Multiconlitron: A General Piecewise Linear Classifier

Li Yujian, Liu Bo, Yang Xinwu, Fu Yaozong, and Li Houjun

***Abstract*— Based on the "convexly separable" concept, we present a solid geometric theory and a new general framework to design piecewise linear classifiers for two arbitrarily complicated nonintersecting classes by using a "multiconlitron," which is a union of multiple conlitrons that comprise a set of hyperplanes or linear functions surrounding a convex region for separating two convexly separable datasets. We propose a new iterative algorithm called the cross distance minimization algorithm (CDMA) to compute hard margin non-kernel support vector machines (SVMs) via the nearest point pair between two convex polytopes. Using CDMA, we derive two new algorithms, i.e., the support conlitron algorithm (SCA) and the support multiconlitron algorithm (SMA) to construct support conlitrons and support multiconlitrons, respectively, which are unique and can separate two classes by a maximum margin as in an SVM. Comparative experiments show that SMA can outperform linear SVM on many of the selected databases and provide similar results to radial basis function SVM on some of them, while SCA performs better than linear SVM on three out of four applicable databases. Other experiments show that SMA and SCA may be further improved to draw more potential in the new research direction of piecewise linear learning.**

***Index Terms*— Conlitron, cross distance minimization algorithm, multiconlitron, piecewise linear classifier, piecewise linear learning, support conlitron algorithm, support multiconlitron algorithm, support vector machine.**

I. Introduction

ONE of the main problems in pattern classification is how to design decision functions that can classify a set of observations correctly with the highest possible level of generalization. An important method for pattern classification is the support vector machine (SVM) [1], [2], by which a linear decision surface separates two classes of data by a maximum margin criterion. This has a very good level of generalization. Since Vapnik [3], [4] proposed the concept of a SVM, it has attracted a large amount of interest with many successful applications to classification problems [5]–[10]. There have been a number of new important achievements and developments in the course of investigating the SVM from the viewpoints of fast algorithm design [11]–[16], new model exploration [17]–[24], kernel function selection [25]–[29], geometric approach analysis [30]–[35], etc. In this paper, we propose a new method that does not need to use kernels and that can solve separable problems easily without scaling and setting any hyperparameter. We try to make it as good as any nonlinear SVM, but it should be noted that we mainly consider two-class commonly separable problems (i.e., there are no common points in the two different classes).

The primal SVM model is formalized as a quadratic optimization problem. If $X, Y \subset \mathbf{R}^n$ ($\mathbf{R}$ stands for the set of all real numbers) denote two classes composed of finite data points, the hard margin SVM (HM-SVM) without kernels can be described as

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t. } \mathbf{w}\cdot\mathbf{x}_i + b \geq 1, \mathbf{x}_i \in X, 1 \leq i \leq |X|; \\ &\mathbf{w}\cdot\mathbf{y}_j + b \leq -1, \mathbf{y}_j \in Y, 1 \leq j \leq |Y|. \end{aligned} \quad (\text{HM}-\text{SVM})$$

Here,"$\|\cdot\|$," "$\cdot$," and "$|\cdot|$," respectively, stand for the L_2 norm, the inner product, and the set cardinality.

In the case where X and Y are linearly separable, there exists a unique solution $(\mathbf{w}^*, b^*)$ to (HM-SVM), whose solution is just the hyperplane $\mathbf{w}^*\cdot\mathbf{x} + b^* = 0$ with the margin $M = 2/\|\mathbf{w}^*\|$, i.e., the maximum distance between two parallel hyperplanes $\mathbf{w}^*\cdot\mathbf{x} + b^* = 1$ and $\mathbf{w}^*\cdot\mathbf{x} + b^* = -1$. However, there exists no solution to (HM-SVM) in the linearly nonseparable case, where the usual strategy to compute the SVM of X and Y is to use the soft margin SVM (SM-SVM) models such as the linear cost soft margin SVM (LCSM-SVM)

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{1\leq i\leq|X|}\xi_i + \sum_{1\leq j\leq|Y|}\zeta_j\right) \\ &\text{s.t. } \mathbf{w}\cdot\phi(\mathbf{x}_i) + b \geq 1 - \xi_i, \mathbf{x}_i \in X, \xi_i \geq 0, 1 \leq i \leq |X|; \\ &\mathbf{w}\cdot\phi(\mathbf{y}_j) + b \leq -1 + \zeta_j, \mathbf{y}_j \in Y, \zeta_j \geq 0, 1 \leq j \leq |Y|. \end{aligned} \quad (\text{LCSM}-\text{SVM})$$

Note that C is a positive regularization constant chosen to control the tradeoff between margin maximization and classification violation, "$\mathbf{w}$" in LCSM-SVM may have a different dimension from "$\mathbf{w}$" in HM-SVM because of the introduced function ϕ, which usually maps input vectors to a very high dimensional feature space, and is implicitly

Manuscript received March 26, 2010; revised November 1, 2010; accepted November 16, 2010. Date of publication December 6, 2010; date of current version February 9, 2011. This work was supported in part by the National Science Foundation of China under Grant 60775010 and Grant 61005001, Beijing University of Technology High-Level Personnel Development Project, the Funding Project for Academic Human Resources Development in Institution of Higher Learning under the Jurisdiction of Beijing Municipality, and the Program of Science Development Beijing Municipal Education Commission under Program KM200810005003 and Program KM201010005012.

The authors are with the College of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China (e-mail: liyujian@bjut.edu.cn; liubo@emails.bjut.edu.cn; yang_xinwu@bjut.edu.cn; fuyaozong@emails.bjut.edu.cn; lihoujun@emails.bjut.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.2010.2094624

defined by a prescribed kernel function $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ such as linear kernel $K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$ and Gaussian radial basis function (RBF), kernel $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \ \|\mathbf{x} - \mathbf{y}\|^2)$. In solving complicated classification problems, it is necessary to choose a good kernel function for the SVMs, which cannot generate any nonlinear separating boundaries without kernels.

Obviously, HM-SVM is a special case ($C = \infty$) of LCSM-SVM, in which an appropriate nonlinear kernel should be selected to get better performance for classification problems especially with a nonlinear boundary that cannot be approximated by a hyperplane. At present, LCSM-SVM is usually solved by applying Platt's SMO and others to their equivalent dual models [11]–[16]. Of course, there are many other methods to compute SVMs, e.g., nearest point algorithm (NPA) [30], relaxed online maximal margin algorithm [31], S–K algorithm [32], and reduced convex hull algorithm [33]–[35]. It is worth noting that the basic idea of NPA, playing a very important role in this paper, is to reformulate HM-SVM as a problem to compute the nearest point pair between two convex polytopes.

Compared to other approaches, SVMs offer many advantages such as global uniqueness, good generalization, and a sound theoretical foundation. With the progress in SVM theory, a number of new models have also been presented to extend the standard SVM, including proximal SVM [17], multisurface proximal SVM [18], twin SVM [19], least square SVM [20], reduced SVM [21], Lagrangian SVM [22], relevance vector machine [23], probabilistic classification vector machine [24], additive SVM [36], etc. Though the great idea of "kernel trick" can be used to solve nonlinear problems with some difficulties in the selection of kernel functions [25]–[29], the requirement of computational resources for large datasets, and the interpretability/transparency of metric changes [36], [37], there remains an important question as to whether margin maximization in the original space is better than that in the extended functional space. Therefore, we consider the problem of how to construct SVM-like classifiers for any complicated datasets without using kernel functions. A good strategy is to develop the theory of piecewise linear classifiers (PLCs), whose related works include CBP-based PLCs [37], locally trainable PLCs [38]–[40], linear programming PLCs [41], [42], decision tree PLCs [43], [44], etc. Conceptually speaking, these PLCs cannot be regarded as a non-kernel extension of SVM because they do not separate two datasets by a maximum margin. To implement such an extension, we may use a set of reflective convex hulls [45], but here we consider a novel and more general geometric framework to construct PLCs from a "multiconlitron," which is a union of multiple conlitrons that comprise a set of hyperplanes or linear functions surrounding a convex region for separating two convexly separable datasets. This framework is fairly simple and takes special advantages over linear SVM in separating "cross planes," which is obtained by perturbing points originally lying on two intersecting planes (lines), and was used to demonstrate the effectiveness of multisurface proximal SVM [18]. For example, a conlitron/multiconlitron can completely separate the 2-D dataset of two "cross planes" illustrated in Fig. 1, but a linear SVM does poorly.

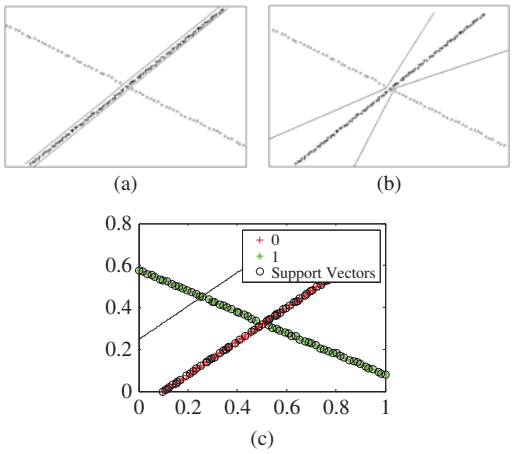


Fig. 1. Two-line conlitron (a) surrounds a convex region and the four-line multiconlitron contains two conlitrons; (b) both of them separate the 2-D cross-plane dataset completely (100% correct), but the linear SVM; and (c) trained with $C = 1$ does poorly (72.5% correct).

In this paper, we mainly establish a solid geometric theory and a general framework to design PLCs from the viewpoint of conlitrons and multiconlitrons, which extend the relationship between SVM and convex hulls. Along with description of basic concepts, definitions, and theorems, we first present a new iterative algorithm, i.e., cross distance minimization algorithm (CDMA) to compute an HM-SVM via the nearest point pair between two convex polytopes. Then, based on the concept of "convexly separable" as an extension of "linearly separable," we build a general linear classifier to separate two arbitrarily complicated nonintersecting classes from a "conlitron" and a "multiconlitron." Additionally, using CDMA we derive two other new algorithms—the support conlitron algorithm (SCA) and the support multiconlitron algorithm (SMA), for constructing unique support conlitrons and unique support multiconlitrons, respectively. These can be regarded as a non-kernel extension of SVM because they separate two arbitrarily complicated nonintersecting classes by certain maximum margin. Finally, we test CDMA, SCA, and SMA on a 3-D cross-plane dataset and a number of UCI benchmark databases, compare their performance with linear SVM and RBF SVM, and discuss their promise for future research.

II. Basic Concepts, Definitions, and Theorems

Throughout this paper, we use X and Y to denote two nonempty classes or datasets in $\mathbf{R}^n$. If $X \cap Y = \varnothing$, we call them "commonly separable." The basic pattern recognition problem is to design a computable function $f : \mathbf{R}^n \to \mathbf{R}$ such that $\forall \mathbf{x} \in X, f(\mathbf{x}) > 0 (< 0)$ and $\forall \mathbf{y} \in Y, f(\mathbf{y}) < 0 (> 0)$. Such a function is called a decision (or discriminant) function of X and Y, where $f(\mathbf{x}) = 0$ represents the decision surface or boundary. Strictly speaking, the boundary $f(\mathbf{x}) = 0$ should separate X and Y completely, but some errors are allowed to improve anti-noise and generalizing abilities in practical applications.

If there exist $\mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}$ such that $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ ($\mathbf{x} \in \mathbf{R}^n$) can separate X and Y without errors, we call them

"linearly separable" and call $f(\mathbf{x})$ their "linear discriminant function."

Definition 1: The distance function d is defined as follows:

1) $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\| = \sqrt{(\mathbf{x}-\mathbf{y})\cdot(\mathbf{x}-\mathbf{y})}$, $\forall \mathbf{x},\mathbf{y} \in \mathbf{R}^n$;
2) $d(X,Y) = \inf\{d(\mathbf{x},\mathbf{y}), \mathbf{x}\in X, \mathbf{y}\in Y\}$, $\forall X, Y \subset \mathbf{R}^n$;
3) $d(\mathbf{x},Y) = d(Y,\mathbf{x}) = \inf\{d(\mathbf{x},\mathbf{y}), \mathbf{y}\in Y\}$, $\forall \mathbf{x} \in \mathbf{R}^n$, $\forall Y \subset \mathbf{R}^n$.

Definition 2: $\forall X \subset \mathbf{R}^n$, the convex hull of X is defined as

$$CH(X) = \left\{ \mathbf{x} \middle| \mathbf{x} = \sum_{1\le i\le |X|} \alpha_i \mathbf{x}_i, \sum_{1\le i\le |X|} \alpha_i = 1, \mathbf{x}_i \in X, \alpha_i \ge 0, \alpha_i \in \mathbf{R} \right\}.$$

Definition 3: The margin of $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b$ with respect to X and Y is defined as

$$D(f|X,Y) = \inf\left\{ \frac{|f(\mathbf{x}) - f(\mathbf{y})|}{\|\mathbf{w}\|}, \mathbf{x}\in CH(X), \mathbf{y}\in CH(Y) \right\}.$$

It is well known that the SVM of X and Y is their linear discriminant function with the maximum margin, if they are linearly separable. Moreover, it could be proven that, if $(\mathbf{w}^*, b^*)$ is the unique nonzero solution of HM-SVM mentioned above, $f(\mathbf{x}) = \mathbf{w}^*\cdot\mathbf{x} + b^*$ must be the SVM of linearly separable X and Y [46]–[48]. However, note that $f_\mu(\mathbf{x}) = \mu\mathbf{w}^*\cdot\mathbf{x} + \mu b^*$ (for $\mu > 0$) is also the SVM of X and Y because it has the same margin as $f(\mathbf{x}) = \mathbf{w}^*\cdot\mathbf{x} + b^*$. Therefore, a SVM $f_\mu(\mathbf{x})$ may not be the solution of HM-SVM.

If $X, Y \subset R^n$ are two finite sets, there exists a very close relationship between their SVM and convex hulls [47], [48]. In fact, if X and Y are linearly separable, the problem of constructing the SVM for them could be converted to the following problem of computing the minimum distance between *CH(X)* and *CH(Y)* [30]:

$$\min \|\mathbf{x}-\mathbf{y}\| \text{ s.t. } \mathbf{x}\in CH(X), \mathbf{y}\in CH(Y). \qquad \text{(NPP)}$$

Although the solution $(\mathbf{x}^*, \mathbf{y}^*)$ of NPP may not be unique, it can be proven that $f(\mathbf{x}) = \mathbf{w}^*\cdot\mathbf{x} + b^*$, with the maximum margin of $\|\mathbf{x}^* - \mathbf{y}^*\|$, is always an SVM of X and Y for $\lambda \ne 0$, $\mathbf{w}^* = \lambda(\mathbf{x}^* - \mathbf{y}^*)$, $b^* = \lambda\left(\|\mathbf{y}^*\|^2 - \|\mathbf{x}^*\|^2\right)/2$. If $\lambda = 2/\|\mathbf{x}^* - \mathbf{y}^*\|^2$, the solution of HM-SVM can be easily expressed as

$$\mathbf{w}^* = \frac{2(\mathbf{x}^* - \mathbf{y}^*)}{\|\mathbf{x}^* - \mathbf{y}^*\|^2}, \qquad b^* = \frac{\left(\|\mathbf{y}^*\|^2 - \|\mathbf{x}^*\|^2\right)}{\|\mathbf{x}^* - \mathbf{y}^*\|^2}.$$

In this section, we will further analyze how convex hulls are related to SVM, summarizing some known results [49], [50] in Theorems 2–4 and presenting new results in Theorems 1, 5, and 6 in order to establish a common mathematical background, where Theorems 5 and 6 are original and significant as far as we know. Moreover, in Appendix A we strictly prove Theorems 1–6.

Theorem 1: $0 \le D(f|X,Y) \le d(CH(X), CH(Y))$ for any linear function $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b$.

Theorem 2: If $X \subseteq \mathbf{R}^n$ is a finite set, $\mathbf{x}_0 \in \mathbf{R}^n$ and $\mathbf{x}_0 \notin CH(X)$, then there exists a unique minimal point $\mathbf{x}^* \in CH(X)$ to $\mathbf{x}_0$ such that $d(\mathbf{x}_0, \mathbf{x}^*) = \min\{d(\mathbf{x}_0, \mathbf{x}), \mathbf{x}\in CH(X)\}$, which holds if and only if $\forall \mathbf{x}\in CH(X)$, $(\mathbf{x} - \mathbf{x}^*)\cdot(\mathbf{x}^* - \mathbf{x}_0) \ge 0$.

Algorithm1–CDMA
Input: X, Y
Output: $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b$
(1) $\mathbf{x}^* \in X$; $\mathbf{y}^* \in Y$;
(2) $\mathbf{x}_1 = \mathbf{x}^*$; $\mathbf{y} = \mathbf{y}^*$;
(3) $\mathbf{x}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \left\{ d(\mathbf{z}, \mathbf{y}^*) \middle| \mathbf{z}\in X \text{ or } \mathbf{z} = \mathbf{x}_1 + \lambda\cdot(\mathbf{x}_2 - \mathbf{x}_1), 0 < \lambda = \frac{(\mathbf{x}_2 - \mathbf{x}_1)\cdot(\mathbf{y}^* - \mathbf{x}_1)}{(\mathbf{x}_2 - \mathbf{x}_1)\cdot(\mathbf{x}_2 - \mathbf{x}_1)} < 1, \mathbf{x}_2 \ne \mathbf{x}_1, \mathbf{x}_2 \in X \right\}$;
(4) $\mathbf{y}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \left\{ d(\mathbf{x}^*, \mathbf{z}) \middle| \mathbf{z}\in Y \text{ or } \mathbf{z} = \mathbf{y}_1 + \mu\cdot(\mathbf{y}_2 - \mathbf{y}_1), 0 < \mu = \frac{(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}^* - \mathbf{y}_1)}{(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1)} < 1, \mathbf{y}_2 \ne \mathbf{y}_1, \mathbf{y}_2 \in Y \right\}$;
(5) **if** $d(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}^*, \mathbf{y}^*) \ge \varepsilon > 0$, **goto** (2);
(6) $\mathbf{w} = \mathbf{x}^* - \mathbf{y}^*$; $b = (\|\mathbf{y}^*\|^2 - \|\mathbf{x}^*\|^2)/2$;
(7) **if** $\exists \mathbf{x}\in X$, $\mathbf{w}\cdot\mathbf{x} + b \le 0$ *or* $\exists \mathbf{y}\in Y$, $\mathbf{w}\cdot\mathbf{y} + b \le 0$, let $f(\mathbf{x}) = 0$:
(8) else $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b$.

Fig. 2. CDMA—the cross distance minimization algorithm.

Theorem 3: Two finite subsets $X, Y \subseteq \mathbf{R}^n$ are linearly separable if and only if $CH(X) \cap CH(Y) = \varnothing$.

Theorem 4: If $X, Y \subseteq \mathbf{R}^n$ are two linearly separable finite subsets given $\mathbf{x}_1, \mathbf{x}_2 \in CH(X)$, $\mathbf{y}_1, \mathbf{y}_2 \in CH(Y)$ satisfying

$$d(\mathbf{x}_1, \mathbf{y}_1) = d(\mathbf{x}_2, \mathbf{y}_2) = d_{\min} = d(CH(X), CH(Y)).$$

$f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b$ is an SVM of X and Y, where

$$\mathbf{w} = \mathbf{x_1} - \mathbf{y_1} = \mathbf{x_2} - \mathbf{y_2}, \qquad b = \frac{\left(\|\mathbf{y}_1\|^2 - \|\mathbf{x}_1\|^2\right)}{2} = \frac{\left(\|\mathbf{y}_2\|^2 - \|\mathbf{x}_2\|^2\right)}{2}.$$

Theorem 5: If $Y \subseteq \mathbf{R}^n$ is a finite set and $d(\mathbf{x}_0, \mathbf{y}^*) = \min\{d(\mathbf{x}_0, \mathbf{y}), \mathbf{y} \in CH(Y)\}$ given $\mathbf{x}_0 \in \mathbf{R}^n$, then $\forall \mathbf{y_1} \in CH(Y)$, $\mathbf{y}_1 \ne \mathbf{y}^*$, $\exists \mathbf{y}_2 \in Y$, $\mathbf{y}_2 \ne \mathbf{y}_1$, such that

$$d(\mathbf{x}_0, \mathbf{y}_2) < d(\mathbf{x}_0, \mathbf{y}_1) \text{ or } d(\mathbf{x}_0, \mathbf{y}_1 + \alpha\cdot(\mathbf{y}_2 - \mathbf{y}_1)) < d(\mathbf{x}_0, \mathbf{y}_1)$$

where

$$0 < \alpha = \frac{(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1)}{(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1)} < 1.$$

Theorem 6: If $X, Y \subseteq \mathbf{R}^n$ are two finite subsets given $\mathbf{x}^* \in CH(X)$, $\mathbf{y}^* \in CH(Y)$ satisfying $d(\mathbf{x}^*, \mathbf{y}^*) = \min\{d(\mathbf{x}^*, y), \mathbf{x} \in CH(Y)\}$ and $d(\mathbf{x}^*, \mathbf{y}^*) = \min\{d(\mathbf{x}, \mathbf{y}^*), x \in CH(X)\}$, then $(\mathbf{x}^*, \mathbf{y}^*)$ is the nearest point pair between *CH(X)* and *CH(Y)*

$$d(\mathbf{x}^*, \mathbf{y}^*) = \min\{d(\mathbf{x}, \mathbf{y}), \mathbf{x}\in CH(X), \mathbf{y}\in CH(Y)\}.$$

In the above-mentioned theorems, Theorem 4 shows that the problem of constructing the SVM of X and Y can be converted to that of computing the nearest point pair $(\mathbf{x}^*, \mathbf{y}^*)$ between their convex hulls *CH(X)* and *CH(Y)*, where the margin of the SVM is just $d(\mathbf{x}^*, \mathbf{y}^*) = d(CH(X), CH(Y))$. Theorems 5 and 6 indicate how to compute $(\mathbf{x}^*, \mathbf{y}^*)$. Based on Theorems 4–6, it is not difficult to design a new iterative algorithm to construct a non-kernel HM-SVM, i.e., the CDMA, detailed in Fig. 2.

In Algorithm 1-CDMA, ε is called "precision parameter," for controlling the convergence condition, "$f(x) = 0$" represents a linear function with a random normal vector, meaning that X and Y are linearly nonseparable or very hard to separate by CDMA at ε-precision, $\mathbf{z} = \mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1)$ actually represents the perpendicular point from $\mathbf{y}^*$ to the line segment

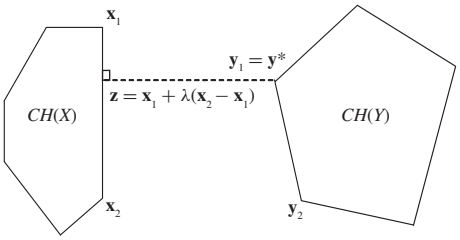


Fig. 3. $\mathbf{z} = \mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1)$ represents the perpendicular point from $\mathbf{y}^*$ to the line segment $CH(\mathbf{x}_1, \mathbf{x}_2)$, if $\mathbf{x}_1$ is not the nearest point from y^* to $CH(X)$, there must exist another point $\mathbf{x}_2$ such that $d(\mathbf{x}_2, \mathbf{y}^*) < d(\mathbf{x}_1, \mathbf{y}^*)$ or $d(\mathbf{z}, \mathbf{y}^*) < d(\mathbf{x}_1, \mathbf{y}^*)$.

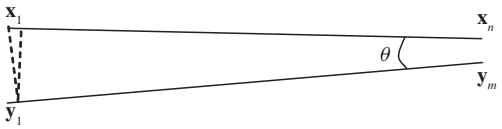


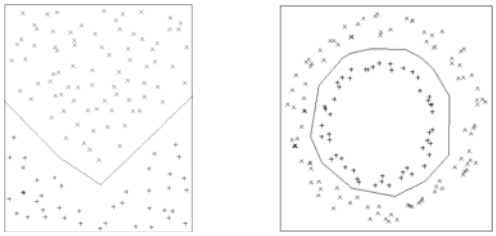
Fig. 4. Example of X and Y on two straight lines intersecting with a small angle of θ.

$CH(\mathbf{x}_1, \mathbf{x}_2)$ if $0 < \lambda < 1$ (see Fig. 3), while $\mathbf{z} = \mathbf{y}_1 + \mu(\mathbf{y}_2 - \mathbf{y}_1)$ represents the perpendicular point from $\mathbf{x}^*$ to the line segment $CH(\mathbf{y}_1, \mathbf{y}_2)$ if $0 < \mu < 1$. According to Theorem 5, if $\mathbf{x}_1$ is not the nearest point from $\mathbf{y}^*$ to $CH(X)$, there must exist another point $\mathbf{x}_2$ such that $d(\mathbf{x}_2, \mathbf{y}^*) < d(\mathbf{x}_1, \mathbf{y}^*)$ or $d(\mathbf{x}_1 + l(\mathbf{x}_2 - \mathbf{x}_1), \mathbf{y}^*) < d(\mathbf{x}_1, \mathbf{y}^*)$; and if $\mathbf{y}_1$ is not the nearest point from $\mathbf{x}^*$ to $CH(Y)$, there must exist another point $\mathbf{y}_2$ such that $d(\mathbf{x}^*, \mathbf{y}_2) < d(\mathbf{x}^*, \mathbf{y}_1)$ or $d(\mathbf{x}^*, \mathbf{y}_1 + \mu(\mathbf{y}_2 - \mathbf{y}_1)) < d(\mathbf{x}^*, \mathbf{y}_1)$. Therefore, a nearer point pair $(\mathbf{x}^*, \mathbf{y}^*)$ can be always found if $(\mathbf{x}_1, \mathbf{y}_1)$ is not the nearest one.

It is not difficult to see that CDMA starts with any point pair $(\mathbf{x}_1, \mathbf{y}_1)$ in $X \times Y$ and repeatedly replaces them with a nearer point pair $(\mathbf{x}^*, \mathbf{y}^*)$ in $CH(X) \times CH(Y)$ until convergence takes place. No matter whether X and Y are linearly separable or not, CDMA always converges to a nearest point pair $(\mathbf{x}^*, \mathbf{y}^*)$ between $CH(X)$ and $CH(Y)$, according to Theorems 4–6. And in the linearly separable case, it will output a unique SVM constructed by $(\mathbf{x}^*, \mathbf{y}^*)$. Otherwise, it may output $f(\mathbf{x}) = 0$ because $(\mathbf{x}^*, \mathbf{y}^*)$ converges to the zero vector $(\mathbf{0}, \mathbf{0})$ in this case. Note that CDMA cannot calculate the distance from a point $\mathbf{x}_0$ inside $CH(Y)$ to the boundary of $CH(Y)$, which is a NP-hard problem [51]. Principally, CDMA has an adaptation rule as simple as the S–K algorithm [32] and simpler than Keerthi's method [30], and it is very easy to be implemented with a linear space complexity of $O(|X| + |Y|)$ and a rough time complexity of $O(I(\varepsilon) \cdot |X| + |Y|)$, where $I(\varepsilon)$, actually involving the distribution of X and Y, represents the total number of running loops for CDMA to converge at ε-precision. For example, if X and Y are on two intersecting straight lines illustrated in Fig. 4, $I(\varepsilon)$ may be estimated as a constant independent of $|X|$ and $|Y|$

$$I(\varepsilon) \approx \frac{\left[\log \varepsilon - \log\left(d(\mathbf{x}_1, \mathbf{y}_1) \sin^2 \theta\right)\right]}{\log\left(\cos^2 \theta\right)}$$

where $0 < \varepsilon < 1.0$ and $\varepsilon \ll d(\mathbf{x}_1, \mathbf{y}_1) \sin^2 \theta$ with the angle θ small enough. Therefore, we guess that $I(\varepsilon)$ is possibly a constant, mainly depending on the relative curvature of X and Y around the nearest point pair $(\mathbf{x}^*, \mathbf{y}^*)$ between them, but its general proof in theory is an open problem.



(a) (b)

Fig. 5. Two artificial examples of convexly separable datasets with "+" standing for X and "×" for Y, where (a) Y is convexly separable to X with a 3-line-segment boundary, and (b) X is convexly separable to Y with a 14-line-segment boundary.

III. Conlitrons and Multiconlitrons

In this section, we will consider the problem of how to separate two arbitrarily complicated nonintersecting classes by a number of linear functions, without using kernels. To solve the problem, we first discuss the concept of "convexly separable."

Given any two finite subsets $X, Y \subseteq \mathbf{R}^n$, X is called convexly separable to Y if there exists a conlitron—convex linear perceptron (CLP) from X to Y, namely, a finite linear function set

$$LFS = \left\{ f_l(\mathbf{x}) = \mathbf{w}_l \cdot \mathbf{x} + b_l, (\mathbf{w}_l, b_l) \in \mathbf{R}^n \times \mathbf{R}, 1 \le l \le L \right\}$$

satisfying the following two conditions:

1) $\forall \mathbf{x} \in X, \forall 1 \le l \le L = |LFS|, f_l(\mathbf{x}) = \mathbf{w}_l \cdot \mathbf{x} + b_l > 0$;
2) $\forall \mathbf{y} \in Y, \exists 1 \le l \le L = |LFS|, f_l(\mathbf{y}) = \mathbf{w}_l \cdot \mathbf{y} + b_l < 0$.

If $CLP = LFS = \{f_l(\mathbf{x}), 1 \le l \le L\}$ is a conlitron from X to Y, we define its decision function as

$$CLP(\mathbf{x}) = \begin{cases} +1, & \forall 1 \le l \le L, f_l(\mathbf{x}) \ge 0; \\ -1, & \exists 1 \le l \le L, f_l(\mathbf{x}) < 0. \end{cases}$$

We call X and Y convexly separable if X is convexly separable to Y or Y is convexly separable to X. Obviously, the two artificial examples in Fig. 5(a) and (b) are linearly nonseparable, but both of them are convexly separable, where the two separating boundaries are determined, respectively, by 3 linear functions and 14 linear functions. In Fig. 5(a), X is convexly nonseparable to Y, while Y is convexly separable to X, in Fig. 5(b), X is convexly separable to Y, while Y is convexly nonseparable to X. It is worth noting that, if X and Y are linearly separable, they must be convexly separable, however, if X and Y are convexly separable, they may not be linearly separable. For instance, provided that two triangles internally intersect with the three vertices of one outside the other, the two 3-vertex sets of them are convexly separable, but linearly nonseparable, as is illustrated in Fig. 6.

We have Theorem 7 about convex separability.

Theorem 7: Given two finite sets $X, Y \subseteq \mathbf{R}^n$, X is convexly separable to Y if and only if $\forall \mathrm{y} \in Y, \mathrm{y} \notin CH(X)$.

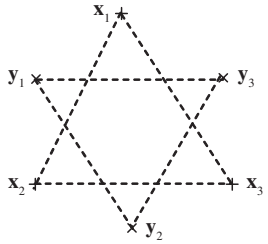


Fig. 6. $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ is convexly separable to $Y = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$, and vice versa, although X and Y are convexly separable, they are linearly nonseparable because *CH(X)* and *CH(Y)* intersect.

Algorithm 2 – SCA

Input: $X = \{\mathbf{x}_i, 1 \le i \le N\}$, $Y = \{\mathbf{y}_j, 1 \le j \le M\}$

Output: $CLP = \{f_l(\mathbf{x}), 1 \le l \le L\}$

(1) **For** $1 \le i \le M$, $g_j(\mathbf{x}) =$ **CDMA** $(X, \{\mathrm{y}_j\})$;

(2) $l = 1$; $Y_1 = Y$:

(3) $q = \arg\max_j \{g_j(\mathbf{y}_j), \mathbf{y}_j \in Y_l\}$

(4) $f_l(\mathbf{x}) = g_q(\mathbf{x})$;

(5) $Y_{l+1} = \{\mathbf{y} | f_l(\mathbf{y}) > g_q(\mathbf{y}_q), \mathbf{y} \in Y_l\}$;

(6) **if** $Y_{l+1} \neq \emptyset$, $l = l + 1$, **goto** (3);

(7) $\mathrm{L} = l$;

(8) **if** $\exists i \exists l$ $f_l(\mathbf{x}_i) \le 0$ *or* $\exists j \forall l$ $f_l(\mathbf{y}_j) \le 0$, **let** $CLP = \emptyset$;

(9) **else** $CLP = \{f_l(\mathbf{x}), 1 \le l \le L\}$.

Fig. 7. SCA—the Support Conlitron Algorithm.

Proof:

1) If $\forall \mathbf{y} \in Y, \mathbf{y} \notin CH(X)$, let $l = 1$ and $Y_l = Y$. Obviously, $\exists \mathbf{y}_l^* \in Y_l, \exists \mathbf{x}_l^* \in CH(X)$, such that

$$d(\mathbf{x}_l^*, \mathbf{y}_l^*) = \min\{d(\mathbf{x}, \mathbf{y}), \mathbf{x} \in CH(X), \mathbf{y} \in Y_l\} > 0$$

by which we get

$$f_l(\mathbf{x}) = \left(\mathbf{x}_l^* - \mathbf{y}_l^*\right) \cdot \left(\mathbf{x} - \frac{\mathbf{x}_l^* + \mathbf{y}_l^*}{2}\right)$$

satisfying

$$\forall \mathbf{x} \in X,\ f_l(\mathbf{x}) \ge \frac{d^2(\mathbf{x}_l^*, \mathbf{y}_l^*)}{2} > 0;$$

$$f_l(\mathbf{y}_l^*) \le -\frac{d^2(\mathbf{x}_l^*, \mathbf{y}_l^*)}{2} < 0.$$

If $Y_{l+1} = \left\{\mathbf{y} \,\middle|\, f_l(\mathbf{y}) > f_l(\mathbf{y}_l^*), \mathbf{y} \in Y_l\right\}$ is nonempty, let $l = l + 1$ and repeat to find $(\mathbf{x}_l^*, \mathbf{y}_l^*)$ for constructing $f_l(\mathbf{x})$ until $Y_{l+1} = \emptyset$. Because $Y_1 = Y$ is a finite set and $|Y_{l+1}| < |Y_l|$, the process must stop for some iterative number L satisfying $Y_{L+1} = \emptyset$.

Let $LFS = \{f_l(\mathbf{x}), 1 \le l \le L\}$, which is a conlitron from X to Y, meaning that X is convexly separable to Y.

2) If X is convexly separable to Y, there exists a finite linear function set

$$LFS = \left\{f_l(\mathbf{x}) = \mathbf{w}_l \cdot \mathbf{x} + b_l, (\mathbf{w}_l, b_l) \in \mathbf{R}^n \times \mathbf{R}, 1 \le l \le L\right\}$$

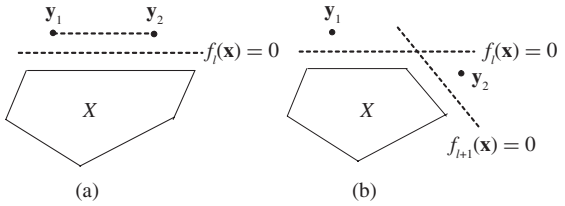


Fig. 8. Example to show the uniqueness for the solutions of SCA. (a) $g_1(\mathbf{y}_2) = g_2(\mathbf{y}_1)$. (b) $g_1(\mathbf{y}_2) \neq g_2(\mathbf{y}_1)$.

satisfying the following two conditions:

1) $\forall \mathbf{x} \in X, \forall 1 \le l \le L$, $f_l(\mathbf{x}) = \mathbf{w}_l \cdot \mathbf{x} + b_l > 0$;

2) $\forall \mathbf{y} \in Y, \exists 1 \le l \le L$, $f_l(\mathbf{y}) = \mathbf{w}_l \cdot \mathbf{y} + b_l < 0$.

If $\exists \mathbf{y} \in Y, \mathbf{y} \in CH(X), \mathbf{y} = \sum \alpha_i \mathbf{x}_i, \sum \alpha_i = 1, \alpha_i \ge 0$, we have $\forall 1 \le l \le L$, $f_l(\mathbf{y}) = \mathbf{w}_l \cdot \mathbf{y} + b_l = \mathbf{w}_l \cdot \left(\sum \alpha_i \mathbf{x}_i\right) + b_l = \sum \alpha_i (\mathbf{w}_l \cdot \mathbf{x}_i + b_l) = \sum \alpha_i f(\mathbf{x}_i) > 0$.

This contradicts $\forall \mathbf{y} \in Y, \exists 1 \le l \le L$, $f_l(\mathbf{y}) = \mathbf{w}_l \cdot \mathbf{y} + b_l < 0$.

Therefore, $\forall \mathbf{y} \in Y, \mathbf{y} \notin CH(X)$. ■

According to Theorem 7, if X is convexly separable to Y, we can use Algorithm 1-CDMA to directly derive an iterative algorithm to compute the conlitron CLP from X to Y, namely, the SCA, detailed in Fig. 7. If $q = argmax_j\{g_j(y_j), y_j \in Y_l\}$ is unique for every l, the solutions of SCA are obviously unique. In the case that q is not unique, e.g., $\exists \mathbf{y}_1 \neq \mathbf{y}_2, g_1(\mathbf{y}_1) = g_2(\mathbf{y}_2) = \max\{g_j(\mathbf{y}_j), \mathbf{y}_j \in Y_l\} < 0$, if $g_1(\mathbf{y}_2) = g_2(\mathbf{y}_1)$ [see Fig. 8(a)], $f_l(\mathbf{x}) = g_1(\mathbf{x}) = g_2(\mathbf{x})$; otherwise [see Fig. 8(b)], either "$f_l(\mathbf{x}) = g_1(\mathbf{x})$ and $f_{l+1}(\mathbf{x}) = g_2(\mathbf{x})$" or "$f_l(\mathbf{x}) = g_2(\mathbf{x})$ and $f_{l+1}(\mathbf{x}) = g_1(\mathbf{x})$." Therefore, the solution of $\mathbf{SCA}(X, Y)$ is always unique, although a general conlitron is not unique for two convexly separable datasets X and Y.

In Algorithm 2-SCA, "$CLP = \varnothing$" represents a conlitron containing linear functions with random normal vectors, which means that X is convexly nonseparable to Y or very hard to be convexly separated to Y at the chosen precision parameter ε for SCA to call CDMA. No matter whether X is convexly separable to Y or not, it is easy to determine that the SCA always converges according to Theorems 4–7, with the same space complexity of $O(|X| + |Y|)$ as CDMA and but with a different rough time complexity of $O(I_{\max}(\varepsilon)|X| \cdot |Y|)$, where $I_{\max}(\varepsilon)$, actually involving the distribution of X and Y, represents the maximum number of running loops for $\mathbf{CDMA}\{X, \{\mathbf{y}_j\}\}(1 \le j \le |Y|)$ called by SCA to converge at the ε-precision. However, if X is convexly nonseparable to Y, SCA may output "$CLP = \varnothing$," as CDMA may output "$f(\mathbf{x}) = 0$."

In addition, we define the convex distance "from X to Y" as

$$d_{CH}(X|Y) = \min\{d(CH(X), \mathbf{y}), \mathbf{y} \in Y\}.$$

It is not difficult to see that the unique conlitron computed by $\mathbf{SCA}(X, Y)$ can separate X and Y by the maximum margin of $d_{CH}(X|Y)$, and thus may be regarded as an extension of the SVM in the convexly separable case. For this reason, we call it "support conlitron." The two separating boundaries given in Fig. 5(a) and (b) are actually determined by their corresponding support conlitrons, the one in Fig. 5(a) is computed by $\mathbf{SCA}(Y, X)$ for the case that Y is convexly separable to X, the other in Fig. 5(b) by $\mathbf{SCA}(X, Y)$ for

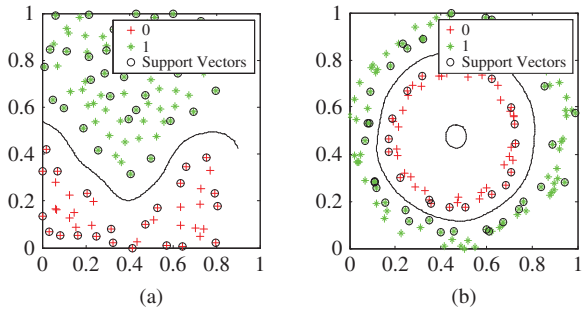


Fig. 9. (a) and (b) Nonlinear separating boundaries determined by a soft margin RBF SVM ($C = 1.0$, $\gamma = 8.0$) for the two datasets illustrated in Fig. 5(a) and (b).

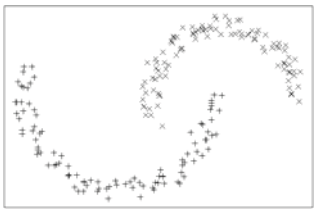


Fig. 10. Two-moon set with "+" standing for X and "×" for Y.

the case that X is convexly separable to Y. For comparison with the kernel SVM, in Fig. 9(a) and (b) we provide the separating boundaries computed by a soft margin RBF SVM ($C = 1.0, \gamma = 8.0$) for the convex-concave set and the two-circle set illustrated in Fig. 5(a) and (b).

In the case where X and Y are convexly separable, we define the support conlitron of X and Y as $\mathbf{SCA}(X, Y)$ if $d_{CH}(X|Y) \geq d_{CH}(Y|X), \mathbf{SCA}(Y, X)$ otherwise. Because the margin of SVM is $d(CH(X), CH(Y))$, i.e., the distance between the convex hulls of X and Y, we can similarly define the margin of the support conlitron of X and Y as

$$d_{CH}(X, Y) = \max\{d_{CH}(X|Y), d_{CH}(Y|X)\}$$

which is called the convex distance "between X and Y," as a measure of generalization for support conlitrons.

Although the convexly separable concept extends the "linearly separable" concept in theory, there remains a problem that any two nonintersecting finite subsets in $\mathbf{R}^n$ may not be convexly separable (e.g., the two moons in Fig. 10). For solving this problem, we need to introduce the concept of a "multiconlitron," which is a union of multiple conlitrons. It can be theoretically proven that a multiconlitron has the ability to separate any two nonintersecting finite subsets in $\mathbf{R}^n$.

Given two finite nonintersecting subsets $X, Y \subseteq \mathbf{R}^n$, a multiconlitron (abbr. MCLP) from X to Y is defined as a finite conlitron set $CLPS = \{CLP_k, 1 \leq k \leq K\}$ which satisfies the following two conditions:

1) $\forall \mathbf{x} \in X, \exists 1 \leq k \leq K = |CLPS|, CLP_k(\mathbf{x}) = +1$;
2) $\forall \mathbf{y} \in Y, \forall 1 \leq k \leq K = |CLPS|, CLP_k(\mathbf{y}) = -1$.

If $MCLP = CLPS = \{CLP_k, 1 \leq k \leq K\}$ is a multiconlitron from X to Y, we define its decision function as

$$MCLP(\mathbf{x}) = \begin{cases} +1, & \exists 1 \leq k \leq K, CLP_k(\mathbf{x}) = +1 \\ -1, & \forall 1 \leq k \leq K, CLP_k(\mathbf{x}) = -1. \end{cases}$$

We have Theorem 8 about multiconlitrons.

Algorithm 3 – SMA

Input: $X = \{\mathbf{x}_i, 1 \leq i \leq N\}$, $Y = \{\mathbf{y}_j, 1 \leq j \leq M\}$

Output: $MCLP = \{CLP_k, 1 \leq k \leq K\}$

(1) $I = \{i, 1 \leq i \leq N\}$;

(2) $k = 1$; $I_1 = I$;

(3) $p = \arg\min_i \{d(\{\mathbf{x}_i\}, Y), i \in I_k\}$;

(4) $CLP_k = \mathbf{SCA}(\{\mathbf{x}_p\}, Y)$;

(5) **if** $CLP_k = \emptyset$, $MCLP = \emptyset$ *and* **stop**;

(5) $I_{k+1} = \{i | \exists f \in CLP_k, f(\mathbf{x}_i) < f(\mathbf{x}_p), i \in I_k\}$;

(6) **if** $I_{k+1} \neq \emptyset$, $k = k + 1$, **goto** (3);

(7) $K = k$;

Fig. 11. SMA—the support multiconlitron algorithm.

Theorem 8: If $X, Y \subseteq \mathbf{R}^n$ are two finite nonintersecting sets, there must exist a multiconlitron from X to Y and a multiconlitron from Y to X.

Proof: Because $X \cap Y = \varnothing$, it is obvious that $\forall \mathbf{x}_k \in X, \{\mathbf{x}_k\}$ is convexly separable to Y. It might be assumed that CLP_k is a support conlitron from $\{\mathbf{x}_k\}$ to Y, namely, $CLP_k = \mathbf{SCA}(\mathbf{x}_k, Y)$.

Let $K = |X|$, $CLPS = \{CLP_k, 1 \leq k \leq K\}$, which satisfies the following two conditions:

1) $\forall \mathbf{x} \in X, \exists 1 \leq k \leq K = |CLPS, | CLP_k(\mathbf{x}) = +1$;
2) $\forall \mathbf{y} \in Y, \forall 1 \leq k \leq K = |CLPS, | CLP_k(\mathbf{y}) = -1$.

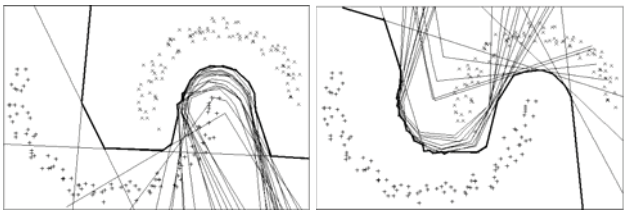
Therefore, $MCLP = CLPS$ is a multiconlitron from X to Y. Similarly, we can get a multiconlitron from Y to X. ■

Obviously, the multiconlitron $MCLP$ constructed in the proof of Theorem 8 may contain a number of redundant conlitrons, which can be deleted to change the MCLP to a smaller multiconlitron. For example, if $\mathbf{x}_i \neq \mathbf{x}_k$ and $CLP_k(\mathbf{x}_i) = +1$, then the conlitron CLP_i can be taken as a redundant conlitron after CLP_k is chosen. By appropriately deleting some of these redundant conlitrons, we can derive an iterative algorithm to compute the multiconlitron $MCLP$ from X to Y, namely, the SMA, detailed in Fig. 11. It is obvious that the solutions of SMA are unique if $p = \arg\min_i \{d(\{\mathbf{x}_i\}, Y), i \in I_k\}$ is unique for every l. In the case that p is not unique

$$\exists \mathbf{x}_1 \neq \mathbf{x}_2, d(\{\mathbf{x}_1\}, Y) = d(\{\mathbf{x}_2\}, Y) = \min\{d(\{\mathbf{x}_i\}, Y), i \in I_k\}$$

one possibility is $CLP_k = \mathbf{SCA}(\{\mathbf{x}_1\}, Y) = \mathbf{SCA}(\{\mathbf{x}_2\}, Y)$, the other is either $CLP_k = \mathbf{SCA}(\{\mathbf{x}_1\}, Y)$ and $CLP_{k+1} = \mathbf{SCA}(\{\mathbf{x}_2\}, Y)$ or $CLP_k = \mathbf{SCA}(\{\mathbf{x}_2\}, Y)$ and $CLP_{k+1} = \mathbf{SCA}(\{\mathbf{x}_1\}, Y)$. Because $\mathbf{SCA}(\mathbf{x}_1, Y)$ and $\mathbf{SCA}(\mathbf{x}_2, Y)$ are both unique, the solution of $\mathbf{SMA}(X, Y)$ is always unique, although a general multiconlitron is not unique for two commonly separable datasets X and Y.

In Algorithm 3-SMA, "$MCLP = \varnothing$" means that $X \cap Y \neq \varnothing$ (commonly nonseparable), i.e., X and Y can not be completely separated by a multiconlitron. So long as $X \cap Y = \varnothing$, it is not difficult to know that both $\mathbf{SMA}(X, Y)$ and $\mathbf{SMA}(Y, X)$ can always converge with the same space complexity of $O(|X| \cdot |Y|)$, but with different time complexities of $O(|X|^2 \cdot |Y|)$



(a) (b)

Fig. 12. Separating boundaries of the support multiconlitrons are marked by the bold broken lines for the two-moon dataset. The support multiconlitron from X to Y in (a) is a union of 18 conlitrons where there are totally 115 linear functions, while the support multiconlitron from Y to X in (b) is also a union of 18 conlitrons, but containing only 72 linear functions in total.

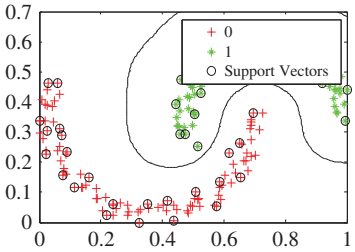


Fig. 13. Nonlinear boundary computed by RBF SVM ($C = 1.0, \gamma = 8.0$).

and $O(|X| \cdot |Y|^2)$ for training. Moreover, the testing time complexity of **SMA** can be estimated as $O(KMn)$, where $M = \max_{1 \le k \le K} \{|CLP_k|\}$, K is the number of conlitrons, and n is the dimension of data.

It is easy to see in Fig. 12 that even the multiconlitron from X to Y, $\mathbf{SMA}(X, Y)$, is usually different from the multiconlitron from Y to X, $\mathbf{SMA}(Y, X)$, where both $\mathbf{SMA}(X, Y)$ and $\mathbf{SMA}(Y, X)$ are a union of 18 conlitrons, but they respectively contain 115 and 75 linear functions as a whole. Nevertheless, both $\mathbf{SMA}(X, Y)$ and $\mathbf{SMA}(Y, X)$, here called a "support multiconlitron," can be regarded as a non-kernel extension of SVM, because they can separate two nonintersecting classes X and Y by the maximum margin of $d(X, Y)$ as a measure of generalization for them.

If $\mathbf{SMA}(X, Y)$ contains a total number of linear functions no greater than $\mathbf{SMA}(Y, X)$, we define the support multiconlitron of X and Y as $\mathbf{SMA}(X, Y)$, otherwise $\mathbf{SMA}(Y, X)$. Thus, the final support multiconlitron of the two-moon dataset in Fig. 10 is $\mathbf{SMA}(Y, X)$, as shown in Fig. 12(b). To compare this to a kernel SVM, in Fig. 13 we also provide the separating boundaries computed by a soft margin RBF SVM ($C = 1.0, \gamma = 8.0$) for the two-moon set.

IV. Experimental Results

To validate and evaluate the performance of CDMA, SCA, and SMA, we report results comparing them with SVM on one synthetic dataset of two cross planes in $\mathbf{R}^3$ as well as 11 publicly available two-class databases from the UCI Repository [52]. The selected databases, including their codes, size, dimensionality, separability, and margin, are presented in Table I, with the "training + test" sets indicated for the last four databases, but not for the first eight. The attribute values of separability and margin in Table I are obtained by computational experiments, where $\varepsilon = 0.000001$ is prescribed for CDMA and $\varepsilon = 0.001$ for SCA to call CDMA (the same choices of ε made in Tables II, V, and VII). The value of separability is one of "CDMA," "SCA," "SMA," and "None," meaning that the corresponding database can be completely separated by one of the algorithms or none of them, where "CDMA" (linearly separable) implies "SCA" and "SMA," and "SCA" (convexly separable) implies "SMA." In addition, "SCA(CDMA)" means that the total database can be completely separated by SCA, the training set by CDMA, while "None(SMA)" means that the total database is "commonly nonseparable" (i.e., the two classes have common points with the distance of 0.000), but the training set can be completely separated by SMA.

For the last four databases including (Spe), (Mo1), (Mo2), and (Mo3) in Table I, we directly compare CDMA, SCA, and SMA with RBF SVM (SVM.rbf) and linear SVM (SVM.lin) on the indicated training and test sets, reporting the accuracies with testing (training) time. For the first eight databases, however, we randomly split each of them into two halves for 50 times, one half for training, the other for testing, and report the mean accuracies and standard deviations with average testing (training) time. We summarize the main results in Table II, where the highest accuracies are in bold and the time data are recorded on the same computer (DELL GX620, 3.2-GHz CPU, 2.0-GB RAM). Table II provides many blank results (i.e., outputs not comparable) for CDMA and SCA because only a few databases in Table I are linearly or convexly separable.

In Table II, the training and testing process of CDMA, SCA, and SMA is very simple, neither choosing kernels and parameters nor normalizing features for data, while it is much more complex for SVM. We scaled each feature to $[0, 1]$ for all the databases before training and testing, then used library of support vector machine [53] to conduct experiments on these normalized databases for C-parameterized SVM.lin and (C, γ)-parameterized SVM.rbf in the soft margin model, where we adopted an exponentially growing grid search scheme recommended in [54] and 10-fold cross validation to identify the optimal values for the two parameters C and γ, respectively, from the candidate sets $\{2^i | i = -4, -3, \ldots, 3, 4\}$ and $\{2^i | i = -7, -6, \ldots, 4, 5\}$ (the same for Tables IV–VII if not indicated). The training time for SVM.lin and SVM.rbf includes in part the time to choose optimal parameters.

From Table II, we can see the following.

1) Without considering blank results, all the training accuracies for CDMA, SCA, and SMA were 100.0%, while those for SVM.rbf and SVM.lin were less than 100.0%. On one hand, this means that the algorithms of CDMA, SCA, and SMA could easily and completely separate two classes with the corresponding separabilities. On the other hand, it means that a "soft margin" with the proper selection of C and γ is very useful for SVM.rbf and SVM.lin to produce error-allowed training models with better generalization.
2) SMA outperforms SVM.lin on many of the selected databases and provides similar results to SVM.rbf on

TABLE I
UCI DATASETS USED IN THE EXPERIMENTS

Database	Code	Size	Dim	Separability	Margin
Cross planes	(Crp)	8000	3	SCA	4.187
Ionosphere	(Ion)	351	34	SMA	0.467
Sonar	(Son)	208	60	SCA	0.324
Musk (Version 1)	(Mus)	476	166	CDMA	7.876
Breast Cancer Wisconsin(Diagnostic)	(Bre)	569	30	SMA	10.922
MAGIC Gamma Telescope	(Mag)	19020	10	SMA	0.773
Parkinsons	(Par)	195	22	SMA	0.132
Pima-indians-diabetes	(Pim)	768	8	SMA	4.275
Spectf Heart	(Spe)	$80 + 187$	44	SCA(CDMA)	13.283 (3.353)
Monks-1	(Mo1)	$124 + 432$	6	None(SMA)	0.000 (1.000)
Monks-2	(Mo2)	$169 + 432$	6	None(SMA)	0.000 (1.000)
Monks-3	(Mo3)	$122 + 432$	6	None(SMA)	0.000 (1.000)

TABLE II
TESTING (TRAINING) ACCURACIES (%) AND TIME (S) IN COMPARATIVE EXPERIMENTS

Accuracy Time	CDMA	SCA	SMA	SVM.rbf	SVM.lin
(Crp)		$97.8 \pm 2.1(100 \pm 0.0)$ $2.66 \times 10^{-4}(187.52)$	$99.7 \pm 0.1(100.0 \pm 0.0)$ $2.219 \times 10^{-3}(52.68)$	$98.1 \pm 0.2(98.1 \pm 0.6)$ 0.078 (1153.06)	$60.5 \pm 3.9(60.9 \pm 3.9)$ $1.25 \times 10^{-4}(125.53)$
(Ion)			$87.3 \pm 2.0(100.0 \pm 0.0)$ $6.09 \times 10^{-4}(0.15)$	$93.5 \pm 1.6(99.1 \pm 0.9)$ 0.016 (8.71)	$85.7 \pm 2.1(93.8 \pm 2.9)$ $1.6 \times 10^{-5}(0.81)$
(Son)		$80.2 \pm 3.2(100 \pm 0.0)$ $3.29 \times 10^{-4}(0.17)$	$79.7 \pm 3.5(100.0 \pm 0.0)$ $9.37 \times 10^{-4}(0.24)$	$82.8 \pm 3.3(99.2 \pm 1.3)$ 0.016 (4.93)	$75.6 \pm 3.5(88.0 \pm 3.4)$ $1.6 \times 10^{-5}(0.28)$
(Mus)	$80.1 \pm 2.1(100.0 \pm 0.0)$ $7.8 \times 10^{-5}(64.02)$	$86.5 \pm 2.3(100 \pm 0.0)$ $6.187 \times 10^{-3}(30.53)$	$83.7 \pm 2.3(100.0 \pm 0.0)$ 0.016 (1.35)	$91.5 \pm 1.6(99.7 \pm 0.6)$ 0.031 (45.96)	$81.4 \pm 2.8(93.1 \pm 3.2)$ $7.8 \times 10^{-5}(2.71)$
(Bre)			$91.9 \pm 1.3(100.0 \pm 0.0)$ $7.18 \times 10^{-4}(0.25)$	$92.2 \pm 1.2(92.9 \pm 1.0)$ $9.5 \times 10^{-3}(13.64)$	$90.9 \pm 1.4(91.1 \pm 1.2)$ $3.1 \times 10^{-5}(0.54)$
(Mag)			$77.6 \pm 0.4(100.0 \pm 0.0)$ 3.468 (624.58)	$83.1 \pm 0.3(83.8 \pm 0.3)$ 6.235 (15 973.00)	$79.1 \pm 0.3(79.1 \pm 0.3)$ $3.43 \times 10^{-4}(560.23)$
(Par)			$82.3 \pm 3.0(100.0 \pm 0.0)$ $1.4 \times 10^{-4}(0.04)$	$81.7 \pm 1.9(86.7 \pm 2.5)$ $9.54 \times 10^{-3}(2.15)$	$78.6 \pm 2.9(81.7 \pm 4.0)$ $1.6 \times 10^{-5}(0.10)$
(Pim)			$66.9 \pm 1.9(100.0 \pm 0.0)$ $2.406 \times 10^{-3}(0.29)$	$76.0 \pm 1.4(78.2 \pm 1.8)$ 0.016 (24.85)	$75.2 \pm 1.3(75.9 \pm 1.3)$ $1.6 \times 10^{-5}(0.80)$
(Spe)	**72.2**(100.00) $1.5 \times 10^{-5}(0.77)$	59.4 (100.00) $2.97 \times 10^{-4}(0.28)$	60.4 (100.0) $1.047 \times 10^{-3}3(0.28)$	71.7 (95.0) $1.5 \times 10^{-3}(2.60)$	70.6 (90.0) $1.5 \times 10^{-5}(0.13)$
(Mo1)			85.7 (100.0) $2.81 \times 10^{-4}(0.08)$	89.6 (97.6) $1.25 \times 10^{-3}(3.61)$	67.8 (69.4) $1.6 \times 10^{-5}(0.14)$
(Mo2)			81.0 (100.0) $6.25 \times 10^{-4}(0.20)$	75.7 (94.7) $1.83 \times 10^{-3}(6.72)$	67.1 (62.1) $1.6 \times 10^{-5}(0.21)$
(Mo3)			83.6 (100.0) $2.661 \times 10^{-4}(0.28)$	95.1 (93.4) $1.22 \times 10^{-3}(3.36)$	81.5 (84.4) $1.6 \times 10^{-5}(0.12)$

some of them, while SCA performs better than linear SVM on three out of four applicable datasets. The performance of SMA may have something to do with the dimensionality of data, for it is often unsatisfactory on high-dimensional cases such as (Mus), (Spe), and (Ion).

3) On the convexly separable databases (Son) and (Mus), SCA has testing accuracies of $80.2 \pm 3.2\%$ and $86.5 \pm 2.3\%$, both higher than SVM.lin ($75.6 \pm 3.5\%$ and $81.4 \pm 2.8\%$), this shows that SCA may be indeed better than SVM.lin in some practical applications, as expected.
4) On the database (Spe), which is convexly separable as a whole but with a linearly separable training set, the testing accuracy of CDMA is 72.2%, slightly higher than SVM.rbf and SVM.lin, while that of SCA (59.4%) and that of SMA (60.4%) are both lower than SVM.rbf (71.7%) and SVM.lin (70.6%). This demonstrates that SCA and SMA may be sometimes influenced by the overfitting phenomenon which leads to bad generalization, as too many linear functions are used in the training process.
5) In the training process, both SCA and SMA run faster than SVM.rbf on all the applicable databases. For example, to finish 50-times training on (Crp), it takes SCA and SMA about $187.52 \times 50 \approx 9376$ s and $52.68 \times 50 \approx 2634$ s, respectively, but it takes

TABLE III
TIME COMPLEXITIES FOR SMA, SVM.rbf AND SVM.lin

Classifer	Time complexity	
	Training	Testing
SMA	$O(\|X\|\|Y\|(\|X\|+\|Y\|))$	$O(KMn)$
SVM.rbf	$O\left((\|X\|+\|Y\|)^3\right)$	$O(Sn)$
SVM.lin	$O(U(\|X\|+\|Y\|))$	$O(n)$

SVM.rbf about $1153.06 \times 50 \approx 57\,650$ s, to finish 50-times training on (Mag), it takes SMA about $624.58 \times 50 \approx 31\,229$ s (<9 h), but it takes SVM.rbf about $15\,973.00 \times 50 \approx 798\,650$ s (>220 h).

6) Compared to training time, most of the testing time on different databases are relatively small and may be neglected, but those for SMA are generally faster than SVM.rbf and slower than SVM.lin. For a large and complicated database such as (Mag), the testing time of SMA (3.468 s) may be in the same order of magnitude as that of SVM.rbf (6.235 s), but several orders of magnitude slower than that of SVM.lin (3.43×10^{-4} s).

It is worth noting that the testing accuracies of SCA and SMA are largely lower than SVM.rbf mainly because they are sometimes sensitive to noise and may be influenced by the overfitting phenomenon that results from using too many linear functions to separate a training set completely. Here we have no intention to solve this problem, but try to show in theory that it may be alleviated or avoided after improvements achieved by using certain techniques. Theoretically speaking, a multiconlitron is a general classifier that can separate two arbitrarily nonintersecting classes, so it can approximate any complicated boundaries such as those generated by SVM.rbf. This means we can always have a multiconlitron performing at least as well as a SVM.rbf, allowing for classification violation. Thus, we may further improve the generalizing ability of SCA and SMA by introducing a "soft margin" (like SVM) and controlling the appropriate number of linear functions in the training process with a good balance between the margin and the number. Usually, we would expect a larger margin and a lower number, and if the balance between them is better, the generalization will be better. A more detailed discussion of this topic will be included in future work, as it is beyond the scope of this paper.

It should also be noticed that the experimental training and testing time for SMA, SVM.rbf, and SVM.lin could largely conform to the theoretical time complexities of them, which are summarized in Table III. From Table III, we can see that $O(|X||Y|(|X|+|Y|))$ is the training time complexity of SMA, which is faster than the worst $O((|X|+|Y|)^3)$ of SVM.rbf with general complexity of $O(S^3)$ [55] and even linear complexity of $O(|X|+|Y|)$ for many algorithms taking SMO strategy in special cases [56], but slower than the $O(U(|X|+|Y|))$ of SVM.lin using the cutting-plane algorithm [57], where S is the number of support vectors and U is the number of nonzero features. Similarly, the testing time complexity of SMA is $O(KMn)$, probably in the same order of magnitude as the $O(Sn)$ of SVM.rbf, but obviously slower than the $O(n)$ of SVM.lin, where K is the number of conlitrons, M is the size of the largest conlitron, and n is the dimension of the data. In addition, we run SMA and SVM.rbf again on the training sets randomly selected for the first eight databases as well as already indicated for the last four databases, and provide the numbers of conlitrons for SMA and support vectors for SVM.rbf in Table IV, which is helpful to confirm the validity of the theoretical testing time complexities for SMA and SVM.rbf.

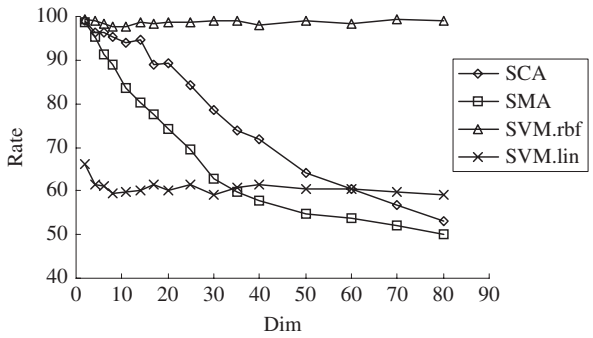


Fig. 14. Test correct rates on n-dimensional unit-sphere problems.

In order to further demonstrate how the performance of SCA and SMA is related to the dimensionality of data, we randomly generate a series of n-dimensional unit spheres containing 2000 samples that satisfy $\|\mathbf{x}\| \leq 1$ and take each of these spheres as a two-class dataset with the separating boundary $\|\mathbf{x}\| = 0.5$. Then we randomly split each of the obtained datasets into two halves only once, one half for training, the other for testing, and report the performance (accuracy or testing correct rate) of SCA, SMA, SVM.rbf, and SVM.lin in Table V with a dim-rate graph illustrated in Fig. 14. On these n-dimensional unit-sphere problems, the experimental results clearly show that the accuracies for SVM.rbf and SVM.lin vary in a relatively small range, but those for SCA and SMA decrease with the dimension numbers, from 99.1% and 98.5% to 53.1% and 50.2%, respectively. This means that SCA and SMA may encounter a serious overfitting problem on high-dimensional databases if not using any soft-margin strategies.

In order to further show that a soft margin with the proper selection of C and γ may play an important role in improving the generalizing ability of SVM.rbf, it is necessary to evaluate the performance of HM-SVM on the selected databases. Because we do not have a publicly available software package for this purpose, we use LCSM-SVM to approximate HM-SVM with a very large $C = 10^8$, run it on the selected datasets and summarize the mean accuracies and average training time in Table VI. On comparing with Table II, it takes SVM.rbf much more time to converge on (Pim) and (Mo2), and so much time on Mag that we cannot finish training on it. This means that SVM.rbf sometimes "suffers from the problem of converging very slowly" even on small databases such as (Pim) if C is set with an inappropriately large number. Although it seems that a soft margin SVM.rbf can approximate a hard margin SVM.rbf very well because many of the training accuracies are 100.0(±0.0)%, actually it just means that $\mathbf{w} \cdot \phi(\mathbf{x}_i) + b \geq 1 - \xi_i > 0$) and $\mathbf{w} \cdot \phi(\mathbf{y}_j) + b \leq -1 + \zeta_j < 0$ hold in the training set with no guarantee for the hard margin inequalities:

TABLE IV
NUMBERS OF CONLITRONS FOR SMA AND SUPPORT VECTORS FOR SVM.rbf

	(Crp)	(Ion)	(Son)	(Mus)	(Bre)	(Mag)	(Par)	(Pim)	(Spe)	(Mo1)	(Mo2)	(Mo3)
SMA	27	56	49	135	26	2933	23	131	37	16	57	22
SVM.rbf	251	95	102	188	53	3753	40	239	73	63	111	42

TABLE V
TESTING CORRECT RATES (%) ON n-DIMENSIONAL UNIT-SPHERE PROBLEMS

n	2	4	6	8	11	14	17	20	25	30	35	40	50	60	70	80
SCA	99.1	96.3	96.3	95.4	93.8	94.5	88.9	89.3	84.2	78.5	74.0	71.7	64.0	60.6	56.7	53.1
SMA	98.5	95.2	91.4	89.1	83.7	80.2	77.4	74.3	69.4	62.7	59.9	57.6	54.6	53.6	52.2	50.2
SVM.rbf	99.4	99.1	98.3	97.5	97.8	98.5	98.3	98.5	98.6	99.1	99.1	98.1	98.9	98.4	99.2	99.0
SVM.lin	66.3	61.5	61.1	59.3	59.9	60.2	61.4	60.0	61.4	59.1	60.9	61.5	60.4	60.6	59.8	59.2

TABLE VI
ACCURACIES (%) AND TIME (S) FOR LCSM-SVM WITH RBF KERNEL AND $C = 10^8$

	(Crp)	(Ion)	(Son)	(Mus)	(Bre)	(Mag)	(Par)	(Pim)	(Spe)	(Mo1)	(Mo2)	(Mo3)
Testing	99.9±0.1	92.9 ± 1.6	83.6 ± 3.6	90.4 ± 2.3	94.8±1.0	–	83.2±3.5	76.1±1.8	74.9	92.8	82.9	91.2
Training	99.9±0.1	100.0±0.0	100.0±0.0	100.0±0.0	96.5±1.2	–	91.6±4.3	79.8±3.1	100.0	100.0	100.0	100.0
Time	778.91	1.38	1.38	9.61	219.25	Too long	232.38	26411	0.57	64.44	2654.9	10.23

TABLE VII
TESTING (TRAINING) ACCURACIES (%) ON MULTICLASS DATABASES

	CDMA	SCA	SMA	SVM.rbf	SVM.lin
iris	96.0 (96.0)	98.7 (98.7)	97.3 (100)	98.7 (97.3)	36.2 (34.9)
glass	89.5 (80.7)	52.4 (66.1)	69.5 (100)	56.2 (52.3)	96.0 (94.7)
wine	88.6 (86.7)	77.3 (83.3)	76.1 (100)	71.6 (72.2)	64.8 (66.7)

$\mathbf{w}\cdot\phi(\mathbf{x}_i)+b \geq 1$ and $\mathbf{w}\cdot\phi(\mathbf{y}_j)+b \leq -1$. Moreover, it can be seen that the accuracies of SVM.rbf become lower on (Ion), (Mus), and (Mo3) in Table II, but they become higher on the other databases except Mag. This means that a better soft margin, well adjusted by γ, may be found for SVM.rbf in the approximation of LCSM-SVM to HM-SVM if the parameter C is inappropriately enlarged at the risk of facing the problem of converging very slowly.

Though CDMA, SCA, and SMA are presented for binary classification in this paper, they are not limited to two-class problems only. Actually, they can solve a p-class problem by combining several binary classifiers if satisfying some separable conditions, and we may consider a number of different methods that have been applied to multiclass SVMs [58]. For simplicity, we would like to just discuss the "maximal margin sequence (MMS)" method, which is described as follows:

1) represent the training set of a p-class problem as $X_1, X_2, \ldots, X_p$ with $k = 1$;
2) let $X = X_i (k \leq i \leq p)$ and $Y = \bigcup_{k \leq j \neq i \leq p} X_j$, and compute the distance d_i between X and Y based on the corresponding linear/convex/common separability;
3) compute $q = \arg\max_{k \leq i \leq p} \{d_i\}$, swap X_k and X_q;
4) let $k = k + 1$, and go to 2 until $k = p$;
5) divide the p-class problem into a MMS of $p - 1$ two-class problems, namely, running CDMA/SCA/SMA to separate X_i and $\bigcup_{i+1 \leq j \leq p} X_j$ for $i = 1, 2, \ldots, p-1$.

Using the above MMS method, we run CDMA, SCA, and SMA on three multiclass databases from the UCI Repository, summarizing the experimental results in Table VII, where "iris" is a three-class 4-D database, "glass" is a six-class 9-D database, and "wine" is a three-class 13-D database. The training and testing sets are obtained by randomly splitting each of them into two halves. Note that in Table VII we do not check the step (7) for CDMA and the step (8) for SCA, so they may produce one or more unstable linear functions in the resultant classifiers, for which training accuracies are possibly less than 100.0% because of violating the strict separability for CDMA or SCA. For comparison, we also report the performance of SVM.rbf and SVM.lin on the three databases in Table VII.

V. CONCLUSION

We have presented some new theorems concerning the concepts of "convex hulls," "SVMs," "convexly separable," "conlitrons," "multiconlitrons," etc., with all of them strictly proven. These theorems greatly advance the state of the art for the geometric theory of SVM, and they help us to establish

a solid geometric theory and a new general framework to design PLCs. Based on them, we proposed a new iterative algorithm—the CDMA, which computes a hard margin non-kernel SVM via the nearest point pair between two convex polytopes. Using the "convexly separable" concept as an extension of the "linearly separable" concept, we also built a general PLC designated as a multiconlitron, which is a union of multiple conlitrons composed of a set of linear functions surrounding a convex region. Moreover, we have theoretically shown that a multiconlitron can separate two arbitrarily complicated nonintersecting classes in $\mathbf{R}^n$, with its special cases—support conlitrons (which can be considered a multiconlitron containing only one conlitron) and support multiconlitrons constructed respectively by SCA and SMA, to solve convexly and commonly separable classification problems without kernels. Additionally, in comparative experiments we have demonstrated that SMA can outperform linear SVM on many of the selected databases and provide similar results to RBF SVM on some of them, while SCA performs better than linear SVM on three out of four applicable databases. Finally, we have indicated that CDMA, SCA, and SMA can be used to solve a multiclass problem by dividing it into a MMS of binary problems if some separable conditions are satisfied.

The most important contribution we have made in this paper is to show that multiconlitron is a general PLC with its special cases—support conlitrons and support multiconlitrons regarded as a non-kernel extension of a HM-SVM. In theory, support conlitrons and support multiconlitrons can solve nonlinear problems without using kernels, providing hyperplanes interpretation with piecewise linear separators. In practice, they are easily constructed and calculated, often performing a little better than a soft margin linear SVM but worse than RBF SVM in most cases. Their main drawback is the sensitiveness to noise, which is affected much by the increase of dimensionality. Although their methodology needs improving to draw more potential, they may open a new research direction as a non-kernel extension of SVM, where many problems remain to be solved, e.g., how to extend their applications to commonly "nonseparable" databases, how to construct equivalent quadratic programming models for them, how to introduce reasonable soft margins for them, how to select an appropriate number of linear functions for them, how to alleviate or avoid the overfitting and noise-sensitive problem for them, how to make a multiconlitron contain as few conlitrons as possible, etc. We will focus future work on these problems forming a new line of research that deserves much attention—piecewise linear learning, the main goal of which is to improve the performance of PLCs in a general framework.

APPENDIX A
PROOFS OF THEOREMS 1–6

Proof of Theorem 1: Because $|\mathbf{w}\cdot(\mathbf{x}-\mathbf{y})| \le ||\mathbf{w}||||\mathbf{x}-\mathbf{y}||$ and

$$D(f\,|X,Y) = \inf\left\{\frac{|\mathbf{w}\cdot(\mathbf{x}-\mathbf{y})|}{\|\mathbf{w}\|}, \mathbf{x}\in CH(X), \mathbf{y}\in CH(Y)\right\},$$
$$0 \le D(f\,|X,Y) \le \inf\{\|\mathbf{x}-\mathbf{y}\|, \mathbf{x}\in CH(X),\ \mathbf{y}\in CH(Y)\}$$
$$= d\left(CH(X), CH(Y)\right).$$ ■

Proof of Theorem 2:

1) Because $CH(X)$ is a bounded closed set, there apparently exists a minimal point $\mathbf{x}^*\in CH(X)$ to $\mathbf{x}_0$ such that $d(\mathbf{x}_0,\mathbf{x}^*) = d_{\min} = \min\{d(\mathbf{x}_0,\mathbf{x}), \mathbf{x}\in CH(X)\}$. Supposing $\mathbf{x}^*$ is not unique, we may have two points $\mathbf{x}_1,\mathbf{x}_2 \in CH(X), \mathbf{x}_1\ne\mathbf{x}_2$ simultaneously satisfying $d(\mathbf{x}_0,\mathbf{x}_1) = d(\mathbf{x}_0,\mathbf{x}_2) = d_{\min}$. Thus
$$d_{\min} = \left\|\mathbf{x}_0 - \frac{\mathbf{x}_1+\mathbf{x}_2}{2}\right\| = \left\|\frac{(\mathbf{x}_0-\mathbf{x}_1)+(\mathbf{x}_0-\mathbf{x}_2)}{2}\right\|$$
$$\le \frac{1}{2}\|\mathbf{x}_0-\mathbf{x}_1\| + \frac{1}{2}\|\mathbf{x}_0-\mathbf{x}_2\| = d_{\min}.$$
Which means $\|(\mathbf{x}_0-\mathbf{x}_1)+(\mathbf{x}_0-\mathbf{x}_2)\| = \|\mathbf{x}_0-\mathbf{x}_1\| + \|\mathbf{x}_0-\mathbf{x}_2\|$ and $\mathbf{x}_0-\mathbf{x}_1 = \lambda(\mathbf{x}_0-\mathbf{x}_2)$.
Because $\|\mathbf{x}_0-\mathbf{x}_1\| = \|\mathbf{x}_0-\mathbf{x}_2\| = d_{\min}$, $\lambda = \pm 1$. If $\lambda = 1$, $\mathbf{x}_1 = \mathbf{x}_2$, contradicting $\mathbf{x}_1\ne\mathbf{x}_2$; if $\lambda = -1$, $\mathbf{x}_0 = (\mathbf{x}_1+\mathbf{x}_2)/2$, contradicting $\mathbf{x}_0\notin CH(X)$. Therefore, the minimal point $\mathbf{x}^*$ is unique.
2) If $\forall\mathbf{x}\in CH(X)$,$(\mathbf{x}-\mathbf{x}^*)\cdot(\mathbf{x}^*-\mathbf{x}_0)\ge 0$, we have
$$(\mathbf{x}_0-\mathbf{x})\cdot(\mathbf{x}_0-\mathbf{x})$$
$$= \left[(\mathbf{x}_0-\mathbf{x}^*)+(\mathbf{x}^*-\mathbf{x})\right]\cdot\left[(\mathbf{x}_0-\mathbf{x}^*)+(\mathbf{x}^*-\mathbf{x})\right]$$
$$= (\mathbf{x}_0-\mathbf{x}^*)\cdot(\mathbf{x}_0-\mathbf{x}^*) + 2(\mathbf{x}-\mathbf{x}^*)\cdot(\mathbf{x}^*-\mathbf{x}_0)$$
$$+(\mathbf{x}^*-\mathbf{x})\cdot(\mathbf{x}^*-\mathbf{x}) \ge (\mathbf{x}_0-\mathbf{x}^*)\cdot(\mathbf{x}_0-\mathbf{x}^*).$$
Thus it holds that $d(\mathbf{x}_0,\mathbf{x}^*) = \min\{d(\mathbf{x}_0,\mathbf{x}), \mathbf{x}\in CH(X)\}$.
3) Suppose $\exists\,\mathbf{x}_1\in CH(X)$,$(\mathbf{x}_1-\mathbf{x}^*)\cdot(\mathbf{x}^*-\mathbf{x}_0) < 0$. We construct a new vector $\mathbf{z} = \mathbf{x}^* + \alpha(\mathbf{x}_1-\mathbf{x}^*) = \alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}^*$, where
$$0 < \alpha = \min\left\{1, \frac{(\mathbf{x}_1-\mathbf{x}^*)\cdot(\mathbf{x}_0-\mathbf{x}^*)}{(\mathbf{x}_1-\mathbf{x}^*)\cdot(\mathbf{x}_1-\mathbf{x}^*)}\right\} \le 1.$$
Obviously, $\mathbf{z}\in CH(X)$. Furthermore, we have
$$(\mathbf{x}_0-\mathbf{z})\cdot(\mathbf{x}_0-\mathbf{z}) = (\mathbf{x}_0-\mathbf{x}^*)\cdot(\mathbf{x}_0-\mathbf{x}^*)$$
$$-2\alpha(\mathbf{x}_0-\mathbf{x}^*)\cdot(\mathbf{x}_1-\mathbf{x}^*) + \alpha^2(\mathbf{x}_1-\mathbf{x}^*)\cdot(\mathbf{x}_1-\mathbf{x}^*),$$
$$(\mathbf{x}_0-\mathbf{z})\cdot(\mathbf{x}_0-\mathbf{z}) < (\mathbf{x}_0-\mathbf{x}^*)\cdot(\mathbf{x}_0-\mathbf{x}^*).$$
This contradicts $d(\mathbf{x}_0,\mathbf{x}^*) = d_{\min}$. Hence, $\forall\mathbf{x}\in CH(X)$, $(\mathbf{x}-\mathbf{x}^*)\cdot(\mathbf{x}^*-\mathbf{x}_0)\ge 0$. ■

Proof of Theorem 3:

1) If $CH(X)\cap CH(Y)=\varnothing$, $\exists\mathbf{x}^*\in CH(X)$, $\exists\mathbf{y}^*\in CH(Y)$ such that
$$d(\mathbf{x}^*,\mathbf{y}^*) = \min\{d(\mathbf{x},\mathbf{y}), \mathbf{x}\in CH(X), \mathbf{y}\in CH(Y)\} > 0.$$
Let $f(\mathbf{x}) = (\mathbf{x}^*-\mathbf{y}^*)\cdot(\mathbf{x}-(\mathbf{x}^*+\mathbf{y}^*)/2)$. We can obtain
$$f(\mathbf{x}) = (\mathbf{x}^*-\mathbf{y}^*)\cdot(\mathbf{x}-\mathbf{x}^*) + \frac{\|\mathbf{x}^*-\mathbf{y}^*\|^2}{2}$$
$$= (\mathbf{x}^*-\mathbf{y}^*)\cdot(\mathbf{x}-\mathbf{y}^*) - \frac{\|\mathbf{x}^*-\mathbf{y}^*\|^2}{2}.$$
According to Theorem 2, $\forall\mathbf{x}\in X$, $f(\mathbf{x}) \ge d^2(\mathbf{x}^*,\mathbf{y}^*)/2 > 0$; $\forall\mathbf{y}\in Y$, $f(\mathbf{y}) \le -d^2(\mathbf{x}^*,\mathbf{y}^*)/2 < 0$. Therefore, X and Y are linearly separable.
2) If X and Y are linearly separable, there exists a linear discriminant function $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x}+b$ such that $\forall\mathbf{x}\in X$, $f(\mathbf{x}) > 0$; $\forall\mathbf{y}\in Y$, $f(\mathbf{y}) < 0$.

Supposing $CH(X) \cap CH(Y) \neq \varnothing$, we will have a contradiction described below: $\exists \mathbf{z} \in CH(X) \cap CH(Y)$ such that

$$\begin{aligned}
\mathbf{z} &= \sum_{1\le i\le |X|} \alpha_i \mathbf{x}_i = \sum_{1\le j\le |Y|} \beta_j \mathbf{y}_j, \sum_{1\le i\le |X|} \alpha_i = 1, \\
&\quad \sum_{1\le j\le |Y|} \beta_j = 1, \alpha_i \ge 0, \beta_j \ge 0, \mathbf{x}_i \in X, \mathbf{y}_j \in Y, \\
f(\mathbf{z}) &= \mathbf{w}\cdot\mathbf{z} + b = \mathbf{w}\cdot \sum_{1\le i\le |X|} \alpha_i \mathbf{x}_i + b \\
&= \sum_{1\le i\le |X|} \alpha_i (\mathbf{w}\cdot\mathbf{x}_i + b) = \sum_{1\le i\le |X|} \alpha_i f(\mathbf{x}_i) > 0, \\
f(\mathbf{z}) &= \mathbf{w}\cdot\mathbf{z} + b = \mathbf{w}\cdot \sum_{1\le j\le |Y|} \beta_j \mathbf{y}_j + b \\
&= \sum_{1\le j\le |Y|} \beta_j (\mathbf{w}\cdot\mathbf{y}_j + b) = \sum_{1\le j\le |Y|} \beta_j f(\mathbf{y}_j) < 0. \qquad \blacksquare
\end{aligned}$$

Proof of Theorem 4: Because $d(\mathbf{x}_1, \mathbf{y}_1) = d(\mathbf{x}_2, \mathbf{y}_2) = d_{\min}$, we have

$$\begin{aligned}
d_{\min} &\le \left\| \frac{\mathbf{x}_1 + \mathbf{x}_2}{2} - \frac{\mathbf{y}_1 + \mathbf{y}_2}{2} \right\| = \left\| \frac{\mathbf{x}_1 - \mathbf{y}_1}{2} + \frac{\mathbf{x}_2 - \mathbf{y}_2}{2} \right\| \\
&\le \frac{1}{2}\|\mathbf{x}_1 - \mathbf{y}_1\| + \frac{1}{2}\|\mathbf{x}_2 - \mathbf{y}_2\| = d_{\min}.
\end{aligned}$$

So, $\|(\mathbf{x}_1 - \mathbf{y}_1) + (\mathbf{x}_2 - \mathbf{y}_2)\| = \|\mathbf{x}_1 - \mathbf{y}_1\| + \|\mathbf{x}_2 - \mathbf{y}_2\|$, $\mathbf{x}_1 - \mathbf{y}_1 = \lambda(\mathbf{x}_2 - \mathbf{y}_2), = 1$.

According to Theorem 3, $CH(X) \cap CH(Y) = \varnothing$, which contradicts $(\mathbf{x}_1 + \mathbf{x}_2)/2 = (\mathbf{y}_1 + \mathbf{y}_2)/2$ obtained when $\lambda = -1$. Thus, $\lambda = 1$, $\mathbf{w} = \mathbf{x}_1 - \mathbf{y}_1 = \mathbf{x}_2 - \mathbf{y}_2$.

According to Theorem 2, $(\mathbf{x}_2 - \mathbf{x}_1)\cdot(\mathbf{x}_1 - \mathbf{y}_1) \ge 0$ and $(\mathbf{x}_1 - \mathbf{x}_2)\cdot(\mathbf{x}_2 - \mathbf{y}_2) \ge 0$. From this, we can derive $(\mathbf{x}_2 - \mathbf{x}_1)\cdot\mathbf{w} = 0$. Similarly, $(\mathbf{y}_2 - \mathbf{y}_1)\cdot\mathbf{w} = 0$. Therefore, $(\mathbf{x}_2 + \mathbf{y}_2 - \mathbf{x}_1 - \mathbf{y}_1)\cdot\mathbf{w} = 0$

$$b = \frac{\left(\|\mathbf{y}_1\|^2 - \|\mathbf{x}_1\|^2\right)}{2} = \frac{\left(\|\mathbf{y}_2\|^2 - \|\mathbf{x}_2\|^2\right)}{2}$$

and $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b$ is a SVM of X and Y because the margin $D(f\,|X, Y)$ reaches the maximum value $d\,(CH(X), CH(Y))$, according to Theorem 1. $\blacksquare$

Proof of Theorem 5: Suppose that $\forall \mathbf{y}_2 \in Y, \mathbf{y}_2 \neq \mathbf{y}_1$, $d(\mathbf{x}_0, \mathbf{y}_2) \ge d(\mathbf{x}_0, \mathbf{y}_1)$. Because it always holds that

$$\begin{aligned}
&d\,(\mathbf{x}_0, \mathbf{y}_1 + \alpha\cdot(\mathbf{y}_2 - \mathbf{y}_1)) \\
&= \sqrt{((\mathbf{x}_0 - \mathbf{y}_1) - \alpha\cdot(\mathbf{y}_2 - \mathbf{y}_1))\cdot((\mathbf{x}_0 - \mathbf{y}_1) - \alpha\cdot(\mathbf{y}_2 - \mathbf{y}_1))} \\
&= \sqrt{(\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) - \frac{\left[(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1)\right]^2}{(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1)}} \\
&\le \sqrt{(\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1)} = d\,(\mathbf{x}_0, \mathbf{y}_1)
\end{aligned}$$

we only need to prove that if $\forall \mathbf{y}_2 \in Y, \mathbf{y}_2 \neq \mathbf{y}_1$, $\alpha \le 0$ or $\alpha \ge 1$, then we are led to a contradiction.

Supposing $\forall \mathbf{y}_2 \in Y, \mathbf{y}_2 \neq \mathbf{y}_1$, $\alpha \le 0$ or $\alpha \ge 1$, we can have $(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) \le 0$ or $(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) \ge (\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1)$.

If $(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) \ge (\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1)$, we can get

$$\begin{aligned}
&(\mathbf{x}_0 - \mathbf{y}_2)\cdot(\mathbf{x}_0 - \mathbf{y}_2) \\
&= (\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) - 2(\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1) \\
&\quad + (\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1) \\
&= (\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) - 2\left[(\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1)\right. \\
&\quad \left. - (\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1)\right] - (\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1) \\
&< (\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1)
\end{aligned}$$

contradicting $d(\mathbf{x}_0, \mathbf{y}_2) \ge d(\mathbf{x}_0, \mathbf{y}_1)$.

Hence, we can only have $\forall \mathbf{y}_2 \in Y, \mathbf{y}_2 \neq \mathbf{y}_1$, $(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) \le 0$ from which we can directly get

$$(\mathbf{z}_i - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) \le 0, \forall \mathbf{z}_i \in Y, 1 \le i \le |Y|\,.$$

Moreover, for any $\mathbf{z} \in CH(Y)$, we can express $\mathbf{z} = \sum_{1\le i\le |Y|} \alpha_i \mathbf{z}_i, \sum_{1\le i\le |Y|} \alpha_i = 1, \forall \alpha_i \ge 0$, which leads to the following inequality:

$$\begin{aligned}
&(\mathbf{x}_0 - \mathbf{z})\cdot(\mathbf{x}_0 - \mathbf{z}) \\
&= \left[\mathbf{x}_0 - \sum_{1\le i\le |Y|} \alpha_i \mathbf{z}_i\right]\cdot\left[\mathbf{x}_0 - \sum_{1\le i\le |Y|} \alpha_i \mathbf{z}_i\right] \\
&= \left[(\mathbf{x}_0 - \mathbf{y}_1) - \sum_{1\le i\le |Y|} \alpha_i\,(\mathbf{z}_i - \mathbf{y}_1)\right] \\
&\quad \cdot \left[(\mathbf{x}_0 - \mathbf{y}_1) - \sum_{1\le i\le |Y|} \alpha_i\,(\mathbf{z}_i - \mathbf{y}_1)\right] \\
&= (\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) - 2\sum_{1\le i\le |Y|} \alpha_i\,(\mathbf{z}_i - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1) \\
&\quad + \left[\sum_{1\le i\le |Y|} \alpha_i\,(\mathbf{z}_i - \mathbf{y}_1)\right]\cdot\left[\sum_{1\le i\le |Y|} \alpha_i\,(\mathbf{z}_i - \mathbf{y}_1)\right] \\
&\ge (\mathbf{x}_0 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1).
\end{aligned}$$

Obviously, the above inequality means that $\mathbf{y}_1 \in CH(Y)$ is the minimal point of $\mathbf{x}_0$. Thus, if $\mathbf{x}_0 \notin CH(Y)$, we have $\mathbf{y}^* = \mathbf{y}_1$ according to Theorem 2, contradicting $\mathbf{y}_1 \neq \mathbf{y}^*$; if $\mathbf{x}_0 \in CH(Y)$, we have $\mathbf{y}^* = \mathbf{y}_1 = \mathbf{x}_0$, also contradicting $\mathbf{y}_1 \neq \mathbf{y}^*$. Therefore, $\exists \mathbf{y}_2 \in Y, \mathbf{y}_2 \neq \mathbf{y}_1$ such that $0 < \alpha = ((\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{x}_0 - \mathbf{y}_1))/(\mathbf{y}_2 - \mathbf{y}_1)\cdot(\mathbf{y}_2 - \mathbf{y}_1) < 1$. $\blacksquare$

Proof of Theorem 6: In the case that $\mathbf{x}^* \in CH(Y)$ or $\mathbf{y}^* \in CH(X)$, $d(\mathbf{x}^*, \mathbf{y}^*) = 0$, meaning the theorem obviously holds. If $\mathbf{x}^* \notin CH(Y)$ and $\mathbf{y}^* \notin CH(X)$, using Theorem 2, we can get

$$\begin{aligned}
&\forall \mathbf{x} \in CH(X), (\mathbf{x} - \mathbf{x}^*)\cdot(\mathbf{x}^* - \mathbf{y}^*) \ge 0; \\
&\forall \mathbf{y} \in CH(Y), (\mathbf{y} - \mathbf{y}^*)\cdot(\mathbf{y}^* - \mathbf{x}^*) \ge 0.
\end{aligned}$$

Hence

$$\begin{aligned}
\|\mathbf{x} - \mathbf{y}\|^2 &= (\mathbf{x} - \mathbf{y})\cdot(\mathbf{x} - \mathbf{y}) \\
&= \left[(\mathbf{x}^* - \mathbf{y}^*) + (\mathbf{x} - \mathbf{x}^* + \mathbf{y}^* - \mathbf{y})\right] \\
&\quad \cdot\left[(\mathbf{x}^* - \mathbf{y}^*) + (\mathbf{x} - \mathbf{x}^* + \mathbf{y}^* - \mathbf{y})\right] \\
&= (\mathbf{x}^* - \mathbf{y}^*)\cdot(\mathbf{x}^* - \mathbf{y}^*) + (\mathbf{x} - \mathbf{x}^* + \mathbf{y}^* - \mathbf{y}) \\
&\quad \cdot(\mathbf{x} - \mathbf{x}^* + \mathbf{y}^* - \mathbf{y}) \\
&\quad + 2(\mathbf{x} - \mathbf{x}^*)\cdot(\mathbf{x}^* - \mathbf{y}^*) + 2(\mathbf{x}^* - \mathbf{y}^*)\cdot(\mathbf{y}^* - \mathbf{y}) \\
&\ge (\mathbf{x}^* - \mathbf{y}^*)\cdot(\mathbf{x}^* - \mathbf{y}^*) = \left\|\mathbf{x}^* - \mathbf{y}^*\right\|^2.
\end{aligned}$$

Therefore, $d(\mathbf{x}^*, \mathbf{y}^*) = \min\{d(\mathbf{x}, \mathbf{y}), \mathbf{x} \in CH(X), \mathbf{y} \in CH(Y)\}$. ■

VI. Acknowledgment

The authors would like to thank M. Jonikas for polishing the language, anonymous reviewers for their comments, and all those who helped us to improve this paper.

References

[1] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[2] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. New York: Academic, 2006.
[3] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.
[4] C. Cortes and V. N. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[5] I. El-Naqa, Y. Yang, M. Wernik, N. Galatsanos, and R. Nishikawa, "A support vector machine approach for detection of microcalcifications," *IEEE Trans. Med. Imag.*, vol. 21, no. 12, pp. 1552–1563, Dec. 2002.
[6] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. 10th Eur. Conf. Mach. Learn.*, Chemnitz, Germany, 1998, pp. 137–142.
[7] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Juan, Puerto Rico, Jun. 1997, pp. 130–136.
[8] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, Jr., and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," in *Proc. Nat. Acad. Sci.*, vol. 97. Jan. 2000, pp. 262–267.
[9] A. Navia-Vasquez, F. Perez-Cruz, and A. Artes-Rodriguez, "Weighted least squares training of support vector classifiers leading to compact and adaptive schemes," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 1047–1059, Sep. 2001.
[10] D. J. Sebald and J. A. Buklew, "Support vector machine techniques for nonlinear equalization," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3217–3226, Nov. 2000.
[11] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Machines*, B. Schölkopf, C. J. C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1998, pp. 185–208.
[12] S. S. Keerthi, S. K. Shevade, C. Bhattachayya, and K. R. K. Murth, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Comput.*, vol. 13, no. 3, pp. 637–649, Mar. 2001.
[13] J.-X. Dong, A. Krzyzak, and C. Y. Suen, "Fast SVM training algorithm with decomposition on very large data sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 603–618, Apr. 2005.
[14] A. Bordes, L. Bottou, and P. Gallinari, "SGD-QN: Careful quasi-Newton stochastic gradient descent," *J. Mach. Learn. Res.*, vol. 10, pp. 1737–1754, Jul. 2009.
[15] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvalis, OR, 2007, pp. 807–814.
[16] O. Chapelle, "Training a support vector machine in the primal," *Neural Comput.*, vol. 19, no. 5, pp. 1155–1178, Mar. 2007.
[17] G. Fung and O. L. Mangasarian, "Proximal support vector machine classifiers," in *Proc. Knowl. Discovery Data Mining*, San Francisco, CA, Aug. 2001, pp. 77–86.
[18] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.
[19] R. J. 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.
[20] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.
[21] K.-M. Lin and C.-J. Lin, "A study on reduced support vector machines," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1449–1459, Nov. 2003.
[22] O. L. Mangasarian and D. R. Musicant, "Lagrangian support vector machines," *J. Mach. Learn. Res.*, vol. 1, no. 3, pp. 161–177, 2001.
[23] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, no. 3, pp. 211–244, Jun. 2001.
[24] H. Chen, P. Tino, and X. Yao, "Probabilistic classification vector machines," *IEEE Trans. Neural Netw.*, vol. 20, no. 6, pp. 901–914, Jun. 2009.
[25] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–202, Mar. 2001.
[26] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[27] S. Sonnenburg, G. Rätsch, and C. Schäfer, "A general and efficient multiple kernel learning algorithm," in *Neural Information Processing Systems*, vol. 15. Cambridge, MA: MIT Press, 2005, pp. 1273–1280.
[28] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Jul. 2006.
[29] Z. Wang, S. Chen, and T. Sun, "MultiK-MHKS: A novel multiple kernel learning algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 348–353, Feb. 2008.
[30] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "A fast iterative nearest point algorithm for support vector machine classifier design," *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 124–136, Jan. 2000.
[31] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," in *Advances in Neural Information Processing Systems*, vol. 12, S. A. Solla, T. K. Leen, and K. R. Müller, Eds. Cambridge, MA: MIT Press, 2000, pp. 498–504.
[32] V. Franc and V. Hlaváč, "An iterative algorithm learning the maximal margin classifier," *Pattern Recognit.*, vol. 36, no. 9, pp. 1985–1996, Sep. 2003.
[33] M. E. Mavroforakis and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 671–682, May 2006.
[34] P. Williams, S. Li, J. Feng, and S. Wu, "A geometrical method to improve performance of the support vector machine," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 942–947, May 2007.
[35] M. E. Mavroforakis, M. Sdralis, and S. Theodoridis, "A geometric nearest point algorithm for the efficient solution of the SVM classification task," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1545–1549, Sep. 2007.
[36] M. Doumpos, C. Zopounidis, and V. Golfinopoulou, "Additive support vector machines for pattern classification," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 37, no. 3, pp. 540–550, Jun. 2007.
[37] O. Pujol and D. Masip, "Geometry-based ensembles: Toward a structural characterization of the classification boundary," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1140–1146, Jun. 2009.
[38] J. Sklansky and L. Michelotti, "Locally trained piecewise linear classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, no. 2, pp. 101–111, Mar. 1980.
[39] Y. Park and J. Sklansky, "Automated design of multiple-class piecewise linear classifiers," *J. Classification*, vol. 6, no. 1, pp. 195–222, Dec. 1989.
[40] H. Tenmoto, M. Kudo, and M. Shimbo, "Piecewise linear classifiers with an appropriate number of hyperplances," *Pattern Recognit.*, vol. 31, no. 11, pp. 1627–1634, Nov. 1998.
[41] O. L. Mangasarian, R. Setono, and W. H. Wolberg, "Pattern recognition via linear programming: Theory and applications to medical diagnosis," in *Large-Scale Numerical Optimization*, T. F. Coleman and Y. Li, Eds. Philadelphia, PA: SIAM, 1990, pp. 22–31.
[42] G. T. Herman and K. T. D. Yeung, "On piecewise-linear classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 782–786, Jul. 1992.
[43] B.-B. Chai, T. Huang, X. Zhuang, Y. Zhao, and J. Sklansky, "Piecewise linear classifiers using binary tree structure and genetic algorithm," *Pattern Recognit.*, vol. 29, no. 11, pp. 1905–1917, Nov. 1996.
[44] A. Kostin, "A simple and fast multi-class piecewise linear pattern classifier," *Pattern Recognit.*, vol. 39, no. 11, pp. 1949–1962, Nov. 2006.
[45] M. Kudo, I. Takigawa, and A. Nakamura, "Classification by reflective convex hulls," in *Proc. 19th Int. Conf. Pattern Recognit.*, Tampa, FL, Dec. 2008, pp. 1–4.
[46] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[47] K. P. Bennett and E. J. Bredensteiner, "Geometry in learning," in *Geometry at Work*, C. Gorini, E. Hart, W. Meyer, and T. Phillips, Eds. Washington D.C.: MAA, 1998.
[48] N. K. Sancheti and S. S. Keerthi, "Computation of certain measures of proximity between convex polytopes: A complexity viewpoint," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3. Nice, France, May 1992, pp. 2508–2513.
[49] D. Naiyang and T. Yingjie, *New Methods in Data Mining: Support Vector Machines*. Duluth, GA: Science Press, 2004.

[50] C. P. Bennett and E. J. Bredensteiner, "Duality and geometry in SVM classifiers," in *Proc. 17th Int. Conf. Mach. Learn.*, San Francisco, CA, 2000, pp. 57–64.
[51] O. L. Mangasarian, "Polyhedral boundary projection," *SIAM J. Optim.*, vol. 9, no. 4, pp. 1128–1134, Apr. 1999.
[52] *UCI Repository of Machine Learning Databases* [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html
[53] C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines* [Online]. Available: http://www.csie.ntu.edu.tw/∼cjlin/libsvm
[54] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. *A Practical Guide to Support Vector Classification* [Online]. Available: http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf
[55] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[56] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training SVM," *J. Mach. Learn. Res.*, vol. 6, pp. 1889–1918, Dec. 2005.
[57] T. Joachins, "Training linear SVMs in linear time," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Philadelphia, PA, 2006, pp. 217–226.
[58] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

Li Yujian was born in Guilin, China, on October 10, 1968. He received the B.S. degree in mathematics from the Huazhong University of Science and Technology, Hubei, China, in 1990, the M.S. degree in mathematics from the Institute of Mathematics, Chinese Academy of Sciences, Beijing, China, in 1993, and the Ph.D. degree in semiconductor devices and microelectronics from the Institute of Semiconductors, Chinese Academy of Sciences, in 1999.

He was with the Institute of Biophysics, Chinese Academy of Sciences, from 1993 to 1996. He was a Post-Doctoral Fellow at the Beijing University of Posts and Telecommunications, Beijing, from 1999 to 2001. He was an Associate Professor at Beijing University of Technology, Beijing, since June 2001, and has been a Professor since December 2007. He has published several papers in leading journals such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition, Pattern Recognition Letters*, and the IEEE TRANSACTIONS ON NEURAL NETWORKS. To this paper, his most important contributions include key ideas, basic concepts, new theorems, theoretical proofs, learning algorithms, some experiments, and a large part of the text. His current research interests include pattern analysis and machine intelligence, especially piecewise linear learning that he has pioneered as a new direction.

Liu Bo received the M.S. degree in theoretical physics in 2000 and the Ph.D. degree in computer science in 2005, both from Beijing University of Technology, Beijing, China.

He is currently an Associate Professor in the College of Computer Science and Technology, Beijing University of Technology. His current research interests include machine learning and computer vision.

Yang Xinwu received the M.S. degree in condensed matter physics in 1999 and the Ph.D. degree in computer science in 2003, both from Beijing University of Technology, Beijing, China.

He is currently an Associate Professor in the College of Computer Science and Technology, Beijing University of Technology. His current research interests include genetic algorithms, data mining, biometrics, and inductive logic programming.

Fu Yaozong is currently pursuing the M.S. degree in the College of Computer Science and Technology, Beijing University of Technology, Beijing, China.

His current research interests include pattern recognition and machine learning.

Li Houjun is currently pursuing the M.S. degree in the College of Computer Science and Technology, Beijing University of Technology, Beijing, China.

His current research interests include pattern recognition and machine learning.