1-NRTSVM via ADMM for Automatical Feature Selection and Classification Simultaneously

HAITAO XU
Liaocheng University
School of Mathematics Sciences
Liaocheng, 252059
P.R. China
xuhaitao@live.com

LIYA FAN
Liaocheng University
School of Mathematics Sciences
Liaocheng, 252059
P.R. China
Corresponding author:fanliya63@126.com

Abstract: In this paper, a novel classifier with linear and nonlinear versions for data classification and automatical feature selection simultaneously is proposed and named as 1-norm regularized twin support vector machine (1-NRTSVM). By means of the alternating direction method of multipliers (ADMM), two implementation algorithms for 1-NRTSVM are presented. A major feature of the proposed method is directly solving primal problems not dual problems. Experiment results show that the proposed 1-NRTSVM is an effective and competitive classifier for data classification and automatical feature selection.

Key–Words: Twin support vector machine, 1-norm of a vector, alternating direction method of multipliers, automatic feature selection

1 Introduction

Support vector machines (SVMs) [1-2], as powerful tools for pattern classification and regression, have already been successfully applied in a wide variety of fields. For a traditional SVM, the classification hyperplane can be obtained by maximizing the margin between two parallel boundary hyperplanes, which involves the minimization of a quadratic programming problem (QPP). Many classification and feature selection methods based on SVM were proposed in recent years, such as 1-norm SVM [3], doubly regularized SVM (DRSVM) [4] and so on. In 2011, Ye et al. [5] proposed a feature selection method based on ADMM to solve DRSVM.

Different from SVM, twin SVM (TSVM) proposed by Jayadeva et al. [6] seeks two nonparallel hyperplanes such that each hyperplane is closer to one of two classes and is at least one distance from the other. It is implemented by solving two smaller quadratic programming problems (QPPs) rather than a single large QPP in SVM, which makes the learning speed of TSVM is more faster than that of SVM. Some extensions for TSVM include least squares TSVM (LSTSVM) [7], parametric-margin TSVM [8], twin bounded SVM [9] and so on, for details see [6-14]. In addition , there also exist some feature selection method based on TSVM, such as 1-norm least squares TSVM [15], in which all the 2-norm terms in LSTSVM are replaced by the 1-norm terms for the purpose of suppressing input features.

Motivated by works above, in this paper, we propose a 1-norm regularized TSVM (1-NRTSVM) with linear and nonlinear versions for data classification and automatic feature selection simultaneously, and present two implementation algorithms for the proposed method by means of the alternating direction method of multipliers (ADMM). It is known that the validation of the quality of a new method seriously depends on a large number of data experiments. So, in this paper, we not only put forward a new classification method but also more focus on calculation.

The rest of the paper is organized as follows. In Section 2, some notations and related works are introduced. In Section 3, 1-NRTSVM with linear and nonlinear versions is proposed and two solving algorithms are provided by means of ADMM. Experiments and results analysis are performed in Section 4 and some conclusions are given in Section 5.

2 Notations and related works

In this section, we briefly recall some basic concepts and results used in the sequel. Let $T = \{(x_i, y_i)\}_{i=1}^{l}$ be a set of vector data, where $x_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$ are the input sample and class label of the ith data, respectively. Let l_1 and l_2 be the numbers of positive and negative samples, respectively, and $l = l_1 + l_2$. We denote by $A = [x_1^+, \cdots, x_{l_1}^+] \in \mathbb{R}^{n\times l_1}$ and $B = [x_1^-, \cdots, x_{l_2}^-] \in \mathbb{R}^{n\times l_2}$ the matrices of samples belonging to the positive and negative classes, respectively. 1-norm and 2-norm of a vector is denoted by $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively.

2.1 Linear TSVM

Linear TSVM seeks a pair of nonparallel hyperplanes $< w_+, x > +b_+ = 0$ and $< w_-, x > +b_- = 0$, where $w_+, w_- \in R^n$ are normal vectors and $b_+, b_- \in R$ are thresholds, by considering the following two quadratic programming problems (QPPs):

$$\min_{w_+,b_+,\xi_2} \frac{c_3}{2}\|w_+\|_2^2 + \frac{1}{2}\left\|A^T w_+ + e_1 b_+\right\|_2^2 + c_1 e_2^T \xi_2$$
$$s.t. \quad -(B^T w_+ + e_2 b_+) \geq e_2 - \xi_2, \quad \xi_2 \geq 0, \quad (1)$$
$$\min_{w_-,b_-,\xi_1} \frac{c_4}{2}\|w_2\|_2^2 + \frac{1}{2}\left\|B^T w_- + e_2 b_-\right\|_2^2 + c_2 e_1^T \xi_1$$
$$s.t. \quad (A^T w_- + e_1 b_-) \geq e_1 - \xi_1, \quad \xi_1 \geq 0, \quad (2)$$

where $c_1, c_2 > 0$ are trade-off parameters, $\xi_1 \in R^{l_1}, \xi_2 \in R^{l_2}$ are slack variables vectors and $e_1 \in R^{l_1}, e_2 \in R^{l_2}$ are vectors of ones. By solving respectively the Wolfe dual forms of the problems (1) and (2), (w_+, b_+) and (w_-, b_-) can be obtained and then a new input $\widetilde{x} \in R^n$ can be assigned the class k depending on which of the two hyperplanes is closer to, that is, $k = \arg\min_{+,-} \frac{|<w_k, \widetilde{x}> + b_k|}{\|w_k\|}$.

2.2 ADMM

ADMM developed in the 1970s [16] has recently become a method of choice for solving many large-scale problems [17-18], which is implemented by solving the following optimization problem:

$$\min_{y,z} f(y) + g(z)$$
$$s.t.\ Fy + Dz = c, \quad (3)$$

where $c \in R^p$ is a constant vector, $F \in R^{p\times n}, D \in R^{p\times m}$ are coefficient matrices and $f : R^n \to R, g : R^m \to R$ are functions. ADMM solves vectors (y, z, α) by using the following iterative procedure: starting from some initial values (y^k, z^k, α^k) with $k = 0$, then they can be updated iteratively by

$$\begin{cases} y^{k+1} = \arg\min_y \widetilde{L}(y, z^k, \alpha^k), \\ z^{k+1} = \arg\min_z \widetilde{L}(y^{k+1}, z, \alpha^k), \\ \alpha^{k+1} = \alpha^k + \mu(Fy^{k+1} + Dz^{k+1} - c), \end{cases}$$

where

$$\widetilde{L}(y, z, \alpha) = f(y) + g(z) + \alpha^T(Fy + Dz - c) + \frac{\mu}{2} \| Fy + Dz - c \|_2^2$$

is the argumented Lagrangian function of the problem (3), $\alpha \in R^p$ is a lagrange multipliers vector and $\mu > 0$ is an adjustable parameter.

3 1-NRTSVM

In this section, linear 1-NRTSVM will be firstly introduced for automatical feature selection and data classification simultaneously and an efficient implementation algorithm will be proposed based on ADMM. Then by means of the kernel skills, the nonlinear case of 1-NRTSVM will be researched. Different from TSVM, 1-NRTSVM can be implemented by solving directly the primal problems. Here, 1 and 2 denote the positive and negative classes, respectively. All notations used in the section are same as in Section 2 unless specially statements.

3.1 Linear 1-NRTSVM

In order to obtain the primal modelings of linear 1-NRTSVM, by means of the vector plus function $(\cdot)_+$ and the idea in [3], we modify the problems (1) and (2) into the following forms:

$$\min_{w_1,b_1} \ c_3\|v_1\|_1 + \frac{1}{2}\left\|A^T w_1 + e_1 b_1\right\|_2^2 + c_1 e_2^T (u_1)_+$$
$$s.t. \quad u_1 = e_2 + (B^T w_1 + e_2 b_1),\ v_1 = w_1, \quad (4)$$
$$\min_{w_2,b_2} \ c_4\|v_2\|_1 + \frac{1}{2}\left\|B^T w_2 + e_2 b_2\right\|_2^2 + c_2 e_1^T (u_2)_+$$
$$s.t. \quad u_2 = e_1 - (A^T w_2 + e_1 b_2),\ v_2 = w_2, \quad (5)$$

where $u_1 \in R^{l_2}, u_2 \in R^{l_1}, v_1, v_2 \in R^n$ are auxiliary variables vectors. Next, we mainly solve the problem (4) by means of ADMM. With the similar way, the problem (5) can be also solved. In order to solve effectively the problem (4), we can decompose it into three optimization problems with respect to (w_1, b_1), u_1 and v_1, respectively, and then obtain

the following iterative procedure by using ADMM:

$$\begin{cases} (w_1^{k+1}, b_1^{k+1}) = \arg\min\limits_{w_1,b_1} \widetilde{L}(w_1, b_1, u_1^k, v_1^k, \alpha_1^k, \beta_1^k), \\ u_1^{k+1} = \arg\min\limits_{u_1} \widetilde{L}(w_1^{k+1}, b_1^{k+1}, u_1, v_1^k, \alpha_1^k, \beta_1^k), \\ v_1^{k+1} = \arg\min\limits_{v_1} \widetilde{L}(w_1^{k+1}, b_1^{k+1}, u_1^{k+1}, v_1, \alpha_1^k, \beta_1^k), \\ \alpha_1^{k+1} = \alpha_1^k + \mu_1(u_1^{k+1} - (e_2 + B^T w_1^{k+1} + e_2 b_1^{k+1})), \\ \beta_1^{k+1} = \beta_1^k + \mu_2(v_1^{k+1} - w_1^{k+1}), \end{cases} \tag{6}$$

where $\widetilde{L}(w_1, b_1, u_1, v_1, \alpha_1, \beta_1)$ is the augmented Lagrangian function of the problem (4), $\alpha_1 \in R^{l_1}, \beta_1 \in R^n$ are multiplier vectors and $\mu_1, \mu_2 > 0$ are parameters. The first problem in (6) can be solved by letting $\frac{\partial \widetilde{L}(w_1,b_1)}{\partial w_1} = 0$ and $\frac{\partial \widetilde{L}(w_1,b_1)}{\partial b_1} = 0$. For solving the second problem in (6), we need the following result introduced in [19].

Proposition 1. Let $S_\lambda(\omega) = \arg\min\limits_{x \in R} \lambda x_+ + \frac{1}{2}(x - \omega)^2$. Then

$$S_\lambda(\omega) = \begin{cases} \omega - \lambda, & \omega > \lambda, \\ 0, & 0 \le \omega \le \lambda, \\ \omega, & \omega < 0. \end{cases}$$

It is easily proven that the second problem is equivalent to the following optimization problem:

$$u_1 = \arg\min_{u_1} c_1 e_2^T (u_1)_+ \frac{\mu_1}{2} \parallel u_1 - (e_2 - \frac{\alpha_1^k}{\mu_1} + (B^T w_1^{k+1} + e_2 b_1^{k+1}) \parallel_2^2,$$

and then by Proposition 1, we can get the iterative formula:

$$u_1^{k+1} = S_{\frac{c_1}{\mu_1}}(e_2 - \frac{\alpha_1^k}{\mu_1} + (B^T w_1^{k+1} + e_2 b_1^{k+1}), \tag{7}$$

where $S_\lambda(\omega) = (S_\lambda(\omega_1), S_\lambda(\omega_2), \cdots, S_\lambda(\omega_{l_1}))^T$ and $\omega = (\omega_1, \cdots, \omega_{l_1})^T \in \mathbb{R}^{l_1}$. Similar to the second problem. the third problem is equivalent to the optimization problem:

$$v_1 = \arg\min_{v_1} c_3 \parallel v_1 \parallel_1 + \frac{\mu_2}{2} \parallel v_1 - (w_1^{k+1} - \frac{\beta_1^k}{\mu_2}) \parallel_2^2,$$

and then by means of the result presented in [5], it can be solved by the iterative formula:

$$v_1^{k+1} = \tau_{\frac{c_3}{\mu_2}}(w_1^{k+1} - \frac{\beta_1^k}{\mu_2}), \tag{8}$$

where $\tau_\lambda(\omega) = (t_\lambda(\omega_1), t_\lambda(\omega_2), \cdots, t_\lambda(\omega_n))^T$ for all $\omega \in R^n$ and $t_\lambda(\omega_i) = \text{sign}(\omega_i)\max(0, |\omega_i| - \lambda)$.

After solving the problem (5) by using the similar way, we can get the following implementation algorithm for linear 1-NRTSVM.

3.2 Nonlinear 1-NRTSVM

In this subsection, we consider the nonlinear version of 1-NRTSVM by means of kernel skills. Let $k : R^n \times R^n \to R$ be a Mercer kernel function and $C = [x_1, \cdots, x_l]$. Put

$$K(A, C) = [k(x_i^1, x_j)], i = 1, \cdots, l_1, j = 1, \cdots, l,$$
$$K(B, C) = [k(x_i^2, x_j)], i = 1, \cdots, l_2, j = 1, \cdots, l,$$
$$K(x, C) = [k(x, x_1), \cdots, k(x, x_l)].$$

The aim of nonlinear 1-NRTSVM is to seek a pair of nonparallel hyperplanes $K(x, C)v_1 + b_1 = 0$ and $K(x, C)v_2 + b_2 = 0$ for automatical feature selection and data classification simultaneously by considering the problems (4) and (5), in which A^T and B^T are replaced by $K(A, C)$ and $K(B, C)$, respectively. With the similar way in Subsection 3.1, we can obtain the following solving algorithm for nonlinear 1-NRTSVM.

4 Experiments

In this section, in order to demonstrate the effectiveness of the proposed 1-NRTSVM with linear and nonlinear versions, a series of comparative experiments with TSVM, LSTSVM and NELSTSVM are performed on the classification accuracy, feature selection and computing time of classifiers and on 9 datasets taken from UCI database [20] and 4 datasets taken from synthetic NDCC database [21]. All the experiments are implemented by using 10-fold cross-validation method and in MATLAB (2013a) [22] running on a PC with system configuration Intel Core2 Celeron (2.6 GHz) with 2 GB of RAM.

It is known that the performance of classifiers seriously depends on the choice of parameters. In order to facilitate the comparison, take $c_1 = c_3 = 1, c_2 = c_4 = 0.1, \mu_1 = \mu_3$ and $\mu_2 = \mu_4$ in all experiments and select μ_1, μ_2 from 2^{-8} to 2^8 by grid search. The selected results of μ_1, μ_2 for linear classifiers are listed in Tables 1-2 and are $\mu_1 = 1, \mu_2 = 2^{-8}$ for nonlinear classifiers. In addition, for nonlinear classifiers, Gaussian RBF kernel is used with the kernel parameter $\sigma = 2^8$. Experiment results are listed in Tables 1-3, in which Num, Dim, CA, NSF and CT denote the numbers of training examples, the dimension of training examples, classification accuracy (%), the number of selected features and computing time (second), respectively. More intuitive

Algorithm 1 Linear 1-NRTSVM

Input: Given a set of data T and a tolerance $\epsilon > 0$. Put $k = 0$ and take arbitrarily $w_i^k, b_i^k, u_i^k, v_i^k, \alpha_i^k, \beta_i^k$ for $i = 1, 2$;

repeat
  Update (w_1, b_1) by solving the linear system of equations $\frac{\partial \widetilde{L}(w_1,b_1)}{\partial w_1} = 0$, $\frac{\partial \widetilde{L}(w_1,b_1)}{\partial b_1} = 0$;
  Update u_1 by (7);
  Update v_1 by (8);
  Update α_1 by $\alpha_1^{k+1} = \alpha_1^k + \mu_1(u_1^{k+1} - (e_2 + (B^T w_1^{k+1} + e_2 b_1^{k+1}))$;
  Update β_1 by $\beta_1^{k+1} = \beta_1^k + \mu_2(v_1^{k+1} - w_1^{k+1})$;
until The stopping criteria is satisfied or the maximum number of iteration is achieved;
Put $w_1^* \leftarrow w_1^{k+1}$, $b_1^* \leftarrow b_1^{k+1}$, $v_1^* \leftarrow v_1^{k+1}$;
Update $(w_2, b_2, u_2, v_2, \alpha_2, \beta_2)$ by the similar procedure;
Construct decision functions $f_1(x) = \frac{<{v_1}^*, x> + b_1^*}{\|v_1^*\|_2}$ and $f_2(x) = \frac{<{v_2}^*, x> + b_2^*|}{\|v_2^*\|_2}$;
For a new pattern $\widetilde{x} \in R^n$, its label can be predicted by label$(\widetilde{x}) = \arg\min_{k=1,2} |f_k(\widetilde{x})|$.

Algorithm 2 Nonlinear 1-NRTSVM

Input: Given a set of data T and a tolerance $\epsilon > 0$. Put $k = 0$ and take arbitrarily $w_i^k, b_i^k, u_i^k, v_i^k, \alpha_i^k, \beta_i^k$ for $i = 1, 2$;

repeat (in the following, A^T and B^T are replaced by $K(A, C)$ and $K(B, C)$, respectively)
  Update (w_1, b_1) by solving the linear system of equations $\frac{\partial \widetilde{L}(w_1,b_1)}{\partial w_1} = 0$ and $\frac{\partial \widetilde{L}(w_1,b_1)}{\partial b_1} = 0$;
  Update u_1 by (7);
  Update v_1 by (8);
  Update α_1 by $\alpha_1^{k+1} = \alpha_1^k + \mu_1(u_1^{k+1} - (e_2 + (K(B, C)w_1^{k+1} + e_2 b_1^{k+1}))$;
  Update β_1 by $\beta_1^{k+1} = \beta_1^k + \mu_2(v_1^{k+1} - w_1^{k+1})$;
until The stopping criteria is satisfied or the maximum number of iteration is achieved;
Put $w_1^* \leftarrow w_1^{k+1}$, $b_1^* \leftarrow b_1^{k+1}$ and $v_1^* \leftarrow v_1^{k+1}$;
Update $(w_2, b_2, u_2, v_2, \alpha_2, \beta_2)$ by the similar procedure;
Construct decision functions $f_1(x) = \frac{|K(x,C)v_1^* + b_1^*|}{\sqrt{{v_1^*}^T K(C,C) v_1^*}}$ and $f_2(x) = \frac{|K(x,C)v_2^* + b_2^*|}{\sqrt{{v_2^*}^T K(C,C) v_2^*}}$;
For a new pattern $\widetilde{x} \in R^n$, its label can be predicted by label$(\widetilde{x}) = \arg\min_{k=1,2} |f_k(\widetilde{x})|$.

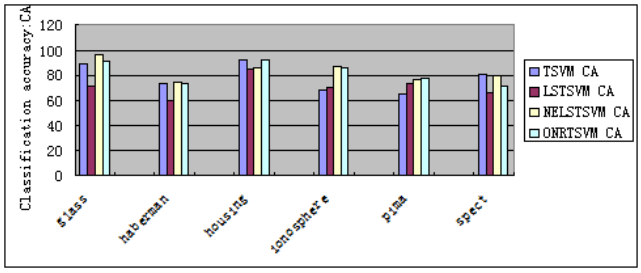

Figure 1: Classification accuracy of linear classifiers on 6 UCI datasets.

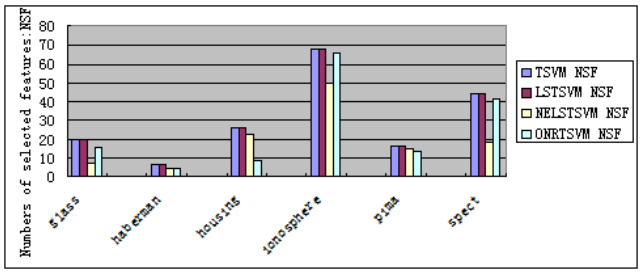

Figure 2: Number of selected features of linear classifiers on 6 UCI datasets.

comparison results can be found in Figure 1-Figure 7. In Figure 5, we changed the computation time to logarithmic computation time to show the figure more clearly.

We can see from the Table 1 that (1) the numbers of selected features of NELSTSVM and 1-NRTSVM are significantly less than that of TSVM and LSTSVM; (2) the classification accuracies of NELSTSVM and 1-NRTSVM are higher than that of TSVM and LSTSVM except spect dataset; (3) for housing, ionosphere and pima three datasets, 1-NRTSVM achieves higher classification accuracy than NELSTSVM; (4) for haberman, housing and pima three datasets, 1-NRTSVM selects less

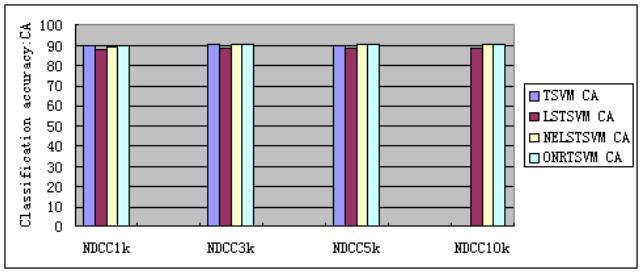

Figure 3: Classification accuracy of linear classifiers on 4 NDCC datasets.

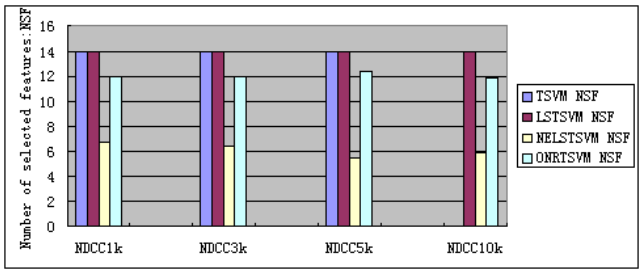

Figure 4: Number of selected features of linear classifiers on 4 NDCC datasets.

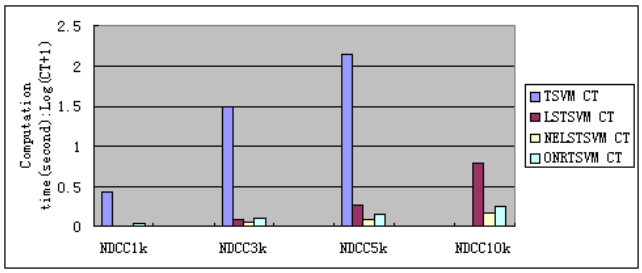

Figure 5: Computing time of linear classifiers on 4 NDCC datasets.

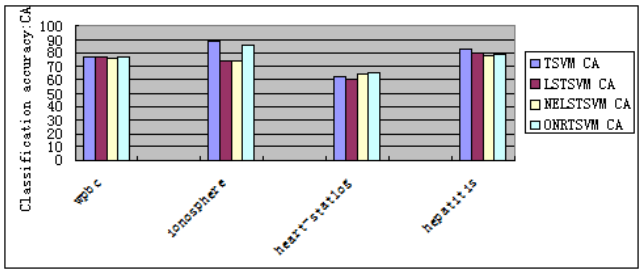

Figure 6: Classification accuracy of nonlinear classifiers on 4 UCI datasets.

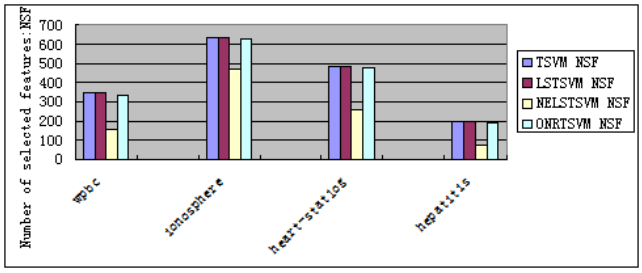

Figure 7: Number of selected features of nonlinear classifiers on 4 UCI datasets.

Table 1: CA and NSF for linear classifiers on 6 UCI datasets

Datasets(Num× Dim)	TSVM	LSTSVM	NELSTSVM	1-NRTSVM	
	CA (%) NSF	CA (%) NSF	CA (%) NSF	CA (%) NSF	μ_1 μ_2
glass (214×10)	88.81±5.91	70.87±12.93	96.26±1.49	91.08±7.47	2
	20	20	7.50±2.60	15.80±1.32	0.25
haberman (306×3)	73.53±0.99	60.11±5.62	74.22±9.80	73.53±0.72	2
	6	6	4.70±0.87	4.30±0.48	0.125
housing (506×13)	93.09±0.96	84.96±4.12	85.57±4.14	93.09±0.98	2
	26	26	22.40±0.93	8.70±1.25	0.5
ionosphere (351×34)	68.38±5.27	70.17±5.86	84.17±8.88	85.48±4.90	2
	68	68	49.20±1.12	65.20±1.55	2
pima (768×8)	65.11±0.36	72.89±5.24	76.16±2.75	77.73±3.21	0.5
	16	16	15.20±4.33	13.50±0.53	2
spect (80×22)	81.25±12.15	66.25±20.45	80.19±6.56	71.25±11.86	1
	44	44	18.60±3.22	40.80±1.40	1

Table 2: CA, NSF and CT for linear classifiers on 4 NDCC datasets

Datasets	TSVM	LSTSVM	NELSTSVM	1-NRTSVM	
	CA (%) NSF CT (s)	CA (%) NSF CT (s)	CA (%) NSF CT (s)	CA (%) NSF CT (s)	μ_1 μ_2
NDCC1k	89.82±3.74	87.80±3.65	89.20±1.77	89.90±2.55	0.5
	14	14	6.80±1.20	12.00±0.00	4
	1.7252	0.0209	0.0295	0.0854	
NDCC3k	90.43±1.78	88.20±1.15	90.30±1.05	90.53±2.03	0.5
	14	14	6.40±0.89	12.00±0.00	4
	29.4924	0.2302	0.1156	0.2562	
NDCC5k	89.98±1.29	88.66±0.87	90.30±1.14	90.52±0.91	0.5
	14	14	5.50±1.55	12.40±0.52	4
	140.749	0.8427	0.2365	0.4001	
NDCC10k	*	88.15±0.93	90.33±0.45	90.34±0.92	0.5
	*	14	5.90±1.58	11.90±0.32	2
	*	5.2188	0.4653	0.7862	

* We stopped experiments as computing time was very high.

Table 3: CA and NSF for nonlinear classifiers on 4 UCI datasets

Datasets(Num× Dim)	TSVM	LSTSVM	NELSTSVM	1-NRTSVM
	CA (%) NSF	CA (%) NSF	CA (%) NSF	CA (%) NSF
wpbc (194×32)	76.29±2.10	76.29±2.10	75.71±4.37	76.29±2.10
	349.2	349.2	155.0±5.73	330.7±1.06
ionosphere (351×34)	88.34±6.77	73.79±6.84	74.22±9.80	85.43±3.78
	631.8	631.8	470.0±0.87	628.7±2.79
heart-statlog (270×14)	62.99±6.04	59.61±5.37	64.44±12.44	65.17±9.25
	486	486	263.0±1.24	482.3±1.64
hepatitis (112×18)	82.18±7.14	80.34±3.58	78.21±8.62	78.53±6.32
	201.6	201.6	75.00±3.94	188.9±1.97

input features than NELSTSVM, for example, 1-NRTSVM selects 8.7 features for housing dataset whereas NELSTSVM, TSVM and LSTSVM select 22.4, 26 and 26 features, respectively.

From the Table 2, we can see that (1) the classification accuracy of 1-NRTSVM are slightly higher than that of TSVM, LSTSVM and NELSTSVM; (2) the number of selected features of 1-NRTSVM is less than that of TSVM and LSTSVM and more than that of NELSTSVM; (3) the computing time of 1-NRTSVM is significantly less than that of TSVM and LSTSVM and more than that of NELSTSVM, this is because that the number of selected features of 1-NRTSVM is nearly two times that of NELSTSVM.

We can find from the Table 3 that the numbers of selected features are significantly more than that in Tables 1-2. It is because that for nonlinear classifiers, selection of input features are performed in the high-dimensional reproduction kernel Hilbert space (RKHS) of the underlying Gaussian RBF kernel. In addition, we can see from the Table 3 that (1) the number of selected features by 1-NRTSVM is less than that by TSVM and LSTSVM and more than that by NELSTSVM; (2) the classification accuracy of 1-NRTSVM is higher than that of LSTSVM and NELSTSVM in general and is higher than that of TSVM on heart-statlog dataset and less than that on ionosphere and hepatitis two datasets.

According to the above analysis, we can conclude that the proposed 1-NRTSVM with linear and nonlinear versions is an effective and competitive classifier for data classification and automatical feature selection.

5 Conclusions

The proposed 1-NRTSVM with linear and nonlinear versions in this paper has two advantages. One is that classification and automatical feature selection of data can be carried out simultaneously. Another is that 1-NRTSVM needs only considering the primal modelings not dual modelings. Two effective algorithms for solving 1-NRTSVM are presented by means of ADMM. Experiment results show that the proposed 1-NRTSVM is an effective and competitive classifier. Along this research direction, there are still a lot of work to do, such as generalization of modelings, improvement of algorithms and selection of kernel functions and kernel parameters. In addition, an extension of 1-NRTSVM for multi-class classification problems can also be considered.

References:

[1] C. Cortes, V. Vapnik, Support-vector network, Machine Learning, 20 (1995) 273–297.

[2] V. Vapnik, The Nature of Statistical Learning, 2nd ed., Springer, New York, (1998).

[3] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines, In Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference, 2003.

[4] L. Wang, J. Zhu, and H. Zou. The doubly regularized support vector machine, Statistica Sinica, 16 (2) (2006) 589–591.

[5] G.B. Ye, Y.F. Chen, X.H. Xie, Efficient variable selection in support vector machines via the alternating direction method of multipliers, Journal of Machine Learning Research, 15 (2011) 832–840.

[6] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machine for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (5) (2007) 905–910.

[7] M.A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Systems with Applications, 36 (4) (2009) 7535–7543.

[8] X. Peng, TPMSVM: A novel twin parametric-margin support vector for pattern recognition, Pattern Recognition, 44 (10–11) (2011) 2678–2692.

[9] Y.H. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, IEEE Transactions on Neural Networks, 22 (6) (2011) 962–968.

[10] W.J. Chen, Y.H. Shao, N. Hong, Laplacian smooth twin support vector machine for semi-supervised classification, International Journal of Machine Learning and Cybernetics, 5 (3) (2014) 459–468.

[11] Y.H. Shao, W.J. Chen, J.J. Zhang, Z. Wang, N.Y. Deng, An efficient weighted Lagrangian twin support vector machine for imbalanced data classification, Pattern Recognition, 47(9) (2014) 3158–3167.

[12] Z. Zhang, L. Zhen, N. Deng, J. Tan, Sparse least square twin support vector machine with adaptive norm, Applied Intelligence, 41(4) (2014) 1097–1107.

[13] D. Tomar, S. Agarwal, Feature selection based least square twin support vector machine for diagnosis of heart disease, International Journal of Bio-Science and Bio-Technology, 6(2) (2014) 69–82.

[14] R. Khemchandani, K. Goyal, S. Chandra, TWSVR: Regression via Twin Support Vector Machine, Neural Networks, Available online 3 November 2015.

[15] S.B. Gao, Q.L. Ye, N. Ye, 1-Norm least squares twin support vector machines, Neurocomputing,74 (17) (2011) 3590–3597.

[16] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Computers and Mathematics with Applications, 2(1) (1976) 17–40.

[17] O.Y. Hua, H. Niao, T. Long, G. Alexander, Stochastic Alternating Direction Method of Multipliers, Journal of Machine Learning Research, 28 (1) (2013) 80–88.

[18] G.Q. Zhang, Bi-alternating direction method of multipliers, IEEE International Conference on: Speech and Signal Processing (ICASSP), (5) (2013) 26–31.

[19] G.B. Ye, X. Xie, Split Bregman method for large scale fused Lasso, Computational Statistics and Data Analysis, 55 (4) (2011) 1552–1569.

[20] P.M. Muphy, D.W. Aha, UCI repository of machine learning databases, 1992.

[21] http://www.cs.wisc.edu/ musicant/data/ndc.

[22] http:// www.mathworks.com.