set $\mathcal{D} = \{\bar{s}_n, y_2, \ldots, y_n, \tilde{\psi}_1, \ldots, \tilde{\psi}_n, \tilde{W}_1, \ldots, \tilde{W}_n \,|\, \sum_{i=1}^{n}(s_i^2 + \tilde{\psi}_i^2 + \|\tilde{W}_i\|^2) + \sum_{i=1}^{n-1} y_{i+1}^2 < (C)/(\gamma)\}$. In addition, the compact set $\mathcal{D}$ can be kept arbitrarily small by adjusting $\lambda_{i,1}$, $\lambda_{i,2}$, $\sigma_{i,1}$, $\sigma_{i,2}$, k_i, and ζ_j $(i = 1, \ldots, n, j = 2, \ldots, n)$.

Case iii): $S_p \in \mathcal{K}_{S_p}$ and $S_q \notin \mathcal{K}_{S_q}$ for $p \neq q$. From the proof of Cases i) and ii), we can easily prove this case. Define subsystems consisting of $S_p \in \mathcal{K}_{S_p}$ and $S_q \notin \mathcal{K}_{S_q}$ as Σ_p and Σ_q, respectively. In addition, Lyapunov candidate functions for Σ_p and Σ_q are defined as V_p and V_q, respectively. Thus, the Lyapunov candidate function (17) can be rewritten by $V = V_p + V_q$. The boundedness of all signals of the subsystem Σ_p is proved using V_p and the proof of Case i). From the proof of Case ii), using V_q, all signals of the subsystem Σ_q are semiglobally uniformly bounded. Therefore, we can show that all closed-loop signals of the total system are semiglobally uniformly bounded.

Accordingly, from all three cases, all signals of the closed-loop system are semiglobally uniformly bounded. This completes the proof of Theorem 1.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. Hoboken, NJ: Wiley, 1995.

[2] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controller for feedback linearizable system," *IEEE Trans. Autom. Control*, vol. 36, no. 11, pp. 1241–1253, Nov. 1991.

[3] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, "Dynamic surface control for a class of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 45, no. 10, pp. 1893–1899, Oct. 2000.

[4] P. P. Yip and J. K. Hedrick, "Adaptive dynamic surface control: A simplified algorithm for adaptive backstepping control of nonlinear systems," *Int. J. Control*, vol. 71, no. 5, pp. 959–979, Nov. 1998.

[5] D. Wang and J. Huang, "Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 195–202, Jan. 2005.

[6] S. S. Ge, F. Hong, and T. H. Lee, "Adaptive neural network control of nonlinear systems with unknown time delays," *IEEE Trans. Autom. Control*, vol. 48, no. 11, pp. 2004–2010, Nov. 2003.

[7] S. S. Ge, F. Hong, and T. H. Lee, "Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 34, no. 1, pp. 499–516, Feb. 2004.

[8] F. Hong, S. S. Ge, and T. H. Lee, "Practical adaptive neural control of nonlinear systems with unknown time delays," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 35, no. 4, pp. 849–854, Aug. 2005.

[9] D. W. C. Ho, J. Li, and Y. Niu, "Adaptive neural control for a class of nonlinearly parametric time-delay systems," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 625–635, May 2005.

[10] S. S. Ge and K. P. Tee, "Approximation-based control of nonlinear MIMO time-delay systems," *Automatica*, vol. 43, no. 1, pp. 31–43, 2007.

[11] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 889–898, Nov. 1992.

[12] J. Zhang, G. Walter, Y. Miao, and W. N. W. Lee, "Wavelet neural networks for function learning," *IEEE Trans. Signal Process.*, vol. 43, no. 6, pp. 1485–1497, Jun. 1995.

[13] Y. Oussar, I. Rivals, L. Personnaz, and G. Dreyfus, "Training wavelet networks for nonlinear dynamic input-output modeling," *Neurocomputing*, vol. 20, no. 1–3, pp. 173–188, Aug. 1998.

[14] S. J. Yoo, J. B. Park, and Y. H. Choi, "Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural network via adaptive learning rates," *Inf. Sci.*, vol. 177, no. 15, pp. 3074–3098, 2007.

[15] S. K. Nguang, "Robust stabilization of a class of time-delay nonlinear systems," *IEEE Trans. Autom. Control*, vol. 45, no. 4, pp. 756–762, Apr. 2000.

[16] S.-L. Niculescu, *Delay Effects on Stability: A Robust Control Approach*. New York: Springer-Verlag, 2001.

[17] M. S. Mahmoud, *Robust Control and Filtering for Time-Delay Systems*. New York: Marcel Dekker, 2000.

[18] W. Lin and C. J. Qian, "Adaptive control of nonlinearly parameterized systems: The smooth feedback case," *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1249–1266, Aug. 2002.

[19] F. J. Lin, T. S. Lee, and C. H. Lin, "Robust H_∞ controller design with recurrent neural network for linear synchronous motor drive," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 456–470, Jun. 2003.

[20] M. M. Polycarpou and M. J. Mears, "Stable adaptive tracking of uncertain systems using nonlinearly parameterized on-line approximators," *Int. J. Control*, vol. 70, no. 3, pp. 363–384, 1998.

[21] P. A. Ioannou and P. V. Kokotovic, *Adaptive Systems With Reduced Models*. New York: Springer-Verlag, 1983.

A Novel Geometric Approach to Binary Classification Based on Scaled Convex Hulls

Zhenbing Liu, J. G. Liu, Chao Pan, and Guoyou Wang

***Abstract*—Geometric methods are very intuitive and provide a theoretical foundation to many optimization problems in the fields of pattern recognition and machine learning. In this brief, the notion of scaled convex hull (SCH) is defined and a set of theoretical results are exploited to support it. These results allow the existing nearest point algorithms to be directly applied to solve both the separable and nonseparable classification problems successfully and efficiently. Then, the popular S-K algorithm has been presented to solve the nonseparable problems in the context of the SCH framework. The theoretical analysis and some experiments show that the proposed method may achieve better performance than the state-of-the-art methods in terms of the number of kernel evaluations and the execution time.**

***Index Terms*—Nearest point problems (NPPs), reduced convex hulls (RCHs), scaled convex hulls (SCHs), S-K algorithm, support vector machines (SVMs).**

I. INTRODUCTION

Geometry provides an intuitive and theoretical framework for the solution of many problems in the pattern recognition and machine learning fields. Support vector machine (SVM) classification is a typical optimization task that has achieved excellent performance because of the sound theoretical foundation based on the statistical learning theory and the clear intuitive geometric interpretation [1], [2].

The geometric properties of SVMs in the feature space have been pointed out by Bennett and Bredensteiner early through the notion of convex hull for the separable case and the notion of reduced convex hull (RCH) for the nonseparable case [3] and [4]. Some important geometric ideas have also been investigated by Crisp and Burges [5]. All these enlighten us to propose some efficient geometric algorithms to

Manuscript received August 07, 2008; revised April 13, 2009; accepted April 24, 2009. First published May 27, 2009; current version published July 09, 2009. This work was supported in part by the National Science Foundation of China under Grant 60672060 and by the Research Fund for the Doctoral Program of Higher Education of China.

The authors are with the State Key Lab for Multispectral Information Processing Technologies, Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Hubei 430074, China (e-mail: liuzb0618@hotmail.com; jgliu@ieee.org).

Digital Object Identifier 10.1109/TNN.2009.2022399

the SVM classification problems [6], [7]. However, almost all the existing geometric algorithms (with the exception of [8]–[11]) are suitable only for solving directly the separable problems. These geometric algorithms have been generalized to solve indirectly the nonseparable case through the use of the technique proposed by Friess and Harisson [12]. However, this technique is equivalent to a quadratic penalty factor, increasing the complexity due to the artificial expansion of the dimension of the input space. In [8]–[10], the authors derived efficient geometric algorithms for the nonseparable problems based on the RCH. These algorithms transformed the nonseparable problems to the nearest point problems (NPPs) between RCHs and reduced the complexity from combinatorial to quadratic.

Inspired by the notion of RCH, a notion of scaled convex hull (SCH) is introduced in this brief, through which the nonseparable classification problems can be transformed to the separable ones. The development and proof of several propositions about the SCH allows the use of the popular S-K algorithm (initially proposed for solving the NPP between convex hulls [7]) to solve the nonseparable classification problems. The derived geometric algorithm, involving the SCH, has almost the same computational cost as the original S-K algorithm.

The rest of this brief is organized as follows. First, some preliminary material on the SVM paradigm and the RCH has been presented in Section II. Then, the notion of SCH is defined and the connection to the nonseparable SVM classification problems is presented in Section III. The S-K algorithm is then rewritten in the context of the SCH framework in Section IV. Finally, the results of the application of the new method solving certain classification tasks are presented in Section V.

II. SVM and Reduced Convex Hulls

Given the training data, the SVM training algorithms find the optimal separating hyperplane between two classes of training samples: $f(x)=w^Tx+t$, where w is the weight vector and t is the bias. The optimal separating hyperplane maximizes the margin, i.e., $2/\|w\|^2$ (or, alternatively, minimizes $\|w\|^2$) [2], to obtain good generalization ability. This classification task, expressed in its dual form, is equivalent to finding the pair of nearest points between the convex hulls (each is generated by the training patterns of each class), and the maximum margin (optimal separating) hyperplane 1) bisects, and 2) is normal to the line segment joining these two nearest points [3]. A convex hull generated by training patterns of one class $X=\{x_i, x_i\in R^d, i=1,2,\ldots,k\}$ is defined as

$$\mathrm{conv}X=\left\{w: w=\sum_{i=1}^{k}a_i x_i, 0\le a_i, \sum_{i=1}^{k}a_i=1, x_i\in X\right\}. \quad (1)$$

For the nonseparable problems, i.e., the convex hulls of the patterns in the feature space are overlapping, the framework of RCH is introduced to transform them to the separable ones [9].

Given the patterns of one class $X=\{x_i, x_i\in R^d, i=1,2,\ldots,k\}$, the RCH of the set X (denoted by $R(X,\mu)$), with the additional constraint that each coefficient a_i is upperbounded by a nonnegative number $\mu<1$, is defined as follows (see [3]):

$$R(X,\mu)=\left\{w: w=\sum_{i=1}^{k}a_i x_i, \sum_{i=1}^{k}a_i=1, x_i\in X, 0\le a_i\le\mu\right\}. \quad (2)$$

It can be seen that the smaller the μ, the smaller size of the RCH, in Fig. 1(a). So by a suitable selection of the reduction factor μ, the initially overlapping convex hulls can be reduced to become separable. It is well known that for the nonseparable case finding the maximum soft margin between two classes is equivalent to finding the pair of nearest points between two RCHs by a suitable selection of the reduction factor [5].

The pair of nearest points between convex hulls depend directly on their extreme points which are some points of the set X for the separable case. However, for the nonseparable case, each extreme point of the RCHs is a reduced convex combination of the original points, thus a direct employment of a nearest point algorithm is impractical. In [9], the authors exploited some results in the RCH and applied them to the S-K algorithm leading to an elegant and efficient solution to the general SVM classification tasks. It was shown that: 1) each extreme point of the RCH is a combination of a specific number $l=\lceil 1/\mu\rceil$ of the original points, with specific coefficients that are analytically computed, and 2) it is not the extreme RCH points themselves that are needed, but rather their projections onto a specific direction. However, to compute the minimum projection, the extra cost of sorting the projections of all the original points in the ascending order and combining appropriately the l smallest of them is required. It can be seen that the complexity increases as μ gets smaller. Besides, the number of extreme points and the shape of the RCH vary with the change of the parameter μ. To address these difficulties, a new method for reducing the convex hull is proposed in the next section.

III. Scaled Convex Hulls

Definition: The SCH of the set $X=\{x_i, x_i\in R^d, i=1,2,\ldots,k\}$ with the nonnegative reduction factor $\lambda\le 1$, denoted as $S(X,\lambda)$, is defined as

$$S(X,\lambda)=\left\{w: w=\lambda\sum_{i=1}^{k}a_i x_i+(1-\lambda)m, \sum_{i=1}^{k}a_i=1, 0\le a_i\le 1\right\}. \quad (3)$$

It can also be rewritten as

$$S(X,\lambda)=\left\{w: w=\sum_{i=1}^{k}a_i(\lambda x_i+(1-\lambda)m), \sum_{i=1}^{k}a_i=1, 0\le a_i\le 1\right\} \quad (4)$$

where $m=(1/k)\sum_{i=1}^{k}x_i$, the mean value of all original points, is called the centroid point of $\mathrm{conv}X$.

1) The Geometric Interpretation of the SCH: For a given λ, each point $\lambda\sum_i a_i x_i+(1-\lambda)m$ of $S(X,\lambda)$ is the convex combination of the centroid m and the point $\sum_{i=1}^{k}a_i x_i$ of the original convex hull $\mathrm{conv}X$, i.e., the point $\lambda\sum_i a_i x_i+(1-\lambda)m$ of the SCH lies on the line segment connecting $\sum_{i=1}^{k}a_i x_i$ and the centroid m [Fig. 1(b)]. In fact, the ratio of the distance between $\lambda\sum_i a_i x_i+(1-\lambda)m$ and the centroid m to the distance between $\sum_{i=1}^{k}a_i x_i$ and the centroid m is the constant λ. So the shape of the SCH $S(X,\lambda)$ is the same as that of the original convex hull $\mathrm{conv}X$. (This is why we call it SCH.) It seems that $S(X,\lambda)$ is obtained by "reducing" $\mathrm{conv}X$ towards the centroid m by λ which controls the size of the SCH. Furthermore, the reduction factor λ can be set different for each class, reflecting the importance of each class.

For convenience, we denote the "reduced" point $\lambda x_i+(1-\lambda)m$ as x_i' and $X'=\{x_i' : i=1,2,\ldots,k\}$. Then, the SCH can be rewritten as

$$S(X,\lambda)=\left\{w: w=\sum_{i=1}^{k}a_i x_i', \sum_{i=1}^{k}a_i=1, 0\le a_i\le 1, x_i'\in X'\right\}=\mathrm{conv}X'. \quad (5)$$

From (5), SCH $S(X,\lambda)$ can be seen as a convex hull generated by the reduced points x_i''s which are the candidate extreme points of the SCH, so it can also be denoted as $\mathrm{conv}X'$. Thus, the candidate extreme

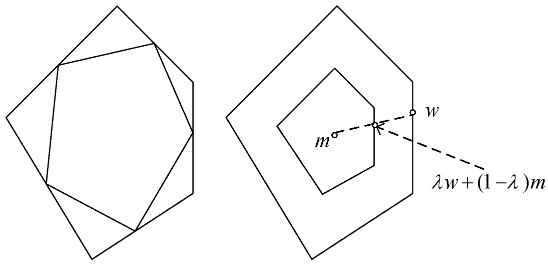

Fig. 1. Geometric interpretation of the RCH and the SCH with respect to the same sample set: (a) RCH (the smaller one) with $\mu = 1/2$; and (b) SCH (the smaller one) with $\lambda = 1/2$. While each (extreme) point of the RCH is the reduced convex combination of $l = \lceil 1/\mu \rceil$ (distinct) ones of the original training set, each (extreme) point of SCH is the convex combination of the centroid and a corresponding (extreme) point of the original convex hull by the reduction factor λ.

point set of $S(X,\lambda)$ is $X' = \{x_i' = \lambda x_i + (1-\lambda)m, i = 1, 2, \ldots, k\}$, having the same number of elements as the original set X when $\lambda \neq 0$.

In this way, the initially overlapping convex hulls can be reduced to become separable by a suitable selection of λ. Once separable, we can find the maximum margin classifiers between the two SCHs through the use of the nearest point algorithms. This viewpoint is the same as the RCH framework in [9] and [11], so it can be seen as a variation of SVMs. But being different from the RCH, the SCH has the same shape and number of candidate extreme points as the original convex hull, resulting in the easier research of the nearest point pair between the SCHs. The comparison of an RCH and an SCH is illustrated in Fig. 1.

Next, we will prove some propositions useful to the SCH notion and form the basis for the development of the novel algorithm proposed in this brief.

Proposition 1: When $\lambda = 1$, SCH $S(X,\lambda)$ is the original convex hull; and when $\lambda = 0$, it becomes the centroid.

Proof: Substituting $\lambda = 1$ and $\lambda = 0$ into (3), respectively, we can get the result.

Proposition 2: The SCH and the original convex hull have the same centroid.

Proof: We know that the SCH $S(X,\lambda)$ is generated by the set $X' = \{x_i' = \lambda x_i + (1-\lambda)m, i = 1, 2, \ldots, k\}$. So the centroid of the SCH is

$$\begin{aligned}\frac{1}{k}\sum_{i=1}^{k} x_i' &= \frac{1}{k}\sum_{i=1}^{k}(\lambda x_i + (1-\lambda)m)\\ &= \frac{1}{k}\sum_{i=1}^{k}\lambda x_i - \lambda m + m\\ &= \frac{1}{k}\sum_{i=1}^{k}\lambda x_i - \lambda\frac{1}{k}\sum_{i=1}^{k}x_i + m\\ &= m \end{aligned} \tag{6}$$

i.e., the SCH and the original convex hull share the same centroid.

Proposition 3: The SCH $S(X,\lambda)$ can be rewritten as

$$S(X,\lambda) = \left\{ w : w = \sum_{i=1}^{k} b_i x_i, \sum_{i=1}^{k} b_i = 1, \frac{1-\lambda}{k} \leq b_i \leq \lambda + \frac{1-\lambda}{k}, x_i \in X \right\}. \tag{7}$$

Proof: By the definition of SCH, each point of $S(X,\lambda)$ can be written as

$$\begin{aligned} w &= \lambda\sum_{i=1}^{k} a_i x_i + (1-\lambda)m\\ &= \lambda\sum_{i=1}^{k} a_i x_i + (1-\lambda)\frac{1}{k}\sum_{i=1}^{k} x_i\\ &= \sum_{i=1}^{k}\left(\lambda a_i + \frac{(1-\lambda)}{k}\right)x_i. \end{aligned} \tag{8}$$

Let $b_i = \lambda a_i + (1-\lambda)/k$, then we get the result.

From the above proposition, it can be seen that when k tends to infinity, the SCH converges to the RCH if $\lambda = \mu$, hence the classifier based on the SCH converges to that based on the RCH.

Suppose $\mathcal{T} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is the training set for a binary classification problem where x_i is the training pattern belonging to one of two classes and $y_i \in \{-1, 1\}$ is its corresponding class label $(i = 1, \ldots, n)$. In the following proposition, we will give a sufficient condition that makes the two SCHs generated by the patterns of each class separable. Before giving the condition, we first define some notations that have the same meaning in the following sections

$$\begin{aligned} I^+ &= \{i : y_i = 1\}, I^- = \{i : y_i = -1\}\\ X^+ &= \{x_i : i \in I^+\}, X^- = \{x_i : i \in I^-\}\\ X^{+\prime} &= \left\{ x_i' = \lambda x_i + (1-\lambda)m^+ : i \in I^+, \; m^+ = \frac{1}{n^+}\sum_{i\in I^+} x_i, n^+ = \left|I^+\right| \right\}\\ X^{-\prime} &= \left\{ x_i' = \lambda x_i + (1-\lambda)m^- : i \in I^-, \; m^- = \frac{1}{n^-}\sum_{i\in I^-} x_i, n^- = \left|I^-\right| \right\}\\ S(X^+,\lambda) &= \left\{ \sum_{i\in I^+} a_i x_i' : x_i' \in X^{+\prime}, 0 \leq a_i \leq 1, \sum_{i\in I^+} a_i = 1 \right\}\\ &= \mathrm{conv}(X^{+\prime})\\ S(X^-,\lambda) &= \left\{ \sum_{i\in I^-} a_i x_i' : x_i' \in X^{-\prime}, 0 \leq a_i \leq 1, \sum_{i\in I^-} a_i = 1 \right\}\\ &= \mathrm{conv}(X^{-\prime})\\ r &= \left\|m^+ - m^-\right\|, r^+ = \max_{i\in I^+}\left\|x_i - m^+\right\|\\ r^- &= \max_{i\in I^-}\left\|x_i - m^-\right\|\\ r_s^+ &= \max_{i\in I^+}\left\|x_i' - m^+\right\|, r_s^- = \max_{i\in I^-}\left\|x_i' - m^-\right\|. \end{aligned} \tag{9}$$

Proposition 4: Convex hulls $S(X^+,\lambda)$ and $S(X^-,\lambda)$ are separable if $\lambda r^+ + \lambda r^- \leq r$.

Proof: By construction, convex hull $S(X^+,\lambda)$ is contained within the r_s^+-radius ball centered in m^+ [see Proposition 2 and the definition of r_s^+ in (9)], and $S(X^-,\lambda)$ is contained within the r_s^--radius ball centered in m^- in Fig. 2. Two balls are nonoverlapping if the Euclidean distance between their means is larger than the sum of their radius.

We can compute

$$\begin{aligned} r_s^+ &= \max_{i\in I^+}\|x_i' - m^+\| = \max_{i\in I^+}\|\lambda x_i + (1-\lambda)m^+ - m^+\|\\ &= \lambda\max_{i\in I^+}\|x_i - m^+\| \end{aligned} \tag{10}$$

and similarly for r_s^-, which completes our proof.

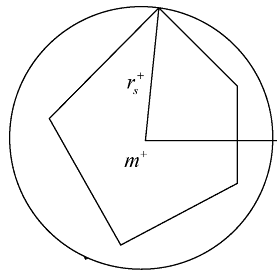

Fig. 2. Two balls including two SCHs, respectively: two balls are separable if $r_s^+ + r_s^- \leq r$.

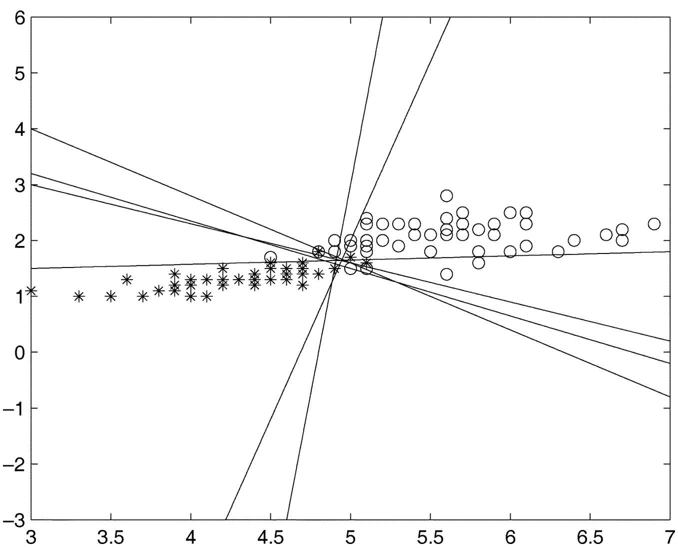

Fig. 3. Separating hyperplanes based on the RCH for the Iris data with different $\mu : 0.8483, 0.7, 0.5, 0.3, 0.2, 0.1$.

In the following theorem, we will give the relation between the C-SVM with slack variables [2] and SCHs, which opens the road of applying the nearest point algorithms to the nonseparable problems through the use of SCHs.

Theorem: The weight vector of the classifier associated with the SCHs converges to that of the C-SVM classifier by a suitable selection of parameters, as the number of training samples tends to infinity.

Proof: The proof is analogous to the RCH case in [5], and can be easily completed using Proposition 3.

This theorem proves that the weight vector of the separating hyperplane associated with the SCHs converges to that of the C-SVM separating hyperplane, but it is not the same case for the biases because the assumptions used to construct the biases differ. It was pointed out that it is not *a priori* evident which assumption for the choice of biases is the best [3]. Our method can be regarded as an approximate alternative to construct SVM classifiers.

IV. S-K Algorithm for Nonseparable Tasks

The so-called S-K algorithm for solving the linearly separable SVM problems has been presented recently in [7]. One important advantage of this algorithm is that it involves only candidate extreme points of the convex hulls. This algorithm is easily generalized to find an ε-optimal separating hyperplane between the SCHs once they are separable.

1) Initialization: set the vector w_1 to any vector (point) $x \in X^{+\prime}$ and w_2 to any vector (point) $x \in X^{-\prime}$.
2) Stopping condition: find the vector x'_t closest to the hyperplane as $t \in \arg\min_{i \in I^+ \cup I^-} m(x'_i)$, where $m(x'_i) = \langle x'_i - w_2, w_1 - w_2 \rangle / \|w_1 - w_2\|$ for $i \in I^+$ and $m(x'_i) = \langle x'_i - w_1, w_2 - w_1 \rangle / \|w_1 - w_2\|$ for $i \in I^-$.
If the ε-optimal condition $\|w_1 - w_2\| - m(x'_t) < \varepsilon$ holds, then the vector $w = w_1 - w_2$ and $b = (\|w_1\|^2 - \|w_2\|^2)/2$ give the ε-solution; otherwise, go to step 3).
3) Adaptation: if $x'_t \in X^{+\prime}$, set $w_2^{\text{new}} = w_2$ and compute $w_1^{\text{new}} = (1 - q) * w_1 + q * x'_t$, where $q = \min(1, \langle w_1 - w_2, w_1 - x'_t \rangle / \|w_1 - x'_t\|^2)$; otherwise, set $w_1^{\text{new}} = w_1$ and compute $w_2^{\text{new}} = (1 - q) * w_2 + q * x'_t$, where $q = \min(1, \langle w_2 - w_1, w_2 - x'_t \rangle / \|w_2 - x'_t\|^2)$.

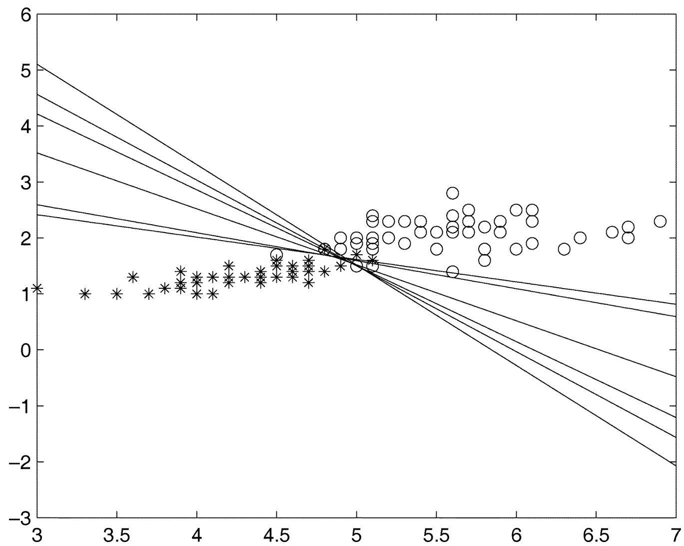

Fig. 4. Separating hyperplanes based on the SCH for the Iris data with different $\lambda : 0.8483, 0.7, 0.5, 0.3, 0.2, 0$.

Next, we mainly compare our algorithm to other algorithms with respect to step 2) which is the main difference in these algorithms. From step 2), it is not the extreme SCH points that are needed, but rather their projections onto a specific direction. Once all x'_i's are computed in advance, the new algorithm has the same complexity as the original S-K algorithm in [7], because the candidate extreme sets $X^{+\prime}$ and $X^{-\prime}$ of the two SCHs have the same number of elements as those of the original convex hulls.

The new algorithm is simpler than the RCH-based S-K algorithm proposed in [9] and [11], because the extra cost of sorting the projections of all the original points in X^+ and X^- in the ascending order and combining appropriately the specific number of the smallest of them is required to compute the minimum projection for the RCH case. Besides, the new algorithm's complexity does not change with the change of the reduction factor λ.

Since the vectors of the feature space are presented in the form of norms and inner product in the above algorithm, the proposed algorithm is easily extended to nonlinear problems through the so-called "kernel trick" technique. But as the mapping is usually unknown, it is difficult to describe the centroid and extreme points in the feature space, resulting in the computational cost of performing the norms and inner products in the S-K algorithm. For computational simplicity, we use a heuristics: directly mapping x'_i's to the feature space. Another important advantage of this heuristics is that it can keep the sparsity of the solution coefficients.

1) Kernel SCH-SK [7]:

1) Initialization: set $\alpha_{i_1} = 1$ for any $i_1 \in I^+$, $\alpha_{i_2} = 1$ for any $i_1 \in I^-$, and the remaining multipliers $\alpha_i = 0, i \in I = I^+ \cup I^-$. Initialize the cache of dot products corresponding to the multipliers α_i as $A = K(x'_{i_1}, x'_{i_1})$, $B = K(x'_{i_2}, x'_{i_2})$, $C = K(x'_{i_1}, x'_{i_2})$, $D_i = K(x'_{i_1}, x'_i)$, $E_i = K(x'_{i_2}, x'_i)$, $i \in I$.

TABLE I
RESULTS ACHIEVED FOR EACH ALGORITHM

Dataset	Algorithms	Training Patterns	Parameters	Success rate (%)	Kernel evaluations	Time (sec)
Diabetes	SMO-K	400	$\sigma = 100, C = 100$	76.70±1.8	1.5×10^6	8.4
Diabetes	RCH-SK	400	$\sigma = 100, \mu = 0.0075$	76.30±1.8	7.0×10^5	6.5
Diabetes	SCH-SK	400	$\sigma = 100, \lambda = 0.94$	76.80±1.8	4.3×10^5	4.6
Thyroid	SMO-K	160	$\sigma = 30, C = 1000$	94.6±2.1	8.3×10^4	1.7
Thyroid	RCH-SK	160	$\sigma = 30, \mu = 0.05$	94.7±2.2	4.1×10^4	1.0
Thyroid	SCH-SK	160	$\sigma = 30, \lambda = 0.84$	94.6±2.0	3.3×10^4	1.0
Waveform	SMO-K	400	$\sigma = 20, C = 1000$	89.20±0.5	2.2×10^6	65.0
Waveform	RCH-SK	400	$\sigma = 20, \mu = 0.02$	88.30±0.8	1.5×10^6	37.0
Waveform	SCH-SK	400	$\sigma = 20, \lambda = 0.62$	88.20±0.7	1.2×10^6	29.0
Heart	SMO-K	170	$\sigma = 120, C = 1000$	83.9±3.3	2.6×10^5	1.5
Heart	RCH-SK	170	$\sigma = 120, \mu = 0.017$	84.2±2.7	4.7×10^4	0.9
Heart	SCH-SK	170	$\sigma = 120, \lambda = 0.76$	83.9±2.7	3.5×10^4	0.9
S Flare	SMO-K	666	$\sigma = 30, C = 1000$	67.6±1.8	1.0×10^7	30.4
S Flare	RCH-SK	666	$\sigma = 30, \mu = 0.0039$	67.6±1.8	2.1×10^6	13.7
S Flare	SCH-SK	666	$\sigma = 30, \lambda = 0.48$	67.5±1.8	1.9×10^6	10.9
German	SMO-K	700	$\sigma = 10, C = 3162$	76.1±2.2	9.0×10^6	31.0
German	RCH-SK	700	$\sigma = 10, \mu = 0.0052$	75.5±0.5	2.7×10^6	3.6
German	SCH-SK	700	$\sigma = 10, \lambda = 0.78$	75.9±0.5	1.6×10^6	3.2
W Cancer	SMO-K	500	$\sigma = 100, C = 1000$	95.6±2.0	8.3×10^4	4.4
W Cancer	RCH-SK	500	$\sigma = 100, \mu = 0.02$	95.3±2.0	4.1×10^4	3.7
W Cancer	SCH-SK	500	$\sigma = 100, \lambda = 0.05$	95.4±2.0	2.3×10^4	2.4
Adult	SMO-K	16000	$\sigma = 1000, C = 1000$	83.3±1.5	2.3×10^8	573.0
Adult	RCH-SK	16000	$\sigma = 1000, \mu = 0.002$	83.2±2.0	1.5×10^7	276.0
Adult	SCH-SK	16000	$\sigma = 1000, \lambda = 0.35$	83.3±2.0	1.2×10^7	218.0

2) Stopping condition: find the vector x'_t closest to the hyperplane as $t \in \arg\min_{i \in I^+ \cup I^-} m_i$, where $m_i = (D_i - E_i + B - C)/\sqrt{A + B - 2C}$ for $i \in I^+$ and $m_i = (E_i - D_i + A - C)/\sqrt{A + B - 2C}$ for $i \in I^-$.
   If the ε-optimal condition $\sqrt{A + B - 2C} - m_t < \varepsilon$ holds, then the multipliers α_i and bias $(B - A)/2$ correspond to the ε-solution; otherwise, go to step 3).
3) Adaptation: if $t \in I^+$, then adapt α_i as: $\alpha_i = \alpha_i \cdot (1 - q) + q \cdot \delta_{i,t}$ for $i \in I^+$, where $q = \min(1, (A - D_t + E_t - C)/(A + K(x'_t, x'_t) - 2 \cdot (D_t - E_t)))$ and $\delta_{i,t}$ is Dirac's delta function; and update cached dot products: $A = A \cdot (1 - q)^2 + 2(1 - q) \cdot q \cdot D_t + q^2 \cdot K(x'_t, x'_t)$, $C = C \cdot (1 - q) + q \cdot E_t$, $D_i = D_i \cdot (1 - q) + q \cdot K(x'_i, x'_t)$, for $i \in I$.
   Otherwise, if $t \in I^-$, then adapt α_i as: $\alpha_i = \alpha_i \cdot (1 - q) + q \cdot \delta_{i,t}$ for $i \in I^-$, where $q = \min(1, (B - E_t + D_t - C)/(B + K(x'_t, x'_t) - 2 \cdot (E_t - D_t)))$, and adapt cached dot products $B = B \cdot (1 - q)^2 + 2(1 - q) \cdot q \cdot E_t + q^2 \cdot K(x'_t, x'_t)$, $C = C \cdot (1 - q) + q \cdot D_t$, $E_i = E_i \cdot (1 - q) + q \cdot K(x'_i, x'_t)$, for $i \in I$. Continue with step 2).

The nonlinear decision function is determined as $f(x) = \sum_{i \in I} \alpha_i y_i K(x'_i, x) + (B - A)/2$.

V. EXPERIMENTS

In this section, we did some experiments to verify the theoretical analysis of the proposed method in Matlab.

A. Comparison of Performance for Linear Problems

In this case, the linear separating hyperplane is used. The Iris data comes from University of California at Irvine (UCI) Repository of machine learning databases which have 150 4-D samples belonging to three classes and each class has 50 samples. For visualization, we chose only the first 2-D samples of the first two classes. The separating hyperplanes obtained by the RCHs with different μ are shown in Fig. 3, and those by the SCHs with different λ are shown in Fig. 4. It can be seen that small changes in the reduction factor λ lead to smaller differences in the final classifier obtained by the SCHs, so our algorithm is insensitive to λ. The reason may be that the SCH has the same shape as the original convex hull, so the position of SCH extreme points varies smoothly with the change of λ.

B. Comparison of Performance for Nonlinear Problems

In order to extensively investigate the performance (both in speed and accuracy) of the new algorithm presented here, several available test data sets from UCI Machine Learning Repository have been used. Three different algorithms were trained, tested, and compared: the modified sequential minimal optimization algorithm presented by Keerthi *et al.* [13] (denoted SMO-K), S-K algorithm based on RCH presented by Mavroforakis *et al.* [8] (denoted RCH-SK), and the algorithm presented here (denoted SCH-SK). Each algorithm was trained and tested for each data set, under the radial basis function (RBF) kernel $K(x, y) = e^{\|x-y\|^2/2\sigma^2}$ in order to achieve the same accuracy referred in the literature [14]. The same validation method and the same data realizations are used. The results of the runs are summarized in Table I.

The results of the new geometric algorithm presented here, compared to other two algorithms, are very encouraging: the differences in the number of kernel evaluations and execution times are noticeable for the same level of accuracy. The enhanced performance of the proposed algorithm is due to the fact that the SCHs have the same shape as the original ones and the extreme points of the SCHs are easier to compute, resulting in the easier computation of the minimum projection of these points onto a specific direction. Besides, the candidate extreme points consider the data distribution (the mean value and the shape) of each class and the SCH separating hyperplane obtains the maximum margin between the SCHs, so the proposed algorithm can achieve good generalization ability. Since the SCH extreme points move smoothly towards the centroid with the change of λ and the pair of nearest points depends directly on these extreme points, the separating hyperplane changes smoothly. Furthermore, the geometric algorithm presented here is a straightforward optimization scheme, with a clear optimization target and always converging to the global minimum.

VI. Conclusion

In this brief, a new method (the SCH framework) for reducing a convex hull is proposed and several theoretical results are presented, through which the nonseparable classification problems are transformed to the separable ones. As a practical application of the SCH framework, the popular S-K algorithm has been generalized to solve the nonseparable classification problems. The derived algorithm provides a clear understanding of the convergence process and the role of the parameters used, resulting in a very promising geometric method of solving the nonseparable classification problems. Furthermore, it provides an easy way to relate each class with a different penalty factor, i.e., to relate each class with a different cost. Compared to the well-known SMO and RCH-SK algorithms, the proposed algorithm achieves enhanced performance with respect to the kernel evaluations and time requirements.

References

[1] V. N. Vapnik*,* *Statistical Learning Theory*. New York: Wiley, 1998.
[2] C. Cortes and V. N. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[3] K. P. Bennett and E. J. Bredensteiner, "Duality and geometry in SVM classifiers," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 57–64.
[4] 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, DC: Mathematical Association of America, 1998.
[5] D. J. Crisp and C. J. C. Burges, "A geometric interpretation of ν-SVM classifiers," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 1999, vol. 12, pp. 244–250.
[6] 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.
[7] V. Franc and V. Hlavác, "An iterative algorithm learning the maximal margin classifier," *Pattern Recognit.*, vol. 36, no. 9, pp. 1985–1996, Sep. 2003.
[8] M. E. Mavroforakis, M. Sdralis, and S. Theodoridis, "A novel SVM geometric algorithm based on reduced convex hulls," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, vol. 2, pp. 564–568.
[9] 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.
[10] 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.
[11] Q. Tao, G. W. Wu, and J. Wang, "A general soft method for learning SVM classifiers with L_1-norm penalty," *Pattern Recognit.*, vol. 41, no. 3, pp. 939–948, Mar. 2008.
[12] T. T. Friess and R. Harisson, "Support vector neural networks: The kernel adatron with bias and soft margin," Dept. Autom. Control Syst. Eng., Univ. Sheffield, Sheffield, U.K., 1998, Tech. Rep. ACSE-TR-752.
[13] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," Dept. Mech. Prod. Eng., Nat. Univ. Singapore, Singapore, Tech. Rep. CD-99-03, 1999.
[14] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Mach. Learn.*, vol. 42, no. 3, pp. 287–320, 2001.
[15] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods-Support Vector Learning*. Cambridge, MA: MIT Press, 1999, pp. 185–208.