Large-scale linear nonparallel support vector machine solver

Yingjie Tian, Yuan Ping

Department of Computer Science and Technology, Xuchang University, Xuchang 461000, China

Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China

Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author at: Department of Computer Science and Technology, Xuchang University, Xuchang 461000, China

E-mail addresses: tyj@ucas.ac.cn (Y. Tian), pingyuan@bupt.edu.cn, pyuan.lhn@xcu.edu.cn (Y. Ping).

Abstract

Twin support vector machines (TWSVMs), as the representative nonparallel hyperplane classifiers, have shown the effectiveness over standard SVMs from some aspects. However, they still have some serious defects restricting their further study and real applications: (1) They have to compute and store the inverse matrices before training, it is intractable for many applications where data appear with a huge number of instances as well as features; (2) TWSVMs lost the sparseness by using a quadratic loss function making the proximal hyperplane close enough to the class itself. This paper proposes a Sparse Linear Nonparallel Support Vector Machine, termed as L1-NPSVM, to deal with large-scale data based on an efficient solver—dual coordinate descent (DCD) method. Both theoretical analysis and experiments indicate that our method is not only suitable for large scale problems, but also performs as good as TWSVMs and SVMs.

© 2013 Elsevier Ltd. All rights reserved.

Keywords: Support vector machines; Classification; Machine learning; Support vector machines; Nonparallel support vector machine

1. Introduction

Support vector machines (SVMs), having their roots in statistical learning theory, are useful for pattern classification (Deng & Tian, 2009; Tian, Shi, & Liu, 2012; Vapnik, 1996, 1998). For a binary classification problem with training set

\[ T = \{(x_1, y_1), \ldots, (x_l, y_l)\} \in (\mathbb{R}^d \times \mathbb{Y})^l \]  

where \( x_i \in \mathbb{R}^d, y_i \in \mathbb{Y} = \{-1, 1\}, i = 1, \ldots, l \), SVM finds the optimal separating hyperplane by maximizing the margin between two parallel support hyperplanes, which involves the minimization of a quadratic programming problem (QPP)

\[ \min_{w, b, \xi} \frac{1}{2} \|w\|^2 + \sum_{i=1}^{l} \xi_i, \quad \text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l, \quad \xi_i \geq 0, \quad i = 1, \ldots, l, \]  

where \( \xi = (\xi_1, \ldots, \xi_l)^\top \), and \( C > 0 \) is a penalty parameter. This SVM is called \( L_1 \)-SVM since the \( L_1 \)-loss function \( \xi_i = \max(1-y_i((w \cdot x_i) + b), 0) \) is adopted. For this primal problem, \( L_1 \)-SVM solves its Lagrangian dual problem

\[ \min_{\alpha} \frac{1}{2} \alpha^\top Q \alpha - e^\top \alpha, \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l, \]  

where \( Q \in \mathbb{R}^{l \times l} \), and \( Q_{ij} = y_jy_i((x_i \cdot x_j) + 1) \). It is also a QPP. An SVM usually maps the training set into a high-dimensional space via a nonlinear function \( \phi(x) \), then the kernel function \( K(x, x') \) is applied to take instead of the inner product \( \langle \phi(x) \cdot \phi(x') \rangle \), such SVM is called a nonlinear SVM.

http://dx.doi.org/10.1016/j.neunet.2013.11.014

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
However, in some applications such as document classification with the data appearing in a high dimensional feature space, linear SVM in which the data are not mapped, has similar performances with nonlinear SVM. For linear SVM, many methods have been proposed in large-scale scenarios (Bottou, 2007; Chang, Hsieh, & Lin, 2008; Chang & Lin, 2001; Collins, Globock, Koo, Carreras, & Bartlett, 2008; Hsieh, Chang, Lin, Keerthi, & Sundararajan, 2008; Joachims, 2006; Keerthi & DeCoste, 2005; Lin, Wang, & Keerthi, 2008; Shalev-Shwartz, Singer, & Srebro, 2011; Smola, Vishwanathan, & Le, 2008; Zhang, 2004).

Recently, some nonparallel hyperplane classifiers have been proposed (Jayadeva, Khemchandani, & Chandra, 2007; Mangasarian & Wild, 2006). For the twin support vector machine (TWSVM) (Jayadeva et al., 2007), it seeks two nonparallel proximal hyperplanes such that each hyperplane is closer to one of the two classes and is at least one distance from the other. Experimental results (Jayadeva, 2007; Kumar & Gopal, 2008) have shown the effectiveness of TWSVM over the standard SVM on UCI data sets. Furthermore, it is implemented by solving two QPPs smaller than the problem (3), which increases the TWSVM’s training speed by approximately fourfold compared with that of SVM. TWSVMs have been studied extensively (Khemchandani, Jayadeva, & Chandra, 2009; Kumar & Gopal, 2009; Peng, 2010; Qi, Tian, & Shi, 2012, 2013; Qi, Tian, & Yong, 2012a, 2012b; Shao, Zhang, Wang, & Deng, 2011).

However, existing TWSVMs have two serious defects which restrict their further studies and real applications: (1) Although TWSVMs solve two smaller QPPs and can be solved by successive overrelaxation (SOR) technique (Shao et al., 2011), they have to compute the inverse of matrices before training, which is in practice intractable for a large dataset; (2) TWSVMs lost the sparseness by using a quadratic loss function making the proximal hyperplane close enough to the class itself.

In this paper, for linear classification issues, we propose a novel nonparallel linear classifier, termed as linear L1-NPSVM, to solve very large linear problems. Our L1-NPSVM has incomparable advantages including: (1) The two problems constructed have the elegant formulation and can be solved efficiently by the dual coordinate descent (DCD) method, more importantly, we do not need to compute the inverses of the large matrices any more before training; (2) It has the valuable sparseness similar with the standard SVMs; (3) L1-NPSVM degenerates to TWSVMs when the corresponding parameters are chosen, and L1-SVM is a special case of L1-NPSVM.

The paper is organized as follows. Section 2 briefly introduces the initial TWSVM and its improved edition TBSVM (Twin Bounded Support Vector Machine) (Shao et al., 2011). Section 3 proposes the linear L1-NPSVM and its corresponding multi-class model, then its efficient solver—DCD method is proposed. Section 4 deals with experimental results and Section 5 contains concluding remarks.

2. Background

In this section, we briefly introduce two variations of the TWSVM.

2.1. TWSVM

Consider the binary classification problem with the training set

\[ T = \{(x_1, +1), \ldots, (x_p, +1), (x_{p+1}, -1), \ldots, (x_{p+q}, -1)\}, \]

where \( x_i \in \mathbb{R}^n, i = 1, \ldots, p + q \). For the linear case, TWSVM (Jayadeva et al., 2007) seeks two nonparallel hyperplanes

\[ (w_+ \cdot x) + b_+ = 0 \quad \text{and} \quad (w_- \cdot x) + b_- = 0 \]

by solving two QPPs

\[
\min_{w_+, b_+} \frac{1}{2} \sum_{i=1}^{p+q} ((w_+ \cdot x_i) + b_+)^2 + c_1 \sum_{j=p+1}^{p+q} \xi_j, \\
\text{s.t.} \quad (w_+ \cdot x_i) + b_+ \leq 1 - \xi_j, \quad j = p + 1, \ldots, p + q, \quad \xi_j \geq 0, \quad j = p + 1, \ldots, p + q.
\] (6)

and

\[
\min_{w_-, b_-} \frac{1}{2} \sum_{i=1}^{p+q} ((w_- \cdot x_i) + b_-)^2 + c_2 \sum_{j=p+1}^{p+q} \xi_j, \\
\text{s.t.} \quad (w_- \cdot x_i) + b_- \geq 1 - \xi_j, \quad j = 1, \ldots, p, \quad \xi_j \geq 0, \quad j = 1, \ldots, p.
\] (7)

where \( c_i, i = 1, 2 \) are the penalty parameters. The solutions \((w_+, b_+)\) and \((w_-, b_-)\) are derived by solving their dual problems

\[
\min_{\alpha} \frac{1}{2} \alpha^\top (H^\top H)^{-1} \alpha - e_1^\top \alpha, \\
\text{s.t.} \quad 0 \leq \alpha \leq c_1 e_2
\] (8)

and

\[
\min_{\gamma} \frac{1}{2} \gamma^\top (G^\top G)^{-1} \gamma - e_1^\top \gamma, \\
\text{s.t.} \quad 0 \leq \gamma \leq c_2 e_1
\] (9)

where \( \alpha = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n, \gamma = (\gamma_1, \ldots, \gamma_p)^\top \in \mathbb{R}^p, H = [A, e_1] \in \mathbb{R}^{n \times (n + 1)}, G = [B, e_2] \in \mathbb{R}^{p \times (n + 1)}, e_1 = (1, \ldots, 1)^\top \in \mathbb{R}^n, e_2 = (1, \ldots, 1)^\top \in \mathbb{R}^p, A = (x_1, x_2, \ldots, x_p)^\top \in \mathbb{R}^{n \times n}, \) and \( B = (x_{p+1}, x_{p+2}, \ldots, x_{p+q})^\top \in \mathbb{R}^{p \times n} \).

We can see that TWSVM solves two smaller QPPs, which claims 4 times faster than the standard SVM (Jayadeva et al., 2007). Unfortunately, it needs to compute and store the inverse matrices \((H^\top H)^{-1} \) and \((G^\top G)^{-1} \) before training. Since both \(H^\top H\) and \((G^\top G)^{-1}\) are all of order \(n + 1\), TWSVM fails frequently in dealing with problems of high dimensions, such as document classification. Furthermore, in order to deal with the case where \(H^\top H\) or \((G^\top G)^{-1}\) is singular and avoid the possible ill conditioning, the inverse matrices \((H^\top H)^{-1}\) and \((G^\top G)^{-1}\) are approximately replaced by \((H^\top H + \epsilon I)^{-1}\) and \((G^\top G + \epsilon I)^{-1}\) respectively, where \(I\) is an identity matrix of appropriate dimensions, \(\epsilon\) is a positive and small scalar to keep the structure of data. After solving the dual problems (8) and (9), the solutions of problems (6) and (7) can be obtained by

\[
(w_+^*, b_+^*) = -(H^\top H)^{-1}G^\top \alpha, \\
(w_-^*, b_-^*) = -(G^\top G)^{-1}H^\top \gamma.
\] (10)

Thus an unknown point \(x \in \mathbb{R}^n\) is predicted to the Class by

\[
\text{Class} = \arg \min_{k = +, -} |(w_k \cdot x) + b_k|,
\] (12)

where \(|\cdot|\) is the vertical distance of point \(x\) from the planes \((w_k \cdot x) + b_k = 0, k = +, -\).

For the nonlinear case, two kernel-generated surfaces instead of hyperplanes are considered and two other primal problems different with problems (6) and (7) are constructed, which can refer to Jayadeva et al. (2007).

2.2. TBSVM

An improved version of TWSVM, termed as TBSVM, is proposed in Shao et al. (2011) whereas the structural risk is claimed to be minimized by adding a regularization term with the idea of
maximizing some margins. For the linear case, they solve the following two primal problems

\[
\begin{align*}
\min_{w_+, b_+, \xi_+} & \quad \frac{c_1}{2} (\|w_+\|^2 + b_+^2) \\
\text{s.t.} & \quad (w_+ \cdot x_i) + b_+ \leq -1 + \xi_i, \\
& \quad j = p + 1, \ldots, p + q, \\
& \quad \xi_j \geq 0, \quad j = 1, \ldots, p.
\end{align*}
\]

(13)

and

\[
\begin{align*}
\min_{w_-, b_-, \xi_-} & \quad \frac{c_4}{2} (\|w_-\|^2 + b_-^2) \\
\text{s.t.} & \quad (w_- \cdot x_i) + b_- \geq 1 - \xi_i, \\
& \quad j = 1, \ldots, p, \\
& \quad \xi_j \geq 0, \quad j = 1, \ldots, p.
\end{align*}
\]

(14)

Their dual problems are

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \alpha^\top G(H^\top H + c_3 I)^{-1} G^\top \alpha - e_2^\top \alpha, \\
\text{s.t.} & \quad 0 < \alpha < c_1 e_2
\end{align*}
\]

(15)

and

\[
\begin{align*}
\min_{\gamma} & \quad \frac{1}{2} \gamma^\top G(H^\top G + c_4 I)^{-1} H^\top \gamma - e_1^\top \gamma, \\
\text{s.t.} & \quad 0 < \gamma < c_4 e_1
\end{align*}
\]

(16)

Different from problems (8) and (9) with the possibility that \(H^\top H\) or \(G^\top G\) is singular, problems (15) and (16) are derived without any extra assumption and need not be modified any more. From this point of view, TBSVM is more rigorous and complete than TWSVM.

However, TBSVM still need to compute and store the inverse matrices \((H^\top H + c_3 I)^{-1}\) and \((G^\top G + c_4 I)^{-1}\). More unfortunately, for different \(c_3\) and \(c_4\), they have to compute different inverse matrices. It costs a huge amount of computation. For the nonlinear case, similar with TWSVM, two kernel-generated surfaces instead of hyperplanes are considered and two other regularized primal problems are constructed.

3. \(L_1\)-NPSVM

In this section, we propose a novel nonparallel classifier, termed as \(L_1\)-NPSVM, which has several unexpected and incomparable advantages compared with the existing TWSVMs.

3.1. Primal problems

We seek the two nonparallel hyperplanes (5) by solving two convex QPPs

\[
\begin{align*}
\min_{w_+, b_+, \eta_+^\top, \xi_+} & \quad \frac{1}{2} (\|w_+\|^2 + b_+^2) + C_1 \sum_{i=1}^p (\eta_i + \eta_i^*) \\
\text{s.t.} & \quad (w_+ \cdot x_i) + b_+ \leq \varepsilon + \eta_i, \quad i = 1, \ldots, p, \\
& \quad -(w_+ \cdot x_i) - b_+ \leq \varepsilon + \eta_i^*, \quad i = 1, \ldots, p, \\
& \quad (w_+ \cdot x_i) + b_+ \leq -1 + \xi_i, \\
& \quad j = p + 1, \ldots, p + q, \\
& \quad \eta_i, \eta_i^* \geq 0, \quad i = 1, \ldots, p, \\
& \quad \xi_j \geq 0, \quad j = p + 1, \ldots, p + q.
\end{align*}
\]

(17)

and

\[
\begin{align*}
\min_{w_-, b_-, \eta_-^\top, \xi_-} & \quad \frac{1}{2} (\|w_-\|^2 + b_-^2) + C_3 \sum_{i=p+1}^{p+q} (\eta_i + \eta_i^*) \\
\text{s.t.} & \quad (w_- \cdot x_i) + b_- \leq \varepsilon + \eta_i, \quad i = p + 1, \ldots, p + q, \\
& \quad -(w_- \cdot x_i) - b_- \leq \varepsilon + \eta_i^*, \quad i = p + 1, \ldots, p + q, \\
& \quad (w_- \cdot x_i) + b_- \geq 1 - \xi_i, \\
& \quad j = 1, \ldots, p, \\
& \quad \eta_i, \eta_i^* \geq 0, \quad i = p + 1, \ldots, p + q, \\
& \quad \xi_j \geq 0, \quad j = 1, \ldots, p.
\end{align*}
\]

(18)

where \(x_i, i = 1, \ldots, p\) are positive inputs, and \(x_i, i = p + 1, \ldots, p + q\) are negative inputs, \(C_1 \geq 0, i = 1, \ldots, 4\) are penalty parameters, \(\xi_+ = (\xi_1, \ldots, \xi_p)^\top, \xi_- = (\xi_{p+1}, \ldots, \xi_{p+q})^\top, \eta_+^\top = (\eta_1^\top, \ldots, \eta_p^\top)^\top, \eta_-^\top = (\eta_1^\top, \ldots, \eta_q^\top)^\top\) are slack variables.

We can see that the differences between the problems (17) and (18) with the problems (13) and (14) are only the loss functions employed, i.e. \(\varepsilon\)-insensitive loss function (\(L_1\)-Loss): \(\eta_i, \eta_i^*\), \(i = 1, \ldots, p\) are employed to replace the quadratic loss function (\(L_2\)-Loss): \((w_+ \cdot x_i) + b_+^2\); Obviously, when the parameter \(\varepsilon\) is set to be zero, the problems (17) and (18) of \(L_1\)-NPSVM degenerate approximately to the problems (13) and (14) of TBSVM.

3.2. Dual problems

In order to get the solutions of problems (17) and (18), we need to derive their dual problems. The Lagrangian of the problem (17) is given by

\[
L(w_+, b_+, \eta_+^\top, \xi_-, \alpha_+, \lambda_-, \beta_-) = \frac{1}{2} (\|w_+\|^2 + b_+^2) + C_1 \sum_{i=1}^p (\eta_i + \eta_i^*) + C_3 \sum_{j=p+1}^{p+q} \xi_j
\]

\[
+ \sum_{i=1}^p \alpha_i ((w_+ \cdot x_i) + b_+ - \eta_i - \varepsilon) \\
+ \sum_{i=1}^p \alpha_i^* ((w_- \cdot x_i) - b_- - \eta_i^* - \varepsilon) \\
+ \sum_{j=p+1}^{p+q} \beta_j ((w_- \cdot x_j) + b_- + 1 - \xi_j)
\]

\[
- \sum_{i=1}^p \gamma_i \eta_i - \sum_{i=1}^p \gamma_i^* \eta_i^* - \sum_{j=p+1}^{p+q} \lambda_j \xi_j,
\]

(19)

where \(\alpha_+^\top = (\alpha_1^\top, \alpha_2^\top)^\top = (\alpha_1, \ldots, \alpha_p, \alpha_1^*, \ldots, \alpha_p^*)^\top, \gamma_+^\top = (\gamma_1^\top, \ldots, \gamma_p^\top, \gamma_1^*, \ldots, \gamma_p^*)^\top, \beta_- = (\beta_{p+1}, \ldots, \beta_{p+q})^\top\), and \(\lambda_- = (\lambda_{p+1}, \ldots, \lambda_{p+q})^\top\) are the Lagrange multiplier vectors. The Karush–Kuhn–Tucker (KKT) conditions for \(w_+, b_+, \eta_+^\top, \xi_-, \alpha_+, \lambda_-, \beta_-\) are given by

\[
\nabla_{w_+} L = w_+ + \sum_{i=1}^p \alpha_i x_i - \sum_{i=1}^p \alpha_i^* x_i + \sum_{j=p+1}^{p+q} \beta_j = 0,
\]

(20)

\[
\nabla_{b_+} L = b_+ + \sum_{i=1}^p \alpha_i - \sum_{i=1}^p \alpha_i^* + \sum_{j=p+1}^{p+q} \beta_j = 0,
\]

(21)

\[
\nabla_{\eta_+} L = \alpha_+ - \gamma_+ = 0.
\]

(22)
\[
\begin{align*}
\mathbf{V}_q L &= \mathbf{C} \mathbf{e}_r - \alpha^*_+ - \gamma^*_+ = 0, \\
\mathbf{V}_r L &= \mathbf{C} \mathbf{e}_r - \beta_+ - \lambda_- = 0, \\
(w_+ \cdot \mathbf{x}_i) + b_+ \leq \varepsilon + \eta_+, \quad i = 1, \ldots, p, \\
-(w_+ \cdot \mathbf{x}_i) - b_- \leq \varepsilon + \eta_-, \quad i = 1, \ldots, p, \\
(w_+ \cdot \mathbf{x}_j) + b_+ \leq -1 + \xi_+ \quad j = p + 1, \ldots, p + q, \\
\eta_+, \eta_- \geq 0, \quad i = 1, \ldots, p, \\
\xi_+ \geq 0, \quad j = p + 1, \ldots, p + q,
\end{align*}
\]

where \(w_+ = (1, \ldots, 1)^\top \in \mathbb{R}^p, e_+ = (1, \ldots, 1)^\top \in \mathbb{R}^p\). Since \(\gamma_+, \gamma_- \geq 0\), from (22)-(24) we have

\[
0 \leq \alpha_+, \quad \alpha^*_+ \subseteq \mathbf{C} \mathbf{e}_r, \\
0 \leq \beta_- \subseteq \mathbf{C} e_-, \\
\text{From (20) and (21), we have}
\]

\[
w_+ = \sum_{i=1}^{p} (\alpha_i^* - \alpha_i) x_i - \sum_{j=p+1}^{p+q} \beta_j x_j, \\
b_+ = \sum_{i=1}^{p} (\alpha_i^* - \alpha_i) - \sum_{j=p+1}^{p+q} \beta_j.
\]

Then putting (32) and (33) into the Lagrangian (19) and using (20)-(29), we obtain the dual problem of the problem (17) as

\[
\begin{align*}
\min_{\alpha^*_+, \beta_-} & \quad \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)((x_i \cdot x_j) + 1) \\
& - \sum_{i=1}^{p} \sum_{j=p+1}^{p+q} (\alpha_i^* - \alpha_i) \beta_j ((x_i \cdot x_j) + 1) \\
& + \frac{1}{2} \sum_{i=p+1}^{p+q} \sum_{j=p+1}^{p+q} \beta_i \beta_j ((x_i \cdot x_j) + 1) \\
& + \varepsilon \sum_{i=1}^{p} (\alpha_i^* + \alpha_i) - \sum_{j=p+1}^{p+q} \beta_j, \\
\text{s.t.} & \quad 0 \leq \alpha^*_+, \quad \alpha^*_+ \subseteq \mathbf{C} \mathbf{e}_r, \\
& \quad 0 \leq \beta_- \subseteq \mathbf{C} e_-
\end{align*}
\]

Problem (34) can be further concisely formulated as

\[
\begin{align*}
\min_{\alpha^*_+, \beta_-} & \quad \frac{1}{2} (\alpha_+^* - \alpha_+)(\mathbf{A}^\top + \varepsilon - \mathbf{A})(\alpha_+^* - \alpha_+) - (\alpha_+^* - \alpha_+)^\top \\
& \times (\mathbf{A}^\top + \varepsilon) \beta_- + \frac{1}{2} \mathbf{b}^\top \varepsilon \mathbf{b} (\mathbf{B}^\top + \varepsilon) \beta_- \\
& + \varepsilon \mathbf{e}_+^\top (\alpha_+^* + \alpha_+) - \mathbf{e}_+^\top \beta_-, \\
\text{s.t.} & \quad 0 \leq \alpha^*_+, \quad \alpha^*_+ \subseteq \mathbf{C} \mathbf{e}_r, \\
& \quad 0 \leq \beta_- \subseteq \mathbf{C} e_-
\end{align*}
\]

where \(\mathbf{e}_+ = (1, \ldots, 1)^\top \in \mathbb{R}^p, \mathbf{e}_- = (1, \ldots, 1)^\top \in \mathbb{R}^p, \mathbf{A} = (x_1, x_2, \ldots, x_p)^\top \in \mathbb{R}^{p \times n}, \mathbf{B} = (x_{p+1}, x_{p+2}, \ldots, x_{p+q})^\top \in \mathbb{R}^{p \times n}, \mathbf{E}
\]

is the matrix of appropriate dimensions with all entries equal to one.

Furthermore, let

\[
\mathbf{Q}_1 = \mathbf{A}^\top + \varepsilon \in \mathbb{R}^{n \times p},
\]

\[
\mathbf{Q}_2 = \mathbf{A}^\top + \varepsilon \in \mathbb{R}^{n \times q},
\]

\[
\tilde{\mathbf{Q}}_2 = \mathbf{AB}^\top + \varepsilon \in \mathbb{R}^{n \times q},
\]

\[
\mathbf{Q}_3 = \mathbf{A}^\top + \varepsilon \in \mathbb{R}^{n \times p},
\]

\[
\mathbf{Q}_4 = \mathbf{B}^\top + \varepsilon \in \mathbb{R}^{n \times p},
\]

\[
\mathbf{Q}_5 = \mathbf{A}^\top + \varepsilon \in \mathbb{R}^{n \times q},
\]

\[
\mathbf{Q}_6 = \mathbf{B}^\top + \varepsilon \in \mathbb{R}^{n \times q},
\]

3.3. **Linear L1-NPSVM**

The solutions of problems (17) and (18) can be obtained from the solutions of problems (43) and (44) based on the following theorems, thus the linear L1-NPSVM is constructed.
Theorem 3.1. Suppose that $\hat{\lambda} = (\alpha_1^+, \alpha_1^-, \beta_1^T)^T$ is a solution of the problem (43), then the solution $(w_+, b_+)$ of the problem (17) can be obtained in the following way:

$$w_+ = \sum_{i=1}^{p+q} (\alpha_i^+ - \alpha_i)x_i - \sum_{j=p+1}^{p+q} \beta_j x_j,$$

and

$$b_+ = \sum_{i=1}^{p+q} (\alpha_i^+ - \alpha_i) - \sum_{j=p+1}^{p+q} \beta_j .$$

Theorem 3.2. Suppose that $\hat{\lambda} = (\alpha_1^+, \alpha_1^-, \beta_1^T)^T$ is a solution of the problem (44), then the solution $(w_-, b_-)$ of the problem (18) can be obtained in the following way:

$$w_- = \sum_{i=1}^{p+q} (\alpha_i^+ - \alpha_i)x_i + \sum_{j=1}^{p} \beta_j x_j ,$$

and

$$b_- = \sum_{i=1}^{p+q} (\alpha_i^+ - \alpha_i) + \sum_{j=1}^{p} \beta_j .$$

Linear $L_1$-NPSVM

(1) Input the training set (4);
(2) Choose appropriate parameters $\varepsilon \geq 0, C_1, C_2 \geq 0$ for problem (43), $C_3, C_4 \geq 0$ for problem (44) respectively;
(3) Solve the two convex QPPs (43) and (44) respectively, get the solutions $\alpha^{(\varepsilon)} = (\alpha_1, \ldots, \alpha_{p+q}, \alpha_1^+, \ldots, \alpha_{p+q}^+)^T$ and $\beta = (\beta_1, \ldots, \beta_{p+q})^T$, get the solutions $(w_+, b_+)$ and $(w_-, b_-)$ by Theorems 3.1 and 3.2;
(4) Construct two decision functions

$$f_+(x) = \langle w_+ \cdot x \rangle + b_+$$

and

$$f_-(x) = \langle w_- \cdot x \rangle + b_- .$$

(5) For any new input $x$, assign it to the class $k(k=+, -)$ by

$$\text{Class} = \arg \min_{k=+, -} |\langle w_k \cdot x \rangle + b_k| ,$$

where $|\cdot|$ is the vertical distance of point $x$ from the hyperplanes $\langle w_k \cdot x \rangle + b_k = 0, k = +, -.$

It is easy to see that if we append each instance with an additional dimension

$$x_i^T \leftarrow (x_i^T, 1),$$

and let

$$\tilde{w}_+ = (w_+ \cdot x) + b_+ \quad \tilde{w}_- = (w_- \cdot x) + b_-,$$

(52) and (53) can be combined as

$$\tilde{w}_+ = \sum_{i=1}^{p} (\alpha_i^+ - \alpha_i)x_i - \sum_{j=p+1}^{p+q} \beta_j x_j ,$$

also (54) and (55) can be combined as

$$\tilde{w}_- = \sum_{i=1}^{p+q} (\alpha_i^+ - \alpha_i)x_i + \sum_{j=1}^{p} \beta_j x_j .$$

3.4. Advantages of linear $L_1$-NPSVM

In this section we discuss the linear $L_1$-NPSVM compared with the linear TW SVM, TBSVM and $L_1$-SVM.

• (i) If we set $\epsilon = 0$, and take the $L_2$-loss function $\sum (\eta_i^2 + \eta_i^*)2$ instead of the $L_1$-loss function $\sum (\eta_i + \eta_i^*)$, the problems (17) and (18) are equivalent to the problems (13) and (14) obviously, i.e. $L_1$-NPSVM degenerates to TBSVM, further to TWSVM when the penalty parameters are large enough; (ii) If we combine the problems (17) and (18) to be the following problem

$$\min_{w_+, w_-\ldots b_+, b_-\ldots \eta_i\eta_i^*} \frac{1}{2} (\|w_+\|^2 + b_+^2) + \frac{1}{2} (\|w_-\|^2 + b_-^2) +$$

$$C_1 \sum_{i=1}^{p} (1 + \eta_i)^2 + C_2 \sum_{j=p+1}^{p+q} \xi_j + C_3 \sum_{i=1}^{p+q} (1 + \eta_i^*)^2,$$

s.t. $(w_+ \cdot x_i) + b_+ \leq \varepsilon + \eta_i, \quad i = 1, \ldots, p, (w_- \cdot x_i) + b_- \leq \varepsilon + \eta_i^*, \quad i = 1, \ldots, p, (w_+ \cdot x_i) + b_+ \geq \varepsilon + \eta_i, \quad i = p+1, \ldots, p+q,$

• (iii) If we set $\epsilon = 0$ and $C_1 = C_3 = 0, C_2 = C_4 = C$, add an additional constraint $w_+ = w_-, b_+ = b_-$, the problem (63) turns to be

$$\min_{w, b, \xi_i} \frac{1}{2} \sum_{j=1}^{p+q} (\|w\|^2 + b^2) + C \sum_{j=1}^{p+q} \xi_j ,$$

s.t. $(w \cdot x_j) + b \leq -1 + \xi_j, \quad j = p+1, \ldots, p+q,$

which is obviously the $L_1$-SVM. In other words, the $L_1$-SVM with parallel hyperplane is a special case of $L_1$-NPSVM with nonparallel hyperplanes.

• Although TWSVM and TBSVM solve small QPPs in which the SOR technique can be applied (Shao et al., 2011), they have to compute the inverse matrices $(H^T H + \epsilon I)^{-1}$ and $(G G^T + \epsilon I)^{-1}$ before training which is in practice intractable or even impossible for large-scale datasets. Since the inverse matrices are avoided, $L_1$-NPSVM can be solved efficiently by the DCD method, which will be discussed in detail in the next section. Suppose the size of negative training set is roughly equal to the size of positive set, i.e. $p \approx q \approx \frac{1}{4} l$, then the variable in $L_1$-NPSVM ($2p + q$ of the problem (43) or $2q + p$ of the problem (44)) is almost 3 times of the variable in TWSVM or TBSVM, which means that $L_1$-NPSVM sacrifices more training time to skillfully avoid the computation of the inverse matrix. The main difference between $L_1$-NPSVM and $L_1$-SVM is that the scale of $L_1$-NPSVM is almost 1.5 times of $L_1$-SVM since it has $2p + q$ variables and $L_1$-SVM has $p + q$ variables. It means that $L_1$-NPSVM sacrifices more training time to get better performance.
• Both TWSVM and TBSVM lost the sparseness because of the quadratic loss function making the proximal hyperplane close enough to the class itself, while $L_1$-NPSVM has the inherent sparseness by applying the $\varepsilon$-insensitive loss function and the soft-margin loss function. We can easily define the support vector of the problem (43) in $L_1$-NPSVM. Suppose that $\lambda_i = (\alpha_i^+, \alpha_i^-, \beta_i)^T$ is a solution of the problem (43). The input $x_i$ associated with the training point $(x_i, y_i)$ is said to be a support vector if the corresponding component $\lambda_i$ of $\lambda$ is nonzero and otherwise it is a non-support vector.

3.5. Multi-class linear $L_1$-NPSVM

It is easy to extend the above linear $L_1$-NPSVM to deal with the multi-class classification problem with training set

$$T = \{(x_1, y_1), \ldots, (x_l, y_l)\} \in (R^d \times R)^l,$$

(65)

where $x_i \in R^d$, $y_i \in \{1, \ldots, l\}$, $i = 1, \ldots, l$. Follow the “one-versus-rest” method, it constructs $M$ nonparallel hyperplanes, one for each class, which results in solving $M$ QPPs simultaneously and assigns the class label according to which hyperplane is nearest to for a new input $x$.

More precisely, taking the $m$-th class $T_m$ as an example, suppose that the $m$-th hyperplane is

$$(w_m \cdot x) + b_m = 0,$$

(66)

with $m = 1, \ldots, M$. It is required that the patterns in the $m$-th class are as far as possible in the $\varepsilon$-band of the $m$-th hyperplane, while the patterns in the rest $M - 1$ classes are as far as possible from the $m$-th hyperplane. This leads to the following QPP

$$\min_{w_m \in R^d, b_m, \xi_m} \frac{1}{2} \left(\|w_m\|^2 + b_m^2\right) + C_m \sum_{i \in T_m} (y_i + \eta_i^+ - \eta_i^-)$$

$$+ \xi_m \sum_{i \in \{T \setminus T_m\}} \xi_i,$$

s.t.

$$(w_m \cdot x_i) + b_m \leq \varepsilon + \eta_i^+, \quad i \in T_m,$$

$$(w_m \cdot x_i) + b_m \geq \varepsilon - \eta_i^- + \xi_m, \quad i \in \{T \setminus T_m\},$$

$$\eta_i^+ \geq 0, \quad i \in T_m,$$

$$\xi_i \geq 0, \quad i \in \{T \setminus T_m\}.$$  (67)

After getting the solution $w_m, b_m$ of the above QPP with $m = 1, \ldots, M$, a new pattern $x \in R^d$ is assigned to class $m(m \in \{1, \ldots, M\}$, depending on which of the $M$ hyperplanes given by (66) it lies nearest to, i.e., the decision function is represented as

$$f(x) = \arg \min_{m = 1, \ldots, M} |(w_m \cdot x) + b_m|,$$

(68)

where $| \cdot |$ is the vertical distance of $x$ from the hyperplanes $(w_m \cdot x) + b_m = 0$.

3.6. DCD method for linear $L_1$-NPSVM

Coordinate descent, a popular optimization technique, updates one variable at a time by minimizing a single-variable subproblem. If one can efficiently solve this sub-problem, then it can be a competitive optimization method. Hsieh et al. (2008) proposed a DCD method for $L_1$-SVM, and pointed out that the DCD method makes crucial advantage of the linear kernel and outperforms other solvers when the numbers of data and features are both large. Since the dual problems (43) and (44) have the same formulations with that of standard $L_1$-SVM, i.e. the problem (3), we can apply the DCD method to solve them with several minor modifications. As the two problems (43) and (44) can be solved similarly, now we describe the coordinate descent method for the problem (43) as an example.

First, let us compare the problem (43) with the problem (3). Based on Theorem 3.1, we get Eq. (69)

$$\tilde{w}_i = \sum_{j = 1}^p (\alpha_{ij}^+ - \alpha_{ij}^-) x_i - \sum_{j = p + 1}^{p+q} \beta_{ij} x_i$$

$$= \sum_{i = 1}^p \alpha_{i}^+ x_i - \left(\sum_{i = 1}^p \alpha_{i}^- x_i + \sum_{j = p + 1}^{p+q} \beta_{ij} x_i\right),$$

(69)

which enlightens us that we can construct a new training set

$$\tilde{T} = \{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_l, \tilde{y}_l)\} = \{(x_1, 1), \ldots, (x_p, 1), (x_1, -1), \ldots, (x_p, -1), (x_{p+1}, -1), \ldots, (x_{p+q}, -1)\},$$

which has $l = 2p + q$ training points (p positive and $p + q$ negative). If we apply $L_1$-SVM for this training set, we will get the dual problem (3)

$$\min_{\tilde{\alpha}^+} \frac{1}{2} \tilde{\alpha}^+ \tilde{Q} \tilde{\alpha} - \tilde{\varepsilon}^T \tilde{\alpha},$$

(70)

s.t.

$$0 \leq \tilde{\alpha}^+ \leq \tilde{C}, \quad i = 1, \ldots, \tilde{l},$$

where $\tilde{\alpha}$ and $\tilde{Q}$ are just the $\lambda$ and $\tilde{A}$ in the problem (43) respectively, and $\tilde{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_l)^T$, $\tilde{C} = (\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_l)^T$. Obviously, the differences between the problem (43) and the problem (70) are only that the $\tilde{k}$ in the problem (43) changed to be the $\tilde{\varepsilon}$ in the problem (70), and also the different $\tilde{C}$ used. Therefore, if we want to apply the DCD method to solve the problem (43), we only need to change the training set (4) to be the new training set $\tilde{T}$, and then follow the process of DCD for $L_1$-SVM with several minor modifications.

The optimization process starts from an initial point $\tilde{x}_0 = (\alpha_{i_0}^+, \alpha_{i_0}^-, \beta_{i_0})^0 \in R^d$ where $\tilde{l} = 2p + q$ and generates a sequence of vectors $(\tilde{x}_k)_{k=0}^\infty$. We refer to the process from $\tilde{x}_k$ to $\tilde{x}_{k+1}$ as an outer iteration. In each outer iteration it has $\tilde{l}$ inner iterations, so that sequentially $\lambda_1, \ldots, \lambda_\tilde{l}$ are updated. Therefore, similar with the DCD method for $L_1$-SVM, we propose a DCD method for solving the problem (43) of Linear $L_1$-NPSVM, where FG means the projected gradient.

4. Experimental results

Since the superiority of TBSVM to TWSVM are proved by Shao et al. (2011), in this section, experimental results related to the proposed $L_1$-NPSVM, TBSVM and $L_1$-SVM are preferred to demonstrate their performance. All experiments reported are carried out on an Intel Dual Core 3.3 GHz, 4 GB RAM with operating system of Fedora Core 16 × 64. To make fair comparisons, we have implemented the proposed method by both MATLAB and C++.
DCD method for problem (43) of Linear $L_1$-NPSVM

(1) For the training set (4), first append each instance with an additional dimension
\[ x_i^T \leftarrow (x_i^T, 1), \]  
then change it to be a new training set
\[ \tilde{x} = ((\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_p, \tilde{y}_p)) \]
\[ = ((x_1, 1), \ldots, (x_p, 1), (x_1, -1), \ldots, (x_p, -1), \]
\[ (x_{p+1}, -1), \ldots, (x_{p+q}, -1)); \]

(2) Given an initial point $\tilde{\lambda} = (\alpha^*_1, \alpha^*_2, \beta) \in \mathbb{R}^l$, compute the corresponding $\tilde{w}_i$ based on (61);

(3) While $\tilde{\lambda} = (\alpha^*_1, \alpha^*_2, \beta)$ is not optimal, i.e., not satisfying the KKT conditions of the problem
   
   For $i = 1, \ldots, l$
   
   (a) $\tilde{\lambda}_i \leftarrow \tilde{\lambda}_i$;
   
   (b) $G = y_i(\tilde{w}_i \cdot \tilde{x}_i) + \tilde{\eta}_i$, where $\tilde{\eta}$ is defined by (40);
   
   (c) $\Delta G = \min\{G, 0\}$, if $\tilde{\lambda}_i = 0,
   
   \max\{G, 0\}$, if $\tilde{\lambda}_i = \tilde{C}_i$,
   
   $G$, if $0 < \tilde{\lambda}_i < \tilde{C}_i$
   
   (d) If $|\Delta G| \neq 0$, $\tilde{\lambda}_i \leftarrow \min(\tilde{\lambda}_i - C/\tilde{\lambda}_i, 0), \tilde{C}_i);$
   
   $\tilde{w}_i \leftarrow \tilde{w}_i + (\tilde{\lambda}_i - \tilde{\eta}_i)\tilde{x}_i$.

Due to the computation of inverse matrices required by TBSVM, comparison on small-scale datasets are conducted in MATLAB 2012b among the three methods, whereas the linear $L_1$-NPSVM and $L_1$-SVM are rewritten with C++ for comparing on large-scale datasets. The performance of these methods is evaluated in terms of the metric of accuracy (denoted by $a$), i.e., $a = (f_{++} + f_{--})/(f_{++} + f_{--} + f_{+-} + f_{-+}$), where $f_{++}$, $f_{--}$, $f_{+-}$ and $f_{-+}$ represent the number of true positive, false positive, true negative and false negative, respectively.

4.2. Comparison on large-scale datasets

Since it is impractical for TBSVM to dealing with large-scale datasets, in this section, we conduct comparisons between $L_1$-NPSVM and $L_1$-SVM. The employed large-scale datasets, with large size in instance, dimension or both, include text, uspset, handwritre, a9a and shuttle from UCI repository (Frank & Asuncion, 2010), news20.binary from Keerthi and DeCoste (2005), rcv1.binary from Lewis, Yang, Rose, and Li (2004), real-sim from McCallum (2012) and kddcup99. Notice that a9a, news20.binary, rcv1.binary and real-sim are pre-processed by Hsieh et al. (2008) while kddcup99 is the 10% of KDD-CUP 99 training data (Hettich & Bay, 1999) with only 9 fields (namely Dst bytes, Logged in, Count, Srv count, Dst host count, Dst host srv count, Dst host diff srv rate, Dst host same src port rate and Dst host srv diff host rate) selected in this study.

Table 2 depicts the fivefold testing percentage accuracy and time cost for $L_1$-NPSVM and $L_1$-SVM. Similar to the previous section, the standard deviation is reported as zero if it is lower than $10^{-5}$. For $L_1$-NPSVM, the value of C selected for each dataset is given. Obviously, in terms of time cost, $L_1$-SVM performs better than the proposed $L_1$-NPSVM. However, $L_1$-NPSVM obtains the best accuracy in seven out of ten cases, except for handwritre, a9a and rcv1.binary. And yet the accuracies achieved by $L_1$-NPSVM on these datasets are very close to that of reached by $L_1$-SVM. All of these results show the significant advantages of $L_1$-NPSVM in terms of accuracy in comparison of $L_1$-SVM. As shown in Table 2, the value of C is suggested to be 1 when dealing with large-scale data.

5. Conclusion

In this paper, we have proposed a novel nonparallel support vector machine, $L_1$-NPSVM, for binary classification. $L_1$-NPSVM has several advantages over existing TWSVMS: (1) $L_1$-NPSVM degenerates to TWSVMS when the corresponding parameters are chosen, and $L_1$-SVM is a special case of $L_1$-NPSVM; (2) The DCD method for large-scale data can be easily applied for it; (3) It can reach the similar sparseness with the standard SVMs. $L_1$-NPSVM is very simple to implement, and possesses sound optimization properties. Experiments show that our method is not only suitable for large scale problems, but also performs no less than linear TWSVMS and SVMs. Since solving the linear $L_1$-NPSVM is mainly concerned in this study, we prefer not to compare it with nonlinear SVMs for fairness, this issue will be considered in our future works.
Table 1  
Tenfold testing percentage accuracy on small-scale data.

| Datasets | $L_1$-NPSVM | |  | $L_1$-SVM | |  | TBSVM | |
|----------|-------------|---|---|------------|---|---|--------|
|          | Accuracy %  | Time (s)/C | $N_{tr}$ (%) | Accuracy %  | Time (s)/C | $N_{tr}$ (%) | Accuracy %  | Time (s) |
| Sonar    | 77.4038 ± 0 | 0.0360/0.8 | 76.11 | 76.9231 ± 0 | 0.0142/*0.5 | 83.07 | 79.5673 ± 2.2215 | 0.3486 |
| (208 × 60) | | | | | | | | |
| Heart-Statlog | 82.8620 ± 1.5237 | 0.2999/2.76214e−03 | 76.54 | 81.1612 ± 2.4758 | 0.2011/*7.8125e−03 | 90.62 | 78.4074 ± 1.2818 | 0.6781 |
| (270 × 13) | | | | | | | | |
| Ionosphere | 89.1738 ± 0 | 0.0027*/1 | 73.75 | 73.7892 ± 0 | 0.0030/8 | 69.12 | 86.6667 ± 0.4293 | 0.0712 |
| (351 × 34) | | | | | | | | |
| Solar | 58.9385 ± 0 | 0.0056/1.5625e−02 | 87.06 | 58.9385 ± 0 | 0.0038*/3.90625e−03 | 90.37 | 60.8939 ± 0.4682 | 0.2196 |
| (358 × 10) | | | | | | | | |
| Image | 99.5000 ± 0 | 0.0438*/6.10352e−05 | 73.83 | 99.7424 ± 0.0826 | 0.0941/128 | 6.08 | 99.6833 ± 0.0151 | 0.3166 |
| (600 × 19) | | | | | | | | |
| German | 75.8000 ± 0.1309 | 0.4402*/7.8125e−03 | 53.42 | 75.2273 ± 0.0965 | 0.6746/0.0625 | 60.67 | 77.1900 ± 0.1832 | 1.2093 |
| (1000 × 24) | | | | | | | | |
| Spice | 84.6103 ± 0 | 0.0634*/6.90534e−04 | 71.76 | 84.7413 ± 0 | 0.1699/7.8125e−03 | 59.68 | 89.1028 ± 0.0402 | 1.3902 |
| (1527 × 60) | | | | | | | | |
| Titanic | 77.6011 ± 0 | 0.0105/3.45267e−04 | 84.35 | 77.1013 ± 0 | 0.0065*/3.90625e−03 | 77.33 | 78.2917 ± 0.0036 | 2.4988 |
| (2201 × 3) | | | | | | | | |
| Waveform | 92.4952 ± 0 | 0.0357/4.88281e−04 | 42.27 | 91.4044 ± 0 | 0.0416*/1.38107e−03 | 50.23 | 92.0097 ± 0.0063 | 5.4230 |
| (3304 × 21) | | | | | | | | |
| Twnorm | 97.8714 ± 0.0002 | 0.1175/3.05176e−05 | 100.00 | 97.8649 ± 0 | 0.1203*/0.125 | 18.80 | 97.8405 ± 0.0012 | 33.8701 |

Table 2  
Fivefold testing percentage accuracy on large-scale data with multiple classes.

| Datasets | # of class $M$ | $L_1$-NPSVM | |  | $L_1$-SVM | |  | |
|----------|---------------|-------------|---|---|------------|---|---|
|          | Accuracy %    | Time (s)/C  | $N_{tr}$ (%) | Accuracy %    | Time (s)/C  | $N_{tr}$ (%) | |
| text     | 2             | 97.7576 ± 0.0188 | 0.1161/0.25 | 97.6362 ± 0 | 0.0449/512 |
| (1946 × 7511) | | | | | | | |
| upsst    | 10            | 92.1774 ± 0.1246 | 365.48/1 | 91.8376 ± 0.0432 | 12.2557/1 |
| (2007 × 256) | | | | | | | |
| handwrite | 10            | 93.2068 ± 0.8795 | 0.1682/3.45267e−04 | 93.8561 ± 0.0174 | 0.0641/1.95312e−03 |
| (3823 × 64) | | | | | | | |
| news20.binary | 20          | 96.9421 ± 0.0071 | 99.7285/1 | 96.7230 ± 0.0033 | 6.7233/1 |
| (19,996 × 1,355,191) | | | | | | | |
| a9a      | 2             | 81.3120 ± 0.0168 | 15.2948/1 | 84.7601 ± 0.0012 | 3.5593/1 |
| (48,842 × 123) | | | | | | | |
| shuttle  | 7             | 91.8511 ± 0.1089 | 1317.8200/3.05176e−05 | 90.8806 ± 0.0901 | 363.4860/1.5625e−02 |
| (59,000 × 9) | | | | | | | |
| ucla     | 2             | 97.479 ± 3.2726 | 79.8357/1 | 91.2491 ± 3.2719 | 3.4958/16 |
| (64,700 × 300) | | | | | | | |
| real-sim | 2             | 97.6774 ± 0.0012 | 12.1824/1 | 97.4094 ± 0.0019 | 2.9033/1 |
| (72,309 × 20,958) | | | | | | | |
| kddcup99 | 5             | 98.2262 ± 0 | 787.1410/1 | 98.2205 ± 0 | 60.6678/1 |
| (494,021 × 9) | | | | | | | |
| rvc1.binary | 2           | 97.9758 ± 0 | 200.863/1 | 97.7844 ± 0 | 39.8385/1 |
| (677,399 × 47,236) | | | | | | | |

Acknowledgments

This work has been partially supported by grants from National Natural Science Foundation of China (No. 11271361, No. 70921061, No. 61303232), the CAS/SAFEA International Partnership Program for Creative Research Teams, Major International (Regional) Joint Research Project (No. 711101070206), the Ministry of Water Resources’ special funds for scientific research on public causes (No. 201301094), the Foundation of He’nan Educational Committee (No. 13A413750, No. 13A413747), and the Natural Science Foundation of He’nan Province of China (No. 132300410393).
References


