Improved Nonparallel Hyperplanes Support Vector Machines for Multi-class Classification

Fusheng Bai and Ruijie Liu
School of Mathematical Sciences
Chongqing Normal University
Chongqing 401331, China
Email: fsbai@cqnu.edu.cn, 969322478@qq.com

***Abstract*—In this paper, we present an improved nonparallel hyperplanes classifier for multi-class classification, termed as INHCMC. As in the nonparallel support vector machine (NPSVM) for binary classification, the ε-insensitive loss function is adopted in the primal problems of multi-class classification to improve the sparseness associated with the nonparallel hyperplanes classifier for multi-class classification (NHCMC) where the quadratic loss function is used. Experimental results on some benchmark datasets are reported to show the effectiveness of our method in terms of sparseness and classification accuracy.**

***Index Terms*—Multi-class classification; Nonparallel hyperplanes classifier; Support vector machine**

I. Introduction

Support vector machines (SVMs) are proposed by Vapnik and his co-workers [1], [11], [12] for classification, regression, or other problems in machine learning. It is well known that there are three essential elements which make SVMs so successful: the principle of maximum margin, dual theory, and kernel trick. The sequential minimization optimization (SMO) can be applied to solve the resulting optimization problems efficiently [10].

In the last decade, some nonparallel hyperplane classifiers which are different from the standard support vector classification have been proposed. The twin support vector machine (TWSVM) [3] seeks two nonparallel proximal hyperplanes such that each hyperplane is closest to one of the two classes and as far as possible to the other class. TWSVMs have been studied extensively [6], [7], [8], [9].

Note that although TWSVM only solves two smaller convex quadratic programming problems, the inverse of matrices need to be computed in the algorithm [8], which tends to be intractable for problems with large datasets. For the nonlinear case, the kernel generated surfaces instead of hyperplanes are considered and two extra primal problems are constructed. In TWSVM a quadratic loss function is adopted to make the proximal hyperplane close to one class and a soft-margin loss function is adopted for the other class, which causes the so called phenomenon of semi-sparseness [13]. The nonparallel support vector machine (NPSVM) proposed in [13] has overcame many drawbacks of TWSVM. The quadratic loss function in TWSVM is replaced by the ε-insensitive loss function in NPSVM to improve the semi-sparsity. Experimental results reported in [13] show the improvement of the sparseness and classification accuracy of NPSVM.

For the multi-class classification problem, a new multi-class classifier, termed as nonparallel hyperplanes classifier for multi-class classification (NHCMC), has been proposed in [5], which seeks K hyperplanes by solving K quadratic programming problems (QPPs). The QPPs are designed to put the k-th class as far as possible to the k-th hyperplane while the rest points are proximal to the k-th hyperplanes. To assign a new point to one class is depending on which of the K hyperplanes it lies farthest to. However, NHCMC lost the sparseness by using a quadratic and a soft-margin loss function for each class to some extent.

In this paper, we propose an improved nonparallel hyperplanes classifier for multi-class classification problems (INHCMC). In order to improve the sparsity of NHCMC, we change the quadratic loss function to the ε-insensitive loss function as in NPSVM. The numerical experiments indicate that the classification accuracy is also improved.

This paper is organized as follows. In Section 2, we briefly introduce the standard support vector classification (SVC), TWSVM, NPSVM and NHCMC. The INHCMC is presented in Section 3. Some experimental results are reported in Section 4. We give concluding remarks in Section 5.

II. Background

In this section, we introduce the standard C-SVC, TWSVM, and NPSVM.

A. C-SVM

Consider the binary classification problem with the training set

$$T = \{(x_1, y_1), \ldots, (x_l, y_l)\},$$

where $x_i \in R^n, y_i \in \gamma = \{1, -1\}, i = 1, \ldots, l$. The standard C-SVC formulates the problem as a convex quadratic programming problem (QPP)

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

where $\xi = (\xi_1, \ldots, \xi_l)^T$, and $C > 0$ is a penalty parameter. C-SVC solves its Lagrangian dual problem

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{l}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{l} y_i\alpha_i = 0 \\ & 0 \le \alpha_i \le C,\ i = 1, \ldots, l. \end{aligned}$$

where $K(x, x')$ is the kernel function, which is also a convex QPP, and then constructs the decision function accordingly.

B. TWSVM

Consider the binary classification problem with the training set

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

where $x_i \in R^n$,$i = 1, \ldots, p+q$. Let $A = (x_1, \ldots, x_p)^T \in R^{p\times n}$, $B = (x_{p+1}, \ldots, x_{p+q})^T \in R^{q\times n}$ and $l = p + q$. For linear classification problem, TWSVM seeks a pair of nonparallel hyperplanes

$$(\omega_+ \cdot x) + b_+ = 0 \text{ and } (\omega_- \cdot x) + b_- = 0$$

by solving two smaller convex QPPs

$$\begin{aligned} \min_{\omega_+, b_+, \xi_-} \quad & \frac{1}{2}\sum_{i=1}^{p}((\omega_+ \cdot x_i) + b_+)^2 + d_1 \sum_{j=p+1}^{p+q}\xi_j \\ \text{s.t.} \quad & (\omega_+ \cdot x_j) + b_+ \le -1 + \xi_j, j = p+1, \ldots, p+q \\ & \xi_j \ge 0, j = p+1, \ldots, p+q \\ \min_{\omega_-, b_-, \xi_+} \quad & \frac{1}{2}\sum_{i=p+1}^{p+q}((\omega_- \cdot x_i) + b_-)^2 + d_2 \sum_{j=1}^{p}\xi_j \\ \text{s.t.} \quad & (\omega_- \cdot x_j) + b_- \ge 1 - \xi_j, j = 1, \ldots, p \\ & \xi_j \ge 0, j = 1, \ldots, p, \end{aligned}$$

where d_i, i=1, 2 are penalty parameters. For nonlinear classification problem, two kernel-based surfaces are constructed and two other primal problem are considered.

C. NPSVM

For training set (1), NPSVM constructs two QPPs as follows:

$$\begin{aligned} \min_{\omega_+, b_+, \eta_+^{(*)}, \xi_-} \quad & \frac{1}{2}\|\omega_+\|^2 + C_1\sum_{i=1}^{p}(\eta_i + \eta_i^*) + C_2\sum_{j=p+1}^{p+q}\xi_j \\ \text{s.t.} \quad & (\omega_+ \cdot x_i) + b_+ \le \varepsilon + \eta_i, i = 1, \ldots, p \\ & -(\omega_+ \cdot x_i) - b_+ \le \varepsilon + \eta_i^*, i = 1, \ldots, p \\ & (\omega_+ \cdot x_j) + b_+ \le -1 + \xi_j, \\ & \qquad j = p+1, \ldots, p+q \\ & \eta_i, \eta_i^* \ge 0,\ i = 1, \ldots, p \\ & \xi_j \ge 0,\ j = p+1, \ldots, p+q, \end{aligned}$$

$$\begin{aligned} \min_{\omega_-, b_-, \eta_-^{(*)}, \xi_+} \quad & \frac{1}{2}\|\omega_-\|^2 + C_3\sum_{i=p+1}^{p+q}(\eta_i + \eta_i^*) + C_4\sum_{j=1}^{p}\xi_j \\ \text{s.t.} \quad & (\omega_- \cdot x_i) + b_- \le \varepsilon + \eta_i, \\ & \qquad i = p+1, \ldots, p+q \\ & -(\omega_- \cdot x_i) - b_- \le \varepsilon + \eta_i^*, \\ & \qquad i = p+1, \ldots, p+q \\ & (\omega_- \cdot x_j) + b_- \le 1 - \xi_j,\ j = 1, \ldots, p \\ & \eta_i, \eta_i^* \ge 0,\ i = p+1, \ldots, p+q \\ & \xi_j \ge 0,\ j = 1, \ldots, p, \end{aligned}$$

where $x_i, i = 1, \ldots, p$ are positive inputs, $x_i, i = p+1, \ldots, p+q$ are negative inputs, $C_i \ge 0$, $i = 1, \ldots, 4$ are penalty parameters, and $\xi_+ = (\xi_1 \ldots, \xi_p)^T, \xi_- = (\xi_{p+1}, \ldots, \xi_{p+q})^T, \eta_+^{(*)} = (\eta_+^T, \eta_+^{*T})^T = (\eta_1, \ldots, \eta_p, \eta_1^*, \ldots, \eta_p^*)^T, \eta_-^{(*)} = (\eta_-^T, \eta_-^{*T})^T = (\eta_{p+1}, \ldots, \eta_{p+q}, \eta_{p+1}^*, \ldots, \eta_{p+q}^*)^T$, are slack variables.

D. NHCMC

Consider the multiple classification problem with the training set:

$$T = \{(x_1, y_1), \ldots, (x_l, y_l)\}, \quad (2)$$

where $x_i \in R^n, i = 1, \ldots, l$, and $y_i \in \{1, \ldots, K\}$ is the corresponding class of x_i. For convenience, denote the number of points belonging to the k-th class as l_k and these points are denoted as $A_k \in R^{l_k \times n}, k = 1, \ldots, K$. Then the matrix

$$B_k = [A_1^T, \ldots, A_{k-1}^T, A_{k+1}^T, \ldots, A_K^T]^T \quad (3)$$

denote all the points except for the points belonging to the k-th class. NHCMC seeks K nonparallel hyperplanes

$$(\omega_k \cdot x) + b_k = 0, k = 1, \ldots, K \quad (4)$$

by solving the following convex QPPs:

$$\begin{aligned} \min_{\omega_k, b_k, \eta_k, \xi_k} \quad & \frac{1}{2}C_1\|\omega_k\|^2 + \frac{1}{2}\eta_k^T\eta_k + C_2 e_{k_2}^T\xi_k \\ \text{s.t.} \quad & B_k\omega_k + e_{k_1}b_k = \eta_k \\ & (A_k\omega_k + e_{k_2}b_k) + \xi_k \ge e_{k_2} \\ & \xi_k \ge 0. \end{aligned}$$

where $\eta_k \in R^{(l-l_k)}$ is a variable, ξ_k is a slack variable, $e_{k_1} \in R^{(l-l_k)}$ and $e_{k_2} \in R^{l_k}$ are the vectors of ones, $C_1 \ge 0$ and $C_2 \ge 0$ are penalty parameters.

III. Improved Nonparallel Hyperplanes Classifier for Multi-Class Classification (INHCMC)

A. Linear INHCMC

For the multi-class classification problem with the train set (2), We aim to construct K nonparallel hyperplanes (4) by

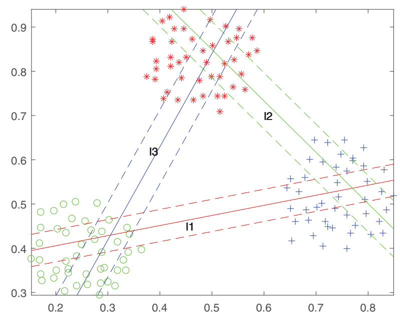

Fig. 1. A toy example learned by INHCMC

solving the following convex QPP for each k:

$$\begin{aligned}
\min_{\omega_k,b_k,\eta_k,\eta_k^*,\xi_k} \quad & \frac{1}{2}\|\omega_k\|^2 + C_1 e_{k_1}^T(\eta_k+\eta_k^*) + C_2 e_{k_2}^T\xi_k \\
\text{s.t.} \quad & B_k\omega_k + e_{k_1}b_k \le \varepsilon e_{k_1} + \eta_k \\
& -B_k\omega_k - e_{k_1}b_k \le \varepsilon e_{k_1} + \eta_k^* \\
& (A_k\omega_k + e_{k_2}b_k) + \xi_k \ge e_{k_2} \\
& \eta_k,\ \eta_k^*,\ \xi_k \ge 0.
\end{aligned} \tag{5}$$

where $\eta_k, \eta_k^* \in R^{(l-l_k)}$ are variables, ξ_k is a slack variable, B_k is given in (3), $e_{k_1} \in R^{(l-l_k)}$ and $e_{k_2} \in R^{l_k}$ are the vectors of ones, $C_1 \ge 0$ and $C_2 \ge 0$ are penalty parameters.

In order to illustrate the primal problem of INHCMC, we generated an artificial two dimensional three-class dataset. The geometric interpretation of above problem with $x \in R^2$ is shown in Figure 1. Take the " $*$ " class in Figure 1 as an example. We hope the hyperplane of the " $*$ " class l_1 is far from the "$*$" points and close to the "$+$" and "$\circ$" points. In order to minimize the misclassification, the points of the k-th class are at distance 1 from the hyperplane, and we minimize the sum of error variables with soft margin loss.

To obtain the solution to problem (5), we need to derive its dual problem. The Lagrangian of problem (5) is given by

$$\begin{aligned}
& L(\omega_k, b_k, \eta_k, \eta_k^*, \xi_k, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) \\
=& \frac{1}{2}\|\omega_k\|^2 + C_1 e_{k_1}^T(\eta_k+\eta_k^*) + C_2 e_{k_2}^T\xi_k \\
& - \alpha_1^T(-B_k\omega_k - e_{k_1}b_k + \varepsilon e_{k_1} + \eta_k) \\
& - \alpha_2^T(B_k\omega_k + e_{k_1}b_k + \varepsilon e_{k_1} + \eta_k^*) \\
& - \alpha_3^T(A_k\omega_k + e_{k_2}b_k + \xi_k - e_{k_2}) \\
& - \alpha_4^T\xi_k - \alpha_5^T\eta_k - \alpha_6^T\eta_k^*,
\end{aligned} \tag{6}$$

where $\alpha_1 = (\alpha_{11}, \ldots, \alpha_{1(l-l_k)})^T$, $\alpha_2 = (\alpha_{21}, \ldots, \alpha_{2(l-l_k)})^T$, $\alpha_3 = (\alpha_{31}, \ldots, \alpha_{3l_k})^T$, $\alpha_4 = (\alpha_{41}, \ldots, \alpha_{4l_k})^T$, $\alpha_5 = (\alpha_{51}, \ldots, \alpha_{5(l-l_k)})^T$, $\alpha_6 = (\alpha_{61}, \ldots, \alpha_{6(l-l_k)})^T$ are the Lagrange multiplier vectors. From the Karush-Kuhn-Tucker (KKT) condition for $\omega_k, b_k, \eta_k, \eta_k^*, \xi_k$ and $\alpha_i (i = 1, \ldots, 6)$, it follows

$$\nabla_{\omega_k} L = \omega_k + B_k^T\alpha_1 - B_k^T\alpha_2 - A_k^T\alpha_3 = 0 \tag{7}$$
$$\nabla_{b_k} L = e_{k_1}^T\alpha_1 - e_{k_1}^T\alpha_2 - e_{k_2}^T\alpha_3 = 0 \tag{8}$$
$$\nabla_{\eta_k} L = C_1 e_{k_1} - \alpha_1 - \alpha_5 = 0 \tag{9}$$
$$\nabla_{\eta_k^*} L = C_1 e_{k_1} - \alpha_2 - \alpha_6 = 0 \tag{10}$$
$$\nabla_{\xi_k} L = C_2 e_{k_2} - \alpha_3 - \alpha_4 = 0 \tag{11}$$
$$B_k\omega_k + e_{k_1}b_k \le \varepsilon e_{k_1} + \eta_k \tag{12}$$
$$-B_k\omega_k - e_{k_1}b_k \le \varepsilon e_{k_1} + \eta_k^* \tag{13}$$
$$(A_k\omega_k + e_{k_2}b_k) + \xi_k \ge e_{k_2} \tag{14}$$
$$\eta_k, \eta_k^*, \xi_k \ge 0. \tag{15}$$

Since $\alpha_4, \alpha_5, \alpha_6 \ge 0$, from (9) – (11) we have

$$0 \le \alpha_1, \alpha_2 \le C_1 e_{k1}, 0 \le \alpha_3 \le C_2 e_{k2},$$

and from (7), we have

$$\omega_k = B_k^T(\alpha_2 - \alpha_1) + A_k^T\alpha_3. \tag{16}$$

Then putting (16) into the Lagrangian and using (7)-(15), we obtain the dual problem of problem (5)

$$\begin{aligned}
\min_{\hat{\pi}} \quad & \frac{1}{2}\hat{\pi}^T\hat{\Lambda}\hat{\pi} + \hat{\kappa}^T\hat{\pi}, \\
\text{s.t.} \quad & \hat{e}^T\hat{\pi} = 0, \\
& 0 \le \hat{\pi} \le \hat{C}
\end{aligned} \tag{17}$$

where

$$\begin{aligned}
\hat{\pi} &= (\alpha_2^T, \alpha_1^T, \alpha_3^T)^T \\
\hat{\kappa} &= (\varepsilon e_{k_1}^T, \varepsilon e_{k_1}^T, -e_{k_2}^T)^T \\
\hat{e} &= (-e_{k_1}^T, e_{k_1}^T, -e_{k_2}^T)^T \\
\hat{C} &= (C_1 e_{k_1}^T, C_1 e_{k_1}^T, C_2 e_{k_2}^T)^T,
\end{aligned}$$

and

$$\hat{\Lambda} = \begin{pmatrix} Q_1 & Q_2 \\ Q_2^T & Q_3 \end{pmatrix}, Q_1 = \begin{pmatrix} B_kB_k^T & -B_kB_k^T \\ -B_kB_k^T & B_kB_k^T \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} B_kA_k^T \\ -B_kA_k^T \end{pmatrix}, Q_3 = A_kA_k^T.$$

For (17), by applying the KKT condition, we can obtain the following conclusions, which is similar to the corresponding conclusions in [13], therefore the proofs are omitted.

Theorem 3.1: Suppose that $\hat{\pi} = (\alpha_2^T, \alpha_1^T, \alpha_3^T)^T$ is a solution to the problem (17), then for $i = 1, \ldots, l - l_k$, each pair of α_{1i} and α_{2i} cannot be both simultaneously nonzero, i.e., $\alpha_{1i}\alpha_{2i} = 0$, $i = 1, \ldots, l - l_k$.

Theorem 3.2: Suppose that $\hat{\pi} = (\alpha_2^T, \alpha_1^T, \alpha_3^T)^T$ is a solution to the problem (17). If there exist components of $\hat{\pi}$ with value in the interval $(0, \hat{C})$, then the solution (ω_k, b_k) to the problem (5) can be computed as follows: Let

$$\omega_k = B_k^T(\alpha_2 - \alpha_1) + A_k^T\alpha_3$$

and choose a component of $\alpha_1, \alpha_{1j} \in (0, C_1)$, B_{kj} is the j-th row of B_k, then let

$$b_k = -B_{kj}\omega_k + \varepsilon,$$

or choose a component of $\alpha_2, \alpha_{2m} \in (0, C_1)$, B_{km} is the m-th row of B_k, then let

$$b_k = -B_{km}\omega_k - \varepsilon,$$

or choose a component of $\alpha_3, \alpha_{3u} \in (0, C_2)$, A_{ku} is the u-th row of A_k, then let

$$b_k = -A_{ku}\omega_k + 1.$$

From Theorem 3.2 it can be seen that the inherent semi-sparseness in the NHCMC is improved due to the introduction of ε-insensitive loss function instead of the quadratic loss function for each class, as pointed out in [14].

Once the solution (ω_k, b_k) to the problem (5) is obtained for each $k = 1, \ldots, K$, a new point $x \in R^n$ is assigned to the class by the decision function

$$f(x) = \arg\max_{k=1,\ldots,K} \frac{|(\omega_k \cdot x) + b_k|}{\|\omega_k\|}$$

where $\frac{|(\omega_k \cdot x)+b_k|}{\|\omega_k\|}$ is the perpendicular distance from x to the hyperplane $(\omega_k \cdot x) + b_k = 0, k = 1, \ldots, K$.

B. Nonlinear INHCMC

As only inner products appear in the dual problem (17), the kernel functions can be applied directly to the problems in order to extend the linear INHCMC to the nonlinear case. The corresponding dual problem to be solved is as follows

$$\begin{aligned} \min_{\hat{\pi}} \quad & \frac{1}{2}\hat{\pi}^T\hat{\Lambda}\hat{\pi} + \hat{\kappa}^T\hat{\pi} \\ \text{s.t.} \quad & \hat{e}^T\hat{\pi} = 0 \\ & 0 \le \hat{\pi} \le \hat{C}, \end{aligned} \tag{18}$$

where

$$\begin{aligned} \hat{\pi} &= (\alpha_2^T, \alpha_1^T, \alpha_3^T)^T \\ \hat{\kappa} &= (\varepsilon e_{k_1}^T, \varepsilon e_{k_1}^T, -e_{k_2}^T)^T \\ \hat{e} &= (-e_{k_1}^T, e_{k_1}^T, -e_{k_2}^T)^T \\ \hat{C} &= (C_1 e_{k_1}^T, C_1 e_{k_1}^T, C_2 e_{k_2}^T)^T, \end{aligned}$$

and

$$\hat{\Lambda} = \begin{pmatrix} Q_1 & Q_2 \\ Q_2^T & Q_3 \end{pmatrix}, Q_1 = \begin{pmatrix} K(B_k, B_k^T) & K(-B_k, B_k^T) \\ K(-B_k, B_k^T) & K(B_k, B_k^T) \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} K(B_k, A_k^T) \\ K(-B_k, A_k^T) \end{pmatrix}, Q_3 = K(A_k, A_k^T).$$

Similar conclusions to Theorems 3.1 and 3.2 can be established accordingly by using $K(x, x')$ instead of $(x \cdot x')$. The INHCMC algorithm is presented as follows.

Algorithm 1

1) Input the training set (2)
2) Choose appropriate kernels $K(x, x')$, appropriate parameters $\varepsilon > 0$, $C_1, C_2 > 0$ for problem (18)
3) Construct and solve the convex QPP (18), get the solution $\hat{\pi} = (\alpha_2^T, \alpha_1^T, \alpha_3^T)^T$
4) Construct the decision functions $f_k(x) = K(x, B_k^T) \cdot (\alpha_2 - \alpha_1) + K(x, A_k^T) \cdot \alpha_3 + b_k, k = 1, \ldots, K$, where b_k are computed by using Theorem 3.2 for kernel cases;
5) For a new input x, assign it to the class k with $k = \arg\max_{k=1,\ldots,K} \frac{|f_k(x)|}{\|\omega_k\|}$

IV. Numerical experiments

In this section, in order to validate the performance of INHCMC, firstly we apply INHCMC to the Iris dataset [2] which contains three classes (Setosa, Versilcolor, Viginica) and four attributes (sepal length, sepal width, petal length and petal width) for an iris. We aim to classify the class of iris based on the four attributes. The varying percentage of support vectors corresponding to problem (17) and (18) is recorded in Fig. 2. Note that there are three problems to be solved for linear or nonlinear case as the number of classes is 3. It can be seen from the figure that with the increasing ε, the number of support vectors decreases, therefore the semi-sparseness ($\varepsilon = 0$) is improved.

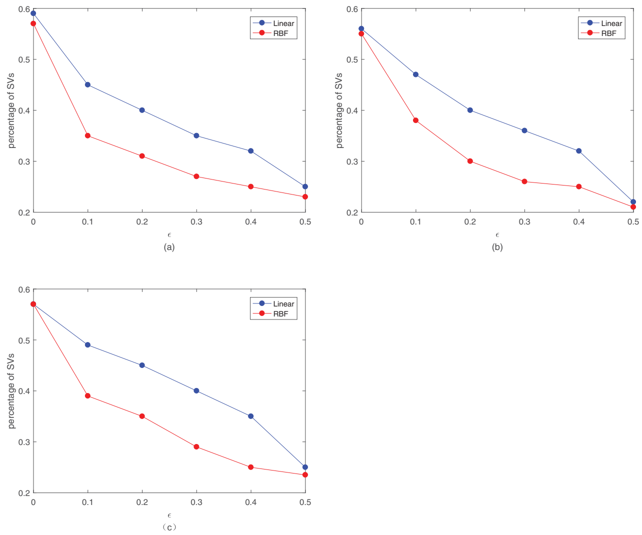

Fig. 2. Increasing of sparseness with the increasing ε

Then we compare the accuracy performance of INHCMC with NHCMC on several publicly available benchmark datasets [2]. These datasets were used in [5] as well. All samples from the datasets are scaled so that the features locate in $[0, 1]$ before training. The RBF kernel $K(x, x') = \exp(\frac{-\|x-x'\|^2}{\sigma})$ is adopted for all the datasets, the parameters C_1, C_2 in INHCMC are tuned for best classification accuracy in the range 2^{-8} to 2^{12}, and the optimal parameter ε in INHCMC is obtained in the range $[0, 0.5]$. The average tenfold cross-validation results of the two methods on the five datasets in terms of accuracy are reported in Table I. It can be seen that the INHCMC produces better accuracy for four out of the five datasets.

TABLE I
AVERAGE RESULTS FOR SOME BENCHMARK DATASETS

Datasets	NHCMC acc(%)	INHCMC acc(%)
Iris	98.45	98.67
Wine	98.62	98.88
Glass	74.55	76.67
Vowel	99.71	98.49
Vehicle	86.91	88.68

V. CONCLUSION

In this paper, we proposed an improved nonparallel hyperplanes classifier for multi-class classification, termed as INHCMC. By using the ε-insensitive loss function instead of the quadratic loss function in the primal problems constructed in NHCMC, the semi-sparsity is improved. It can also be regarded as an extension of NPSVM into the case of multi-class classification. Some preliminary experimental results show the improvement on sparseness and classification accuracy of the proposed method.

REFERENCES

[1] C. Cortes and V. Vapnik (1995), *Support-vector networks*, Machine Learning, 20(3): 273-297.

[2] D. Dua, and E. Karra Taniskidou (2017), UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science.

[3] Jayadeva, R. Khemchandani and S. Chandra (2007), *Twin Support Vector Machines for Pattern Classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(5):905-910.

[4] X. C. Ju and Y. J. Tian, *Efficient Implementation of Nonparallel Hyperplanes Classifier*, Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, 3: 5-9.

[5] X. Ju, Y. Tian, D. Liu and Z. Qi (2015), *Nonparallel Hyperplanes Support Vector Machine for Multi-class Classification*, Procedia Computer Science, 51(1): 1574-1582.

[6] R. Khemchandani, Jayadeva and S. Chandra (2009), *Optimal kernel selection in twin support vector machines*, Optimization Letters, 3(1): 77-88.

[7] M. A. Kumar and M. Gopal (2008), *Application of smoothing technique on twin support vector machines*, Pattern Recognition Letters, 29(13): 1842-1848.

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

[9] X. Peng (2010), *TSVR: An efficient twin support vector machine for regression*, Neural Networks, 23(3): 365-372.

[10] J. C. Platt (1999), *Fast training of support vector machines using sequential minimal optimization*, Advances in Kernel Methods, pp. 185-208, MIT Press, Cambridge, MA, USA.

[11] V. Vapnik (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

[12] V. Vapnik (1999), *An overview of statistical learning theory*, IEEE Transactions on Neural Networks, 10(5): 988-999.

[13] Y. Tian, Z. Qi, X. Ju, Y. Shi, and X. Liu (2014), *Nonparallel support vector machines for pattern classification*, IEEE Transactions on Cybernetics, 44(7): 1067-1079.

[14] Y. Tian, Z. Qi, X. Ju and Y. Shi (2014), *Improved twin support vector machine*. Science China Mathematics, 57(2): 417-432.

[15] Z.-X. Yang, Y.-H. Shao, and X.-S. Zhang (2013), *Multiple birth support vector machine for multi-class classification*, Neural Computing and Applications, 22(Supplement 1): 153-161.