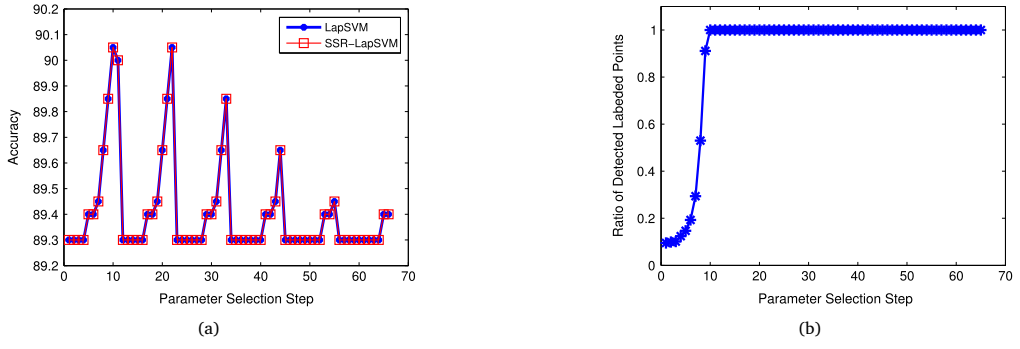

Fig. 6. Results of LapSVM and SSR-LapSVM on Toy dataset.

Take the last dataset "Mushroom" as an example. This dataset includes 8124 samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (Lichman, 2013). Each species is identified as edible (3916 positive samples) or poisonous (4208 negative samples). In our experiment, 3133 positive data (4/5 of 3916) and 3366 negative data (4/5 of 4208) are randomly collected as the training set (in total 6499 samples) and the rest samples are reserved as testing set. To make the unlabeled ratio as 50%, the labels of 3250 (50% of 6499) training data are randomly deleted.

Table 2 gives the detailed comparisons of our SSR-LapSVM and LapSVM on 24 benchmark datasets. Where "Average" denotes average time of each parameter selection step; "Initialization" represents the initialization time which is the cost for constructing matrixes; "Inverse" denotes average time of matrix inverse operation in each step; "Solver" represents the average time of solving QPP in each parameter selection step of LapSVM, and in SSR-LapSVM it contains the time of both screening process and solving QPP; "SSR Ratio(%)" is only for SSR-LapSVM, which represents the average ratio of detected samples in each

Table 2
Detailed comparisons of SSR-LapSVM and LapSVM on 24 benchmark datasets. Bold type shows the best result.

Dataset	Algorithm	Time(s)				SSR Ratio(%)	Speedup
		Average	Initialization	Inverse	Solver		
DBWorld	LapSVM	0.0225	0.0013	0.0204	0.0006		11.84
	SSR-LapSVM	**0.0019**	**0.0001**	**0.0013**	**0.0003**	79.08	
Fertility	LapSVM	0.0533	0.0016	0.0192	0.0231		16.15
	SSR-LapSVM	**0.0033**	**0.0004**	**0.0011**	**0.0012**	63.02	
LSVT	LapSVM	**0.0012**	**0.0002**	**0.0004**	**0.0004**		0.26
	SSR-LapSVM	0.0047	0.0003	0.0019	0.0018	65.34	
PlanningRelax	LapSVM	0.2722	0.0047	0.1195	0.1082		47.75
	SSR-LapSVM	**0.0057**	**0.0013**	**0.0013**	**0.0027**	68.02	
Sonar	LapSVM	0.0063	0.0013	0.0023	0.0023		2.52
	SSR-LapSVM	**0.0025**	**0.0004**	**0.0009**	**0.0008**	70.70	
SpectHeart	LapSVM	0.2727	0.0021	0.0763	0.1784		37.88
	SSR-LapSVM	**0.0072**	**0.0017**	**0.0026**	**0.0024**	68.27	
BCIa	LapSVM	0.5988	0.0045	0.1870	0.3672		72.14
	SSR-LapSVM	**0.0083**	**0.0015**	**0.0034**	**0.0029**	72.91	
Haberman	LapSVM	1.5035	0.0063	0.2068	1.2576		68.97
	SSR-LapSVM	**0.0218**	**0.0022**	**0.0044**	**0.0145**	54.54	
LiverDisorder	LapSVM	0.0139	0.0037	0.0055	0.0038		1.81
	SSR-LapSVM	**0.0077**	**0.0009**	**0.0034**	**0.0006**	73.92	
Ionosphere	LapSVM	0.4721	0.0049	0.1481	0.2966		45.39
	SSR-LapSVM	**0.0104**	**0.0019**	**0.0041**	**0.0040**	65.13	
Monks	LapSVM	0.0244	0.0064	0.0092	0.0076		2.54
	SSR-LapSVM	**0.0096**	**0.0011**	**0.0026**	**0.0030**	62.88	
BreastCancer569	LapSVM	0.6752	0.0122	0.2506	0.3811		35.54
	SSR-LapSVM	**0.0190**	**0.0036**	**0.0071**	**0.0069**	58.62	
BreastCancer683	LapSVM	0.5109	0.0113	0.2120	0.2637		19.80
	SSR-LapSVM	**0.0258**	**0.0048**	**0.0100**	**0.0092**	53.07	
Australian	LapSVM	0.8132	0.0173	0.3936	0.3678		31.64
	SSR-LapSVM	**0.0257**	**0.0047**	**0.0098**	**0.0093**	65.77	
Pima	LapSVM	0.8269	0.0156	0.3984	0.3789		27.94
	SSR-LapSVM	**0.0296**	**0.0049**	**0.0121**	**0.0104**	52.00	
Banknote	LapSVM	**0.1720**	0.0196	**0.0579**	0.0824		0.85
	SSR-LapSVM	0.2014	**0.0056**	0.1129	**0.0126**	44.93	
USPS-3VS8	LapSVM	0.3511	0.0297	0.2511	0.0503		4.73
	SSR-LapSVM	**0.0742**	**0.0094**	**0.0611**	**0.0029**	78.09	
USPS-3VS5	LapSVM	0.3168	0.0300	0.2214	0.0463		5.51
	SSR-LapSVM	**0.0575**	**0.0096**	**0.0441**	**0.0030**	78.18	
PCMAC	LapSVM	1.2427	0.1313	0.8970	0.1803		2.81
	SSR-LapSVM	**0.4423**	**0.0838**	**0.3055**	**0.0492**	69.00	
USPS-0VS6	LapSVM	0.9014	0.0713	0.7281	0.0722		3.22
	SSR-LapSVM	**0.2796**	**0.0178**	**0.2559**	**0.0044**	77.75	
Adult	LapSVM	3.5515	0.4356	0.5056	2.5181		1.19
	SSR-LapSVM	**2.9751**	**0.1745**	**0.5054**	**1.7946**	34.83	
Spambase	LapSVM	1.9868	**0.0511**	0.9296	0.9664		1.96
	SSR-LapSVM	**1.0155**	0.0557	**0.8788**	**0.0554**	66.61	
Musk	LapSVM	17.6137	0.3324	3.5774	13.1059		1.07
	SSR-LapSVM	**16.5305**	**0.1081**	**3.0824**	**12.7604**	20.04	
Mushroom	LapSVM	20.3176	0.3588	6.5672	12.3418		2.36
	SSR-LapSVM	**8.5976**	**0.0119**	**5.5649**	**2.0454**	42.21	

step. "Speedup" denotes

$$\text{Speedup} = \frac{\text{Time of SSR-LapSVM}}{\text{Time of LapSVM}}.$$

From the columns of "Average" and "Speedup" in Table 2, the average speed of each step of SSR-LapSVM is many times faster than that of LapSVM. From the column of "Solver", we observe that the solver of proposed SSR-LapSVM is obviously faster than LapSVM. It indicates that SSR truly contributes to improving the speed of solving QPP. From the column of "Inverse", we find that our SSR-LapSVM runs faster than LapSVM in the inverse operation. It indicates that the fast inverse operation technique indeed works efficiently. From the column of "SSR Ratio", we see that SSR-LapSVM can generally detect most of the label samples before solving QPP. By contrast, the original LapSVM does not have this function.

Table 3 shows the performance comparisons of four algorithms on 24 benchmark datasets. Where "Accuracy(%)" represents the prediction accuracy with the optimal values of parameters, and "Time(s)" denotes its corresponding time.

From Table 3, we find that the accuracy of our SSR-LapSVM is almost the same as original LapSVM on each dataset. It demonstrates that SSR-LapSVM is truly a safe algorithm to guarantee the same solution as solving the original LapSVM. By contrast, LapSVM-Newton and LapSVM-PCG do not have this property. In terms of running time, SSR-LapSVM and LapSVM-PGC are generally faster than LapSVM-Newton and LapSVM.

In general, our proposed SSR-LapSVM could greatly accelerate the LapSVM. Moreover, it guarantees to obtain the same predication accuracy as the original LapSVM.

5. Conclusion

In this paper, an efficient SSR for LapSVM is proposed via variational inequalities. It can greatly reduce the number of training samples and effectively improve the computational speed. Moreover the same solution can be achieved as solving the original problem, so it is safe. In comparison with the existing DVI-SVM, it effectively addresses the problem of multiple parameters. Experiments on both 3 artificial datasets and 24 real world benchmark datasets demonstrate its feasibility and validity. The SSR only screens the labeled samples, how to screen the unlabeled instances is our further work.

Table 3
Performance comparisons of four algorithms on 24 benchmark datasets. Bold type shows the best result.

Dataset	LapSVM-Newton		LapSVM-PCG		LapSVM		SSR-LapSVM	
	Accuracy(%)	Time(s)	Accuracy(%)	Time(s)	Accuracy(%)	Time(s)	Accuracy(%)	Time(s)
DBWorld	**61.54**	0.0044	**61.54**	0.0026	**61.54**	0.0073	**61.54**	**0.0022**
Fertility	90.00	0.0043	90.00	0.0029	**95.00**	0.0204	**95.00**	**0.0020**
LSVT	**88.00**	0.0041	**88.00**	0.0034	**88.00**	**0.0013**	**88.00**	0.0065
PlanningRelax	**72.22**	0.0058	**72.22**	**0.0032**	**72.22**	0.0231	**72.22**	0.0101
Sonar	70.00	0.0062	75.00	**0.0042**	**100.00**	0.0089	**100.00**	0.0059
SpectHeart	**86.79**	0.0074	**86.79**	**0.0061**	**86.79**	0.0408	**86.79**	0.0073
BCIa	72.22	0.0550	72.22	0.0106	**77.78**	0.8034	**77.78**	**0.0081**
Haberman	**80.33**	0.0089	**80.33**	**0.0074**	**80.33**	2.7046	**80.33**	0.0227
LiverDisorder	**71.01**	**0.0041**	**71.01**	0.0077	**71.01**	0.0204	**71.01**	0.0063
Ionosphere	88.57	0.0117	88.57	**0.0071**	**90.00**	0.0101	**90.00**	0.0110
Monks	**89.66**	**0.0058**	**89.66**	0.0224	82.76	0.0223	82.76	0.0096
BreastCancer569	94.78	0.2065	93.91	**0.0206**	**98.26**	0.5041	**98.26**	0.0861
BreastCancer683	97.79	0.2249	**98.53**	0.0266	**98.53**	0.2070	**98.53**	**0.0234**
Australian	64.03	0.1516	64.03	**0.0225**	**87.77**	0.0900	**87.77**	0.0311
Pima	75.82	0.0383	73.86	**0.0358**	**76.47**	0.0409	**76.47**	0.0362
Banknote	**100.00**	**0.1637**	**100.00**	0.2107	**100.00**	0.1920	**100.00**	0.1974
USPS-3VS8	99.02	0.2420	99.02	0.2467	**100.00**	0.1127	**100.00**	**0.0797**
USPS-3VS5	95.45	0.2388	95.45	0.2513	**100.00**	0.2227	**100.00**	**0.0536**
PCMAC	83.03	0.7980	83.03	0.8469	**89.46**	0.4310	**89.46**	**0.3963**
USPS-0VS6	99.73	**0.5235**	99.70	0.5771	**100.00**	0.4131	**100.00**	**0.2206**
Adult	**84.93**	2.3286	**84.93**	**2.1533**	84.30	3.4354	84.30	2.1750
Spambase	78.56	2.6920	78.56	2.6847	**92.06**	1.8951	**92.06**	**1.0294**
Musk	95.60	**6.1524**	95.60	6.1611	**98.64**	35.8633	**98.64**	31.9621
Mushroom	**100.00**	86.1518	**100.00**	179.4507	**100.00**	21.6847	**100.00**	**7.8820**

Acknowledgments

The authors would like to thank the reviewers for the helpful comments and suggestions, which have improved the presentation. This work was supported in part by National Natural Science Foundation of China (No. 11671010) and Beijing Natural Science Foundation (No. 4172035).

References

Altınel, B, Ganiz, M.C., Dirib, B., 2017. Instance labeling in semi-supervised learning with meaning values of words. Eng. Appl. Artif. Intell. 62, 152–163.
Belkin, M., Niyogi, P., Sindhwani, V., 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. 7, 2399–2434.
Chang, C.C., Lin, C.J., 2011. LIBSVM: A library for support vector machines. 2, 27:1–27:27 Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.
Chapelle, O., Schölkopf, B., Zien, A., 2006. Semi-Supervised Learning. MIT Press, Cambridge, MA.
Chapelle, O., Sindhwani, V., Keerthi, S.S., 2008. Optimization techniques for semi-supervised support vector machines. J. Mach. Learn. Res. 9, 203–233.
Chova, L., Valls, G., Mari, J., Calpe, J., 2008. Semi-supervised image classification with laplacian support vector machines. IEEE Geosci. Remote Sens. Lett. 5, 336–340.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297.
Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge, UK.
Golub, G., Loan, C., 1996. Matrix Computations. The Johns Hopkins University Press, Baltimore and London.
Güler, , 2010. Matrix Computations. Springer, New York.
Hastie, T., Rosset, S., Tibshirani, R., Zhu, J., 2004. The entire regularization path for the support vector machine. J. Mach. Learn. Res. 5, 1391–1415.
Hsieh, C., Chang, K., Lin, C., Keerthi, S., Sundararajan, S., 2008. A dual coordinate descent method for large-scale linear SVM, In: Proceedings of the 25th international conference on Machine learning, Helsinki, Finland, pp. 408–415.
Hu, L., Lu, S., Wang, X., 2013. A new and informative active learning approach for support vector machine. Inform. Sci. 244, 142–160.
Jayadeva, , Khemchandani, R., Chandra, S., 2007. Twin support vector machines for pattern classification. IEEE Trans. Neural Netw. Learn. Syst. 29, 905–910.
Li, Y., Kwok, J., Zhou, Z., 2010. Cost-sensitive semi-supervised support vector machine, In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI'10, Atlanta, GA, pp. 500–505.
Lichman, M., 2013. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences . http://archive.ics.uci.edu/ml.
Liu, J., Zhao, Z., Wang, Z., Wang, J., Ye, J., 2014. Safe screening with variational inequalities and its application to lasso, In: Proceedings of the 31st International Conference on Machine Learning, Beijing, China, pp. 289–297.
Melacci, S., Belkin, M., 2011. Laplacian support vector machines trained in the primal. J. Mach. Learn. Res. 12, 1149–1184.
Ogawa, K., Suzuki, Y., Takeuchi, I., 2013. Safe screening of non-support vectors in pathwise SVM computation. J. Mach. Learn. Res. 28, 1382–1390.
Pan, X., Yang, Z., Xu, Y., Wang, L., 2017. Safe screening rules for accelerating twin support vector machine classification. IEEE Trans. Neural Netw. Learn. Syst. 1–12. http://dx.doi.org/10.1109/TNNLS.2017.2688182.
Platt, John, C., 1999. Fast training of support vector machines using sequential minimal optimization, In: Advances in Kernel Methods-support Vector Learning, Cambridge, MA, pp. 185–208.
Qi, Z., Tian, Y., Shi, Y., 2012. Laplacian twin support vector machine for semi-supervised classification. Neural Netw. 35, 46–53.
Qi, Z., Tian, Y., Shi, Y., 2014. Successive overrelaxation for Laplacian support vector machine. IEEE Trans. Neural Netw. Learn. Syst. 26, 674–683.
Steinwart, I., Christmann, A., 2008. Support Vector Machines. Springer Verlag, New York.
Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge University Press, New York.
Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.
Wang, J., Wonka, P., Ye, J., 2014. Scaling SVM and least absolute deviations via exact data reduction. J. Mach. Learn. Res. 32, 523–531.
Wang, J., Wonka, P., Ye, J., 2015. Lasso screening rules via dual polytope projection. J. Mach. Learn. Res. 16, 1063–1101.
Xu, Y., 2017. Maximum margin of twin spheres support vector machine for imbalanced data classification. IEEE Trans. Cybern. 47, 1540–1550.
Xu, Y., Yang, Z., Pan, X., 2017. A novel twin support-vector machine with pinball loss. IEEE Trans. Neural Netw. Learn. Syst. 28, 359–370.
Xu, Y., Yang, Z., Zhang, Y., Pan, X., Wang, L., 2016. A maximum margin and minimum volume hyper-spheres machine with pinball loss for imbalanced data classification. Knowl.-based Syst. 95, 75–85.
Yang, L., Wang, L., Gao, Y., Sun, Q., Zhao, T., 2014. A convex relaxation framework for a class of semi-supervised learning methods and its application in pattern recognition. Eng. Appl. Artif. Intell. 35, 335–344.
Yang, T., Wang, J., Sun, Q., Hibar, D., Jahanshad, N., Liu, L., Wang, Y., Zhan, L., Thompson, P., Ye, J., 2015. Detecting genetic risk factors for alzheimers disease in whole genome sequence data via lasso screening, In: Proceedings of 12th IEEE International Symposium on Biomedical Imaging, New York City, NY, United States, pp. 985–989.
Yang, Z., Xu, Y., 2016. Laplacian twin parametric-margin support vector machine for semi-supervised classification. Neurocomputing 171, 325–334.
Zhu, X., 2005. Semi-Supervised Learning with Graphs. Ph.D Thesis. School of Computer Science, Language Technologies Institute, Carnegie Mellon University, America.
Zhu, X., 2008. Semi-Supervised Learning Literature Survey Technical Report 1530. Computer Sciences, University of Wisconsin-Madison.