[11] D. J. C. MacKay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, Sep. 1992.
[12] M. H. Law and J. T. Kwok, "Bayesian support vector regression," in *Proc. 8th Int. Workshop Artif. Intell. Stat.*, 2001, pp. 239–244.
[13] J. B. Gao, S. R. Gunn, C. J. Harris, and M. Brown, "A probabilistic framework for SVM regression and error bar estimation," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 71–89, 2002.
[14] W. Chu, S. S. Keerthi, and C. J. Ong, "Bayesian support vector regression using a unified loss function," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 29–44, Jan. 2004.
[15] C. J. Lin and R. C. Weng, "Simple probabilistic predictions for support vector regression," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2004.
[16] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, Nov. 1995.
[17] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.
[18] F. H. Long, H. C. Peng, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[19] E. S. Page, "A note on generating random permutations," *Appl. Stat.*, vol. 16, no. 3, pp. 273–274, 1967.
[20] W. D. Penny, "KL divergences of normal, gamma, Dirichlet and Wishart densities," Dept. Cognit. Neurol., Univ. College London, London, U.K., Tech. Rep., Mar. 2001.
[21] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert–Schmidt norms," in *Proc. 16th Int. Conf. Algorith. Learn. Theory*, Oct. 2005, pp. 63–78.
[22] A. P. A. Silva, V. H. Ferreira, and R. M. Velasquez, "Input space to neural network based load forecasters," *Int. J. Forecast.*, vol. 24, no. 4, pp. 616–629, Oct.–Dec. 2008.
[23] C.-C. Chang and C.-J. Lin. (2001). *LIBSVM: A Library for Support Vector Machines* [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm
[24] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.
[25] A. Asuncion and D. J. Newman. (2007). *UCI Machine Learning Repository* [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[26] J. C. Platt, "Using sparseness and analytic QP to speed training of support vector machines," in *Advances in Neural Information Processing Systems 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1998.

# Improvements on Twin Support Vector Machines

Yuan-Hai Shao, Chun-Hua Zhang, Xiao-Bo Wang, and Nai-Yang Deng

***Abstract*— For classification problems, the generalized eigenvalue proximal support vector machine (GEPSVM) and twin support vector machine (TWSVM) are regarded as milestones in the development of the powerful SVMs, as they use the nonparallel hyperplane classifiers. In this brief, we propose an improved version, named twin bounded support vector machines (TBSVM), based on TWSVM. The significant advantage of our TBSVM over TWSVM is that the structural risk minimization principle is implemented by introducing the regularization term. This embodies the marrow of statistical learning theory, so this modification can improve the performance of classification. In addition, the successive overrelaxation technique is used to solve the optimization problems to speed up the training procedure. Experimental results show the effectiveness of our method in both computation time and classification accuracy, and therefore confirm the above conclusion further.**

***Index Terms*— Machine learning, maximum margin, structural risk minimization principle, support vector machines.**

## I. Introduction

Support vector machines (SVMs), being computationally powerful tools for supervised learning [1]–[3], have already outperformed most other systems in a wide variety of applications [4]–[6]. For the standard support vector classification (SVC), its primal problem can be understood in the following way: construct two parallel support hyperplanes such that, on one hand, the band between the two parallel hyperplanes separates the two classes (the positive and negative data points) well, on the other hand, the width between the two hyperplanes is maximized, leading to the introduction of a regularization term. Thus, the structural risk minimization principle is implemented. The final separating hyperlane is selected to be the "middle one" between the two hyperplanes. Different from SVC with two parallel hyperplanes, some nonparallel hyperplane classifiers such as the generalized eigenvalue proximal support vector machine (GEPSVM) and twin support vector machine (TWSVM) have been proposed in [7] and [8]. TWSVM seeks two nonparallel proximal hyperplanes such that each hyperplane is closest to one of two classes and as far as possible from the other class. A fundamental difference between TWSVM and SVC is that TWSVM solves two smaller sized quadratic programming problems (QPPs), whereas SVC solves one larger QPP. Therefore, TWSVM works faster than SVC. Experimental results in [8], and [9] have shown the effectiveness of TWSVM over both standard SVC and GEPSVM on UCI datasets. In addition, TWSVM is excellent at dealing with the "Cross Planes" dataset. Thus, the methods of constructing the nonparallel hyperplanes have been studied extensively [9]–[12].

It is well known that one significant advantage of SVC is the implementation of the structural risk minimization principle [13], [14]. However, only the empirical risk is considered in the primal problems of TWSVM. In addition, we noticed that the inverse matrices $(G^\top G)^{-1}$ and $(H^\top H)^{-1}$ appear in the dual problems. This implies that, in order to obtain the dual problems, TWSVM must assume that the inverse matrices $(G^\top G)^{-1}$ and $(H^\top H)^{-1}$ exist or the matrices $G^\top G$ and $H^\top H$ are nonsingular. However, this extra prerequisite cannot always be satisfied. So the duality theory in TWSVM is not perfect from the theoretical point of view, although these inverse matrices have been handled by modifying the dual problems technically and elegantly.

Manuscript received September 13, 2009; revised February 11, 2011; accepted March 6, 2011. Date of publication May 5, 2011; date of current version June 2, 2011. This work is supported in part by the National Natural Science Foundation of China, under Grant 10971223 and Grant 11071252.

Y.-H. Shao and N.-Y. Deng are with the College of Science China Agricultural University, Beijing 100083, China (e-mail: shaoyuanhai21@163.com; dengnaiyang@cau.edu.cn).

C.-H. Zhang is with the Department of Mathematics, Information School, Renmin University of China, Beijing 100872, China (zhangchunhua@ruc.edu.cn).

X.-B. Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: xb-wang10@mails.tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2130540

This brief makes some improvements on TWSVM and proposes a modified version, named the twin bounded support vector machines (TBSVM). Similar to TWSVM, TBSVM constructs two nonparallel hyperplanes by solving two smaller QPPs. However, there are some differences: 1) in the primal problems of TWSVM, the empirical risk is minimized, whereas in our TBSVM the structural risk is minimized by adding a regularization term with the idea of maximizing some margin; 2) the dual problems of our primal problems can be derived without any extra assumption and need not be modified any more. From this point of view, we think that our method is more rigorous and complete than TWSVM; and 3) in order to shorten training time, an effective method (successive overrelaxation, SOR) is applied to our TBSVM.

This brief is organized as follows. Section II briefly dwells on the standard SVC and TWSVM. Section III proposes our TBSVM. In Section IV, the SOR method is used to solve the optimization problems in TBSVM. Experimental results are described in Section V, and concluding remarks are given in Section VI.

## II. Background

In this section, we give a brief outline of SVC and TWSVM.

### A. SVC

Consider the classification problem with the training set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$, where $x_i \in R^n$ are inputs and $y_i \in \{+1, -1\}$ are the corresponding outputs.

Linear SVC [13] searches for a separating hyperplane

$$f(x) = w^\top x + b = 0 \tag{1}$$

where $w \in R^n$ and $b \in R$. To measure the empirical risk, the soft margin loss function $\sum_{i=1}^{l} \max(0, 1 - y_i(w^\top x_i + b))$ is used. By introducing the regularization term $1/2||w||^2$ and the slack variable $\xi = (\xi_1, \ldots, \xi_l)$, the primal problem of SVC can be expressed as

$$\begin{array}{ll} \min\limits_{w,b,\xi} & \frac{1}{2}||w||^2 + C\sum\limits_{i=1}^{l}\xi_i \\ \text{s.t.} & y_i(w^\top x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \quad i = 1, \ldots, l \end{array} \tag{2}$$

where $C > 0$ is a parameter. Note that the minimization of the regularization term $1/2\|w\|^2$ is equivalent to the maximization of the margin between two parallel supporting hyperplanes $w^\top x + b = 1$ and $w^\top x + b = -1$. And the structural risk minimization principle is implemented in this problem.

### B. TWSVM

Consider the following classification problem. Suppose that all of the data points in class $+1$ are denoted by a matrix $A \in R^{m_1 \times n}$, where the $i$th row $A_i \in R^n$ represents a data point. Similarly, the matrix $B \in R^{m_2 \times n}$ represents the data points of class $-1$.

Different from SVC, linear TWSVM [8] seeks a pair of nonparallel hyperplanes

$$f_1(x) = w_1^\top x + b_1 = 0 \text{ and } f_2(x) = w_2^\top x + b_2 = 0 \tag{3}$$

such that each hyperplane is proximal to the data points of one class and far from the data points of the other class, where $w_1 \in R^n$, $w_2 \in R^n$, $b_1 \in R$ and $b_2 \in R$. Here, the empirical risks are measured by

$$\begin{aligned} R_1 = &\frac{1}{2}(Aw_1 + e_1b_1)^\top(Aw_1 + e_1b_1) \\ &+ c_1e_2^\top \max(0, e_2 + Bw_1 + e_2b_1) \end{aligned} \tag{4}$$

and

$$\begin{aligned} R_2 = &\frac{1}{2}(Bw_2 + e_2b_2)^\top(Bw_2 + e_2b_2) \\ &+ c_2e_1^\top \max(0, e_1 - Aw_2 - e_1b_2) \end{aligned} \tag{5}$$

where $c_1 > 0$ and $c_2 > 0$ are parameters. By introducing the slack variables $\xi$, $\xi^*$, $\eta$, and $\eta^*$, the primal problems are expressed as

$$\begin{array}{ll} \min\limits_{w_1,b_1,\xi,\xi^*} & \frac{1}{2}\xi^{*\top}\xi^* + c_1e_2^\top\xi \\ \text{s.t.} & Aw_1 + e_1b_1 = \xi^* \\ & -(Bw_1 + e_2b_1) + \xi \geq e_2, \quad \xi \geq 0 \end{array} \tag{6}$$

and

$$\begin{array}{ll} \min\limits_{w_2,b_2,\eta,\eta^*} & \frac{1}{2}\eta^{*\top}\eta^* + c_2e_1^\top\eta \\ \text{s.t.} & Bw_2 + e_2b_2 = \eta^* \\ & (Aw_2 + e_1b_2) + \eta \geq e_1, \quad \eta \geq 0. \end{array} \tag{7}$$

In order to derive the corresponding dual problems, TWSVM assumes that the matrices $H^TH$ and $G^TG$ are nonsingular, where $G = [B \quad e_2]$ and $H = [A \ e_1]$. Under this extra condition, the dual problems are

$$\begin{array}{ll} \max\limits_{\alpha} & e_2^\top\alpha - \frac{1}{2}\alpha^\top G(H^\top H)^{-1}G^\top\alpha \\ \text{s.t.} & 0 \leq \alpha \leq c_1 \end{array} \tag{8}$$

and

$$\begin{array}{ll} \max\limits_{\gamma} & e_1^\top\gamma - \frac{1}{2}\gamma^\top H(G^\top G)^{-1}H^\top\gamma \\ \text{s.t.} & 0 \leq \gamma \leq c_2 \end{array} \tag{9}$$

respectively.

In order to deal with the case when $H^\top H$ or $G^\top G$ is singular and avoid the possible ill conditioning, the inverse matrices $(H^TH)^{-1}$ and $(G^TG)^{-1}$ are approximately replaced by $(H^TH + \epsilon I)^{-1}$ and $(G^TG + \epsilon I)^{-1}$ respectively, where $I$ is an identity matrix of appropriate dimensions, $\epsilon$ is a positive scalar, small to keep the structure of data. Thus the above dual problems are modified artificially as

$$\begin{array}{ll} \max\limits_{\alpha} & e_2^\top\alpha - \frac{1}{2}\alpha^\top G(H^\top H + \epsilon I)^{-1}G^\top\alpha \\ \text{s.t.} & 0 \leq \alpha \leq c_1 \end{array} \tag{10}$$

and

$$\begin{array}{ll} \max\limits_{\gamma} & e_1^\top\gamma - \frac{1}{2}\gamma^\top H(G^\top G + \epsilon I)^{-1}H^\top\gamma \\ \text{s.t.} & 0 \leq \gamma \leq c_2. \end{array} \tag{11}$$

The nonparallel proximal hyperplanes are obtained from the solution $\alpha$ and $\gamma$ of (10) and (11) by

$$v_1 = -(H^\top H + \epsilon I)^{-1}G^\top\alpha \text{ and } v_2 = (G^\top G + \epsilon I)^{-1}H^\top\gamma \tag{12}$$

where $v_1 = [w_1,\ b_1]$, $v_2 = [w_2,\ b_2]$. It should be pointed out that, strictly speaking, $[w_1,\ b_1]$ and $[w_2,\ b_2]$ obtained by (12) are no longer solutions to the problem (6)–(7) due to the difference between (6) and (10) as well as (7) and (11), they are only approximate solutions. The case of nonlinear kernels is handled on lines similar to linear kernels [8], [10].

## III. TBSVM

### A. Linear TBSVM

Following the basic idea of SVC and TWSVM, we propose our method: TBSVM. It finds the two nonparallel proximal hyperplanes

$$f_1(x) = w_1^\top x + b_1 = 0 \text{ and } f_2(x) = w_2^\top x + b_2 = 0 \quad (13)$$

by considering the following primal problems:

$$\begin{array}{ll} \min\limits_{w_1,b_1,\xi,\xi^*} & \frac{1}{2}\ c_3(||w_1||^2 + b_1^2) + \frac{1}{2}\xi^{*\top}\xi^* + c_1 e_2^\top \xi \\ \text{s.t.} & Aw_1 + e_1 b_1 = \xi^* \\ & -(Bw_1 + e_2 b_1) + \xi \geq e_2, \quad \xi \geq 0 \end{array} \quad (14)$$

and

$$\begin{array}{ll} \min\limits_{w_2,b_2,\eta,\eta^*} & \frac{1}{2}c_4(||w_2||^2 + b_2^2) + \frac{1}{2}\eta^{*\top}\eta^* + c_2 e_1^\top \eta \\ \text{s.t.} & Bw_2 + e_2 b_2 = \eta^* \\ & (Aw_2 + e_1 b_2) + \eta \geq e_1, \quad \eta \geq 0 \end{array} \quad (15)$$

where $c_1$, $c_2$, $c_3$, and $c_4$ are positive parameters.

Now we discuss the difference between the primal problems of TWSVM and our TBSVM, by comparing problems (14) and (6). Obviously, there is an extra regularization term $(1/2)c_3(||w_1||^2+b_1^2)$ in (14). We shall show that the structural risk is minimized in (14) due to this term. Remember the primal problem of SVC, where the structural risk minimization is implemented by maximizing the margin between two classes and this margin is measured by the Euclidian distance between two supporting hyperplanes. Correspondingly, the margin between two classes can be measured by some kind of distances between the proximal hyperplane $w_1^\top x + b_1 = 0$ and the bounding hyperplane $w_1^\top x + b_1 = -1$ here. Now we show that one of the reasonable distances can be expressed by

$$\frac{1}{\sqrt{\|w_1\|^2 + b_1^2}} \quad (16)$$

called the one side margin between two classes with respect to the hyperplane $w_1^\top x + b_1 = 0$. In fact, by introducing the transformation from $R^n$ to $R^{n+1}$: $\mathbf{x} = [x^\top, 1]^\top$, the proximal and bounding hyperplanes are expressed as $\mathbf{w_1}^\top \mathbf{x} = [w_1, b_1]^\top [x^\top, 1] = 0$ and $\mathbf{w_1}^\top \mathbf{x} = [w_1, b_1]^\top [x^\top, 1] = -1$ respectively. It is easy to see that the distance between the above two hyperplanes is the quantity (16). Therefore, minimizing the first term in (14) is equivalent to maximizing the one side margin with respect to the hyperplane $w_1^\top x + b_1 = 0$. Thus, the structural risk minimization principle is implemented.

In order to get the solutions to problems (14) and (15), we need to derive their dual problems. The Lagrangian of the problem (14) is given by

$$\begin{aligned} L(w_1, b_1, \xi, \alpha, \beta) = \frac{1}{2}c_3(\|w_1\|^2 + b_1^2) + \frac{1}{2}\|Aw_1 \\ + e_1 b_1\|^2 + c_1 e_2^\top \xi + \alpha^\top (Bw_1 + e_2 b_1 - \xi + e_2) - \beta^\top \xi \end{aligned} \quad (17)$$

where $\alpha = (\alpha_1, \ldots, \alpha_{m_2})$ and $\beta = (\beta_1, \ldots, \beta_{m_2})$ are the vectors of Lagrange multipliers. The Karush–Kuhn–Tucker (KKT) conditions [2] for $w_1, b_1, \xi_2$ and $\alpha, \beta$ are given by

$$\nabla_{w_1} L = c_3 w_1 + A^\top (Aw_1 + e_1 b_1) + B^\top \alpha = 0 \quad (18)$$

$$\nabla_{b_1} L = c_3 b_1 + e_1^\top (Aw_1 + e_1 b_1) + e_2^\top \alpha = 0 \quad (19)$$

$$\nabla_{\xi} L = c_1 e_2^\top - \beta^\top - \alpha^\top = 0 \quad (20)$$

$$-(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \quad \xi \geq 0 \quad (21)$$

$$\alpha^\top (Bw_1 + e_2 b_1 - \xi + e_2) = 0,\ \beta^\top \xi = 0 \quad (22)$$

$$\alpha \geq 0,\ \beta \geq 0. \quad (23)$$

Since $\beta \geq 0$, from (20) we have

$$0 \leq \alpha \leq c_1. \quad (24)$$

Obviously, (18) and (19) imply that

$$([A^\top\ e_1^\top][A\ \ e_1] + c_3 I)[w_1\ b_1]^\top + [B^\top\ e_2^\top]\alpha = 0 \quad (25)$$

where $I$ is an identity matrix of appropriate dimensions. Defining $v_1 = [w_1,\ b_1]^\top$, (25) can be rewritten as

$$(H^\top H + c_3 I)v_1 + G^\top \alpha = 0 \text{ or } v_1 = -(H^\top H + c_3 I)^{-1} G^\top \alpha. \quad (26)$$

Then putting (26) into the Lagrangian and using (18)–(23), we obtain the dual problem of the problem

$$\begin{array}{ll} \max\limits_{\alpha} & e_2^\top \alpha - \frac{1}{2}\alpha^\top G(H^\top H + c_3 I)^{-1} G^\top \alpha \\ \text{s. t.} & 0 \leq \alpha \leq c_1. \end{array} \quad (27)$$

At first glance, it can be found that the formulation of the problem (27) is the same as that of problem (10) when the parameter $c_3$ in (27) is replaced by $\epsilon$. However, there exists an essential difference in their significance. The parameter $\epsilon$ in (27) is just a fixed small scalar, while $c_3$ is a weighting factor which determines the tradeoff between the regularization term and the empirical risk. Therefore, selecting a proper $c_3$, either small or large, reflects the structure risk minimization principle. The experimental results in Section V will confirm that adjusting the value of $c_3$ can improve the classification accuracy.

In the same way, the dual of the problem (15) is obtained

$$\begin{array}{ll} \max\limits_{\gamma} & e_1^\top \gamma\ - \frac{1}{2}\gamma^\top H(G^\top G + c_4 I)^{-1} H^\top \gamma \\ \text{s.t.} & 0 \leq\ \gamma\ \leq c_2 \end{array} \quad (28)$$

where $\gamma$ is the Lagrange multiplier. The augmented vector $v_2 = [w_2,\ b_2]^\top$ is given by

$$v_2 = (G^\top G + c_4 I)^{-1} H^\top \gamma. \quad (29)$$

Once the solutions $(w_1, b_1)$ and $(w_2, b_2)$ of the problems (14) and (15) are obtained from the solutions of (27) and (28), a new point $x \in R^n$ is assigned to class $i$ ($i = +1, -1$),

depending on which of the two hyperplanes in (13) it is closer to

$$\text{Class } i = \arg\min_{k=1,2} \quad \frac{|w_k^\top x + b_k|}{\|w_k\|} \tag{30}$$

where $|\cdot|$ is the absolute value.

### B. Nonlinear Kernel Classifier

In order to extend our results to nonlinear classifiers, consider the following kernel-generated surfaces instead of hyperplanes:

$$K(x^\top, C^\top)u_1 + b_1 = 0 \text{ and } K(x^\top, C^\top)u_2 + b_2 = 0 \tag{31}$$

where $C^\top = [A \;\; B]^\top$ and $K$ is an appropriately chosen kernel. For the surface $K(x^\top, C^\top)u_1 + b_1 = 0$, we construct the primal problem

$$\begin{array}{ll} \min\limits_{u_1,b_1,\xi,\xi^*} & \frac{1}{2}c_3(\|u_1\|^2 + b_1^2) + \frac{1}{2}\xi^{*\top}\xi^* + c_1 e_2^\top \xi \\ \text{s.t.} & K(A, C^\top)u_1 + e_1 b_1 = \xi^* \\ & -(K(B, C^\top)u_1 + e_2 b_1) + \xi \geq e_2, \xi \geq 0 \end{array} \tag{32}$$

where $c_1 > 0$ and $c_3 > 0$ are parameters. Its Lagrangian function is

$$\begin{aligned} L(u_1, b_1, \xi, \alpha, \beta) = \frac{1}{2}\, c_3(\|u_1\|^2 + b_1^2) + \frac{1}{2}\|K(A, C^\top)u_1 \\ + e_1 b_1\|^2 + c_1 e_2^\top \xi + \alpha^\top (K(B, C^\top)u_1 \\ + e_2 b_1 - \xi + e_2) - \beta^\top \xi \end{aligned} \tag{33}$$

where $\alpha = (\alpha_1, \ldots, \alpha_{m_2})$ and $\beta = (\beta_1, \ldots, \beta_{m_2})$ are the vectors of Lagrange multipliers. The KKT conditions for $u_1,\ b_1, \xi$ and $\alpha, \beta$ are given by

$$\begin{aligned} &\nabla_{u_1} L = c_3 u_1 + K(A, C^\top)^\top (K(A, C^\top)u_1 + e_1 b_1) \\ &\qquad + K(B, C^\top)^\top \alpha = 0 && (34) \\ &\nabla_{b_1} L = c_3 b_1 + e_1^\top (K(A, C^\top)u_1 + e_1 b_1) + e_2^\top \alpha = 0 && (35) \\ &\nabla_{\xi} L = c_1 e_2^\top - \beta^\top - \alpha^\top = 0 && (36) \\ &\qquad -(K(B, C^\top)u_1 + e_2 b_1) + \xi \geq e_2, \xi \geq 0 && (37) \\ &\qquad \alpha^\top (K(B, C^\top)u_1 + e_2 b_1 - \xi + e_2) = 0 && (38) \\ &\beta^\top \xi = 0, \alpha \geq 0,\ \beta \geq 0. && (39) \end{aligned}$$

Since $\beta \geq 0$, from (36) we have

$$0 \leq \alpha \leq c_1. \tag{40}$$

Obviously, (34) and (35) imply that

$$\begin{aligned} &([K(A, C^\top)^\top \; e_1^\top][K(A, C^\top) \; e_1] + c_3 I)[u_1 \; b_1]^\top \\ &\quad + [K(B, C^\top)^\top \; e_2^\top]\alpha = 0. \end{aligned} \tag{41}$$

Defining

$$R = [K(B \; C^\top)e_2], \;\; S = [K(A \; C^\top)e_1] \tag{42}$$

and the augmented vector $z_1 = [u_1, \;\; b_1]^\top$, (41) can be rewritten as

$$(S^\top S + c_3 I)z_1 + R^\top \alpha = 0 \text{ or } z_1 = -(S^\top S + c_3 I)^{-1} R^\top \alpha. \tag{43}$$

Then we obtain the dual problem

$$\begin{array}{ll} \max\limits_{\alpha} & e_2^\top \alpha - \frac{1}{2}\alpha^\top R(SS^\top + c_3 I)^{-1} R^\top \alpha \\ \text{s.t.} & 0 \leq \alpha \leq c_1. \end{array} \tag{44}$$

In the same way, for the surface $K(x^\top, C^\top)u_2 + b_2 = 0$, we construct the primal problem

$$\begin{array}{ll} \min\limits_{u_2,b_2,\eta,\eta^*} & \frac{1}{2}c_4(\|u_2\|^2 + b_2^2) + \frac{1}{2}\eta^{*\top}\eta^* + c_2 e_1^\top \eta \\ \text{s.t.} & K(B, C^\top)u_2 + e_2 b_2 = \eta^* \\ & (K(A, C^\top)u_2 + e_1 b_2) + \eta \geq e_1, \quad \eta \geq 0 \end{array} \tag{45}$$

where $c_2 > 0$ and $c_4 > 0$ are parameters, and obtain the dual problem

$$\begin{array}{ll} \max\limits_{\gamma} & e_1^\top \gamma - \frac{1}{2}\gamma^\top S(RR^\top + c_4 I)^{-1} S^\top \gamma \\ \text{s.t.} & 0 \leq \gamma \leq c_2 \end{array} \tag{46}$$

with the augmented vector $z_2 = [u_2, \;\; b_2]^\top$ given by

$$z_2 = (RR^\top + c_4 I)^{-1} S^\top \gamma. \tag{47}$$

Once the solutions $(u_1, b_1)$ and $(u_2, b_2)$ of the problems (32) and (45) are obtained from the solutions of (44) and (46), a new point $x \in R^n$ is assigned to class $i$ $(i = +1, -1)$ by

$$\text{Class } i = \arg\min_{k=1,2} \quad \frac{|K(x^\top, C^\top)u_k + b_k|}{\sqrt{u_k^\top K(C, C^\top)u_k}} \tag{48}$$

where $|\cdot|$ is the absolute value.

## IV. Fast TBSVM Solver—Successive Overrelaxation Technique

In our TBSVM, there are four strictly convex quadratic problems to be solved: (27), (28), (44), and (46). It is easy to see that these problems can be rewritten in the following unified form:

$$\begin{array}{ll} \min\limits_{\alpha} & \frac{1}{2}\alpha^\top Q\alpha - e^\top \alpha \\ \text{s.t.} & 0 \leq \alpha \leq c \end{array} \tag{49}$$

where $Q \in R^{m \times m}$ is positive definite. For example, the above problem becomes the problem (27), when $Q = G(H^\top H + c_3 I)^{-1} G^\top$, $c = c_1$.

The above problem (49) can be solved efficiently by the following SOR technique, see [15], [16].

**Algorithm 4.1**: Choose $t \in (0, 2)$. Start with any $\alpha^0 \in R^n$. Having $\alpha^i$, compute $\alpha^{i+1}$ as follows:

$$\alpha^{i+1} = (\alpha^i - tE^{-1}(Q\alpha^i - e + L(\alpha^{i+1} - \alpha^i))) \tag{50}$$

until $||\alpha^{i+1} - \alpha^i||$ is less than some prescribed tolerance, where the nonzero elements of $L \in R^{m \times m}$ constitute the strictly lower triangular part of the symmetric matrix $Q$, and the nonzero elements of $E \in R^{m \times m}$ constitute the diagonal of $Q$.

SOR is an excellent TBSVM solver, because it can process efficiently very large datasets that need not reside in memory. Furthermore, it has been proved that this algorithm converges linearly to a solution [15]. It should be pointed out that we also improve the original TWSVM by applying the SOR technique to solve the problems (10) and (11). The experimental results in the following section will show that the SOR technique has a remarkable acceleration effect on both our TBSVM and TWSVM.

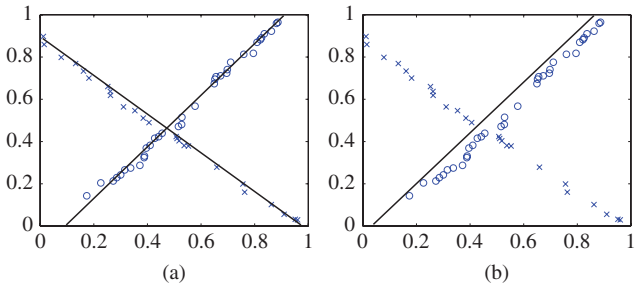


Fig. 1. Classification results of (a) TBSVM and (b) SVC for "Cross planes" dataset.

## V. Experimental Results

In this section, some experiments are made to demonstrate the performance of our TBSVM. All methods are implemented by using MATLAB 7.0 [17] on a PC with an Intel P4 processor (2.9 GHz) with 1 GB RAM. Our TBSVM and SVC are solved by the SOR technique and the optimization toolbox QP in MATLAB, respectively. In order to show the effectiveness of SOR, TWSVM is solved by both QP and SOR. The "Accuracy" used to evaluate methods is defined as follows. $Accuracy = (TP+TN)/(TP+FP+TN+FN)$, where TP, TN, FP, and FN are the number of true positive, true negative, false positive, and false negative, respectively. Classification accuracy of each method is measured by the standard tenfold cross-validation methodology [18].

First, consider a simple 2-D "Cross Planes" dataset, which was tested in [7] to indicate that their TWSVM can handle the cross planes dataset much better compared with SVC. Now we show that our TBSVM also has this advantage. The "Cross Planes" dataset is generated by perturbing points lying on two intersecting lines. Fig. 1 shows the dataset and the linear classifiers obtained by our TBSVM and SVC. It is easy to see that the result of our TBSVM is more reasonable than that of SVC. In addition, we get very different accuracies for these two methods: 100% (TBSVM), 76% (SVM).

Second, in order to compare our TBSVM with TWSVM and SVC, we choose the same datasets as the ones in [8], which are from the UCI machine learning repository [19]. In Table I, the classification accuracy, computation time, and optimal values of $c_3$ and $c_4$ in our TBSVM are listed. For SVC, the optimal parameter $C$ is obtained through searching in the range $2^{-8}$ to $2^8$. The optimal values of $c_i (i = 1, 2)$ in TWSVM and $c_i (i = 1, 2, 3, 4)$ in our TBSVM are obtained in the same range by using a tuning set comprising of 10% of the dataset. Once the parameters are selected, the tuning set is returned to learn the final classifier.

In order to compare the behavior of our linear TBSVM with the linear TWSVM and linear SVC, the results of numerical experiments are summarized in Table I, where $\epsilon$ is selected according to [8], [11]. In Table I, the best accuracy is shown by bold figures. It is easy to see that the accuracy of our linear TBSVM is significantly better than that of the linear TWSVM on all datasets. For example, for BUPA liver the accuracy of our TBSVM is 70.12% with $c_3 = 0.0078$ and $c_4 = 8$, respectively, while the accuracy of TWSVM is 66.40% with $\epsilon$ is $10^{-6}$. In other datasets also we obtain similar results.

TABLE I
Tenfold Testing Percentage Accuracy of Linear Classifiers

| Datasets | TBSVM Accuracy % Time (s) $c_3/c_4$ | TWSVM Accuracy % Time (s) $\epsilon = 10^{-6}$ | SVC Accuracy % Time (s) |
|---|---|---|---|
| Hepatitis | 83.23 ± 5.94 | 82.89 ± 6.30 | **84.13 ± 5.58** |
| (155 × 19) | 0.011 | 0.012/0.281 | 1.170 |
| | 0.0039/0.0039 | | |
| BUPA liver | **70.12 ± 7.94** | 66.40 ± 7.74 | 67.78 ± 5.51 |
| (345 × 6) | 0.010 | 0.011/0.840 | 3.540 |
| | 0.0078/8 | | |
| Heart-Statlog | **85.27 ± 4.95** | 84.44 ± 6.80 | 83.12 ± 5.41 |
| (270 × 14) | 0.025 | 0.023/0.454 | 1.584 |
| | 0.0078/64 | | |
| Heart-c | **85.02 ± 8.04** | 84.86 ± 6.27 | 83.33 ± 5.64 |
| (303 × 14) | 0.034 | 0.042/0.516 | 2.193 |
| | 32/256 | | |
| Votes | **96.33 ± 4.62** | 95.85 ± 2.75 | 95.80 ± 2.65 |
| (435 × 16) | 0.062 | 0.797/1.851 | 3.192 |
| | 64/4 | | |
| WPBC | **84.14 ± 3.33** | 83.68 ± 5.73 | 83.30 ± 4.53 |
| (198 × 34) | 0.012 | 0.012/0.560 | 2.094 |
| | 0.0156/0.0039 | | |
| Sonar | 78.94 ± 5.54 | 77.00 ± 6.10 | **80.13 ± 5.43** |
| (208 × 60) | 0.004 | 0.007/0.375 | 0.941 |
| | 64/256 | | |
| Ionosphere | **89.68 ± 6.74** | 88.48 ± 5.74 | 88.20 ± 4.51 |
| (351 × 34) | 0.049 | 0.047/0.969 | 4.120 |
| | 0.0156/4 | | |
| Australian | 86.70 ± 5.47 | 85.94 ± 5.84 | **88.51 ± 4.85** |
| (690 × 14) | 0.235 | 0.346/6.907 | 21.93 |
| | 256/128 | | |
| Pima-Indian | 76.86 ± 3.28 | 73.80 ± 4.97 | **77.34 ± 4.37** |
| (768 × 8) | 0.126 | 0.121/8.281 | 29.94 |
| | 4/64 | | |
| CMC | **72.98 ± 3.92** | 68.28 ± 2.21 | 67.82 ± 2.63 |
| (1473 × 9) | 1.001 | 1.247/8.625 | 41.17 |
| | 0.0039/1 | | |

It is observed that the choice of $c_3$ and $c_4$ affects the results significantly, and the values $c_3$ and $c_4$ chosen vary, instead of always taking a fixed small value. This shows that adjusting the parameters $c_3$ and $c_4$ is useful in practice.

Table I also depicts the training CPU time for these three methods. Note that there are two CPU times for TWSVM: the first one is the CPU time for the SOR technique, the second one is the CPU time for TWSVM with the QP method. It is easy to see that TWSVM (using the SOR technique) is much far faster than the original TWSVM, indicating that the SOR technique can improve the calculation speed. It can also be seen that our TBSVM is the fastest on most of datasets, and TWSVM (using the SOR technique) has a similar behavior.

TABLE II
TENFOLD TESTING PERCENTAGE TEST SET ACCURACY OF NONLINEAR (RBF) CLASSIFIERS

| Datasets | TBSVM Accuracy % Time (s) $c_3/c_4$ | TWSVM Accuracy % Time (s) $\epsilon = 10^{-6}$ | SVC Accuracy % Time (s) |
|---|---|---|---|
| Hepatitis | **84.52 ± 5.23** | 83.39 ± 7.31 | 84.13 ± 6.25 |
| (155 × 19) | 0.013 | 0.016/0.797 | 1.300 |
| | 0.0078/0.0039 | | |
| BUPA liver | **73.04 ± 5.55** | 67.83 ± 6.49 | 68.32 ± 7.20 |
| (345 × 6) | 0.028 | 0.033/2.700 | 5.248 |
| | 2/1 | | |
| Heart-Statlog | **86.30 ± 5.24** | 82.96 ± 4.67 | 83.33 ± 9.11 |
| (270 × 14) | 0.028 | 0.029/1.130 | 6.100 |
| | 0.25/0.0039 | | |
| Heart-c | **85.27 ± 4.95** | 83.83 ± 5.78 | 83.68 ± 5.67 |
| (303 × 14) | 0.035 | 0.052/2.141 | 3.800 |
| | 0.5/16 | | |
| Votes | **96.40 ± 4.57** | 94.91 ± 4.37 | 95.64 ± 7.23 |
| (435 × 16) | 0.068 | 0.072/3.540 | 7.783 |
| | 0.25/0.5 | | |
| WPBC | **82.47 ± 3.18** | 81.28 ± 5.92 | 80.18 ± 6.90 |
| (198 × 34) | 0.035 | 0.029/1.305 | 3.141 |
| | 0.125/0.0156 | | |
| Sonar | **90.0 ± 7.5** | 89.64 ± 6.11 | 88.93 ± 10.43 |
| (208 × 60) | 0.008 | 0.014/2.630 | 5.302 |
| | 0.125/0.0156 | | |
| Ionosphere | 87.75 ± 3.34 | 87.46 ± 3.40 | **90.2 ± 4.51** |
| (351 × 34) | 0.088 | 0.064/5.576 | 15.71 |
| | 0.0039/0.0039 | | |
| Australian | 76.32 ± 4.51 | 75.8 ± 4.91 | **85.51 ± 4.85** |
| (690 × 14) | 0.297 | 0.420/33.43 | 48.11 |
| | 128/1 | | |
| Pima-Indian | **77.86 ± 6.84** | 73.74 ± 5.2 | 76.09 ± 3.58 |
| (768 × 8) | 0.424 | 0.427/41.92 | 112.04 |
| | 16/0.0039 | | |
| CMC | **75.15 ± 2.51** | 73.95 ± 3.48 | 68.98 ± 3.44 |
| (1473 × 9) | 1.557 | 1.708/58.36 | 127.80 |
| | 1/0.0078 | | |

Table II is concerned with our kernel TBSVM, TWSVM, and SVC. The Gaussian kernel $K(x, x') = e^{-\mu\|x-x'\|^2}$ is used. The kernel parameter $\mu$ is also obtained by searching from the range $2^{-8}$ to $2^8$. The training CPU times for these three methods are also listed. The results in Table II are similar to those in Table I, thereby confirming the above conclusion further.

## VI. CONCLUSION

For binary classification, an improved version of TBSVM based on the TWSVM in [8] was proposed in this brief. The main contribution is that the structural risk minimization principle is implemented by adding the regularization term in the primal problems of our TBSVM. This embodies the marrow of statistical learning theory. The parameters $c_3$ and $c_4$ introduced are the weights between the regularization term and the empirical risk, so they can be chosen flexibly, improving the original TWSVM where $\epsilon$ is a fixed scalar. In addition, the application of the SOR technique is also an excellent contribution, since it overcomes the drawback of TWSVM, namely, the training time. Computational comparisons between our TWSVM and other methods including SVC and TWSVM have been made on several datasets, indicating that our TBSVM is not only faster but also shows better generalization. We believe that its nice classification accuracy mainly comes from the fact that the parameters $c_3$ and $c_4$ are adjusted properly. Our TBSVM MATLAB codes can be downloaded from http://math.cau.edu.cn/dengnaiyang.html.

It should be pointed out that there are four parameters in our TBSVM, so the parameter selection is a practical problem and should be addressed in the future further. Besides, other acceleration algorithms such as [20], [21], and the extension to semisupervised learning such as [22] are also interesting.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for valuable suggestions.

## REFERENCES

[1] C. Cortes and V. N. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[2] O. L. Mangasarian, *Nonlinear Programming*. Philadelphia, PA: SIAM, 1994.
[3] S. Abe, *Support Vector Machines for Pattern Classification*. Berlin, Germany: Springer-Verlag, 2005.
[4] W. S. Noble, "Support vector machine applications in computational biology," in *Kernel Methods in Computational Biology*, B. Schölkopf, K. Tsuda, and J.-P. Vert, Eds. Cambridge, MA: MIT Press, 2004.
[5] T. N. Lal, M. Schröder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schölkopf, "Support vector channel selection in BCI," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1003–1010, Jun. 2004.
[6] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw.*, vol. 6. Como, Italy, Jul. 2000, pp. 348–353.
[7] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.
[8] R. K. Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.
[9] M. A. Kumar and M. Gopal, "Application of smoothing technique on twin support vector machines," *Pattern Recognit. Lett.*, vol. 29, no. 13, pp. 1842–1848, Oct. 2008.
[10] R. Khemchandani, R. K. Jayadeva, and S. Chandra, "Optimal kernel selection in twin support vector machines," *Optim. Lett.*, vol. 3, no. 1, pp. 77–88, 2009.
[11] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7535–7543, May 2009.
[12] S. Ghorai, A. Mukherjee, and P. K. Dutta, "Nonparallel plane proximal classifier," *Signal Process.*, vol. 89, no. 4, pp. 510–522, Apr. 2009.
[13] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.
[14] C. H. Zhang, Y. J. Tian, and N. Y. Deng, "The new interpretation of support vector machines on statistical learning theory," *Sci. China*, vol. 53, no. 1, pp. 151–164, Jan. 2010.
[15] Z.-Q. Luo and P. Tseng, "Error bounds and convergence analysis of feasible descent methods: A general approach," *Ann. Oper. Res.*, vols. 46–47, no. 1, pp. 157–178, 1993.

[16] O. L. Mangasarian and D. R. Musicant, "Successive overrelaxation for support vector machines," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1032–1037, Sep. 1999.
[17] MathWorks. (2007) [Online]. Available: http:// www.mathworks.com
[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.
[19] C. L. Blake and C. J. Merz. (1998). *UCI Repository for Machine Learning Databases*. Dept. Inf. Comput. Sci., Univ. California, Irvine [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[20] P.-H. Chen, R.-E. Fan, and C.-J. Lin, "A study on SMO-type decomposition methods for support vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 893–908, Jul. 2006.
[21] T. Martinetz, K. Labusch, and D. Schneegass, "SoftDoubleMaxMinOver: Perceptron-like training of support vector machines," *IEEE Trans. Neural Netw.*, vol. 20, no. 7, pp. 1061–1072, Jul. 2009.
[22] M. M. Adankon, M. Cheriet, and A. Biem, "Semisupervised least squares support vector machine," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1858–1870, Dec. 2009.

## Hyperellipsoidal Statistical Classifications in a Reproducing Kernel Hilbert Space

Xun Liang, *Senior Member*, *IEEE*, and Zhihao Ni

***Abstract*— Standard support vector machines (SVMs) have kernels based on the Euclidean distance. This brief extends standard SVMs to SVMs with kernels based on the Mahalanobis distance. The extended SVMs become a special case of the Euclidean distance when the covariance matrix in a reproducing kernel Hilbert space is degenerated to an identity. The Mahalanobis distance leads to hyperellipsoidal kernels and the Euclidean distance results in hyperspherical ones. In this brief, the Mahalanobis distance-based kernel in a reproducing kernel Hilbert space is developed systematically. Extensive experiments demonstrate that the hyperellipsoidal kernels slightly outperform the hyperspherical ones, with fewer SVs.**

***Index Terms*— Classification accuracy, hyperellipsoid, Mahalanobis distance, reproducing kernel Hilbert space, support vector machines.**

### I. Introduction

Support vector machines (SVMs) have been successfully applied in many science and engineering fields. By manipulating the tricky kernels that insinuate an implicit reproducing kernel Hilbert space $\mathcal{H}$, SVMs gain conspicuous advantages. These advantages include the absence of local minima and the optimal separation or the largest margin width between two clusters, simply by solving linearly constrained quadratic programming problems [1]–[8].

A standard SVM is based on Euclidean distance, and data are handled with a hyperspherical assumption. However, data are more likely to be distributed within a hyperellipsoidal region. In particular, a hypersphere is a special case of a hyperellipsoid. Hyperspherical SVMs are based on the Euclidean distance, while the hyperellipsoidal kernels SVMs are based on the Mahalanobis distance [9], [10]. The intuitive explanation is that the Mahalanobis framework not only incorporates the size of hyperellipsoids but also reflects the directions of the major and minor axis of the hyperellipsoids.

In recent years, a significant number of efforts have been made toward classifying data based on the Mahalanobis distance in a reproducing kernel Hilbert space [6], [7], [11]–[22]. In many data classification problems, researchers have found that the Mahalanobis distance serves as a better measure of the distance than the Euclidean distance [9], [10], [17], [19]. Two representatives were given by [15], [17], and [19]. Fig. 1 portrays a typical position of the separating hyperplane as in [15] and [19]. A standard Euclidean distance-based SVM has a separating hyperplane located in the middle between the two clusters. In [15] and [19], it was argued that the separating hyperplane should be closer to the right hyperellipsoid, as shown in Fig. 1. Furthermore, they claimed that their Mahalanobis distance-based hyperellipsoidal SVMs outperform standard SVMs. By employing the Mahalanobis distance, [17] also made an improvement. However, by separating the hyperplane, [15], [17], and [19] reached opposite conclusions. Whereas [15] and [19] believed that it should be closer to the right cluster, [17] favored a different opinion. We infer that the better experimental performances in [17] may be due largely to their use of the weighted factor $N_1^2/(N_1^2+N_2^2)$ for separating the hyperplane, where $N_1$ and $N_2$ are the numbers of data in the two clusters. This factor directly controls the separating hyperplane in $\mathcal{H}$, as opposed to the Mahalanobis distance. Alternative useful approaches on kernel principal component analysis framework for kernelizing a hyperellipsoidal SVM were discussed in [6]–[7]. In [21], an efficient algorithm for learning a Mahalanobis distance metric with the principle of margin maximization was presented. In this brief, we propose a simple algorithm for learning a Mahalanobis distance metric.

To extend the shape into hyperellipsoid, some scholars experimented with hyperellipsoid-based classifications in input spaces. For example, [14] first decomposed the covariance matrix with $\sum^{-1} = \boldsymbol{P}^T\boldsymbol{D}\boldsymbol{P}$, with $\boldsymbol{P}$ orthogonal and $\boldsymbol{D}$ diagonal, and then used $x_i\prime = \boldsymbol{D}^{1/2}\boldsymbol{P}x_i$, $i = 1, \ldots, l$, as new input data, where $x_i$'s are the original input data. The proposed input space hyperellipsoid-based method demonstrated a more improved performance than the nearest neighbor classifier [14, Tab. I and II]. Unfortunately, such decomposition is not unique. The Cholesky decomposition $\sum^{-1} = \boldsymbol{B}\boldsymbol{B}^T$ may also be tried. Subsequently, $\boldsymbol{B}$ is used to multiply the original input data. More importantly, the decompositions are only in the input spaces, not in the reproducing kernel Hilbert space $\mathcal{H}$, which is what we actually desire. Another example of hyperellipsoidal operations in input spaces can be found in [17]. Since the distances from (18) to (16) in [17] are for the input space, and kernel (19) in [17] does not actually carry these relations into $\mathcal{H}$, [17] only advanced one step ahead of

Manuscript received September 9, 2010; revised December 24, 2010; accepted March 14, 2011. Date of publication May 5, 2011; date of current version June 2, 2011. This work was supported by the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China (10XNI029, Research on Financial Web Data Mining and Knowledge Management), the Natural Science Foundation of China under Grant 70871001, and the 863 Project of China under Grant 2007AA01Z437.

X. Liang and Z. Ni are with the School of Information, Remin University of China, Beijing 100872, China (e-mail: xliang@ruc.edu.cn; alexni@ruc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2130539