Soft Comput (2018) 22:1945–1957
https://doi.org/10.1007/s00500-016-2455-9

METHODOLOGIES AND APPLICATION

Large-scale linear nonparallel SVMs

Dalian Liu[1,2] · **Dewei Li**[3,4] · **Yong Shi**[1,4,5,6] · **Yingjie Tian**[4,5]

Published online: 19 December 2016

Abstract Large-scale problems have been a very active topic in machine learning area. In the time of big data, it is a challenge and meaningful work to solve such problems. Standard SVM can make linear classification on large-scale problems effectively, with acceptable training time and excellent prediction accuracy. However, nonparallel SVM (NPSVM) and ramp loss nonparallel SVM (RNPSVM) are proposed with better performance than SVM on benchmark datasets. It is motivated to introduce NPSVMs into the area of large-scale issues. In this paper, we propose large-scale linear NPSVMs, solved by the alternating direction method of multipliers (ADMM), to handle large-scale classification problems. ADMM breaks large problems into smaller pieces, avoiding solving intractable problems and leading to higher training speed. The primal problems of NPSVM are convex and differentiable, and they can be managed directly by ADMM. But the objective functions of RNPSVM, composed of convex ones and concave ones, should first be processed by CCCP algorithm and transformed as a series of convex programs. Then, we apply ADMM to solve these programs in every iteration. Experiments of NPSVMs on large-scale problems verify that the algorithms can classify large-scale tasks effectively.

Keywords Large-scale · Nonparallel SVM · Ramp loss function · ADMM

Communicated by V. Loia.

✉ Yong Shi
yshi@ucas.ac.cn

Dalian Liu
ldlluck@sina.com

Dewei Li
lidewei15@mails.ucas.ac.cn

Yingjie Tian
tyj@ucas.ac.cn

1 School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

2 Department of Basic Course Teaching, Beijing Union University, Beijing 100101, China

3 School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

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

5 Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China

6 College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182, USA

1 Introduction

Support vector machines (SVMs), constructed on the bases of statistical learning theory and VC-dimensional theory, are popular methods in handling problems of classification, regression, clustering, etc. SVM has attracted many interests since its original proposed (Cortes and Vapnik 1995; Vapnik 1998, 2000; Deng and Tian 2004; Deng et al. 2012). As a hot spot in data mining, SVM has been applied in many real applications (Akbani et al. 2004; Weston et al. 1999; Bradley and Mangasarian 1998; Guyon et al. 2002; Zhang et al. 2006; Tan et al. 2010). For a binary classification problem, SVM seeks for two best parallel support hyperplanes with maximal margin by constructing and solving a quadratic programming problem with hinge loss function. Recently, SVM has been improved in two sides, including hyperplanes and loss function. To advance the generalization ability of SVM, twin support vector machine (TWSVM)

(Jayadeva et al. 2007; Shao et al. 2011; Tian et al. 2014a) was proposed to find two nonparallel hyperplanes, satisfying that each hyperplane be close to its corresponding class as much as possible and far from the other class beyond one unit distance. TWSVM employs the information of all the inputs to make exacter prediction, and it is nearly four time faster than SVM since it solves two smaller convex problems. TWSVMs have been researched widely owing to their advantages in both classification performance and training speed (Qi et al. 2013, 2012; Kumar and Gopal 2008; Arun Kumar and Gopal 2009; Naik et al. 2010). Based upon TWSVM, the nonparallel support vector machine (NPSVM) (Tian et al. 2014b,c, 2016; Tian and Ping 2014; Qi et al. 2014) was presented to overcome the drawbacks existing in TWSVMs, including computing inverse matrices, lack of sparseness, not considering structural risk, etc. NPSVM applies hinge loss function and $\varepsilon-$insensitive loss function to make improvements on TWSVM. But TWSVM and NPSVM are both sensitive to outliers. Ramp loss function has been introduced into NPSVM to replace the hinge loss function, and then a new robust NPSVM, called RNPSVM, was proposed (Liu et al. 2015). RNPSVM can eliminate the negative impact of outliers and noises and obtain higher classification accuracy with less support vectors. CCCP algorithm is implemented to solve RNPSVM since the primal problems are nonconvex and nondifferentiable.

However, in the big data era, some applications appear with a large number of instances or high-dimensional features which cannot be managed by traditional quadratic programming problems (QPPs). Within the scope of SVM, many algorithms for large-scale problems are proposed. LIBLINEAR is a fast method for large-scale linear classification (Fan et al. 2008). It applies trust region method for optimization and runs much faster than QPPs when dealing with large-scale datasets. Many applications have been benefited from LIBLINEAR (Deng et al. 2010; Maas et al. 2011; Ma et al. 2009). However, up to now, the research on large-scale problems using nonparallel support vector machines is not active. L_1-NPSVM (Tian and Ping 2014) applies dual coordinate descent(DCD) method to solve NPSVM, and the experiments on large-scale datasets have shown its strong ability in classifying such problems. DCNPSVM (Tian et al. 2016) constructs a multi-level structure with a division step and a combination step. Large-scale original problems have first been divided into smaller subproblems, and then the subsolutions are combined to form the final solution for the large problems. The method converges quickly and performs better than state-of-the-art methods. Recently, the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) was studied comprehensively and claimed to be suited to convex optimization. ADMM solves convex problems by breaking them into smaller-scale pieces, each of which can be solved more efficiently. The characteristics make great contribution to solving large-scale problems well. ADMM has been extensively applied in many areas (Li et al. 2012; Bhaskar et al. 2013; Kasiviswanathan et al. 2011).

In this paper, ADMM is applied for linear NPSVM and RNPSVM to solve large-scale problems in light of their advantages in classification. ADMM can solve the primal problems of NPSVM directly for they are both convex ones. For RNPSVM, CCCP procedure is first applied to make the primal problems convex and differentiable. Then, a sequence of convex QPPs is obtained and every transformed problem can be solved by ADMM. Our methods have the following advantages: (1) They can solve large-scale problems effectively, with the first time to introduce ADMM into nonparallel SVMs; (2) RNPSVM with ADMM is insensitive to noises and outliers; (3) They have the property of sparseness to get faster prediction for test points. Numerical experiments are made to verify the ability of our novel algorithms.

The paper is structured as follows. Background on the standard SVM, NPSVM and RNPSVM are introduced in Sect. 2. Section 3 proposes ADMM for linear NPSVM, and Sect. 4 presents ADMM for RNPSVM with CCCP. Experimental results and conclusions are summarized in Sects. 5 and 6, respectively.

2 Background

In this section, we will give a brief introduction of some previous works, including standard SVM, NPSVM, RNPSVM and ADMM. The main principles and analyses will be provided.

2.1 Standard SVM

Consider a binary classification problem with the following training set

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

where $x_i \in R^n, i = 1, \ldots, p+q$. The standard SVM seeks for the best decision hyperplanes $f(x) = w^\top x + b = 0$ by solving a convex QPP as follows

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{p+q}\xi_i \tag{2}$$

$$s.t. \ \ y_i(w^\top x_i + b) \geq 1 - \xi_i, i = 1, \ldots, p+q, \tag{3}$$

$$\xi_i \geq 0, i = 1, \ldots, p+q \tag{4}$$

The above problem can be reformulated as an unconstrained convex problem with hinge loss function

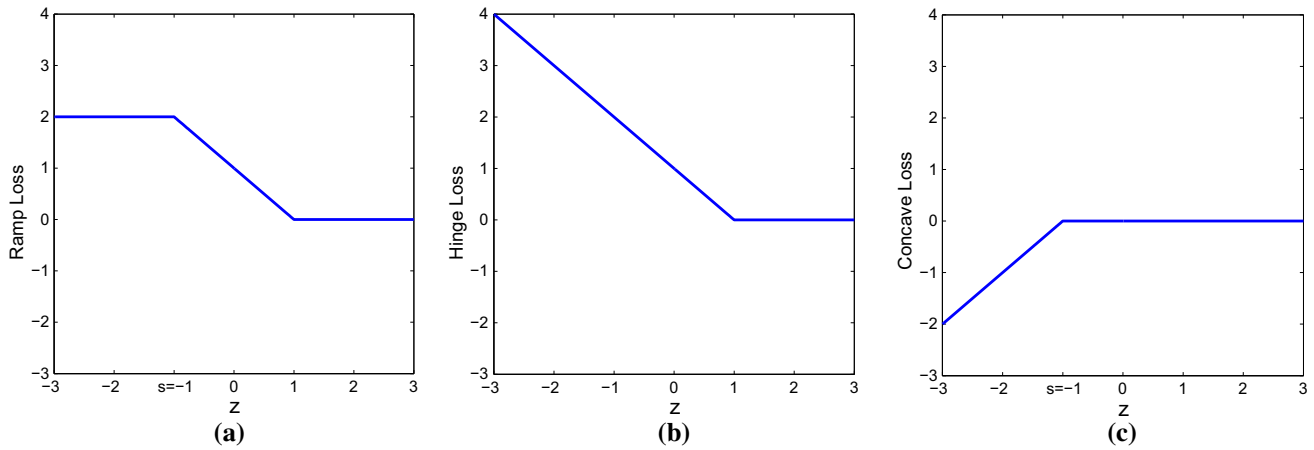

Fig. 1 A ramp loss function can be decomposed into the sum of a hinge loss function and a concave loss function. **a** Ramp loss. **b** Hinge loss. **c** Concave loss

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{p+q} max(0, 1 - y_i(w^\top x_i + b)) \tag{5}$$

The classical hinge loss function $H_\varepsilon(z) = max(0, \varepsilon - z) \triangleq [\varepsilon - z]_+$(Fig. 1b) penalizes the points which invade the margin between two support hyperplanes. Standard SVM has been extensively studied since it has graceful formation and excellent performance in classification. However, it has two drawbacks related to hinge loss function. First, its prediction is only decided by the support vectors, resulting in insufficient extraction of the data information. Second, the classification performance will decline when there exist outliers because SVM is sensitive to outlier patterns.

2.2 NPSVM

Instead of solving a single QPP, NPSVM aims to find two nonparallel hyperplanes $f_+(x) = w_+^\top x + b_+ = 0$ and $f_-(x) = w_-^\top x + b_- = 0$ for the classification problem with the training set (1) which lead to the construction of two smaller convex QPPs

$$\min_{w_+,b_+} \frac{1}{2}(\|w_+\|^2 + b_+^2) + C_1\sum_{i=1}^{p} I_\varepsilon(f_+(x_i)) + C_2\sum_{j=p+1}^{p+q} H_1(-f_+(x_j)) \tag{6}$$

and

$$\min_{w_-,b_-} \frac{1}{2}(\|w_-\|^2 + b_-^2) + C_3\sum_{i=p+q}^{p+q} I_\varepsilon(f_-(x_i)) + C_4\sum_{j=1}^{p} H_1(f_-(x_j)) \tag{7}$$

where $C_i \geq 0, i = 1, \ldots, 4$ are penalty parameters, and

$$I_\varepsilon(z) = \max(0, |z| - \varepsilon) \tag{8}$$

is the ε-insensitive($\varepsilon > 0$) loss function (Fig. 2b). In fact, $I_\varepsilon(z)$ is the sum of two hinge loss functions $H_{-\varepsilon}(-z)$ and $H_{-\varepsilon}(z)$. NPSVM is an improved version of TWSVM. Compared with TWSVM, NPSVM has several advantages in the following aspects: (1) It does not need to compute inverse matrices, making it tractable for large-scale problems; (2) It absorbs all the data information, but not lose sparseness; (3) NPSVM is a generalized version of TWSVM and can degenerate to TWSVM when the proper parameters are selected; (4) The principle of structural risk minimization is implemented. But similar as traditional SVM, NPSVM still has the drawbacks of being sensitive to outliers.

2.3 RNPSVM

RNPSVM is a robust version of NPSVM which is proposed to weaken the negative impact of noise points. It constructs two problems

$$\min_{w_+,b_+} J_+(w_+, b_+) = \frac{1}{2}(\|w_+\|^2 + b_+^2) + C_1\sum_{i=1}^{p} L_{\epsilon,t}(f_+(x_i)) + C_2\sum_{j=p+1}^{p+q} R_s(-f_+(x_j)) \tag{9}$$

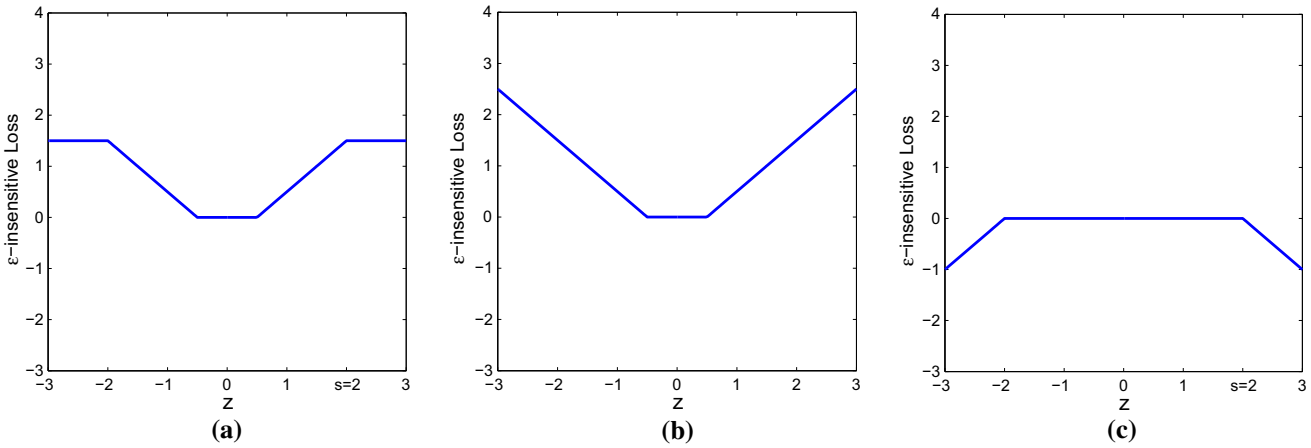

Fig. 2 A ε-insensitive ramp loss function can be decomposed into the sum of a convex ε-insensitive loss function and a concave loss function. **a** ε-insensitive Ramp loss. **b** ε-insensitive loss. **c** Concave loss

and

$$\min_{w_-,b_-} J_-(w_-,b_-) = \frac{1}{2}(\|w_-\|^2 + b_-^2) + C_3 \sum_{i=p+q}^{p+q} L_{\varepsilon,t}(f_-(x_i)) + C_4 \sum_{j=1}^{p} R_s(f_-(x_j)) \tag{10}$$

where $C_i \geq 0, i = 1, \ldots, 4$ are penalty parameters, $L_{\varepsilon,t}(z)$ is ε-insensitive ramp loss function (Fig. 2a),

$$L_{\varepsilon,t}(z) = \begin{cases} t-\varepsilon, & |z| > t \\ |z|-\varepsilon, & \varepsilon \leq |z| \leq t \\ 0, & |z| < \varepsilon \end{cases} \tag{11}$$

and $R_s(z)$ is standard ramp loss function (Fig. 1a)

$$R_s(z) = \begin{cases} 0, & z > 1 \\ 1-z, & s \leq z \leq 1 \\ 1-s, & z < s \end{cases} \tag{12}$$

Due to the nonconvexity of ramp loss function, RNPSVM cannot be directly solved by traditional interior point or trust region reflective method. CCCP is an efficient solver to deal with such problem with the objective function formulated as the sum of a convex function and a concave function.

2.4 ADMM

ADMM is proposed to solve convex problems with much higher speed than traditional methods. It splits convex problems into smaller ones, all of which can be handled in short time in virtue of much smaller scale.

The algorithm deals with such problems as below

$$\min_{u,v} \quad f(u) + g(v) \tag{13}$$

$$s.t. \quad Fu + Gv = c \tag{14}$$

where f, g are convex function with respect to different variables, $u \in R^{n_u}, v \in R^{n_v}$ and F, G are matrices with appropriate dimension. The augmented Lagrangian can be formed as

$$L_\rho(u,v,\lambda) = f(u) + g(v) + \lambda^\top(Fu+Gv-c) + \frac{\rho}{2}\|Fu+Gv-c\|^2 \tag{15}$$

Defining the residual $\boldsymbol{r} = Fu + Gv - c$ and the scaled dual variable $h = \frac{1}{\rho}\lambda$, then

$$L_\rho(u,v,\lambda) = f(u) + g(v) + \lambda^\top r + \frac{\rho}{2}\|r\|^2 \tag{16}$$

$$= f(u) + g(v) + \frac{\rho}{2}\|r + \frac{1}{\rho}\lambda\|^2 - \frac{1}{2\rho}\|\lambda\|^2 \tag{17}$$

$$= f(u) + g(v) + \frac{\rho}{2}\|r+h\|^2 - \frac{\rho}{2}\|h\|^2 \tag{18}$$

The solution of the primal problems (13), (14) can be obtained by the following iterations

$$u^{k+1} = arg\min_u (f(u) + \frac{\rho}{2}\|Fu + Gv^k - c + h^k\|^2) \tag{19}$$

$$v^{k+1} = arg\min_v (g(v) + \frac{\rho}{2}\|Fu^{k+1} + Gv - c + h^k\|^2) \tag{20}$$

$$h^{k+1} = h^k + Fu^{k+1} + Gv^{k+1} - c \tag{21}$$

where $\rho > 0$. The iterations stop at a maximum iteration number or a predefined threshold. Define the primal residual

$\boldsymbol{r}^{k+1} = Fu^{k+1} + Gv^{k+1} - c$ and the dual residual $\boldsymbol{s}^{k+1} = \rho F^\top G(v^{k+1} - v^k)$. Then, a reasonable termination criterion is that $\|\boldsymbol{r}^{\boldsymbol{k}}\| \le \varepsilon_p^k$, $\|\boldsymbol{s}^{\boldsymbol{k}}\| \le \varepsilon_d^k$, where $0 < \varepsilon_p^k, \varepsilon_d^k \ll 1$ are feasibility tolerances for the primal and dual feasibility conditions, respectively. The tolerances can be set using an absolute tolerance ε_1 and relative tolerance ε_2,

$$\varepsilon_p^k = \sqrt{n_h}\varepsilon_1 + \varepsilon_2 \max\{\|Au^k\|, \|Bh^k\|, \|c\|\} \tag{22}$$

$$\varepsilon_d^k = \sqrt{n_u}\varepsilon_1 + \varepsilon_2 \rho \|A^\top h^k\| \tag{23}$$

where n_u, n_h denotes the dimension of u and h, respectively.

3 ADMM for linear NPSVM

Since the primal problems of NPSVM are convex, they can be solved by ADMM directly. In linear case, NPSVM can be easily converted into the standard formulation of ADMM. Then, it can handle large-scale problems because the kernel matrix does not needed to be computed. The primal problems (6) and (7) can be written in explicitly formulation with the sum of regularization term and the hinge loss function.

$$\begin{aligned}\min_{w_+,b_+} \ & \frac{1}{2}(\|w_+\|^2 + b_+^2) + C_1 \sum_{i=1}^{p}([w_+^\top x_i + b_+ - \varepsilon]_+ \\ & + [-w_+^\top x_i - b_+ - \varepsilon]_+) \\ & + C_2 \sum_{i=p+1}^{p+q} [w_+^\top x_j + b_+ + 1]_+ \end{aligned} \tag{24}$$

and

$$\begin{aligned}\min_{w_-,b_-} \ & \frac{1}{2}(\|w_-\|^2 + b_-^2) + C_3 \sum_{i=p+1}^{p+q} ([w_-^\top x_i + b_- - \varepsilon]_+ \\ & + [-w_-^\top x_i - b_- - \varepsilon]_+) \\ & + C_4 \sum_{i=1}^{p} [1 - w_-^\top x_j - b_-]_+ \end{aligned} \tag{25}$$

Introducing additional variables $\alpha_+, \beta_+ \in R^p, \gamma_- \in R^q$ and $\alpha_-, \beta_- \in R^q, \gamma_+ \in R^p$, the above problems can be reformulated as

$$\begin{aligned}\min_{w_+,b_+,\alpha_+,\beta_+,\gamma_-} \ & \frac{1}{2}(\|w_+\|^2 + b_+^2) + C_1 \sum_{i=1}^{p}([\alpha_{+,i}]_+ + [\beta_{+,i}]_+) \\ & + C_2 \sum_{i=p+1}^{p+q} [\gamma_{-,i}]_+ \end{aligned} \tag{26}$$

$$s.t. \quad \alpha_{+,i} = w_+^\top x_i + b_+ - \varepsilon, i = 1, \ldots, p, \tag{27}$$

$$\beta_{+,i} = -w_+^\top x_i - b_+ - \varepsilon, i = 1, \ldots, p, \tag{28}$$

$$\gamma_{-,j} = w_+^\top x_j + b_+ + 1, \quad j = p+1, \ldots, p+q \tag{29}$$

and

$$\begin{aligned}\min_{w_-,b_-,\alpha_-,\beta_-,\gamma_+} \ & \frac{1}{2}(\|w_-\|^2 + b_-^2) \\ & + C_3 \sum_{i=p+1}^{p+q} ([\alpha_{-,i}]_+ + [\beta_{-,i}]_+) \\ & + C_4 \sum_{i=1}^{p} [\gamma_{+,i}]_+ \end{aligned} \tag{30}$$

$$s.t. \quad \alpha_{-,i} = w_-^\top x_i + b_- - \varepsilon,$$

$$i = p+1, \ldots, p+q, \tag{31}$$

$$\beta_{-,i} = -w_-^\top x_i - b_- - \varepsilon,$$

$$i = p+1, \ldots, p+q, \tag{32}$$

$$\gamma_{+,j} = 1 - w_-^\top x_j - b_-, j = 1, \ldots, p \tag{33}$$

For simplicity, the above problems can be rewritten in matrix formation

$$\begin{aligned}\min_{w_+,b_+,\alpha_+,\beta_+,\gamma_-} \ & \frac{1}{2}(\|w_+\|^2 + b_+^2) + C_1 e_+^\top([\alpha_+]_+ + [\beta_+]_+) \\ & + C_2 e_-^\top [\gamma_-]_+ \end{aligned} \tag{34}$$

$$s.t. \quad \alpha_+ = Aw_+ + e_+ b_+ - \varepsilon e_+, \tag{35}$$

$$\beta_+ = -Aw_+ - e_+ b_+ - \varepsilon e_+, \tag{36}$$

$$\gamma_- = Bw_+ + e_- b_+ + e_- \tag{37}$$

and

$$\begin{aligned}\min_{w_-,b_-,\alpha_-,\beta_-,\gamma_+} \ & \frac{1}{2}(\|w_-\|^2 + b_-^2) + C_3 e_-^\top([\alpha_-]_+ + [\beta_-]_+) \\ & + C_4 e_+^\top [\gamma_+]_+ \end{aligned} \tag{38}$$

$$s.t. \quad \alpha_- = Bw_- + e_- b_- - \varepsilon e_-, \tag{39}$$

$$\beta_- = -Bw_- - e_- b_- - \varepsilon e_-, \tag{40}$$

$$\gamma_+ = -Aw_- - b_- + e_+ \tag{41}$$

where $A = (x_1, x_2, \ldots, x_p)^\top$, $B = (x_{p+1}, x_{p+2}, \ldots, x_{p+q})^\top$. For any column vector $z \in R^d$, $[z]_+ = ([z_1]_+, \ldots, [z_d]_+)^\top$. If let $u_+ = (w_+^\top, b_+)^\top$, $z_+ = (\alpha_+^\top, \beta_+^\top, \gamma_-^\top)^\top$, the problems (34)–(37) can be transformed as

$$\min_{u_+,z_+} \ \frac{1}{2}u_+^\top u_+ + C_+^\top [z_+]_+ \tag{42}$$

$$s.t. \quad Tu_+ + z_+ = c_+ \tag{43}$$

where $C_+ = \left(C_1 e_+^\top, C_1 e_+^\top, C_2 e_-^\top\right), c_+ = \left(-\varepsilon e_+^\top, -\varepsilon e_+^\top, e_-^\top\right)$ and

$$T = \begin{pmatrix} -A & -e_+^\top \\ A & e_+^\top \\ -B & -e_-^\top \end{pmatrix} \tag{44}$$

Similarly, the problems (38)–(41) can be transformed as

$$\min_{u_-, z_-} \quad \frac{1}{2} u_-^\top u_- + C_-^\top [z_-]_+ \tag{45}$$

$$s.t. \quad Qu_- + z_- = c_- \tag{46}$$

where $C_- = \left(C_3 e_-^\top, C_3 e_-^\top, C_4 e_+^\top\right), c_+ = \left(-\varepsilon e_-^\top, -\varepsilon e_-^\top, e_-^\top\right)$ and

$$Q = \begin{pmatrix} -B & -e_-^\top \\ B & e_-^\top \\ A & e_+^\top \end{pmatrix} \tag{47}$$

The procedures of solving problems (42), (43) and (43), (45) are shown in Algorithms 1 and 2, respectively.

Algorithm 1 ADMM for the problem (42), (43)

1. Given the training set (1) and the parameters $\varepsilon, C_i, i = 1, \ldots, 4$, initialize u_+^0, z_+^0, h_+^0, set $k = 0$, convergence threshold $\delta (0 < \delta \ll 1)$;
2. Solve the problems

$$u_+^{k+1} = arg \min_{u_+} \left(\frac{1}{2} u_+^\top u_+ + \frac{\rho}{2} \| Tu_+ + z_+^k - c_+ + h_+^k \|^2 \right) \tag{48}$$

$$z_+^{k+1} = arg \min_{z_+} \left(C_+^\top [z_+]_+ + \frac{\rho}{2} \| Tu_+^{k+1} + z_+ - c_+ + h_+^k \|^2 \right) \tag{49}$$

$$h_+^{k+1} = h_+^k + Tu_+^{k+1} + z_+^{k+1} - c_+ \tag{50}$$

and get the solution u_+^{k+1}, z_+^{k+1};

3. Compute the primal and dual residual. If $\| Tu_+^{k+1} + z_+^{k+1} - c_+ \| > \delta$, $\| \rho T^\top (z_+^{k+1} - z_+^k) \| > \delta$, set $k = k + 1$, go to step 2.

Then we can construct linear NPSVM with ADMM in Algorithm 3.

4 ADMM for linear RNPSVM with CCCP

Since the primal problems of RNPSVM are not convex, ADMM cannot deal with them directly. It is distinct that $L_{\varepsilon,t}(z)$ and $R_s(z)$ can be decomposed as follows

$$L_{\varepsilon,t}(z) = I_\varepsilon(z) - I_t(z) \tag{55}$$

$$R_s(z) = H_1(z) - H_s(z) \tag{56}$$

Algorithm 2 ADMM for the problem (45), (46)

1. Given the training set (1) and the parameters $\varepsilon, C_i, i = 1, \ldots, 4$, initialize u_-^0, z_-^0, h_-^0, set $k = 0$, convergence threshold $\delta (0 < \delta \ll 1)$;
2. Solve the problems

$$u_-^{k+1} = arg \min_{u_-} \left(\frac{1}{2} u_-^\top u_- + \frac{\rho}{2} \| Qu_- + z_-^k - c_- + h_-^k \|^2 \right) \tag{51}$$

$$z_-^{k+1} = arg \min_{z_-} \left(C_-^\top [z_-]_+ + \frac{\rho}{2} \| Qu_-^{k+1} + z_- - c_- + h_-^k \|^2 \right) \tag{52}$$

$$h_-^{k+1} = h_-^k + Qu_-^{k+1} + z_-^{k+1} - c_- \tag{53}$$

and get the solution u_+^{k+1}, z_+^{k+1};

3. If $\| Qu_-^{k+1} + z_-^{k+1} - c_- \| > \delta$, $\| \rho Q^\top (z_-^{k+1} - z_-^k) \| > \delta$, set $k = k + 1$, go to step 2.

Algorithm 3 Linear NPSVM

1. Given the training set (1) and the parameters $\varepsilon, C_i, i = 1, \ldots, 4$, initialize $u_+^0, z_+^0, h_+^0, u_-^0, z_-^0, h_-^0$, set $k = 0$, convergence threshold $\delta (0 < \delta \ll 1)$;
2. Use Algorithm 1 and 2 to get the solutions $u_+^*, z_+^*, u_-^*, z_-^*$;
3. The label of a new point $x \in R^n$ is predicted by

$$label = arg \min_{k=+,-} |w_k^\top x + b_k| \tag{54}$$

where $\cdot$ is the perpendicular distance of point x from the planes $w_k^\top x + b_k = 0, k = +, -$.

The geometric explanations of the above two decompositions are provided in Figs. 1 and 2, respectively. Then the problems (9) and (10) can be reformulated as

$$\min_{w_+, b_+} \quad J_+(w_+, b_+) = \check{P} + \hat{P} \tag{57}$$

$$\min_{w_-, b_-} \quad J_-(w_-, b_-) = \check{N} + \hat{N} \tag{58}$$

where

$$\check{P} = \frac{1}{2} (\|w_+\|^2 + b_+^2) + C_1 \sum_{i=1}^{p} I_\varepsilon (f_+(x_i)) + C_2 \sum_{j=p+1}^{p+q} H_1 (-f_+(x_j)) \tag{59}$$

$$\hat{P} = -C_1 \sum_{i=1}^{p} I_t (f_+(x_i)) - C_2 \sum_{j=p+1}^{p+q} H_s (-f_+(x_j)) \tag{60}$$

and

$$\check{N} = \frac{1}{2} (\|w_-\|^2 + b_-^2) + C_3 \sum_{i=p+q}^{p+q} I_\varepsilon (f_-(x_i))$$

$$+C_4\sum_{j=1}^{p}H_1(f_-(x_j)) \tag{61}$$

$$\hat{N}=-C_3\sum_{i=p+q}^{p+q}I_t(f_-(x_i))-C_4\sum_{j=1}^{p}H_s(f_-(x_j)) \tag{62}$$

It can be seen that $\check{P}$, $\check{N}$ are convex functions and $\hat{P}$, $\hat{N}$ are concave functions. Then, the problems (57) and (58) can both be solved by CCCP algorithm. The detailed solving procedures are seen in Algorithms 4 and 5.

Algorithm 4 CCCP for the problem (57)

1. Given the training set (1) and the parameters $\varepsilon, t, s, C_i, i = 1,\ldots,4$, initialize w_+^0, b_+^0, set $k=0$, convergence threshold $\delta(0<\delta\ll 1)$;
2. Solve the problems

$$\min_{w_+,b_+}\check{P}(w_+,b_+)+\hat{P}'(w_+^k,b_+^k)\cdot(w_+,b_+) \tag{63}$$

and get the solution w_+^{k+1}, b_+^{k+1};
3. If $\|w_+^{k+1}-w_+^k\|>\delta, \|b_+^{k+1}-b_+^k\|>\delta$, set $k=k+1$, go to step 2.

Algorithm 5 CCCP for the problem (58)

1. Given the training set (1) and the parameters $\varepsilon, t, s, C_i, i = 1,\ldots,4$, initialize w_-^0, b_-^0, set $k=0$, convergence threshold $\delta(0<\delta\ll 1)$;
2. Solve the problems

$$\min_{w_-,b_-}\check{N}(w_-,b_-)+\hat{N}'(w_-^k,b_-^k)\cdot(w_-,b_-) \tag{64}$$

and get the solution w_-^{k+1}, b_-^{k+1};
3. If $\|w_-^{k+1}-w_-^k\|>\delta, \|b_-^{k+1}-b_-^k\|>\delta$, set $k=k+1$, go to step 2.

The objective functions of (63) and (64) are both convex, and then they can be solved by ADMM. Rewrite the problems in explicit formulation

$$\begin{aligned}\min_{w_+,b_+,\alpha_+,\beta_+,\gamma_-}\ &\frac{1}{2}(\|w_+\|^2+b_+^2)\\&+C_1\sum_{i=1}^{p}([\alpha_{+,i}]_+ +[\beta_{+,i}]_+)\\&+C_2\sum_{j=p+1}^{p+q}[\gamma_{-,j}]_+\\&+\sum_{i=1}^{p}\theta_i(w_+^\top x_i+b_+)\\&+\sum_{j=p+1}^{p+q}\delta_j y_j(w_+^\top x_j+b_+)\end{aligned} \tag{65}$$

$$s.t.\alpha_{+,i}=w_+^\top x_i+b_+-\varepsilon, i=1,\ldots,p, \tag{66}$$

$$\beta_{+,i}=-w_+^\top x_i-b_+-\varepsilon, \quad i=1,\ldots,p, \tag{67}$$

$$\gamma_{-,j}=w_+^\top x_j+b_++1, \quad j=p+1,\ldots,p+q \tag{68}$$

and

$$\begin{aligned}\min_{w_-,b_-,\alpha_-,\beta_-,\gamma_+}\ &\frac{1}{2}(\|w_-\|^2+b_-^2)\\&+C_3\sum_{j=p+1}^{p+q}([\alpha_{-,j}]_+ +[\beta_{-,j}]_+)\\&+C_4\sum_{i=1}^{p}[\gamma_{+,i}]_+\\&+\sum_{j=p+1}^{p+q}\theta_j(w_-^\top x_j+b_-)\\&+\sum_{i=1}^{p}\delta_i y_i(w_-^\top x_i+b_-)\end{aligned} \tag{69}$$

$$s.t.\alpha_{-,j}=w_-^\top x_j+b_--\varepsilon, \quad j=p+1,\ldots,p+q, \tag{70}$$

$$\beta_{-,j}=-w_-^\top x_j-b_--\varepsilon, \quad j=p+1,\ldots,p+q, \tag{71}$$

$$\gamma_{+,i}=1-w_-^\top x_i-b_-, i=1,\ldots,p \tag{72}$$

where

$$\begin{aligned}\theta_i&=-C_1\frac{\partial I_t(f_+(x_i))}{\partial f_+(x_i)}\\&=\begin{cases}-C_1, & \text{if } f_+(x_i)>t\\ C_1, & \text{if } f_+(x_i)<-t\\ 0, & \text{otherwise}\end{cases}, i=1,\ldots,p\end{aligned} \tag{73}$$

$$\begin{aligned}\theta_j&=-C_3\frac{\partial I_t(f_-(x_j))}{\partial f_-(x_j)}\\&=\begin{cases}-C_3, & \text{if } f_-(x_i)>t\\ C_3, & \text{if } f_-(x_i)<-t\\ 0, & \text{otherwise}\end{cases}, \quad i=p+1,\ldots,p+q\end{aligned} \tag{74}$$

and

$$\begin{aligned}\delta_i&=-C_4y_i\frac{\partial H_s(y_i f_-(x_i))}{\partial f_-(x_i)}\\&=\begin{cases}C_4, & \text{if } y_i f_-(x_i)<s\\ 0, & \text{otherwise}\end{cases}, \quad i=1,\ldots,p,\end{aligned} \tag{75}$$

$$\delta_j = -C_2 y_j \frac{\partial H_s(y_j f_+(x_j))}{\partial f_+(x_j)} = \begin{cases} C_2, & \text{if } y_j f_+(x_j) < s \\ 0, & \text{otherwise} \end{cases}, \quad j = p+1, \ldots, p+q, \tag{76}$$

Let $\theta_+ = (\theta_1, \ldots, \theta_p), \theta_- = (\theta_{p+1}, \ldots, \theta_{p+q}), \delta_+ = (\delta_1, \ldots, \delta_p), \delta_- = (\delta_{p+1}, \ldots, \delta_{p+q})$, the above problems in matrix expression are

$$\min_{u_+, z_+} \quad \frac{1}{2} u_+^\top u_+ + \kappa_+^\top u_+ + C_+^\top [z_+]_+ \tag{77}$$

$$s.t. \quad Tu_+ + z_+ = c_+ \tag{78}$$

and

$$\min_{u_-, z_-} \quad \frac{1}{2} u_-^\top u_- + \kappa_-^\top u_- + C_-^\top [z_-]_+ \tag{79}$$

$$s.t. \quad Qu_- + z_- = c_- \tag{80}$$

where

$$\kappa_+ = \begin{pmatrix} A^\top \theta_+ + B^\top (\delta_- \circ Y_B) \\ e_+^\top \theta_+ + e_-^\top (\delta_- \circ Y_B) \end{pmatrix} \tag{81}$$

$$\kappa_- = \begin{pmatrix} B^\top \theta_- + A^\top (\delta_+ \circ Y_A) \\ e_-^\top \theta_- + e_+^\top (\delta_+ \circ Y_A) \end{pmatrix} \tag{82}$$

After the above transformations, we can use ADMM to solve RNPSVM in every iteration of CCCP. The algorithms of ADMM for (77), (78) and (79), (80), named as Algorithms R5 and R6, are the same as Algorithms 1 and 2, under the condition that the iterations (48) and (51) should be replaced by

$$u_+^{k+1} = arg \min_{u_+} \left(\frac{1}{2} u_+^\top u_+ + \kappa_+^\top u_+ + \frac{\rho}{2} \| Tu_+ + z_+^k - c_+ + h_+^k \|^2 \right) \tag{83}$$

and

$$u_-^{k+1} = arg \min_{u_-} \left(\frac{1}{2} u_-^\top u_- + \kappa_-^\top u_- + \frac{\rho}{2} \| Qu_- + z_-^k - c_- + h_-^k \|^2 \right) \tag{84}$$

respectively. The comprehensive solving procedures for linear RNPSVM are displayed in Algorithm 6.

Algorithm 6 Linear RNPSVM

1. Given the training set (1) and the parameters $\varepsilon, t, s, C_i, i = 1, \ldots, 4$, initialize $w_+^0, b_+^0, z_+^0, h_+^0, \theta_+^0, \delta_+^0, w_-^0, b_-^0, z_-^0, h_-^0, \theta_-^0, \delta_-^0$, set $k = 0$, convergence threshold $\delta (0 < \delta \ll 1)$;
2. Solve Algorithms R5 and R6, and get the solution $w_+^{k+1}, b_+^{k+1}, w_-^{k+1}, b_-^{k+1}$;
3. Compute $\theta_+^{k+1}, \delta_+^{k+1}, \theta_-^{k+1}, \delta_-^{k+1}$;
4. If $\|(\theta_+^{k+1}, \delta_+^{k+1}, \theta_-^{k+1}, \delta_-^{k+1}) - (\theta_+^k, \delta_+^k, \theta_-^k, \delta_-^k)\| > \delta$, set $k = k+1$, go to step 2, else get the solution $(w_+^*, b_+^*) = (w_+^{k+1}, b_+^{k+1}), (w_-^*, b_-^*) = (w_-^{k+1}, b_-^{k+1})$;
5. A new point $x \in R^n$ is predicted by the Eq. (54).

5 Numerical experiments

In this section, experiments on benchmark datasets and large-scale datasets are made to validate the classification ability of ADMM on NPSVM and RNPSVM. We compare them with SVM, solved by LIBLINEAR and ADMM, respectively. All the methods are implemented in MATLAB 2015a (Lenovo PC, Intel Core I5 processor, 8GB RAM). (The codes can be downloaded from the Web site: https://github.com/henryvivid/ADMMforNPSVMs/tree/henryvivid-patch-SVM.)

Table 1 Error rate on small-scale datasets

Dataset	SVM		NPSVM		RNPSVM	
	Error (%)	SVs (%)	Error (%)	SVs (%)	Error (%)	SVs (%)
WPBC (194 × 34)	23.70 ± 7.16	87.61	19.63 ± 3.64	69.84	**18.68** ± 3.34	**64.79**
Sonar (208 × 60)	**17.88** ± 3.64	**79.68**	22.17 ± 4.29	83.53	19.67 ± 6.30	82.69
Spectf (267 × 44)	18.27 ± 7.66	**52.62**	19.05 ± 4.13	84.65	**17.58** ± 3.78	72.10
Heart (270 × 13)	16.30 ± 4.79	82.31	15.56 ± 6.49	**76.85**	**14.44** ± 5.30	79.07
Bupa_liver (345 × 6)	31.30 ± 3.34	96.59	31.59 ± 6.01	**58.48**	**29.86** ± 7.71	63.04
Ionosphere (351 × 34)	10.26 ± 1.24	**70.52**	9.98 ± 3.37	78.13	**9.98** ± 2.70	77.42
Dermatology (366 × 34)	2.73 ± 1.36	**45.63**	2.18 ± 2.84	94.06	**2.18** ± 2.84	93.37
Votes (435 × 16)	4.37 ± 1.26	**42.59**	4.37 ± 1.26	90.34	**3.22** ± 1.50	90.34
Australian (690 × 14)	**13.33** ± 2.79	76.04	13.33 ± 4.18	**67.61**	13.33 ± 3.61	83.91
German (1000 × 20)	23.10 ± 1.43	70.00	24.10 ± 2.95	73.50	**22.80** ± 1.30	**53.13**

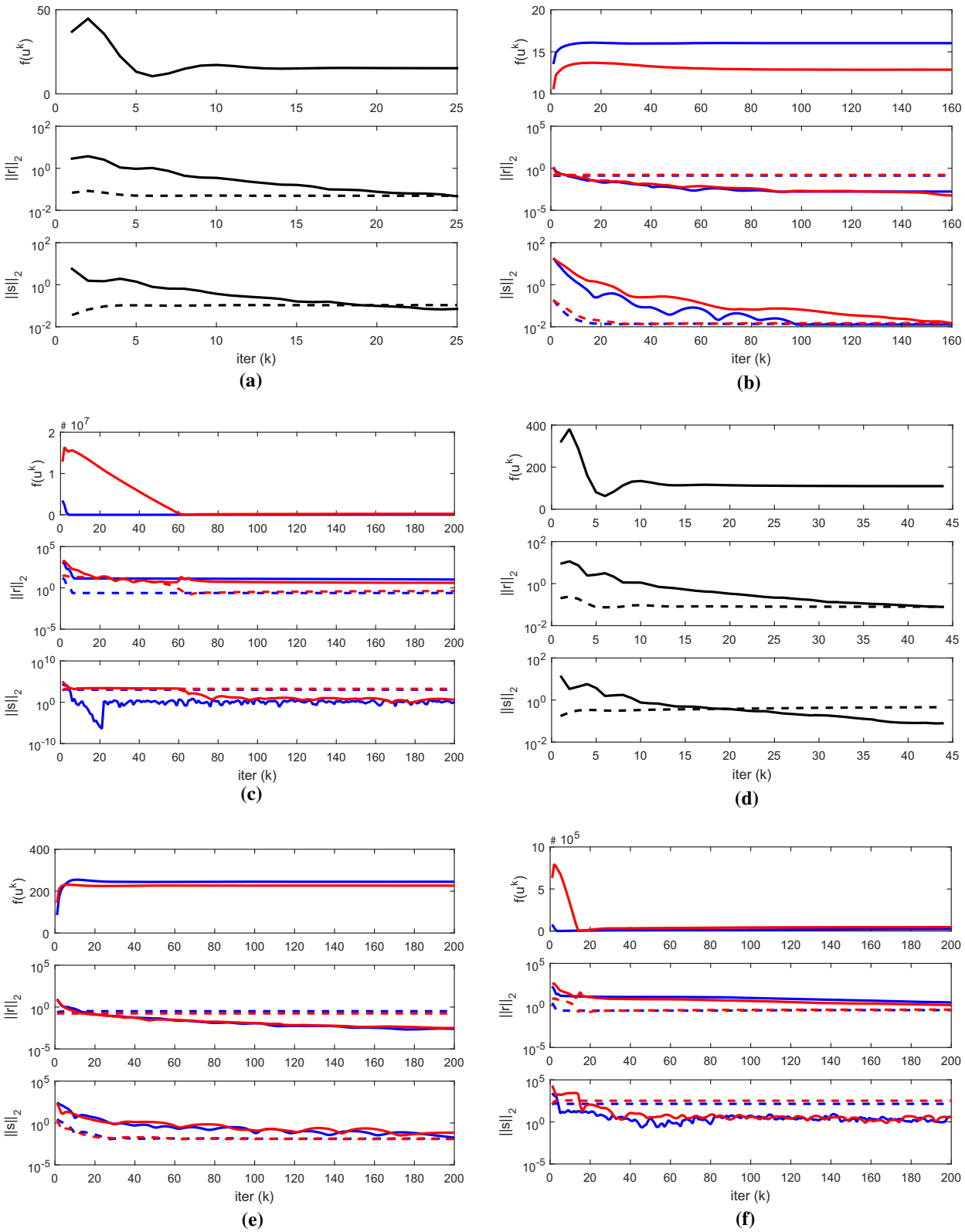

Fig. 3 Convergence of ADMM on Votes and German. The *solid lines* denote the value of the objective function, the norm of primal residual or dual residual. The *dashed lines* denote the corresponding tolerance. For NPSVM and RNPSVM, the *blue* and *red lines* denote the value in positive and negative programming, respectively. **a** Votes-SVM. **b** Votes-NPSVM. **c** Votes-RNPSVM. **d** German-SVM. **e** German-NPSVM. **f** German-RNPSVM

Table 2 Characteristics of large-scale datasets

Dataset	Instance	Attribute	Class	Class distribution
USPS	9298	256	10	{1553, 1269, 929, 824, 852, 716, 834, 792, 708, 821}
a9a	48842	123	2	{11687, 37155}
shuttle	58000	9	7	{45586, 50, 171, 8903, 3267, 10, 13}
codrna	58535	8	2	{19845, 39690}
w8a	64700	300	2	{1479, 48270}
ijcnn1	141691	2	2	{13565, 128126}
skin	245057	4	2	{50859, 194198}
webspam	350000	254	2	{212189, 137811}
covtype	581012	54	7	{211840, 283301, 35754, 2747, 9493, 17367, 20510}

Table 3 Error rates on large-scale datasets

Dataset	SVM-L		SVM-A		NPSVM		RNPSVM	
	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)
USPS	6.23 ± 0.43	6.05	5.37 ± 0.49	9.70	$\mathbf{4.78 \pm 0.34}$	12.64	4.83 ± 0.33	44.87
a9a	15.13 ± 0.11	5.46	16.82 ± 0.11	2.17	15.13 ± 0.15	5.60	$\mathbf{15.09 \pm 0.05}$	47.65
shuttle	8.08 ± 0.07	4.25	2.73 ± 0.27	4.95	$\mathbf{2.28 \pm 0.15}$	24.90	2.62 ± 0.23	111.54
codrna	9.88 ± 2.22	3.90	$\mathbf{6.11 \pm 0.13}$	0.51	6.13 ± 0.06	6.32	6.16 ± 0.04	11.00
w8a	1.73 ± 0.05	1.15	1.36 ± 0.08	3.30	1.32 ± 0.09	7.40	$\mathbf{1.31 \pm 0.11}$	71.52
ijcnn1	8.09 ± 0.12	8.86	$\mathbf{6.28 \pm 0.15}$	4.75	8.06 ± 0.09	16.08	7.06 ± 0.11	144.07
skin	7.86 ± 0.43	31.48	7.42 ± 0.03	3.07	6.50 ± 0.08	18.55	$\mathbf{5.67 \pm 0.10}$	120.23
webspam	7.31 ± 0.09	34.51	6.95 ± 0.15	72.98	7.05 ± 0.13	89.14	$\mathbf{6.95 \pm 0.11}$	1949.20
covtype	28.73 ± 0.16	355.98	29.46 ± 0.37	102.62	27.41 ± 0.10	498.42	$\mathbf{27.18 \pm 0.15}$	3604.50

5.1 Benchmark datasets

To verify the performance of ADMM, we first make experiments on small-scale benchmark datasets. ADMM is used to solve SVM, NPSVM, RNPSVM to test its convergence and precision of the solutions. The benchmark datasets are all from UCI repository. To obtain the best values of all the parameters, fivefold cross-validation is implemented. The penalty parameters, C for SVM, $c_i (i = 1, 2, 3, 4)$ for NPSVM and RNPSVM are all searched from the set $\{2^{-8}, \ldots, 2^8\}$. But $C_1 = C_2 = C_3 = C_4$ is set for simplicity. The parameter ε is set to 0.2. The optimal values of t and s are selected through searching the set $\{0.4, 0.6, 0.8, 1.0\}$ and $\{-1.0, -0.8, -0.6, -0.4, -0.2, 0\}$, respectively. The experimental results of classification error and ratio of support vectors (SVs%) are listed in Table 1. The best results are displayed in boldface. RNPSVM obtains 8 minimal classification errors on 10 datasets, which proves its superiority on classification. NPSVM performs better than SVM on 5 datasets. NPSVM and RNPSVM get the lowest ratio of support vectors in a percent of 3/10 and 2/10, respectively. It is obvious that ADMM can deal with QPPs well and get a good approximate solution. To understand the convergence process of ADMM clearly, the values of objective function f, primal residual $\boldsymbol{r}$, dual residual $\boldsymbol{s}$ according to the iteration k are displayed in Fig. 3. The $\varepsilon_1, \varepsilon_2$ in the Eqs. (22), (23) is set to be $10^{-4}, 10^{-2}$, respectively. Due to the space limitation, only the results of **Votes** and **German** are exhibited. It is noted that the values of f, $\boldsymbol{r}$, $\boldsymbol{s}$ become smaller than the tolerances or changeless after certain iterations, which proves that the solutions of ADMM on SVM, NPSVM, RNPSVM can all converge to stable points. The experimental results demonstrate that: (1) Ramp loss function is very useful in improving NPSVM since it is superior to hinge loss function in some aspects, including its insensitivity to outliers; (2) nonparallel SVMs can extract more data information to make more accurate prediction; (3) ADMM is effective in solving quadratic programming problems by finding good approximate solutions.

5.2 Large-scale datasets

In this subsection, nine large-scale datasets with different sizes and dimensions are selected to test the classification capability of NPSVM and RNPSVM solved by ADMM on large-scale problems. The characteristics, including the number of instances and attributes, class distribution, of the datasets are listed in Table 2. The experiments of SVM solved

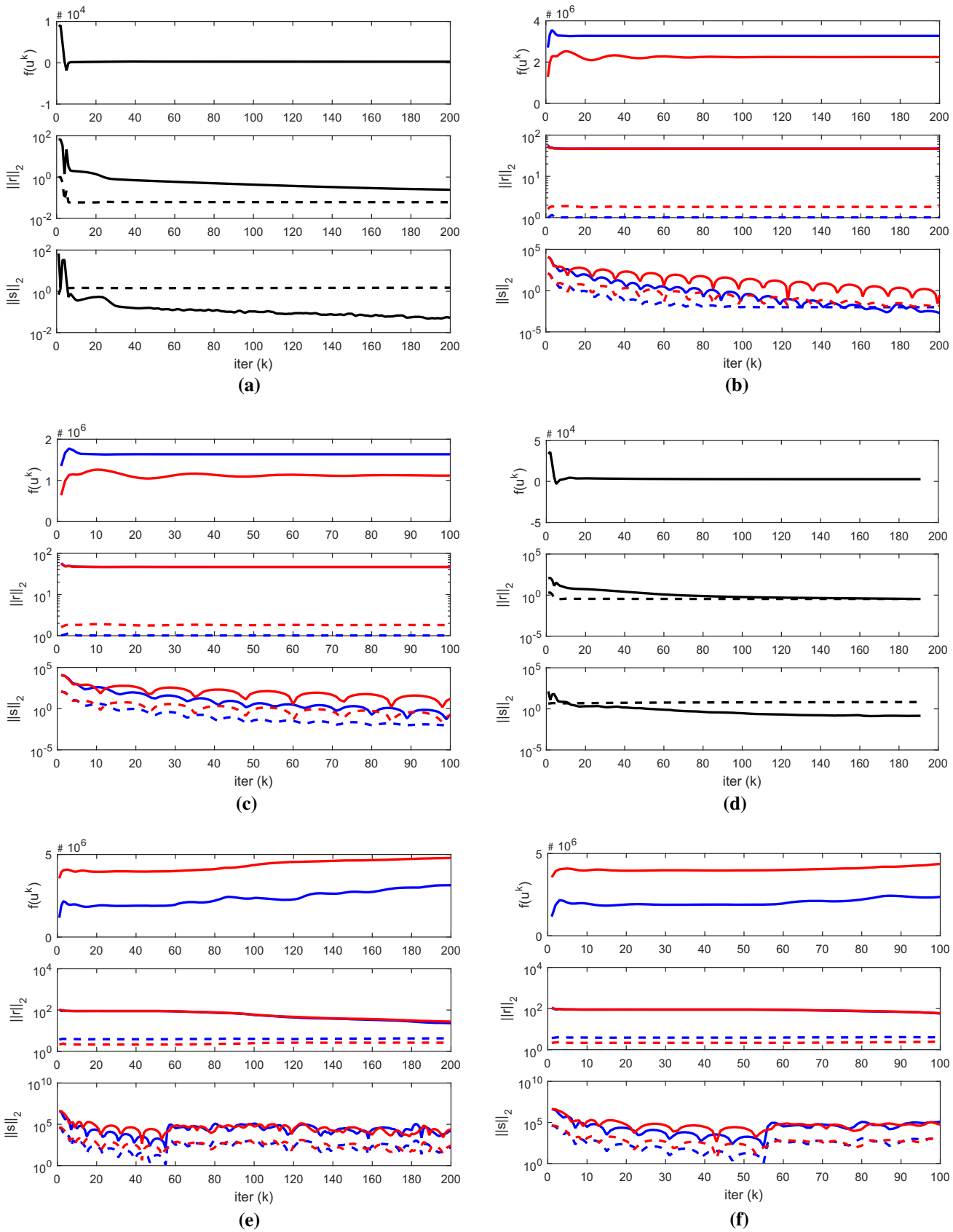

Fig. 4 Convergence of ADMM on a9a and skin. **a** a9a-SVM. **b** a9a-NPSVM. **c** a9a-RNPSVM. **d** skin-SVM. **e** skin-NPSVM. **f** skin-RNPSVM

by LIBLINEAR(SVM-L) and ADMM(SVM-A) are used to make comparison with NPSVM and RNPSVM. The settings of searching for the penalty parameters and ramp loss parameters are the same as in the benchmark datasets. The best values of the parameters are achieved by threefold cross-validation. The experimental results, classification error and training time included, are shown in Table 3. For SVM, making comparison of LIBLINEAR and ADMM, SVM-A gets seven lower classification errors than SVM-L. And SVM-A is faster than SVM-L on five datasets. It can be seen that ADMM is more effective than LIBLINEAR in solving SVM. In the nine datasets, NPSVM and RNPSVM obtain two and five lowest errors, respectively. Obviously, RNPSVM shows strong ability in making classification on large-scale problems. Similarly, the varying values of objective function, primal residual and dual residual for ADMM are depicted in Fig. 4. After a certain iteration, the objective functions do not change again, and the residuals meet the termination criterion or fluctuate lightly. The case demonstrates that ADMM converges to stable level finally. The experimental results prove that NPSVM and RNPSVM perform better than SVM on large-scale problems and ADMM can deal with large-scale convex programming effectively.

6 Conclusions

In this paper, a new kind of large-scale version of NPSVMs has been proposed, which aims to manage large-scale problems in an efficient way. LIBLINEAR is a well-performed solver for the standard SVM in linear classification on both benchmark and large-scale datasets. However, NPSVMs, including the original NPSVM and ramp loss NPSVM, have shown stronger ability in classification than SVM. ADMM is applied in solving NPSVMs with the purpose of classifying large-scale problems. ADMM can split a large convex programming problem into several smaller-scale convex problems and solve these problems iteratively until the stop criterion is satisfied. The results of numerical experiments demonstrate that ADMM can always converge into stable solutions and NPSVMs still perform better than SVM on large-scale classification problems.

Acknowledgements This work has been partially supported by grants from National Natural Science Foundation of China (Nos. 61472390, 11271361, 71331005, 11226089 and 91546201), Major International (Regional) Joint Research Project (No. 71110107026) and the Beijing Natural Science Foundation (No. 1162005).

Compliance with ethical standards

Conflict of interest We declare that we have no conflicts of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors

References

Akbani R, Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. In: Machine learning: ECML 2004, Springer, Berlin, p 39–50

Arun Kumar M, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543

Bhaskar BN, Tang G, Recht B (2013) Atomic norm denoising with applications to line spectral estimation. IEEE Trans Signal Process 61(23):5987–5999

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and trends® in. Mach Learn 3(1):1–122

Bradley PS, Mangasarian OL (1998) Feature selection via concave minimization and support vector machines. ICML 98:82–90

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Deng J, Berg AC, Li K, Fei-Fei L (2010) What does classifying more than 10,000 image categories tell us? In: Computer vision–ECCV 2010, Springer, Berlin, pp 71–84

Deng N, Tian Y (2004) New method in data mining: support vector machines. Science Press, Beijing

Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. CRC Press, Boca Raton

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach learn 46(1–3):389–422

Jayadeva RK, C S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910

Kasiviswanathan SP, Melville P, Banerjee A, Sindhwani V (2011) Emerging topic detection using dictionary learning. In: Proceedings of the 20th ACM international conference on Information and knowledge management, ACM, pp 745–754

Kumar MA, Gopal M (2008) Application of smoothing technique on twin support vector machines. Pattern Recognit Lett 29(13):1842–1848

Li W, Zhao R, Wang X (2012) Human reidentification with transferred metric learning. In: ACCV (1), pp 31–44

Liu D, Shi Y, Tian Y (2015) Ramp loss nonparallel support vector machine for pattern classification. Knowl-Based Syst 85:224–233

Ma J, Saul LK, Savage S, Voelker GM (2009) Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th annual international conference on machine learning, ACM, pp 681–688

Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, pp 142–150

Naik GR, Kumar DK et al (2010) Twin SVM for gesture classification using the surface electromyogram. IEEE Trans Inf Technol Biomed 14(2):301–308

Qi Z, Tian Y, Shi Y (2012) Twin support vector machine with universum data. Neural Netw 36:112–119

Qi Z, Tian Y, Shi Y (2013) Robust twin support vector machine for pattern classification. Pattern Recognit 46(1):305–316

Qi Z, Tian Y, Shi Y (2014) A nonparallel support vector machine for a classification problem with universum learning. J Comput Appl Math 263:288–298

Shao Y, Zhang C, Wang X, Deng N (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
Tan J, Zhang C, Deng N (2010) Cancer related gene identification via p-norm support vector machine. In: The 4th international conference on computational systems biology, vol 1, pp 101–108
Tian Y, Ping Y (2014) Large-scale linear nonparallel support vector machine solver. Neural Netw 50:166–174
Tian Y, Ju X, Qi Z, Shi Y (2014a) Improved twin support vector machine. Sci China Math 57(2):417–432
Tian Y, Qi Z, Ju X, Shi Y, Liu X (2014b) Nonparallel support vector machines for pattern classification. IEEE Trans Cybern 44(7):1067–1079
Tian Y, Zhang Q, Liu D (2014c) ν-nonparallel support vector machine for pattern classification. Neural Comput Appl 25(5):1007–1020
Tian Y, Ju X, Shi Y (2016) A divide-and-combine method for large scale nonparallel support vector machines. Neural Netw 75:12–21
Vapnik V (1998) Statistical learning theory. Wiley, New York
Vapnik V (2000) The nature of statistical learning theory. Springer, Berlin
Weston J, Watkins C et al (1999) Support vector machines for multi-class pattern recognition. ESANN 99:219–224
Zhang HH, Ahn J, Lin X, Park C (2006) Gene selection using support vector machines with non-convex penalty. Bioinformatics 22(1):88–95