Pattern Recognition 46 (2013) 2531–2537

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

An iterative SVM approach to feature selection and classification in high-dimensional datasets

Dehua Liu, Hui Qian*, Guang Dai, Zhihua Zhang

College of Computer Science & Technology, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 11 June 2012
Received in revised form
31 January 2013
Accepted 5 February 2013
Available online 19 February 2013

Keywords:
Feature selection
SVM
DrSVM
Sparse learning

ABSTRACT

Support vector machine (SVM) is the state-of-the-art classification method, and the doubly regularized SVM (DrSVM) is an important extension based on the elastic net penalty. DrSVM has been successfully applied in handling variable selection while retaining (or discarding) correlated variables. However, it is challenging to solve this model. In this paper we develop an iterative ℓ_2-SVM approach to implement DrSVM over high-dimensional datasets. Our approach can significantly reduce the computation complexity. Moreover, the corresponding algorithms have global convergence property. Empirical results over the simulated and real-world gene datasets are encouraging.

1. Introduction

In this paper we are concerned with a binary classification problem. Assume that we are given a training data set $\mathcal{X}=\{(\mathbf{x}_i,y_i),i=1,\ldots,n\}$, where $\mathbf{x}_i\in\mathbb{R}^p$ is the input instance and $y_i\in\{-1,1\}$ is the corresponding label. This problem tends to seek a decision function so that we can predict the labels of test input instances.

Support vector machine (SVM) is among the best off-the-shelf supervised learning algorithm. It produces a hyperplane that partition the input datasets into two categories. Particularly, for a binary classification problem, one considers the optimization problem as follows:

$$\begin{aligned}\min_{\mathbf{w},b,\xi}\quad & \lambda\|\mathbf{w}\|_q^q+\sum_{i=1}^{n}\xi_i\\ \text{s.t.}\quad & y_i(\mathbf{x}_i^T\mathbf{w}+b)\geq 1-\xi_i,\\ & \xi_i\geq 0,\ i=1,\ldots,n,\end{aligned}$$

where $\mathbf{w}=(w_1,\ldots,w_p)^T$ is the vector of regression coefficients, $\xi=(\xi_1,\ldots,\xi_n)^T$ is the vector of slack variables, and $\lambda>0$ is a regularization parameter (or tuning parameter). Note that in the problem, the original input vector $\mathbf{x}$ can be replaced by $\phi(\mathbf{x})$, where ϕ is a feature map that transforms $\mathbf{x}$ into high-dimensional feature space.

The classical SVM arises when $q=2$ and can be solved by transforming it to a quadratic optimization problem. In this case, a direct merit is that SVM can be extended easily to the kernel framework. This formulation makes it unnecessary to find the explicit feature map ϕ and the coefficient vector $\mathbf{w}$.

However, selecting the true underlying original features is fundamental in some applications such as the microarray data analysis. In such cases, the number of instances is usually small but the dimension is very high.

There are many methods for feature selection under the SVM framework in the literature. For example, Rakotomamonjy [1] investigated the efficiency of feature selection criteria that derived from SVM. They suggested to select several best features based on the ranking criterion. In [2,3] the authors employed a scalar factor to assess the relevance of features.

There is another popular approach for achieving the purpose of feature selection in high-dimensional space, the use of sparsity-inducing penalties [4]. This approach usually has much simpler model and lower computational complexity compared to the methods mentioned above.

An intuitive and natural approach is to impose the ℓ_0-norm penalty to $\mathbf{w}$, i.e., $\|\mathbf{w}\|_0=\text{card}\{w_i|w_i\neq 0\}$. However, the resulting problem is typically NP hard, it is required to use tractable functions to approximate $\|\mathbf{w}\|_0^0$, e.g., a negative exponential function $\sum_{i=1}^{p}(1-\epsilon^{-\alpha v_i})$ with $|w_i|\leq v_i$ [5], or a nonconvex logarithmic function $\sum_{i=1}^{p}\ln(\epsilon+|w_i|)$ [6], etc.

A sparse SVM model using the ℓ_1-norm penalty is another choice for handling feature selection and classification simultaneously, called ℓ_1-SVM [7–9]. The ℓ_1-SVM has some advantages over the classical SVM when there are redundant noise features.

* Corresponding author. Tel.: +86 13758246168.
E-mail addresses: dehualiu0427@gmail.com (D. Liu), qianhui@zju.edu.cn (H. Qian), daiguang116@gmail.com (G. Dai), zhzhang@cs.zju.edu.cn (Z. Zhang).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.02.007

Unfortunately, the ℓ_1-SVM cannot select correlated features simultaneously. In order to further overcome this limitation, Wang et al. [10] proposed a doubly regularized SVM (DrSVM) by employing the elastic net penalty [11], due to its ability in capturing group effect. DrSVM has also appeared in the framework of knowledge based SVM model [12]. It automatically performs feature selection and encourages highly correlated features to be selected (or removed) together. Formally, it is defined as

$$\begin{aligned}&\min_{\mathbf{w},b,\xi} \quad \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\lambda_1\|\mathbf{w}\|_1+\frac{1}{n}\sum_{i=1}^{n}\xi_i,\\ &\quad \text{s.t.} \quad y_i(\mathbf{x}_i^T\mathbf{w}+b)\ge 1-\xi_i, \quad \xi_i\ge 0, i=1,\ldots,n,\end{aligned} \tag{1}$$

where both λ_1 and λ_2 are the regularization parameters. DrSVM can also be equivalently formulated as the minimization of the hinge loss function plus the elastic net penalty. That is

$$\min_{\mathbf{w},b} \quad \frac{1}{n}\sum_{i=1}^{n}[1-y_i(\mathbf{x}_i^T\mathbf{w}+b)]_+ +\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\lambda_1\|\mathbf{w}\|_1, \tag{2}$$

where $[z]_+=\max\{z,0\}$. $[1-y_i(\mathbf{x}_i^T\mathbf{w}+b)]_+$ is the hinge loss function for SVMs.

The ℓ_1-norm penalty makes this model have the feature selection property. Theorem 1 in [10] tells us that the ℓ_2-norm penalty makes it enjoy the group effect property. That is, the lager λ_2 corresponds to more obvious group effect.

Since both the hinge loss and ℓ_1-norm are not differentiable, Wang et al. [13] applied a huberized hinge loss to replace the original one, developing a so-called hybrid huberized SVM (HHSVM). As well, they devised an algorithm to calculate a whole solution path for HHSVM. Since the algorithm needs to track disappearance of features along a regularization path, its implementation is not easy. In fact, the nondifferentiability of hinge loss and the ℓ_1-norm is not an obstacle for solving the DrSVM. Recently, Ye et al. [14] proposed an alternating direction method of multipliers (ADMM) to efficiently solve DrSVM. The main idea is to re-express the problem (1) as an equivalent form

$$\begin{aligned}&\min_{\mathbf{w},b,\mathbf{u}} \quad \frac{1}{n}\sum_{i=1}^{n}[\eta_i]_+ +\lambda_1\|\mathbf{u}\|_1+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2\\ &\quad \text{s.t.} \quad \boldsymbol{\eta}=\mathbf{1}-\mathbf{Y}(\mathbf{X}\mathbf{w}+b\mathbf{1}), \quad \mathbf{w}=\mathbf{u},\end{aligned} \tag{3}$$

where $\boldsymbol{\eta}=(\eta_1,\ldots,\eta_n)^T$, $\mathbf{1}$ is the $n\times 1$ vector of ones, and $\mathbf{Y}$ is the $n\times n$ diagonal matrix with y_i on the ith diagonal element. $\mathbf{X}$ is a matrix of size $n\times p$, where each row of $\mathbf{X}$ is an input instance vector. Hence, the corresponding Lagrangian function of the problem (3) can be further expressed as follows:

$$\begin{aligned}\mathcal{L}_0(\mathbf{w},b,\boldsymbol{\eta},\mathbf{u},\mathbf{c},\mathbf{v})=&\frac{1}{n}\sum_{i=1}^{n}(\eta_i)_+ +\lambda_1\|\mathbf{u}\|_1+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2\\ &+\langle\mathbf{c},\mathbf{1}-\mathbf{Y}(\mathbf{X}\mathbf{w}+b\mathbf{1})-\boldsymbol{\eta}\rangle+\langle\mathbf{v},\mathbf{w}-\mathbf{u}\rangle,\end{aligned}$$

where $\mathbf{c}\in\mathbb{R}^n$ and $\mathbf{v}\in\mathbb{R}^p$ are the dual variables for the constraints. Then, one solves the following augmented Lagrangian function:

$$\mathcal{L}(\mathbf{w},b,\boldsymbol{\eta},\mathbf{u},\mathbf{c},\mathbf{v})=\mathcal{L}_0(\mathbf{w},b,\boldsymbol{\eta},\mathbf{u},\mathbf{c},\mathbf{v})+\frac{\rho_1}{2}\|\mathbf{1}-\mathbf{Y}(\mathbf{X}\mathbf{w}+b\mathbf{1})-\boldsymbol{\eta}\|_2^2+\frac{\rho_2}{2}\|\mathbf{w}-\mathbf{u}\|_2^2.$$

The corresponding solution can be calculated by an iterative optimization procedure (see the details in [14]). However, in each kth iteration, the high computational complexity is involved by solving $(\mathbf{w},b)$ with the other parameters fixed. That is

$$\begin{aligned}(\mathbf{w},b)=\operatorname*{argmin}_{\mathbf{w},b}\Big\{&\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\langle\mathbf{v}^{(k)},\mathbf{w}-\mathbf{u}^{(k)}\rangle+\langle\mathbf{c}^{(k)},\mathbf{1}-\mathbf{Y}(\mathbf{X}\mathbf{w}+b\mathbf{1})-\boldsymbol{\eta}^{(k)}\rangle\\ &+\frac{\rho_1}{2}\|\mathbf{1}-\mathbf{Y}(\mathbf{X}\mathbf{w}+b\mathbf{1})-\boldsymbol{\eta}^{(k)}\|_2^2+\frac{\rho_2}{2}\|\mathbf{w}-\mathbf{u}^{(k)}\|_2^2\Big\}.\end{aligned} \tag{4}$$

Optimizing function in (4) is equivalent to solving a linear equation of the form $(\mathbf{I}_{p+1}+\mathbf{A})\mathbf{x}=\mathbf{f}$, where $\mathbf{I}_m$ is the $m\times m$ identical matrix, $\mathbf{A}$ is a $(p+1)\times(p+1)$ positive semidefinite matrix with rank at most $(n+1)$, and $\mathbf{f}$ is a $(p+1)\times 1$ vector. Hence, it can be solved at most $n+2$ step by a conjugate gradient method [15], i.e., the complexity of this step is $O((n+2)p^2)$.

However, according to the complexity of solving $(\mathbf{w},b)$ at every iteration, the total running time still increases fast if p have a large increase.

In this paper, we discuss the large p and small n problems. We find that by reformulating the problem and using the ADMM technique, DrSVM can be optimized by solving a classical SVM at every iteration. Our formulation is different from [14] because their frameworks can only be solved in the primal space. Our new formulation can be easily solved in the dual space. As a result, the large p problem is transformed into a sequence of small n problems. Thus the computational time does not seriously depend on p any more. Moreover, our algorithms converge to the global optimal solution.

The rest parts of the paper are organized as follows. In Section 2, we derive three efficient algorithms to calculate the solution of high-dimensional DrSVM problems. Then, we present some experimental results in Section 3 to demonstrate the efficiency and effectiveness of our algorithms. Finally, the conclusion is made in Section 4.

2. Algorithms and convergence analysis

In this section we develop the efficient algorithms for DrSVM in high-dimensional datasets and then present convergence analysis for the global optimal solution.

2.1. Efficient algorithms for DrSVM

The formulation in (2) can be rewritten as

$$\begin{aligned}&\min_{\mathbf{w},b} \quad \sum_{i=1}^{n}[1-y_i(\mathbf{x}_i^T\mathbf{w}+b)]_+ +\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\lambda_1\|\mathbf{u}\|_1\\ &\quad \text{s.t.} \quad \mathbf{w}=\mathbf{u}.\end{aligned} \tag{5}$$

The corresponding Lagrangian function is then

$$\begin{aligned}L_0(\mathbf{w},b,\mathbf{u},\boldsymbol{\mu})=&\sum_{i=1}^{n}[1-y_i(\mathbf{x}_i^T\mathbf{w}+b)]_+ +\lambda_1\|\mathbf{u}\|_1\\ &+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\boldsymbol{\mu}^T(\mathbf{w}-\mathbf{u}).\end{aligned} \tag{6}$$

Furthermore, the augmented Lagrangian is

$$\begin{aligned}L(\mathbf{w},b,\mathbf{u},\boldsymbol{\mu})=&\sum_{i=1}^{n}[1-y_i(\mathbf{x}_i^T\mathbf{w}+b)]_+ +\lambda_1\|\mathbf{u}\|_1\\ &+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\boldsymbol{\mu}^T(\mathbf{w}-\mathbf{u})+\frac{\rho}{2}\|\mathbf{w}-\mathbf{u}\|_2^2.\end{aligned} \tag{7}$$

Let $f_1(\mathbf{w},b)=\sum_{i=1}^{n}[1-y_i(\mathbf{x}_i^T\mathbf{w}+b)]_+ +(\lambda_2/2)\|\mathbf{w}\|_2^2$ and $f_2(\mathbf{u})=\lambda_1\|\mathbf{u}\|_1$. Then, we can employ the method of multipliers to solve the optimization problem (5), it consists of the following two steps at every iteration:

$$(\mathbf{w}_*^{(k+1)},b_*^{(k+1)},\mathbf{u}_*^{(k+1)})=\operatorname*{argmin}_{\mathbf{w},b,\mathbf{u}}L(\mathbf{w},b,\mathbf{u},\boldsymbol{\mu}_*^{(k)}), \tag{8}$$

and

$$\boldsymbol{\mu}_*^{(k+1)}=\rho(\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)})+\boldsymbol{\mu}_*^{(k)}. \tag{9}$$

As we see, for high-dimensional problems, it is not efficient to calculate the optimization problem within the primal step (8). We now attempt to overcome this problem. Let $\xi_i=[1-y_i(\mathbf{x}_i^T\mathbf{w}+b)]_+$ and $\boldsymbol{\xi}=(\xi_1,\ldots,\xi_n)^T$. Then the optimization problem (8) is transformed into

$$\underset{\xi,b,\mathbf{w},\mathbf{u}}{\operatorname{argmin}}\left\{\mathbf{1}^T\xi+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\lambda_1\|\mathbf{u}\|_1+\delta(\mathbf{w},b,\xi|\mathbf{Y}(\mathbf{X}\mathbf{w}+b\mathbf{1}\geq\mathbf{1}-\xi)\right.$$
$$\left.+\delta(\xi|\xi\geq\mathbf{0})+\boldsymbol{\mu}_*^{(k)T}(\mathbf{w}-\mathbf{u})+\frac{\rho}{2}\|\mathbf{w}-\mathbf{u}\|_2^2\right\}, \qquad (10)$$

where $\delta(x|A)=0$ if $x\in A$, otherwise $\delta(x|A)=\infty$.

Unfortunately, it is not easy to calculate $(\mathbf{w},\xi,b,\mathbf{u})$ in the problem (10) simultaneously. Alternatively, the problem (10) can be iteratively solved via the two optimization steps as follows:

$$(\mathbf{w}_{j+1}^{(k+1)},b_{j+1}^{(k+1)},\xi_{j+1}^{(k+1)})=\underset{\mathbf{w},b,\xi}{\operatorname{argmin}}\left\{\mathbf{1}^T\xi+\frac{\lambda_2+\rho}{2}\|\mathbf{w}\|_2^2\right.$$
$$+(\boldsymbol{\mu}_*^{(k)}-\rho\mathbf{u}_j^{(k+1)})^T\mathbf{w}+\delta(\mathbf{w},b,\xi|\mathbf{Y}(\mathbf{X}\mathbf{w}+\mathbf{1}b)$$
$$\left.\geq\mathbf{1}-\xi)+\delta(\xi|\xi\geq\mathbf{0})\right\}, \qquad (11)$$

and

$$\mathbf{u}_{j+1}^{(k+1)}=\underset{\mathbf{u}}{\operatorname{argmin}}\left\{\lambda_1\|\mathbf{u}\|_1+\frac{\rho}{2}\left\|\mathbf{u}-\left(\mathbf{w}_{j+1}^{(k+1)}+\frac{\boldsymbol{\mu}_*^{(k)}}{\rho}\right)\right\|_2^2\right\}. \qquad (12)$$

In addition, (11) is equivalent to the following optimization problem:

$$\min_{\mathbf{w},b,\xi}\quad \mathbf{1}^T\xi+\frac{\lambda_2+\rho}{2}\|\mathbf{w}\|_2^2+\mathbf{w}^T(\boldsymbol{\mu}_*^{(k)}-\rho\mathbf{u}_j^{(k+1)})$$
$$\text{s.t.}\quad \mathbf{Y}(\mathbf{X}\mathbf{w}+b\mathbf{1})\geq\mathbf{1}-\xi,\quad \xi\geq\mathbf{0}, \qquad (13)$$

For high-dimensional problems with $p>n$, (13) can be efficiently solved by using its dual problem in the following form:

$$\min_{\mathbf{a}}\quad \frac{\mathbf{a}^T\mathbf{Y}\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{a}}{2(\lambda_2+\rho)}+\left(\frac{\mathbf{Y}\mathbf{X}(\rho\mathbf{u}_j^{(k+1)}-\boldsymbol{\mu}_*^{(k)})}{\lambda_2+\rho}-\mathbf{1}\right)^T\mathbf{a}$$
$$\text{s.t.}\quad \mathbf{a}^T\mathbf{y}=\mathbf{0}\quad\text{and}\quad \mathbf{0}\leq\mathbf{a}\leq\mathbf{1}, \qquad (14)$$

where $\mathbf{a}=(a_1,\ldots,a_n)^T$. Now we find that it shares the same form as the quadratic programming (QP) in classical SVM. Subsequently, based on the solution $\mathbf{a}_{j+1}^{(k+1)}=((a_1)_{j+1}^{(k+1)},\ldots,(a_n)_{j+1}^{(k+1)})^T$ for (14), we can calculate the solution $\mathbf{w}_{j+1}^{(k+1)}$ for (11) as follows:

$$\mathbf{w}_{j+1}^{(k+1)}=\frac{\mathbf{X}^T\mathbf{Y}\mathbf{a}_{j+1}^{(k+1)}+\rho\mathbf{u}_j^{(k+1)}-\boldsymbol{\mu}_*^{(k)}}{\lambda_2+\rho}.$$

Note that $b_{j+1}^{(k+1)}$ for (11) must satisfy $y_i(\mathbf{x}_i^T\mathbf{w}_{j+1}^{(k+1)}+b_{j+1}^{(k+1)})=1$ for such i that $0<(a_i)_{j+1}^{(k+1)}<1$. As for (12), it is easily obtained that

$$\mathbf{u}_{j+1}^{(k+1)}=\operatorname{sign}(\rho\mathbf{w}_{j+1}^{(k+1)}+\boldsymbol{\mu}_*^{(k)})\max\{0,|\rho\mathbf{w}_{j+1}^{(k+1)}+\boldsymbol{\mu}_*^{(k)}-\lambda_1\mathbf{1}|\}/\rho.$$

Here, ρ can either be fixed or increased monotonically.

Algorithm 1. Iterative support vector machine algorithm 1 (ISVM1).

```
Input: $\mathcal{X}=\{(\mathbf{x}_i,y_i)\}_{i=1}^n$, $\lambda_1$, $\lambda_2$, $\rho^{(0)}$, $\alpha$, $k=0$, $\mathbf{w}=\mathbf{w}_*^{(0)}$,
    $\mathbf{u}=\mathbf{u}_*^{(0)}$, and $\boldsymbol{\mu}=\boldsymbol{\mu}_*^{(0)}$.
repeat
  Set $j=0$, $\mathbf{w}_0^{(k+1)}=\mathbf{w}_*^{(k)}$ and $\mathbf{u}_0^{(k+1)}=\mathbf{u}_*^{(k)}$;
  while not convergence do
    Calculate $\mathbf{a}_{j+1}^{(k+1)}$:
      $\underset{\mathbf{a}}{\operatorname{argmin}}\ \frac{1}{2(\lambda_2+\rho^{(k)})}\mathbf{a}^T\mathbf{Y}\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{a}+\left(\frac{\mathbf{Y}\mathbf{X}(\rho^{(k)}\mathbf{u}_j^{(k+1)}-\boldsymbol{\mu}_*^{(k)})}{\lambda_2+\rho^{(k)}}-\mathbf{1}\right)^T\mathbf{a}$
      subject to  $\mathbf{a}^T\mathbf{y}=\mathbf{0}$, and $\mathbf{0}\leq\mathbf{a}\leq\mathbf{1}$;
    Calculate $\mathbf{w}_{j+1}^{(k+1)}=\frac{\mathbf{X}^T\mathbf{Y}\mathbf{a}_{j+1}^{(k+1)}+\rho^{(k)}\mathbf{u}_j^{(k+1)}-\boldsymbol{\mu}_*^{(k)}}{\lambda_2+\rho^{(k)}}$;

    Calculate
      $\mathbf{u}_{j+1}^{(k+1)}=\operatorname{sign}(\rho^{(k)}\mathbf{w}_{j+1}^{(k+1)}+\boldsymbol{\mu}_*^{(k)})\max\{0,|\rho^{(k)}\mathbf{w}_{j+1}^{(k+1)}+\boldsymbol{\mu}_*^{(k)}-\lambda_1\mathbf{1}|\}/\rho^{(k)}$;
    $j=j+1$;
  end while
  Set $\mathbf{w}_*^{(k+1)}=\mathbf{w}_j^{(k+1)}$, $\mathbf{u}_*^{(k+1)}=\mathbf{u}_j^{(k+1)}$ and $\mathbf{a}_*^{(k+1)}=\mathbf{a}_j^{(k+1)}$;
  Calculate $b_*^{(k+1)}=\frac{1}{N_{\mathcal{M}}}\sum_{i\in\mathcal{M}}(y_i-\mathbf{x}_i^T\mathbf{w}_*^{(k+1)})$, where
    $\mathcal{M}=\{0<a_{*i}^{(k+1)}<1\}$ with size $N_{\mathcal{M}}$.
  Calculate $\boldsymbol{\mu}_*^{(k+1)}=\boldsymbol{\mu}_*^{(k)}+\rho^{(k)}(\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)})$;
  Calculate $\rho^{(k+1)}=\alpha\times\rho^{(k)}$;
  $k=k+1$;
until convergence
Output: the solution: $\mathbf{w}_*=\mathbf{w}_*^{(k)}$, $\mathbf{u}_*=\mathbf{u}_*^{(k)}$, $b_*=b_*^{(k)}$; the
    classifier: $h(\mathbf{x})=\operatorname{sign}(\mathbf{x}^T\mathbf{w}_*+b_*)$.
```

Observe that in the iterative optimization scheme between (11) and (12), the calculation of $\mathbf{w}_{j+1}^{(k+1)}$ and $\mathbf{u}_{j+1}^{(k+1)}$ is independent of the calculation of $b_{j+1}^{(k+1)}$ and $\xi_{j+1}^{(k+1)}$. Further, calculation of ξ is unnecessary, and b can be calculated outside the inner loop, i.e., we can calculate $b_*^{(k)}=(1/N_{\mathcal{M}})\sum_{i\in\mathcal{M}}(y_i-\mathbf{x}_i^T\mathbf{w}_*^{(k)})$, where $\mathcal{M}=\{0<a_{*i}^{(k)}<1\}$ with the size $N_{\mathcal{M}}$. We summarize the above calculation in Algorithm 1 in which ρ exponentially increases. We call the algorithm *iterative SVM*1 (ISVM1).

However, the computational cost of ISVM1 is still high because it requires to solve the quadratic optimization problem within the inner loop many times. In fact, in order to calculate the solution for DrSVM, it is unnecessary to obtain an exact solution for the subproblem (8). Hence, the inner iteration in ISVM1 can be implemented only once. Finally, ISVM1 can be further speeded up, and it is described in ISVM2 with ρ being fixed and ISVM3 with ρ being increased exponentially (see Algorithms 2 and 3).

Algorithm 2. ISVM2.

```
Input: $\mathcal{X}=\{(\mathbf{x}_i\ ,y_i)$
}$i=1n$

, $\lambda_1$, $\lambda_2$, $\rho$, $\mathbf{u}=\mathbf{u}^{(0)}$
,$\boldsymbol{\mu}=\boldsymbol{\mu}^{(0)}$

, $k=0$.
repeat
  Calculate $\mathbf{a}^{(k+1)}$:
$\mathbf{a}^{(k+1)}=\underset{\mathbf{a}}{\operatorname{argmin}}\frac{1}{2(\lambda_2+\rho)}\mathbf{a}^T\mathbf{Y}\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{a}$
  $+(\frac{\mathbf{Y}\mathbf{X}(\rho\mathbf{u}^{(k)}-\boldsymbol{\mu}^{(k)})}{\lambda_2+\rho}-\mathbf{1})^T\mathbf{a}$

  subject to  $\mathbf{a}^T\mathbf{y}=\mathbf{0}$, and $\mathbf{0}\leq\mathbf{a}\leq\mathbf{1}$.

  Calculate $\mathbf{w}^{(k+1)}=\frac{\mathbf{X}^T\mathbf{Y}\mathbf{a}^{(k+1)}+\rho\mathbf{u}^{(k)}-\boldsymbol{\mu}^{(k)}}{\lambda_2+\rho}$
  Calculate $\mathbf{u}^{(k+1)}=\operatorname{sign}(\rho\mathbf{w}^{(k+1)}+\boldsymbol{\mu}^{(k)})$
$\max\{0,|\rho\mathbf{w}^{(k+1)}+\boldsymbol{\mu}^{(k)}-\lambda_1\mathbf{1}|\}/\rho$
  Calculate $\boldsymbol{\mu}^{(k+1)}=\boldsymbol{\mu}^{(k)}+\rho(\mathbf{w}^{(k+1)}-\mathbf{u}^{(k+1)})$
  $k=k+1$.
until convergence
Output: the solution $\mathbf{a}^*$,$\mathbf{w}^*$,$\mathbf{u}^*$ and
$b^*=\frac{1}{N_{\mathcal{M}}}\sum_{i\in\mathcal{M}}(y_i-\mathbf{x}_i^T\mathbf{w}^*)$, where
  $\mathcal{M}=\{i|0<a_i^*<1\}$, $N_{\mathcal{M}}$ the size of $\mathcal{M}$

The classifier is $h(\mathbf{x})=\operatorname{sign}(y(\mathbf{x}^T\mathbf{w}^*+b^*))$
```

Algorithm 3. ISVM3.

```
Let $\rho$ increase exponentially in Algorithm 2, i.e., line 6 in Algorithm 2 is replaced with
$\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \rho^{(k)}(\mathbf{w}^{(k+1)} - \mathbf{u}^{(k+1)})$,
    $\rho^{(k+1)} = \alpha \times \rho^{(k)}$   where $\alpha > 1$.
```

2.2. Computational cost analysis

In this paper, we only focus on the algorithms ISVM2 and ISVM3. In every loop, the main computational burden is the QP (14), this is a scalar of size n problem and does not depend on p. It can be solved efficiently with the cost $O(n^3)$. Although the number of iterations is unknown, experiments have shown that usually less than 100 iterations will achieve convergence. Additionally, we can find from the description of algorithms that the only parts of computational cost depending on p are the computation of $\mathbf{XX}^T$ and $\mathbf{Xf}$ with $\mathbf{f}$ a vector, the former is implemented only once before the loop, the latter is implemented in every iteration with complexity $O(np)$. Thus the total computational cost of algorithm ISVM2 or ISVM3 is $O(K(n^3+np))$ with K the number of iteration.

2.3. Convergence analysis

In order to reveal the efficiency and feasibility of our algorithms, we further discuss their convergence analysis in what follows:

Theorem 1. *For algorithm ISVM1, any limit point* $(\mathbf{w}_*, b_*, \mathbf{u}_*)$ *of* $(\mathbf{w}_*^{(k)}, b_*^{(k)}, \mathbf{u}_*^{((k)})$ *is an optimal solution to* (5) *and the convergence rate is at least* $O(1/\rho^{(k-1)})$, *i.e.,*

$$|f_1(\mathbf{w}_*^{(k)}, b_*^{(k)}) + \lambda_1 \|\mathbf{u}_*^{(k)}\|_1 - p^*| = O\left(\frac{1}{\rho^{(k-1)}}\right),$$

where p^* *is the optimal value of the problem* (5).

In order to generate the global convergence property of ISVM2, we rewrite the optimization problem (5) as

$$\min_{\mathbf{w},b,\mathbf{u}} \quad f_1(\mathbf{w},b) + f_2(\mathbf{u})$$

$$\text{s.t.} \quad \mathbf{w} - \mathbf{u} = \mathbf{0}.$$

The iteration between $\mathbf{w}, \mathbf{u}, b, \boldsymbol{\mu}$ constitutes an ADMM procedure that solves the problem (5). It has global convergence property if the following conditions are satisfied [16]:

(A) The extended real valued functions $f_1 : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $f_2 : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are closed, proper, and convex.
(B) The unaugmented Lagrangian (6) has a saddle point (but not necessarily unique), i.e., there exists $(\mathbf{w}^*, b^*, \xi^*, \mathbf{u}^*, \boldsymbol{\mu}^*)$ such that $L_0(\mathbf{w}^*, b^*, \xi^*, \mathbf{u}^*, \boldsymbol{\mu}) \le L_0(\mathbf{w}^*, b^*, \xi^*, \mathbf{u}^*, \boldsymbol{\mu}^*) \le L_0(\mathbf{w}, b, \xi, \mathbf{u}, \boldsymbol{\mu}^*)$.

Obviously, Condition (A) is trivial and Condition (B) is also easily satisfied, see [17]. We summarize the following theorem [16].

Theorem 2. *The iterative sequence* $(\mathbf{w}^{(k)}, \mathbf{u}^{(k)}, \boldsymbol{\mu}^{(k)})$ *from algorithm ISVM2 satisfies the following properties*:

(1) $\mathbf{u}^{(k)} - \mathbf{w}^{(k)} \to \mathbf{0}$, i.e., *the iteration approaches feasibility of the problem* (5).
(2) *The objective function in* (5) *approaches the optimal value* p^* *as* $k \to \infty$.
(3) *Dual variable convergence*: $\boldsymbol{\mu}^{(k)} \to \boldsymbol{\mu}^*$, $\boldsymbol{\mu}^*$ *is a dual optimal point.*

Finally, the following theorem establishes the global convergence of algorithm ISVM3.

Theorem 3. *Let* $(\mathbf{w}^\infty, b^\infty, \mathbf{u}^\infty)$ *be the limit point of* $(\mathbf{w}^{(k)}, b^{(k)}, \mathbf{u}^{(k)})$ *from algorithm ISVM3.*

If $\lim_{k\to\infty} \rho^{(k)}(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}) = 0$ *and* $\sum_{k=1}^{\infty} 1/\rho^{(k)} < \infty$, *then* $(\mathbf{w}^\infty, b^\infty, \mathbf{u}^\infty)$ *is an optimal solution of the problem* (5).

The proof of Theorems 1 and 3 is given in Appendix A.

Table 1
Experimental results with the three methods on the simulation datasets, Here, n—the size of training dataset; m—the size of test dataset; p—the dimension of the input vector; r—the relative coefficient; CAR—the classification accuracy rate; NI—the number of iteration; and CT—the computational time (s). Parameter setting: $\lambda_1 = 10\|\mathbf{w}_0\|_\infty$ and $\lambda_2 = 10$ in all the algorithms (include ADMM-DrSVM, since this choice can largely improve the speed), $\rho = 50$ in ISVM2, and $\rho^{(0)} = 10$ and $\alpha = 1.2$ in ISVM3.

Data size	r	ISVM2			ISVM3			ADMM-DrSVM		
		CAR	CT(s)	NI	CAR	CT(s)	NI	CAR	CT(s)	NI
$n=50$	0.1	0.975	0.19	29	0.976	0.20	28	0.982	0.36	79
$m=500$	0.5	0.873	0.17	25	0.873	0.17	25	0.894	0.35	74
$p=2000$	0.9	0.821	0.17	23	0.822	0.16	22	0.834	0.33	70
$n=50$	0.1	0.967	0.26	43	0.967	0.28	24	0.967	0.75	97
$m=500$	0.5	0.875	0.24	21	0.875	0.25	22	0.876	0.96	53
$p=5000$	0.9	0.819	0.24	20	0.819	0.24	20	0.818	0.88	51
$n=50$	0.1	0.948	0.47	18	0.949	0.47	20	0.950	1.21	44
$m=500$	0.5	0.871	0.40	18	0.872	0.42	19	0.872	1.10	43
$p=10{,}000$	0.9	0.799	0.40	17	0.800	0.43	18	0.800	1.02	40

Table 2
Experimental results of NTSF for ISVM2 with respect to the different λ_2 on the simulation datasets. Here, n—the size of training dataset; m—the size of test dataset; p—the dimension of the input vector; and r—the relative coefficient.

Data size	r	NTSF for different λ_2 (algorithm ISVM2)									
		1	2	3	4	5	10	20	30	40	50
$n=50$	0.1	8.8	9.3	9.5	9.5	9.9	9.9	10	10	10	10
$m=500$	0.5	6.5	7.8	8.3	9.4	9.5	9.7	9.9	10	10	10
$p=500$	0.9	8.0	9.2	9.7	9.7	10	10	10	10	10	10
$n=50$	0.1	9.0	9.5	9.8	10	10	10	10	10	10	10
$m=500$	0.5	8.2	9.1	9.7	9.9	9.9	10	10	10	10	10
$p=1000$	0.9	8.9	9.9	9.9	9.9	10	10	10	10	10	10
$n=50$	0.1	9.5	9.9	10	10	10	10	10	10	10	10
$m=500$	0.5	8.6	9.5	9.9	10	10	10	10	10	10	10
$p=2000$	0.9	9.9	10	10	10	10	10	10	10	10	10

Table 3
Summary of the benchmark datasets: C—the number of classes; n—the size of the training dataset; m—the size of the test data; and p—the dimension of the input vector.

Data set	C	n	m	p
Colon	2	31	31	2000
Duke breast	2	22	22	7129
Leukemia	2	38	34	7129

Table 4
Experimental results for the three methods on the three gene expression datasets: CAR, CT, NT have the same meaning as in Table 1 and NNC—the number of nonzero coefficients. Parameter setting: $\lambda_1 = 10|\mathbf{w}_0|$ and $\lambda_2 = 1$ for all algorithms (we changed the value of λ_1 in ADMM-DrSVM to improve its performance), $\rho = 50$ in ISVM2, and $\rho^{(0)} = 10$ and $\alpha = 1.2$ in ISVM3.

Dataset	ISVM2				ISVM3				ADMM-DrSVM				SVM
	CAR	CT	NI	NNC	CAR	CT	NI	NNC	CAR	CT	NI	NNC	CAR
`Colon`	0.830	0.65	79	1851	0.830	0.31	28	1735	0.819	6.16	308	1877	0.810
`Duke breast`	0.825	0.27	54	6773	0.813	0.16	27	6500	0.802	10.52	201	6861	0.845
`Leukemia`	0.824	0.84	53	6738	0.824	0.46	27	6450	0.824	17.5	238	6823	0.882

3. Experimental evaluation

In this section, we only consider algorithms ISVM2 and ISVM3, because ISVM1 needs an inner loop to solve a subproblem exactly which is time costly. Generally, all the parameters ρ, λ_1, λ_2, and α can be selected by the cross-validation (CV) technique, while giving rise to the high computational cost. For the sake of simplicity, in our experiments, the parameter ρ is set to $\rho = 50$ in ISVM2, parameter $\lambda_2 \in [1,50]$ in ISVM2 and ISVM3, α and $\rho^{(0)}$ are set as 1.2 and 10, respectively, in ISVM3. In addition, since the parameter λ_1 have a great impact on sparsity of the model, a proper parameter λ_1 is empirically set to $\lambda_1 = 10 \times \|\mathbf{w}_0\|_\infty$ in ISVM2 and ISVM3, where $\mathbf{w}_0$ is the solution of traditional SVM on the training dataset. The experimental results show that this is a proper choice. We find in the experiments that the accuracy is insensitive to the different value of parameters. However, the varying of parameter values have great influence on the speed of the algorithms and sparseness of coefficients $\mathbf{w}$.

Since Theorems 2 and 3 reveal that $\mathbf{u}^{(k)} - \mathbf{w}^{(k)} \to \mathbf{0}$ as $k \to \infty$, the stopping criteria for convergence in our algorithms can be set to $\|\mathbf{u}^{(k)} - \mathbf{w}^{(k)}\|_2 \le \tau$, where $\tau = 10^{-4}$ in our experiments. Note that this stop condition is not sufficient to guarantee the convergence theoretically, however, it still works well in practice.

Recall that ADMM-DrSVM in [14] can efficiently solve the optimization problem for DrSVM. Moreover, it outperforms HHSVM [13] and other algorithms for DrSVM, significantly reducing the computational time for high-dimensional datasets with competitive classification accuracy. For simplicity, we only compare our algorithms with ADMM-DrSVM.

In order to validate the efficiency of our algorithms, we conduct experiments on the simulation and real-world datasets. Several evaluation criteria for comparison are reported, including average classification accuracy rate, average number of nonzero coefficients, average number of iteration, and average computational time.

3.1. Simulation data

We are going to implement the experiment on simulation datasets. The two classes '+' and '−' are drawn from the distributions $\mathcal{N}(\boldsymbol{\mu}_+, \boldsymbol{\Sigma})$ and $\mathcal{N}(\boldsymbol{\mu}_-, \boldsymbol{\Sigma})$, respectively, where $\boldsymbol{\mu}_+ = (1, \ldots, 1, 0, \ldots, 0)^T$ with the first 10 components being 1 and the rest being 0, $\boldsymbol{\mu}_- = (-1, \ldots, -1, 0, \ldots, 0)^T$ with the first 10 components being -1 and the rest being 0, and the covariance matrix is

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^*_{10\times 10} & \mathbf{0}_{10\times(p-10)} \\ \mathbf{0}_{(p-10)\times 10} & \mathbf{I}_{(p-10)\times(p-10)} \end{pmatrix}.$$

Here $\boldsymbol{\Sigma}^*_{10\times 10}$ is a 10×10 matrix with all diagonal terms being 1 and the rest being $r < 1$, and $\mathbf{0}_{k\times l}$ is the $k \times l$ zero matrix. It should be noted that the larger r implies the higher correlation between the first 10 features of the input data. We implement our experiments on r=0.1, 0.5, and 0.9. Obviously, the first 10 variables play a significant role on the classification decision. We report the averaged results of 10 runnings.

Firstly, we compare our algorithms with ADMM-DrSVM based on the number of iterations and computational time, while keeping the test accuracy and sparsity almost the same for all the compared algorithms. Table 1 reports the corresponding results. Table 1 shows that the speeds of both ISVM2 and ISVM3 are almost the same, and they are more efficient in comparison with ADMM-DrSVM.

Secondly, we implement the simulation experiments in order to reveal how does the varying of λ_2 affect the number of true selected features (NTSF). Obviously, the ideal NTSF for our simulation datasets is 10. The corresponding results are given in Table 2. For the sake of brief, we only report the result of ISVM2, because we find that λ_2 varying in the interval [1,50] does not affect the NTSF in algorithm ISVM3. From Table 2, we can see that as the value of λ_2 increases, the NTSF increases, i.e., the group effect becomes more apparent.

3.2. Microarray classification and gene selection

Typically, a microarray gene expression dataset consists of thousands of genes and a small number of instances, i.e., gene selection in this setting is a high-dimension problem with $p \gg n$.

We conduct our experiments on three microarray gene expression datasets,[1] including `colon`, `duke breast` and `leukemia`. The description of datasets is given in Table 3. In our experiments, each dataset was standardized. Each dataset is randomly partitioned into two disjoint subsets as training and test datasets, see Table 3.

Table 4 shows the experimental results, also including the standard SVM classification accuracy rate. According to Table 4, it should be obvious that our algorithms are more efficient than ADMM-DrSVM, and that they achieve convergence within a small number of iterations. Moreover, the classification accuracies of our algorithms are very competitive with the other compared algorithms and the standard SVM. Moreover, the tuning parameter α in Algorithm 3 efficiently speeds up the convergence of the algorithms.

We have previously mentioned that although taking values of the tuning parameters in a certain interval does not affect the test accuracy, it does have great influence on the speed. We discuss this issue on the real-world datasets. From the experiments, for both algorithms ISVM2 and ISVM3, we find that a larger λ_1 will increase the number of iterations, leading to the higher computational time. The increase of λ_2 will quickly decrease the number of iterations, leading to the lower computational time. We also find that in ISVM3, the appropriate increase of α can also decrease the computational time to some extent.

[1] The datasets is available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

We implemented the experiments to analyze the sparsity ability of our algorithms with respect to the different values of the parameters λ_1, λ_2, and ρ. Here, the degree of sparsity is evaluated by using the number of zero elements in estimated $\mathbf{w}$. (We mention that in some papers, degree of sparsity is measured as the number of nonzero elements which is opposite to our definition.) We find that the degree of sparsity increases with respect to the parameter λ_1, and decreases with respect to λ_2. Additionally, the parameters λ_1 and λ_2 in ISVM3 have similar influence on sparsity. We find that degree of sparsity is insensitive to the different values of ρ in algorithms ISVM2. In addition, we also mention that increase of α in ISVM3 leads to the decrease of the degree of sparsity.

4. Conclusion

In this paper we have developed three efficient algorithms for DrSVM in high-dimensional datasets. Our proposed algorithms proceed by solving the QP at every iteration, which largely reduce the scale of the original problem and accelerate the speed when $p \gg n$. We have shown that our algorithms have complexity $O(K(n^3+np))$, which is much lower than $O(Knp^2)$ given by ADMM-DrSVM algorithm. Moreover, our algorithms are able to achieve global convergence. The experimental results on the simulation and real-world gene datasets illustrate the efficiency of our algorithms. In the future work, we will extend the binary classification setting to the multi-class case.

Conflict of interest statement

None declared.

Acknowledgments

This work has been supported in part by the Natural Science Foundations of China (Nos. 61070239 and 61272303).

Appendix A

A.1. Proof of Theorem 1

We denote

$$[1-y_i(\mathbf{w}^T\mathbf{x}_i+b)]_+ = \phi(y_i(\mathbf{w}^T\mathbf{x}_i+b)),$$

and

$$(\mathbf{w}_*^{(k+1)},b_*^{(k+1)},\mathbf{u}_*^{(k+1)}) = \operatorname{argmin}\sum_{i=1}^n \phi(y_i(\mathbf{w}^T\mathbf{x}_i+b))+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2$$

$$+\lambda_1\|\mathbf{u}\|_1+\frac{\rho_*^{(k)}}{2}\|\mathbf{w}-\mathbf{u}\|_2^2+\langle \boldsymbol{\mu}_*^{(k)},\mathbf{w}-\mathbf{u}\rangle$$

in the $(k+1)$th iterative optimization procedure. Since $(\mathbf{w}_*^{(k+1)},b_*^{(k+1)},\mathbf{u}_*^{(k+1)})$ is the optimal solution for the optimization problem above, we have that

$$\mathbf{0} \in \sum_{i=1}^n \partial\phi(y_i(\mathbf{w}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)}))y_i\mathbf{x}_i$$
$$+\lambda_2\mathbf{w}_*^{(k+1)}+\rho_*^{(k)}(\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)})+\boldsymbol{\mu}_*^{(k)}, \tag{15}$$

$$0 \in \sum_{i=1}^n \partial\phi(y_i(\mathbf{w}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)}))y_i, \tag{16}$$

$$\mathbf{0} \in \partial\|\mathbf{u}_*^{(k+1)}\|_1+\rho_*^{(k)}(\mathbf{u}_*^{(k+1)}-\mathbf{w}_*^{(k+1)})-\boldsymbol{\mu}_*^{(k)}. \tag{17}$$

Let

$$\boldsymbol{\mu}_*^{(k+1)} = \boldsymbol{\mu}_*^{(k)}+\rho_*^{(k)}(\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)}). \tag{18}$$

Then, Eqs. (15) and (17) can be re-expressed as

$$-\boldsymbol{\mu}_*^{(k+1)}-\lambda_2\mathbf{w}_*^{(k+1)} \in \sum_{i=1}^n \partial\phi(y_i(\mathbf{w}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)}))y_i\mathbf{x}_i, \tag{19}$$

$$\boldsymbol{\mu}_*^{(k+1)} \in \lambda_1\partial\|\mathbf{u}_*^{(k+1)}\|_1. \tag{20}$$

Obviously, $\partial\phi(\cdot)$ and $\partial\|\mathbf{u}\|$ are bounded. Then, (19) and (20) tell us that $\boldsymbol{\mu}_*^{(k+1)}$ and $\boldsymbol{\mu}_*^{(k+1)}+\lambda_2\mathbf{w}_*^{(k+1)}$ are bounded. Hence, $\mathbf{w}_*^{(k+1)}$ is bounded. If we denote $(\mathbf{w}_*,\mathbf{u}_*)$ the limit point of $(\mathbf{w}_*^{(k)},\mathbf{u}_*^{(k)})$, from the fact that there exists M, such that $\|\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)}\|_2 \le (M/\rho_*^{(k)})$, we have $\mathbf{w}_* = \mathbf{u}_*$. So the limit point is feasible.

Since

$$L(\mathbf{w}_*^{(k+1)},b_*^{(k+1)},\mathbf{u}_*^{(k+1)},\boldsymbol{\mu}_*^{(k)},\rho_*^{(k)})$$
$$=\min_{\mathbf{w},b,\mathbf{u}} L(\mathbf{w},b,\mathbf{u},\boldsymbol{\mu}_*^{(k)},\rho_*^{(k)})$$
$$\le \min_{\mathbf{w}=\mathbf{u}} L(\mathbf{w},b,\mathbf{u},\boldsymbol{\mu}_*^{(k)},\rho_*^{(k)})$$
$$=\min \sum_{i=1}^n \partial\phi(y_i(\mathbf{w}^T\mathbf{x}_i+b))+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2+\lambda_1\|\mathbf{w}\|_1 = p^*,$$

then we have

$$\sum_{i=1}^n \phi(y_i(\mathbf{w}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)}))+\frac{\lambda_2}{2}\|\mathbf{w}_*^{(k+1)}\|_2^2+\lambda_1\|\mathbf{u}_*^{(k+1)}\|_1$$
$$=L(\mathbf{w}_*^{(k+1)},b_*^{(k+1)},\mathbf{u}_*^{(k+1)},\boldsymbol{\mu}_*^{(k)},\rho_*^{(k)})-\langle \boldsymbol{\mu}_*^{(k)},\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)}\rangle \tag{21}$$
$$-\frac{\rho_*^{(k)}}{2}\|\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)}\|_2^2$$
$$=L(\mathbf{w}_*^{(k+1)},b_*^{(k+1)},\mathbf{u}_*^{(k+1)},\boldsymbol{\mu}_*^{(k)},\rho_*^{(k)})-\left\langle \boldsymbol{\mu}_*^{(k)},\frac{\boldsymbol{\mu}_*^{(k+1)}-\boldsymbol{\mu}_*^{(k)}}{\rho_*^{(k)}}\right\rangle \tag{22}$$
$$-\frac{1}{2\rho_*^{(k)}}\|\boldsymbol{\mu}_*^{(k+1)}-\boldsymbol{\mu}_*^{(k)}\|_2^2$$
$$\le p^*-\frac{1}{2\rho_*^{(k)}}(\|\boldsymbol{\mu}_*^{(k+1)}\|_2^2-\|\boldsymbol{\mu}_*^{(k)}\|_2^2) = p^*+O((\rho_*^{(k)})^{-1}). \tag{23}$$

On the other hand, from the boundedness of $\mathbf{w}_*^{(k+1)}$, $\mathbf{u}_*^{(k+1)}$, and $\|\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)}\|_2 = O((\rho_*^{(k)})^{-1})$, we have

$$\sum_{i=1}^n \phi(y_i(\mathbf{w}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)}))+\frac{\lambda_2}{2}\|\mathbf{w}_*^{(k+1)}\|_2^2+\lambda_1\|\mathbf{u}_*^{(k+1)}\|_1$$
$$=\sum_{i=1}^n \phi(y_i(\mathbf{u}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)}))+\frac{\lambda_2}{2}\|\mathbf{u}_*^{(k+1)}\|_2^2+\lambda_1\|\mathbf{u}_*^{(k+1)}\|_1$$
$$+\frac{\lambda_2}{2}\|\mathbf{w}_*^{(k+1)}\|_2^2-\frac{\lambda_2}{2}\|\mathbf{u}_*^{(k+1)}\|_2^2+\sum_{i=1}^n \phi(y_i(\mathbf{w}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)}))$$
$$-\sum_{i=1}^n \phi(y_i(\mathbf{u}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)})) \ge p^*-\sum_{i=1}^n |(\mathbf{w}_*^{(k+1)T}-\mathbf{u}_*^{(k+1)T})\mathbf{x}_i|$$
$$+\frac{\lambda_2}{2}(\mathbf{w}_*^{(k+1)}+\mathbf{u}_*^{(k+1)})^T(\mathbf{w}_*^{(k+1)}-\mathbf{u}_*^{(k+1)}) \ge p^*+O((\rho_*^{(k)})^{-1}). \tag{24}$$

From (23) and (24), these prove that $|f_1(\mathbf{w}_*^{(k)},b_*^{(k)})+\lambda_1\|\mathbf{u}_*^{(k)}\|_1-p^*| = O((\rho_*^{(k)})^{-1})$.

A.2. Proof of Theorem 3

In the $(k+1)$th iterative optimization procedure, let

$$(\mathbf{w}^{(k+1)},b^{(k+1)}) = \operatorname{argmin} \sum_{i=1}^{n} \phi(y_i(\mathbf{w}^T\mathbf{x}_i+b)) + \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 + \frac{\rho^{(k)}}{2}\|\mathbf{w}-\mathbf{u}^{(k)}\|_2^2 + \langle \boldsymbol{\mu}^{(k)},\mathbf{w}-\mathbf{u}^{(k)}\rangle, \tag{25}$$

$$\mathbf{u}^{(k+1)} = \operatorname{argmin} \lambda_1\|\mathbf{u}\|_1 + \frac{\rho^{(k)}}{2}\left\|\mathbf{u}-\left(\mathbf{w}^{(k+1)}+\frac{\boldsymbol{\mu}^{(k)}}{\rho^{(k)}}\right)\right\|_2^2, \tag{26}$$

and

$$\hat{\boldsymbol{\mu}}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \rho^{(k)}(\mathbf{w}^{(k+1)}-\mathbf{u}^{(k)}). \tag{27}$$

Since $(\mathbf{w}^{(k+1)},b^{(k+1)})$ and $\mathbf{u}^{(k+1)}$ are the optimal solutions for (25) and (26), similar to the proof of Theorem 2, we have

$$-\hat{\boldsymbol{\mu}}^{(k+1)} - \lambda_2\mathbf{w}^{(k+1)} \in \sum_{i=1}^{n} \partial\phi(y_i(\mathbf{w}^{(k+1)T}\mathbf{x}_i+b^{(k+1)}))y_i\mathbf{x}_i, \tag{28}$$

$$0 \in \sum_{i=1}^{n} \partial\phi(y_i(\mathbf{w}^{(k+1)T}\mathbf{x}_i+b^{(k+1)}))y_i, \tag{29}$$

and

$$\boldsymbol{\mu}^{(k+1)} \in \lambda_1\partial\|\mathbf{u}^{(k+1)}\|_1. \tag{30}$$

Obviously, $\boldsymbol{\mu}^{(k+1)}$ is bounded. According to $\hat{\boldsymbol{\mu}}^{(k+1)}-\boldsymbol{\mu}^{(k+1)} = \rho^{(k)}(\mathbf{u}^{(k+1)}-\mathbf{u}^{(k)})$, $\lim_{k\to\infty}\rho^{(k)}(\mathbf{u}^{(k+1)}-\mathbf{u}^{(k)})\to\mathbf{0}$ and Eq. (27), it is not difficult to find that $\hat{\boldsymbol{\mu}}^{(k)}$ is bounded and $\|\mathbf{u}^{(k+1)}-\mathbf{u}^{(k)}\| = o((\rho^{(k)})^{-1})$. In addition, from (28), we can find $\mathbf{w}^{(k)}$ is bounded. Hence, $\mathbf{u}^{(k)}$ and $b^{(k)}$ are bounded as well.

Since $\rho^{(k)}$ grows exponentially and $\mathbf{u}^{(k)}$ is a Cauchy sequence, $\mathbf{u}^{(k)}$ converges to a limit $\mathbf{u}^{(\infty)}$. In addition, from $(1/\rho^{(k)})(\boldsymbol{\mu}^{(k+1)}-\boldsymbol{\mu}^{(k)}) = \mathbf{w}^{(k+1)}-\mathbf{u}^{(k+1)}$, we can see that the limit point satisfy $\mathbf{w}^{\infty} = \mathbf{u}^{\infty}$.

Since all terms in the objective function in (5) are convex, together with (28)–(30), we have

$$\begin{aligned}
\sum_{i=1}^{n} \phi(y_i(\mathbf{w}^{(k+1)T}\mathbf{x}_i+b^{(k+1)})) + \frac{\lambda_2}{2}\|\mathbf{w}^{(k+1)}\|_2^2 + \lambda_1\|\mathbf{u}^{(k+1)}\|_1 \\
\le \sum_{i=1}^{n} \phi(y_i(\mathbf{w}_*^{(k+1)T}\mathbf{x}_i+b_*^{(k+1)})) + \frac{\lambda_2}{2}\|\mathbf{w}_*^{(k+1)}\|_2^2 + \lambda_1\|\mathbf{u}_*^{(k+1)}\|_1 \\
+ \langle \hat{\boldsymbol{\mu}}^{(k+1)},\mathbf{w}_*^{(k+1)}-\mathbf{w}^{(k+1)}\rangle - \langle \boldsymbol{\mu}^{(k+1)},\mathbf{u}_*^{(k+1)}-\mathbf{u}^{(k+1)}\rangle \\
\le p^* + O((\rho^{(k)})^{-1}) + \langle \hat{\boldsymbol{\mu}}^{(k+1)},\mathbf{w}_*^{(k+1)}-\mathbf{w}^{(k+1)}\rangle \\
- \langle \boldsymbol{\mu}^{(k+1)},\mathbf{u}_*^{(k+1)}-\mathbf{u}^{(k+1)}\rangle
\end{aligned}$$

$$\begin{aligned}
= p^* + O((\rho^{(k)})^{-1}) + \frac{1}{\rho_*^{(k)}}\langle \boldsymbol{\mu}^{(k+1)},\boldsymbol{\mu}_*^{(k+1)}-\boldsymbol{\mu}_*^{(k)}\rangle \\
- \frac{1}{\rho^{(k)}}\langle \boldsymbol{\mu}^{(k+1)},\boldsymbol{\mu}^{(k+1)}-\boldsymbol{\mu}^{(k)}\rangle + \langle \rho^{(k)}(\mathbf{u}^{(k+1)}-\mathbf{u}^{(k)}),\mathbf{w}_*^{(k+1)}-\mathbf{w}^{(k+1)}\rangle
\end{aligned}$$

From the boundedness of $(\boldsymbol{\mu}^{(k)},\hat{\boldsymbol{\mu}}^{(k)},\boldsymbol{\mu}_*^{(k)},\mathbf{w}_*^{(k)},\mathbf{w}^{(k)})$ and $\rho^{(k)}(\mathbf{u}^{(k+1)}-\mathbf{u}^{(k)})\to\mathbf{0}$, we find that the limit point $(\mathbf{w}^{\infty},b^{\infty},\mathbf{u}^{\infty})$ must satisfy $f_1(\mathbf{w}^{\infty},b^{\infty})+\lambda_1\|\mathbf{u}^{\infty}\|_1 \le p^*$. Then, we obtain

$$f_1(\mathbf{w}^{\infty},b^{\infty})+\lambda_1\|\mathbf{u}^{\infty}\|_1 = p^*.$$

References

[1] A. Rakotomamonjy, Variable selection using SVM-based criteria, Journal of Machine Learning Research 3 (2003) 1357–1370.

[2] Y. Grandvalet, S. Ganu, Adaptive scaling for feature selection in SVMs, in: NIPS, vol. 15, 2002.

[3] T. Jebara, T. Jaakkola, Feature selection and dualities in maximum entropy discrimination, in: 16th Annual Conference on Uncertainty in Artificial Intelligence, 2000.

[4] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, Optimization with sparsity-inducing penalties, Foundations and Trends in Machine Learning 4 (1) (2012) 1–106.

[5] P.S. Bradley, O.L. Mangasarian, Feature selection via concave minimization and support vector machines, in: ICML, 1998, pp. 82–90.

[6] J. Weston, A. Elisseeff, B. Scholkopf, M. Tipping, Use of the zero-norm with linear models and2 kernel methods, Journal of Machine Learning Research 3 (2003) 1439–1461.

[7] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-norm support vector machines, in: NIPS, vol. 16, 2003.

[8] J. Bi, Y. Chen, J.Z. Wang, A sparse support vector machine approach to region-based image categorization, in: CVPR, 2005, pp. 1121–1128.

[9] J. Bi, K.P. Bennett, M. Embrechts, C. Breneman, M. Song, Dimensionality reduction via sparse support vector machines, Journal of Machine Learning Research 3 (2003) 1229–1243.

[10] L. Wang, J. Zhu, H. Zou, The doubly regularized support vector machine, Statistica Sinica 16 (2) (2006) 589–615.

[11] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of Royal Statistical Society B (2005) 301–320.

[12] V. Jeyakumar, G. Li, S. Suthaharan, Support vector machine classifiers with uncertain knowledge sets via robust optimization, Optimization: A Journal of Mathematical Programming and Operations Research (2012) 1–18.

[13] L. Wang, J. Zhu, H. Zou, Hybrid huberized support vector machines for microarray classification and gene selection, Bioinformatics 24 (3) (2008) 412–419.

[14] G.B. Ye, Y. Chen, X. Xie, Efficient variable selection in support vector machines via the alternating direction methods of multipliers, in: AISTATS, 2011.

[15] G.H. Golub, C.F.V. Loan, Matrix Computations, third edition, Johns Hopkins University Press, New York, 1996.

[16] S. Boyd, N. Parikh, E. Chu, J. Peleato, B. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[17] R.T. Rockafellar, Convex Analysis, second edition, Princeton University Press, New York, 1970.

Dehua Liu is the PhD student at Computer Science & Technology departent, Zhejiang University.

Hui Qian is the associate professorat Computer Science & Technology departent, Zhejiang University.

Guang Dai is the PhD student at Computer Science & Technology departent, Zhejiang University.

Zhihua Zhang is the professor at Computer Science & Technology departent, Zhejiang University.