Fuzzy Kernel Perceptron

Jiun-Hung Chen and Chu-Song Chen

***Abstract*—A new learning method, the fuzzy kernel perceptron (FKP), in which the fuzzy perceptron (FP) and the Mercer kernels are incorporated, is proposed in this paper. The proposed method first maps the input data into a high-dimensional feature space using some implicit mapping functions. Then, the FP is adopted to find a linear separating hyperplane in the high-dimensional feature space. Compared with the FP, the FKP is more suitable for solving the linearly nonseparable problems. In addition, it is also more efficient than the kernel perceptron (KP). Experimental results show that the FKP has better classification performance than FP, KP, and the support vector machine (SVM).**

***Index Terms*—Classification, fuzzy perceptron (FP), kernel-based method, Mercer kernel, supervised learning, support vector machine (SVM).**

I. Introduction

COVER'S theorem [2] on the separability of patterns states that a complex pattern classification problem projected into a high-dimensional feature space nonlinearly is more likely to be linearly separable than that in a low-dimensional space. However, the computations involved in a high-dimensional feature space are very time-consuming. Mercer kernels [19] have recently been adopted to make this idea practical. Mercer kernels induce implicit nonlinear mapping functions from input spaces to high-dimensional feature spaces. In addition, the inner product of any two feature vectors in the high-dimensional space can be computed by using the kernel function of the two associated input vectors in the low-dimensional space. Such a technique was adopted in many studies such as support vector machine (SVM) [19], kernel principal component analysis (PCA) [18] and others [5], [12], [13].

SVM [19] is a classification and regression tool. It first maps the input data into a high-dimensional feature space using some kernel functions. Then, a linear separating hyperplane with the maximal margin between its closet positive and negative examples in the mapped space is found. Generally, the learning process of the SVM is formulated as solving a constrained quadratic optimization problem that minimizes a weighted sum of two terms, where the first term is related to the reciprocal of the margin described above and the second is the sum of the classification errors. Although this optimization problem can be solved by quadratic programming techniques, it is typically a large dense quadratic programming one while the amount of associated data is large. Some research works have focused on the ways for solving this optimization problem efficiently and effectively [15], [8], [10]. As a powerful kernel-based learning algorithm, SVM has been successfully applied to handwritten digit recognition problems [19], face detection [14], object recognition [16], and others.[1]

The kernel-based concept has also been adopted for unsupervised learning. PCA [6], which finds a set of orthogonal axes that captures most variations of the data set, can be used to reduce the dimensionality of the data set, such that each datum is represented as a linear combination of its projections onto these axes with least quadratic errors. However, PCA cannot be used to extract nonlinear features and the kernel PCA [18] is proposed to overcome this drawback. The kernel PCA also maps data into some high-dimensional feature space induced by a kernel function in advance, and the standard PCA is then performed on the high-dimensional feature space. Hence, nonlinear structures existing in the data set can be better extracted with kernel PCA.

This paper focuses on supervised learning. In particular, the fuzzy perceptron (FP) proposed by Keller and Hunt [9], [11] is adopted as a basic learning tool. As an iterative refinement scheme, perceptron [11] is an efficient method for learning a linear classifier from training examples. Although the learning rule of the perceptron is simple, it fails to converge for linearly nonseparable cases. The FP solves the above convergence problem using the fuzzy theory so that vectors of high uncertainty have less influence on the training results. In this paper, the FP is extended to become the fuzzy kernel perceptron (FKP) with the help of Mercer kernels. There are two advantages of such an extension from both the analytical and the experimental points of views. First, better classification accuracy is achieved for linearly nonseparable cases. Second, faster convergence property is also obtained. In addition, the performance of the FKP is compared with that of the SVM in this paper. Experimental results show that, by choosing appropriate models or parameters for both FKP and SVM, the FKP consistently outperforms the SVM in either synthetic or real data sets.

The remainder of this paper is organized as follows. Section I-A gives the formulation of our problem. Section II reviews the concept of the FP. Then, in Section III, the FP is generalized to the FKP. Some experimental results are shown in Section IV. Finally, conclusions and discussion are presented in Section V.

A. Problem Formulation

Consider a two-class classification problem. Let $X = \{x_1, x_2, \ldots, x_N | x_i \in R^d, i = 1, 2, \ldots, N\}$ be a set of N d-dimensional input vectors, and its associated label set is $C = \{c_{x_1}, c_{x_2}, \ldots, c_{x_N} | c_i \in \{-1, 1\}, i = 1, 2, \ldots, N\}$. The subsets of examples $X_1 = \{x_i | c_{x_i} = 1\}$ and $X_{-1} = \{x_i | c_{x_i} = -1\}$ are referred to as the sets of

Manuscript received February 7, 2001; revised January 3, 2002.

The authors are with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C. (e-mail: jhchen@iis.sinica.edu.tw; song@iis.sinica.edu.tw).

Digital Object Identifier 10.1109/TNN.2002.804311

[1] Many other applications that use the SVM for learning can be found in http://clopinet.com/isabelle/Projects/SVM/applist.html.

positive and negative examples. Let N_1 and N_{-1} be the respective number of positive and negative examples with $N_1 + N_{-1} = N$, respectively. For each x_i, its augmented input vector is denoted by $\mathbf{x}_i = [x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(d)}, -1]^T$, where $x_i^{(j)}$ is the value of the jth component of x_i, and x_i is also called the unaugmented vector of $\mathbf{x}_i$. Given a linear separating hyperplane $\mathbf{w} = [w_1, w_2, \ldots, w_{d+1}]^t$, its associated classifier $\mathbf{C}_{\mathbf{w}}(.)$ is defined by

$$\mathbf{C}_{\mathbf{w}}(\mathbf{x}_p) = \begin{cases} 1, & \text{if } sgn(\mathbf{w}^T\mathbf{x}_p) \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

for each augmented input vector, $\mathbf{x}_p$, where $sgn(.)$ is the sign function. A pattern $\mathbf{x}_p$ is classified correctly by $\mathbf{w}$ if $\mathbf{C}_{\mathbf{w}}(\mathbf{x}_p) = c_{x_p}$. If there exists a linear separating hyperplane that classifies correctly all the patterns in a two-class classification problem, the problem is called linearly separable.

II. Fuzzy Perceptron

This section reviews the FP, which plays an important role in the derivation of the FKP. The membership function $u_k(x_i) \in [0, 1]$ describes the degree that a given datum x_i belongs to class k. Since this paper focuses on 2-class classification problems, it is assumed that $u_1(x_i) + u_{-1}(x_i) = 1$ for $i = 1, 2, \ldots, N$, where $u_1(x_i)$ and $u_{-1}(x_i)$ are the associate degrees that x_i belongs to positive and negative classes, respectively.

Keller and Hunt [9] suggested the following way for assigning fuzzy membership values such that a fuzzy 2-partition is formed. Given an $x_i \in X_1$

$$\begin{cases} u_1(x_i) = 0.5 + \dfrac{\exp(f(d_{-1}(x_i) - d_1(x_i))/d) - \exp(-f)}{2(\exp(f) - \exp(-f))} \\ u_{-1}(x_i) = 1 - u_1(x_i). \end{cases} \tag{2}$$

Given an $x_i \in X_{-1}$

$$\begin{cases} u_{-1}(x_i) = 0.5 + \dfrac{\exp(f(d_1(x_i) - d_{-1}(x_i))/d) - \exp(-f)}{2(\exp(f) - \exp(-f))} \\ u_1(x_i) = 1 - u_{-1}(x_i). \end{cases} \tag{3}$$

In the above, $d_1(x_i)$ is the distance between vector x_i and the mean of the positive class $M_1 = (1/N_1) \sum_{x_j \in X_1} x_j$

$$d_1(x_i) = ||x_i - M_1||. \tag{4}$$

$d_{-1}(x_i)$ is the distance between vector x_i and the mean of the negative class $M_{-1} = (1/N_{-1}) \sum_{x_j \in X_{-1}} x_j$

$$d_{-1}(x_i) = ||x_i - M_{-1}||. \tag{5}$$

d is the distance between the two means of each class

$$d = ||M_1 - M_{-1}||. \tag{6}$$

Note that $||.||$ is the 2-norm and f is a constant controlling the rate at which memberships decrease toward 0.5. Given a two-class classification problem, the FP learns iteratively a linear separating hyperplane as follows. Given an augmented input vector $\mathbf{x}(t)$ at time step t with its unaugmented vector being $x(t) \in X$, the FP adapts its linear separating hyperplane $\mathbf{w}(t)$ by

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta |u_1(x(t)) - u_{-1}(x(t))|^m \cdot (c_{x(t)} - \mathbf{C}_{\mathbf{w}(t)}(\mathbf{x}(t)))\mathbf{x}(t) \tag{7}$$

where η is a learning rate and m is a positive constant controlling the fuzziness. Hence, the fuzzier the training example, the smaller its influence on the weight adaptation is. An interesting property of the FP is that it retains the property (associated with the standard perceptron) of finding a separating hyperplane in a finite number of iterations in the linearly separable case. Note that by setting $|u_1(x(t)) - u_{-1}(x(t))|^m$ to one, (7) degenerates to the standard perceptron learning rule.

A drawback of the standard perceptron is that it will not terminate in finite times if the classification problem is not linearly separable. Note that a common way for dealing with the termination problem in the standard perceptron is to force it to stop by setting the number of maximal iterations allowed. Hence, the behavior of the standard perceptron can be very erratic in the linearly nonseparable case. On the other hand, the FP provides an elegant way to deal with the problem about termination. In the FP, a training example is *very fuzzy* if its membership value falls within the following region: $u_1(x_i) \in [0.5 - \xi, 0.5 + \xi]$, (where ξ is a selected constant). An additional terminating condition of the FP is that if the misclassifications are all caused by the very fuzzy training examples, then the learning algorithm of the FP should terminate. The parameter ξ is better to be selected when the training examples with the membership values outside $[0.5 - \xi, 0.5 + \xi]$ are linearly separable. Note that it is ensured that the FP will terminate in finite times in this case. Therefore, the FP provides a more reasonable stopping criterion using the fuzziness of the training examples.

In practice, the parameter ξ can be determined by trying a set of different values and then selecting the one associated with the largest correct ratio of classification (if the training process is allowed to be off-line). On the other hand, ξ can also be determined according the data distribution [9]

$$\xi = \frac{1 - \exp(-f)}{2(\exp(f) - \exp(-f))} + \epsilon \tag{8}$$

where ϵ is a positive constant controlling the area in which the separating hyperplane can lie, and the left side of (8) is obtained by setting $d_1(x_i) = d_{-1}(x_i)$ in (2).

III. Fuzzy Kernel Perceptron

From the Mercer theorem [19], it is known that a Mercer kernel induces an implicit function that projects nonlinearly the original input vectors into a very high-dimensional feature space. In addition, the value obtained from a Mercer kernel associated with any two vectors in the low-dimensional space can be interpreted as the inner product of the mapped vectors in the high-dimensional feature space. For example, two commonly adopted Mercer kernels are the polynomial kernel $K^{P,h}$ and the radial basis kernel $K^{R,\gamma}$ as shown in the following:

$$K^{P,h}(z_1, z_2) = (z_1^t z_2 + 1)^h \tag{9}$$

$$K^{R,\,\gamma}(z_1,\, z_2) = \exp(-\gamma \|z_1 - z_2\|^2) \tag{10}$$

where z_1 and z_2 are two vectors in the low-dimensional space, and h and γ are two constants.

The main idea of the FKP is described below. Our method projects nonlinearly the input data into a high-dimensional feature space. From Cover's theorem, projecting into a high-dimensional feature space is assumed to make linearly nonseparable problems more likely be linearly separable. In view of this assumption, the perceptron is a good choice for classification in the high-dimensional feature space because it provides a simple and efficient way to deal with the linearly separable problem. Compared with the SVM, the kernel perceptron can be more easily implemented by the associated simple learning rule. However, it is not guaranteed that all problems cast from the input space to a high-dimensional feature space are linearly separable. Hence, it is also important to handle the convergence for the possible cases that are linearly nonseparable in high-dimensional feature spaces. Therefore, the FP is adopted, as a better choice than the perceptron, to solve the corresponding classification problem in the high-dimensional feature space in this paper.

However, direct computation in the high-dimensional feature space is very time-consuming. Therefore, the Mercer kernels are used to make it practical. In the following, the projection of a training example x_p in the high-dimensional feature space is denoted by $\hat{x}_p$. The augmented vector of $\hat{x}_p$ is denoted by $\hat{\mathbf{x}}_p$. In particular, $\hat{x}_p$ and $\hat{\mathbf{x}}_p$ are also called the *image* and the *augmented image* of x_p, respectively, and x_p is called the *preimage* of both $\hat{\mathbf{x}}_p$ and $\hat{x}_p$. Given an FKP with the Mercer kernel K, the corresponding learning rule of the FKP in the high-dimensional feature space is

$$\begin{aligned}\hat{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(t) + \eta |u_1\left(\hat{x}(t)\right) - u_{-1}\left(\hat{x}(t)\right)|^m \\ \cdot \left(c_{x(t)} - \mathbf{C}_{\hat{\mathbf{w}}(t)}\left(\hat{\mathbf{x}}(t)\right)\right) \hat{\mathbf{x}}(t) \text{ for } t = 0,\, 1,\, \ldots .\end{aligned} \tag{11}$$

From (11), it is realized that $\hat{\mathbf{w}}(t+1)$ is a linear combination of the augmented images $\hat{\mathbf{x}}(j)$, $j = 0, \ldots, t$ and $\hat{\mathbf{w}}(0)$. More specifically, $\hat{\mathbf{w}}(t+1)$ can be expressed by

$$\hat{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(0) + \sum_{j=0,\, \ldots,\, t} a(j) \hat{\mathbf{x}}(j), \quad \text{for } t = 0,\, 1,\, \ldots \tag{12}$$

where $a(j)$ is a coefficient associated with the training example $x(j)$. When some augmented image $\hat{\mathbf{x}}(t)$ is presented at time step t, its associated coefficient $a(t)$ is obtained by

$$\begin{aligned}a(t) = \eta |u_1\left(\hat{x}(t)\right) - u_{-1}\left(\hat{x}(t)\right)|^m \\ \cdot \left(c_{x(t)} - \mathbf{C}_{\hat{\mathbf{w}}(t)}\left(\hat{\mathbf{x}}(t)\right)\right), \quad \text{for } t = 0,\, 1,\, \ldots .\end{aligned} \tag{13}$$

Let $\hat{\mathbf{w}}(0) = [\hat{w}(0)^T \theta]^T$, where θ is a random number and $\hat{w}(0)$ is a feature vector whose preimage $w(0)$ is selected randomly in the low-dimensional space. To compute $sgn(\hat{\mathbf{w}}(t)^T \hat{\mathbf{x}}(t))$, the following equation is used:

$$\begin{aligned} & sgn\left(\hat{\mathbf{w}}(t)^T \hat{\mathbf{x}}(t)\right) \\ &= sgn\left(\hat{\mathbf{w}}(0)^T \hat{\mathbf{x}}(t) + \sum_{j=0,\, \ldots,\, t-1} a(j) \hat{\mathbf{x}}(j)^T \hat{\mathbf{x}}(t)\right) \\ &= sgn\left(\left[\hat{w}(0)^T \theta\right] \left[\hat{x}(t)^T - 1\right]^T \right. \\ &\qquad \left. + \sum_{j=0,\, \ldots,\, t-1} a(j) \left[\hat{x}(j)^T - 1\right] \left[\hat{x}(t)^T - 1\right]^T \right) \\ &= sgn\left(K(w(0),\, x(t)) - \theta \right. \\ &\qquad \left. + \sum_{j=0,\, \ldots,\, t-1} a(j)(K(x(j),\, x(t)) + 1)\right). \end{aligned} \tag{14}$$

Hence, the learned classifier of the FKP at time step t is the following.

1) If $t = 0$

$$\mathbf{C}_{\hat{\mathbf{w}}(0)}\left(\hat{\mathbf{x}}(0)\right) = \begin{cases} 1, & \text{if } sgn(K(w(0),\, x(0)) - \theta) \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{15}$$

2) If $t = 1,\, 2,\, \ldots$

$$\begin{aligned} & \mathbf{C}_{\hat{\mathbf{w}}(t)}\left(\hat{\mathbf{x}}(t)\right) \\ &= \begin{cases} 1, & \text{if } sgn\left(\displaystyle\sum_{j=0,\, \ldots,\, t-1} a(j)(K(x(j),\, x(t)) + 1) \right. \\ & \qquad \left. + K(w(0),\, x(t)) - \theta \right) \geq 0 \\ -1, & \text{otherwise.} \end{cases} \end{aligned} \tag{16}$$

During the learning process, it is also required to compute the fuzzy memberships $u_1(\hat{x}(t))$ and $u_{-1}(\hat{x}(t))$. The set of the latest N images $\hat{X}(t) = \{\hat{x}(t-N),\, \ldots,\, \hat{x}(t-1)\}$ is used to estimate the fuzzy memberships in our method. According to (2) and (3), the assignment of fuzzy membership values $u_1(\hat{x}(t))$ and $u_{-1}(\hat{x}(t))$ can be obtained by computing $d_1(\hat{x}(t))$, $d_{-1}(\hat{x}(t))$, and $d(t)$ in the high-dimensional feature space. Let $X_1(t) = \{x_j | \hat{x}_j \in \hat{X}(t) \text{ and } c_{x_j} = 1\}$ and $X_{-1}(t) = \{x_j | \hat{x}_j \in \hat{X}(t) \text{ and } c_{x_j} = -1\}$, respectively. The means of the images of the positive examples $X_1(t)$ and the negative examples $X_{-1}(t)$ in the feature spaces are denoted by $\hat{M_1}(t)$ and $\hat{M_{-1}}(t)$, respectively

$$\hat{M_1}(t) = \frac{1}{N_1(t)} \sum_{x_j \in X_1(t)} \hat{x}_j$$

$$\hat{M_{-1}}(t) = \frac{1}{N_{-1}(t)} \sum_{x_j \in X_{-1}(t)} \hat{x}_j$$

where $N_1(t)$ and $N_{-1}(t)$ is the number of positive and negative elements contained in $X_1(t)$ and $X_{-1}(t)$, respectively, at time

t with $N_1(t) + N_{-1}(t) = N$. Consider that $d_1(\hat{x}(t))$ is the distance between vector $\hat{x}(t)$ and $\hat{M}_1(t)$

$$\begin{aligned} d_1(\hat{x}(t)) &= \left\| \hat{x}(t) - \hat{M}_1(t) \right\| \\ &= \left(\hat{x}(t)^t \hat{x}(t) - 2\hat{x}(t)^t \hat{M}_1(t) + \hat{M}_1(t)^t \hat{M}_1(t) \right)^{1/2} \\ &= \left(K(x(t), x(t)) - \frac{2}{N_1(t)} \sum_{x_j \in X_1(t)} K(x(t), x_j) \right. \\ &\quad \left. + \frac{1}{N_1(t)^2} \sum_{x_j, x_k \in X_1(t)} K(x_j, x_k) \right)^{1/2} \end{aligned} \tag{17}$$

where $d_{-1}(\hat{x}(t))$ is the distance between vector $\hat{x}(t)$ and $\hat{M}_{-1}(t)$

$$\begin{aligned} d_{-1}(\hat{x}(t)) &= \left\| \hat{x}(t) - \hat{M}_{-1}(t) \right\| \\ &= \left(\hat{x}(t)^t \hat{x}(t) - 2\hat{x}(t)^t \hat{M}_{-1}(t) + \hat{M}_{-1}(t)^t \hat{M}_{-1}(t) \right)^{1/2} \\ &= \left(K(x(t), x(t)) - \frac{2}{N_{-1}(t)} \sum_{x_j \in X_{-1}(t)} K(x(t), x_j) \right. \\ &\quad \left. + \frac{1}{N_{-1}(t)^2} \sum_{x_j, x_k \in X_{-1}(t)} K(x_j, x_k) \right)^{1/2} \end{aligned} \tag{18}$$

$d(t)$ is the distance between $\hat{M}_1(t)$ and $\hat{M}_{-1}(t)$

$$\begin{aligned} d(t) &= \left\| \hat{M}_1(t) - \hat{M}_{-1}(t) \right\| \\ &= \left(\hat{M}_1(t)^t \hat{M}_1(t) - 2\hat{M}_1(t)^t \hat{M}_{-1}(t) + \hat{M}_{-1}(t)^t \hat{M}_{-1}(t) \right)^{1/2} \\ &= \left(\frac{1}{N_1(t)^2} \sum_{x_j, x_k \in X_1(t)} K(x_j, x_k) \right. \\ &\quad - \frac{2}{N_1(t) N_{-1}(t)} \sum_{x_j \in X_1(t),\, x_k \in X_{-1}(t)} K(x_j, x_k) \\ &\quad \left. + \frac{1}{N_{-1}(t)^2} \sum_{x_j, x_k \in X_{-1}(t)} K(x_j, x_k) \right)^{1/2}. \end{aligned} \tag{19}$$

If a fixed set of training examples, $\Gamma = \{(x_1, c_{x_1}), (x_2, c_{x_2}), \ldots, (x_N, c_{x_N})\}$, is used, then an alternative way is to compute the values of $d_1(\hat{x}_i)$, $d_{-1}(\hat{x}_i)$, and $d(t)$ in (17)–(19), respectively, in advance, and use them throughout the learning process. Algorithm 1 shows the detailed learning procedure of the FKP if the training examples contained in Γ are fed into it iteratively. In particular, a special case of the FKP, obtained by setting $|u_1(\hat{x}(t)) - u_{-1}(\hat{x}(t))|^m$ to one, is called the kernel perceptron (KP) in this paper. The concept of KP can be found in [3] and [5]. They mainly focused on transforming the perceptron into a kernel-based version, converting the online learning algorithm of the KP to a batch learning one, or analyzing theoretical error bounds. However, they did not consider the important issue about how to terminate the KP when it is used for a nonlinear separable problem in high-dimensional feature spaces. In this paper, by using the FKP, this issue is well considered. After the training is finished, x_i is called a *representation vector* if $a_i \neq 0, i = 1, \ldots, N$. All the representation vectors and nonzero a_is, together with $w(0)$ and θ, are then recorded for the learned classifier since only nonzero a_is have to be involved in run-time decision.

As we know, the learning process of SVM is formulated as solving a large-scale constrained optimization problem, which requires very tricky and sophisticated optimization techniques. On the other hand, the learning process of FKP (or KP) is simpler than that of SVM since very regular operations are performed. Such a simple learning rule makes it more suitable for hardware implementations, and, thus, also endows it with much more potential for real-time applications.

IV. Experimental Results

In this section, some experimental results are presented to show the effectiveness of the FKP. First, a linear nonseparable problem was used for comparing the FKP with the FP and the KP, respectively, under the situation that a series of random trials were performed. Then, real-world problems were used to compare the FKP with the SVM, where model/parameter selections were taken into account.

Algorithm 1 FKP learning algorithm [Input: $X = \{x_1, \ldots, x_N\}$, $C = \{c_1, \ldots, c_N\}$; Output: $w(0), a_1, \ldots, a_N, \theta$]

```
0.    Initialize $w(0)$ and $\theta$ randomly.
1.    Set $\eta, m, f, \epsilon, T$. Let $a_i = 0 \forall i = 1, \ldots, N$. Set $run = 0$.
2.    Let $A(x_i) = (2/N_1) \sum_{x_j \in X_1} K(x_i, x_j)$,
      $B = (1/N_1^2) \sum_{x_j, x_k \in X_1} K(x_j, x_k)$,
      $C(x_i) = (2/N_{-1}) \sum_{x_j \in X_{-1}} K(x_i, x_j)$,
      $D = (1/N_{-1}^2) \sum_{x_j, x_k \in X_{-1}} K(x_j, x_k)$, and
      $E = (2/N_1 N_{-1}) \sum_{x_j \in X_1, x_k \in X_{-1}} K(x_j, x_k)$.
3.    Compute $d_1(\hat{x}_i) = (K(x_i, x_i) - A(x_i) + B)^{1/2}$
      $\forall i = 1, \ldots, N$.
4.    Compute $d_{-1}(\hat{x}_i) = (K(x_i, x_i) - C(x_i) + D)^{1/2}$
      $\forall i = 1, \ldots, N$.
5.    Compute $d = (B - E + D)^{1/2}$.
6.    Compute $u_1(\hat{x}_i), u_{-1}(\hat{x}_i)$ by substituting
      $d_1(\hat{x}_i), d_{-1}(\hat{x}_i), d$ into (2) and (3) $\forall i = 1, \ldots, N$.
7.    repeat
7.1.      Set $CFlag = 0$, and set $p$ to be a random permutation from 1 to $N$.
7.2.      for $j = 1, \ldots, N$ do
```

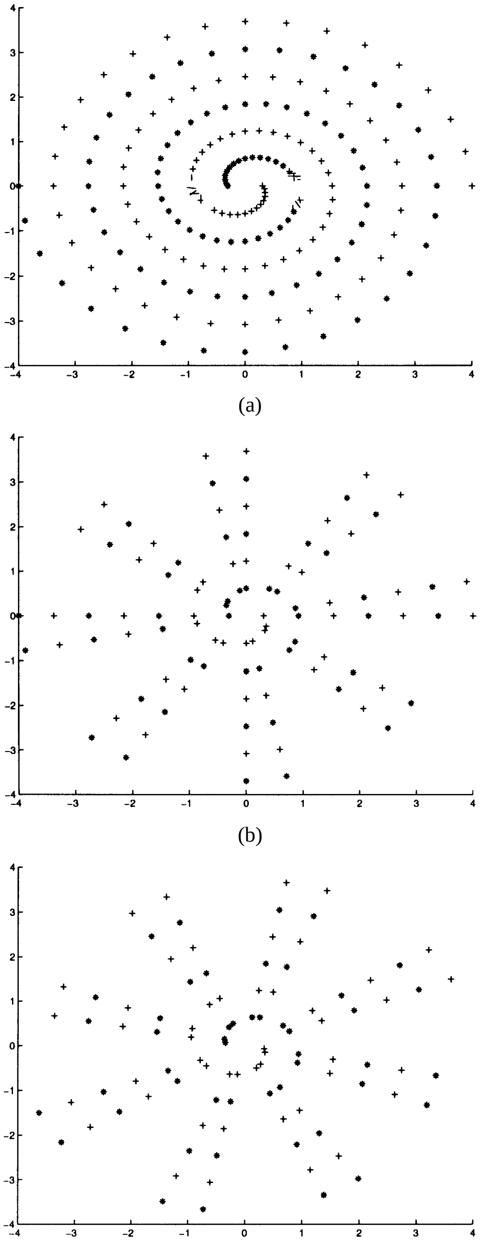
```
7.2.1.  Obtain the classification result $\mathbf{C}_{\hat{\mathbf{w}}}(\hat{\mathbf{x}}_{p[j]})$ by

$$\mathbf{C}_{\hat{\mathbf{w}}}\left(\hat{\mathbf{x}}_{p[j]}\right)=\begin{cases} 1, & \text{if } sgn\left(\sum_{l=1,\ldots,N} a_l\left(K\left(x_l,\, x_{p[j]}\right)+1\right) + K\left(w(0),\, x_{p[j]}\right)-\theta\right)\geq 0 \\ -1, & \text{otherwise.} \end{cases}$$

7.2.2.      if $\mathbf{C}_{\hat{\mathbf{w}}}(\hat{\mathbf{x}}_{p[j]}) \neq c_{x_{p[j]}}$ then
7.2.2.1.        $a_{p[j]} += \eta|u_1(\hat{x}_{p[j]}) - u_{-1}(\hat{x}_{p[j]})|^m (c_{x_{p[j]}} - \mathbf{C}_{\hat{\mathbf{w}}}(\hat{\mathbf{x}}_{p[j]}))$
7.2.2.2.        if $u_1(\hat{x}_{p[j]}) \notin [0.5-\xi,\, 0.5+\xi]$ then set $CFlag = 1$ end if
            end if
        end for
7.3.    set $run = run + 1$
        until $run < T$ and $CFlag == 1$
```

The testing data of the former experiment is a synthetic two-spiral two-dimensional (2-D) point set [4] downloaded from the CMU learning benchmark archive,[2] and those of the latter include two real data sets, ionosphere and sonar [1], both of which were commonly used machine learning benchmarks.[3] Some properties of these three data sets are listed in Table I. The former experiment was done on an ASUS Ultra 1 workstation, and the latter on a Sun UltraSparcIIi workstation.

The input data points of the two-spiral problem are shown in Fig. 1(a). They are partitioned into a training set and a testing set as shown in Fig. 1(b) and (c), respectively. The radial basis functions with $\gamma = 0.5$ was used in this experiment. The initial augmented image $\hat{\mathbf{w}}(0)$ was obtained by setting each component of $w(0)$ as a random number between zero and one, under the situation that $\theta = 0$. The parameters f, m, ϵ, η, and the maximum loop were 0.4, 1.05, 0.007, 0.05, and 2500, respectively. Fifty tests were performed by choosing $w(0)$ randomly for FP, KP, and FKP, respectively. The average training time and average correct ratios for both training and testing data sets are shown in Table II, where the average correct ratio for the training (testing) data set is the ratio of the number of training (testing) examples that are correctly classified to the number of training (testing) examples. According to the experimental results listed in this table, the FKP considerably outperforms the FP for both the training and testing sets. It is because the FKP performs better when dealing with the linearly nonseparable problems. Compared with the KP, the FKP also has better correct ratio and training efficiency. It reveals that the convergence problems for the linearly nonseparable cases in the high-dimensional space were handled more effectively by using the FKP.

TABLE I
PROPERTIES OF THE THREE DATA SETS USED IN OUR EXPERIMENTS

Data set	No. of Examples	No. of Attributes
Spirals	194	2
Ionosphere	351	34
Sonar	208	60



(a)

(b)

(c)

Fig. 1. (a) Input examples. (b) Training examples. (c) Testing examples. The symbols $+$ and $*$ are used for different classes.

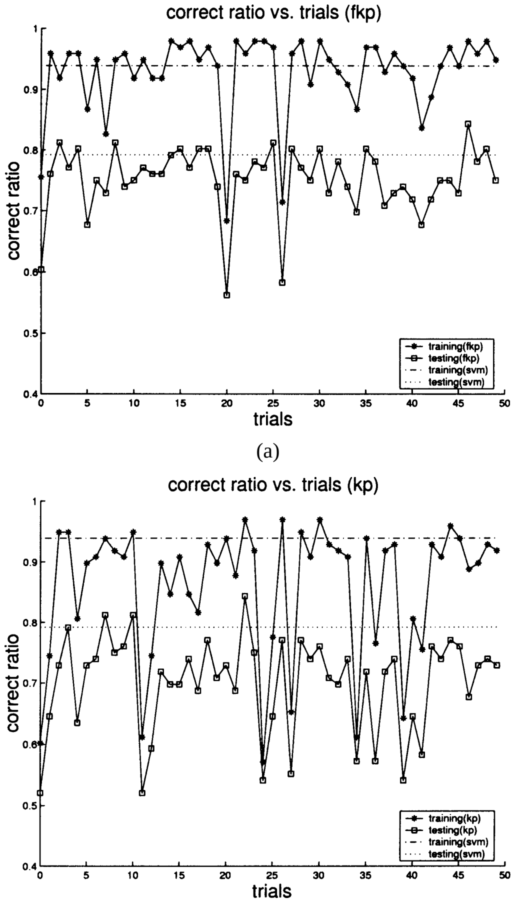
A more detailed list of the results of all the 50 tests in this experiment is shown in Fig. 2. In particular, an additional test with the SVM, where the same kernel function (or equivalently, the same model [13]) was adopted, is also shown in Fig. 2. In this figure, each of the correct ratios of the above 50 random tests are shown with different lines. Unlike that of the SVM, the

[2] http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/neural/bench/cmu/.

[3] These real data sets were downloaded from http://www.ics.uci.edu/~mlearn/MLRepository.html. Many comparisons of results using these data sets can be found in http://www.phys.uni.torun.pl/kmk/projects/datasets.html.

TABLE II
AVERAGE CORRECT RATIOS AND AVERAGE TRAINING TIMES FOR BOTH THE TRAINING AND TESTING SETS OF FKP, KP, AND FP

	FKP	KP	FP
Average correct ratio for the training set	92.9%	86.1%	50%
Average correct ratio for the testing set	75.2%	70%	50%
Average training time	64.45 sec	78.66 sec	17.31 sec



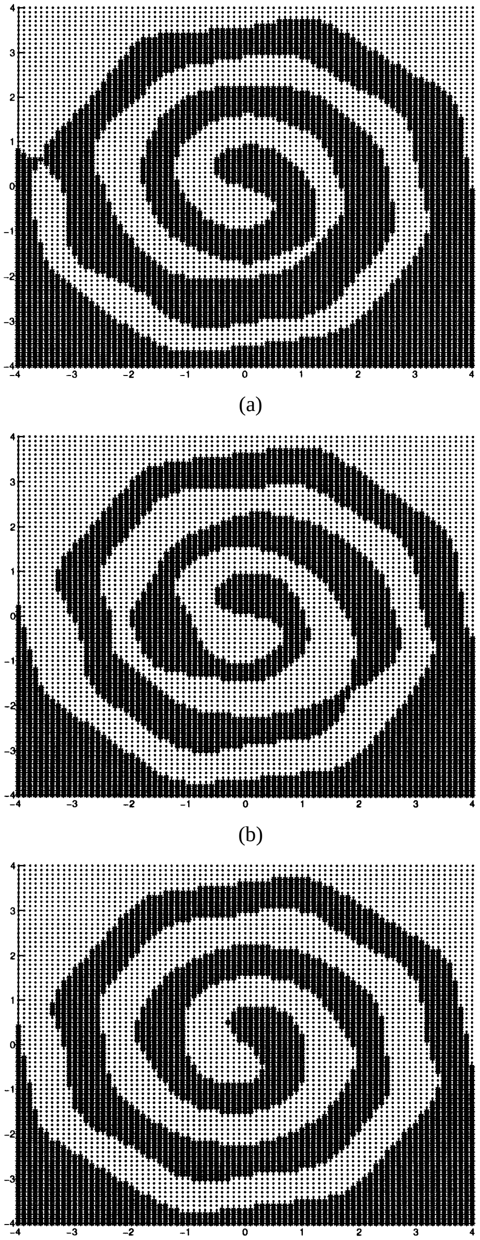
(a)

(b)

Fig. 2. (a) Correct ratios of FKP of the 50 random trials. (b) Correct ratios of KP of the 50 random trials.

classification performances of both the FKP and the KP depend further on the initial configurations even when the same kernel function is used. As seen in Fig. 2(a) and (b), both the FKP and the KP outperform the SVM in some random trials. Fig. 3(a) and (b) show the classifying boundaries of the FKP and the KP with the largest correct ratios for the training set among the 50 tests, respectively, and the classifying boundary of the SVM is shown in Fig. 3(c). The correct ratios of the FKP, the KP, and the SVM (associated with this figure) for the training sets are 98%, 96.9%, and 93.9%, respectively, while those of the testing sets are 79.2%, 84%, and 79.2%, respectively.

Furthermore, to investigate the sensitivities of the parameters (f, m, ϵ, η), the performance of the FKP around the center $(f, m, \epsilon, \eta) =$ (0.4,1.05,0.007,0.05) is shown in Fig. 4. In this analysis, the average of the training and testing correct ratios (ASR) is chosen to be the performance measure. In addition, for each (f, m, ϵ, η), 50 trials were run and the largest ASR observed in these 50 trials is recorded in this analysis. If the largest variation in average correct ratios among the chosen region is considered for sensitivity analyses, then the learning rate η is the most sensitive parameter and ϵ is the next sensitive one according to our tests because their correct ratios vary more than those of the others. In addition, from Fig. 4, the classification results associated with $(f, m, \epsilon, \eta) =$ (0.4,1.05,0.007,0.05) is a good choice but can still be improved with further investigations, which will be considered later through a model/parameter selection process.



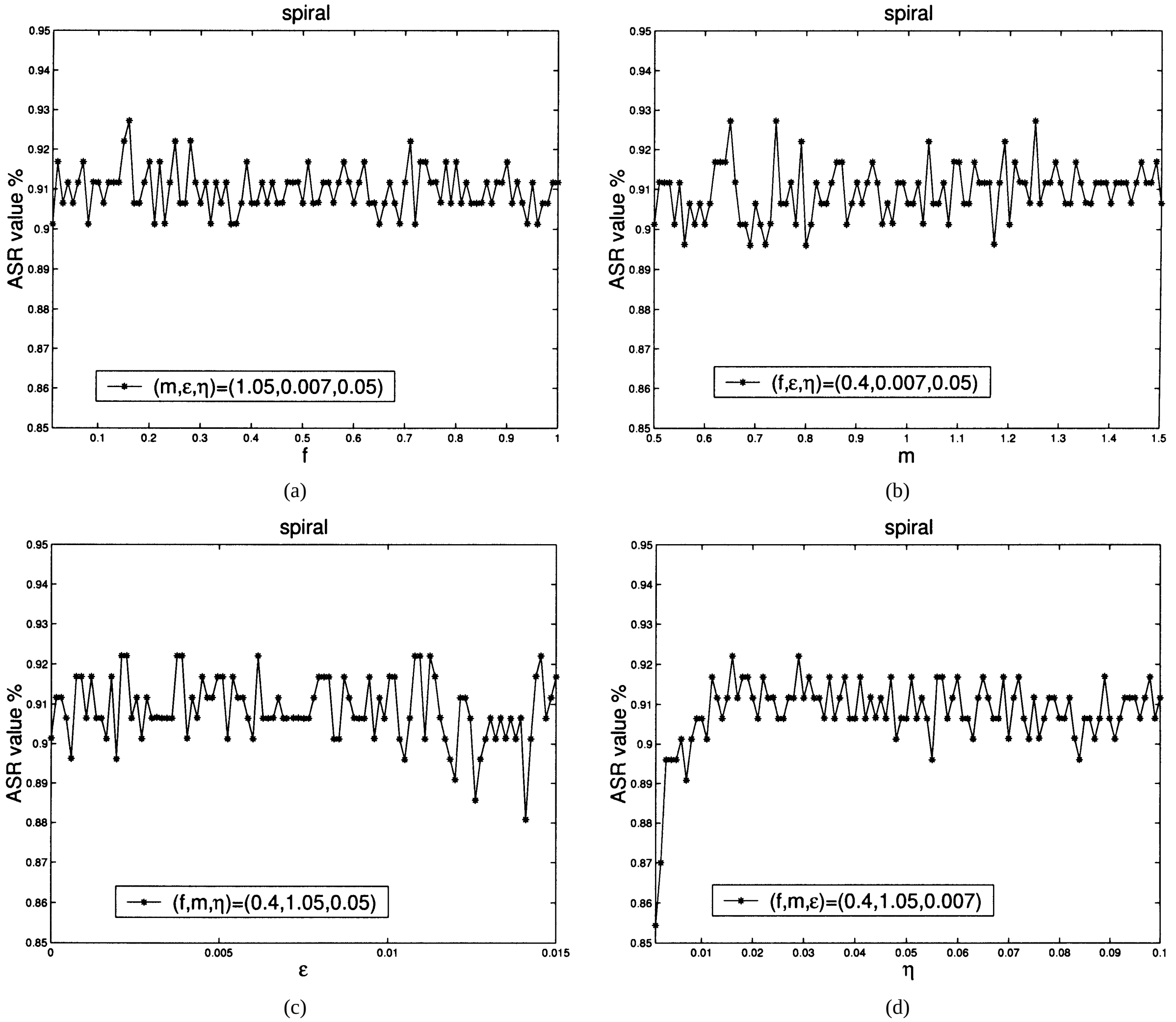
(a)

(b)

(c)

Fig. 3. (a) Classifying boundary of the FKP with the largest correct ratio for the training set among 50 experiments. (b) Classifying boundary of the KP with the largest correct ratio for the training set among 50 experiments. (c) Classifying boundary of the SVM.

An important issue inherent in the above observations is that the FKP can outperform the SVM if appropriate initial conditions are set. In fact, the choice of the kernel also has crucial



(a) (b) (c) (d)

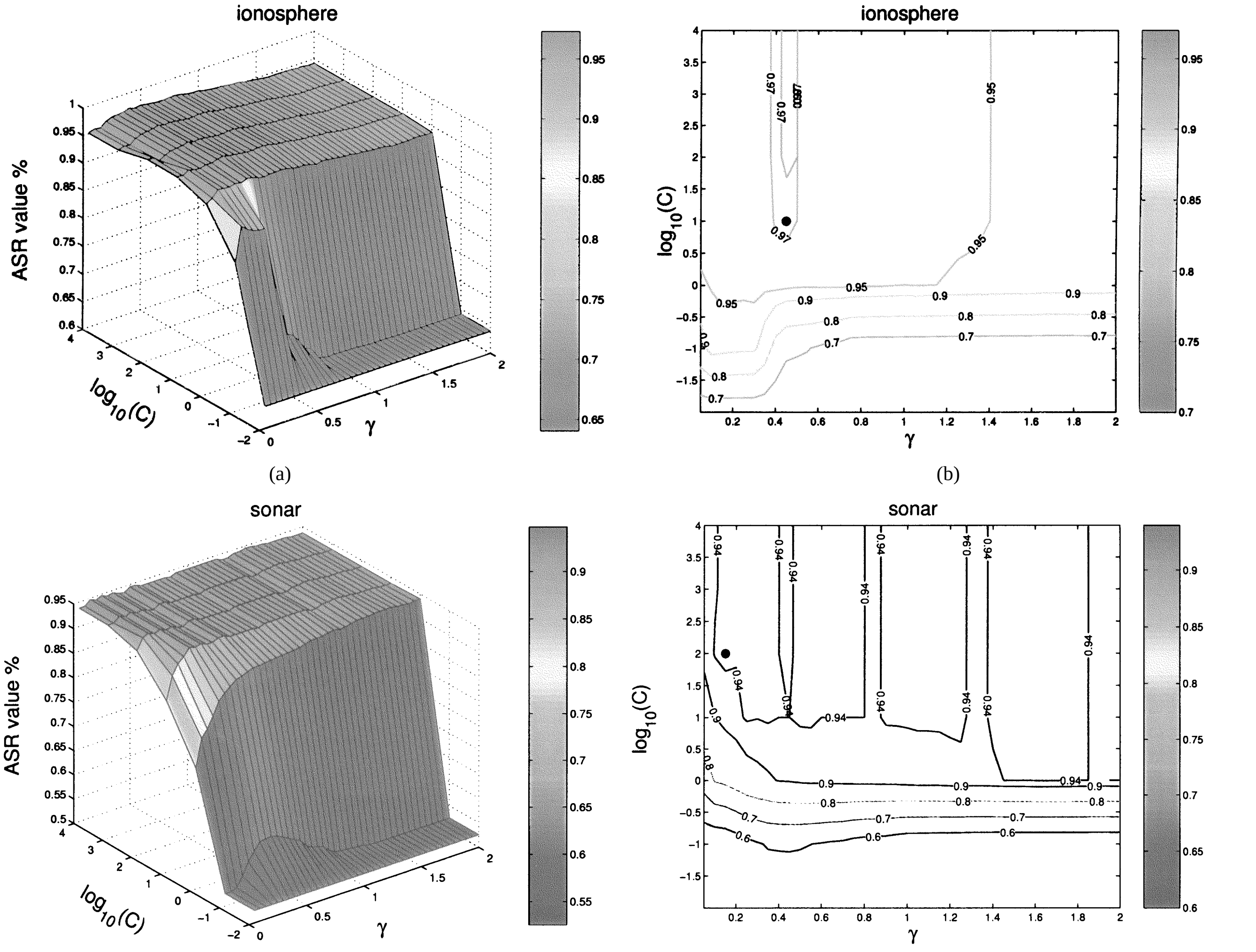
Fig. 4. Sensitivity analyses of the introduced parameters $(f,\ m,\ \epsilon,\ \eta)$ in the FKP around the center $(f,\ m,\ \epsilon,\ \eta)$ = (0.4,1.05,0.007,0.05). (a) ASR versus f with the other three parameters fixed. (b) ASR versus m with the other three parameters fixed. (c) ASR versus ϵ with the other three parameters fixed. (d) ASR versus η with the other three parameters fixed.

effects on the performances for any kernel-based methods [13]. In the second experiment, the performance of FKP is compared with that of the SVM where real data sets were used. Unlike the previous experiment, appropriate kernel function and learning parameters (including η, f, m, ϵ and initial weights) have been investigated through a so-called *model* (*or parameter*) *selection process* [17], [13], [7] in this experiment, instead of being given randomly. Tenfold cross validation [17] combined with the ASR performance measure was used to select a good model for SVM. That is, the parameters with the largest average ASR over an investigated parameter space were chosen and used for the subsequent comparisons.

To make the comparisons between the FKP and the SVM fair, the kernel function was selected according to the performance of the SVM. In this experiment, the radial basis kernel was also adopted. Hence, the kernel parameter γ and the cost parameter C [19] were required to be chosen for SVM. The parameter space investigated in this experiment for both γ and C were $\gamma = \{0.05,\ 0.1,\ 0.15,\ \ldots,\ 2.0\}$ and $C = \{10^{-2},\ \ldots,\ 10^{4}\}$, respectively. The model-selection process computed exhaustively the ASRs associated with all the combinations of the values $(\gamma,\ C)$ contained in the spaces defined above for the SVM. The investigation results are shown in Fig. 5. Finally, the best $(\gamma,\ C)$, that is, $(\gamma,\ C)$ associated with the largest ASR, was selected for the ionosphere and the sonar problems, which turned out to be (0.45,10) and (0.15,100), respectively.

After selecting a proper model for the SVM, the radial basis kernel function with the same γ was used for the FKP in the subsequent tests for comparison purpose. The maximum loop of the FKP was fixed to be 30 000 and a parameter-selection process was further performed for finding appropriate values for the parameters f, m, ϵ, and η of the FKP. The ranges of f, m, ϵ and η were sampled within [0, 3], [0, 2], [0, 0.5], and $[10^{-3},\ 10^{3}]$, respectively, and there are totally 18 720 sample points. For each combination of $(f,\ m,\ \epsilon,\ \eta)$, 50 initial augmented images, $\hat{\mathbf{w}}(0)$, were tried for classification, where $\hat{\mathbf{w}}(0)$ was obtained by setting each component of $w(0)$ as a random number between zero and one under the situation that $\theta = 0$. The combination $(w(0),\ f,\ m,\ \epsilon,\ \eta)$ associated with the best ASR among the above trials was chosen as the parameter set for FKP.

The performance comparisons between the FKP and the SVM on both ionosphere and sonar data sets are summarized in Tables III and IV, respectively. From these tables, both



(a)

(b)

(c)

(d)

Fig. 5. Model selection process for SVM over two different real data sets. The black dots show the positions of the best parameters. (a) ASR versus the parameters (γ, C) for the ionosphere problem. (b) Level-curve diagram of (a). (c) ASR versus the parameter (γ, C) for the sonar problem. (d) Level-curve diagram of (c).

TABLE III
PERFORMANCE COMPARISON BETWEEN THE FKP AND THE SVM ON THE IONOSPHERE DATA SET

	FKP	SVM
Average correct ratio for the training set	100%	100%
Average correct ratio for the testing set	98.3%	94.6%
Average training time	0.643 sec	0.408 sec
Average number of support/representation vectors	85.4	171.8

TABLE IV
PERFORMANCE COMPARISON BETWEEN THE FKP AND THE SVM ON THE SONAR DATA SET

	FKP	SVM
Average correct ratio for the training set	100%	100%
Average correct ratio for the testing set	94%	89.5%
Average training time	0.477 sec	1.496 sec
Average number of support/representation vectors	167	82.6

methods achieve 100% correct ratios for the training sets after model/parameter selections. However, the FKP has better testing correct ratios (Both tests show that the testing correct ratio of the FKP is at least 3.5% higher than that of the SVM.) Compared with those of SVM, the training times of the FKP are faster on the sonar data set but slower on the ionosphere data set. As for the number of support/representation vectors, the FKP has fewer vectors on the ionosphere data set but more vectors on the sonar data set. The parameters (f, m, ϵ, η) associated with the results shown in the Tables III and IV are $(0.0001, 0.1, 0, 10.0)$ and $(0.0001, 0.2, 0.01, 0.01)$ for the ionosphere and sonar data sets, respectively. Finally, by using the same tenfold cross validation and model/parameter selection process introduced above, Fig. 6 and Table V show a renewed result of the spiral problem (which was done on the same UltraSparcIIi Sun workstation). In this case, the FKP also has better classification performance than the SVM did. From Fig. 6, it can be observed that the FKP has learned a more suitable extrapolation than the SVM did if the data are expected to be extrapolated as a spiral shape.

V. CONCLUSION AND DISCUSSION

In this paper, we propose a new learning method, the FKP, for training a 2-classifier. Training with the FKP is equivalent to training with the FP, except that the training vectors are first projected into a high-dimensional space. It is well known that the projection is likely to make a linearly nonseparable problem become linearly separable in the high-dimensional space. Hence,

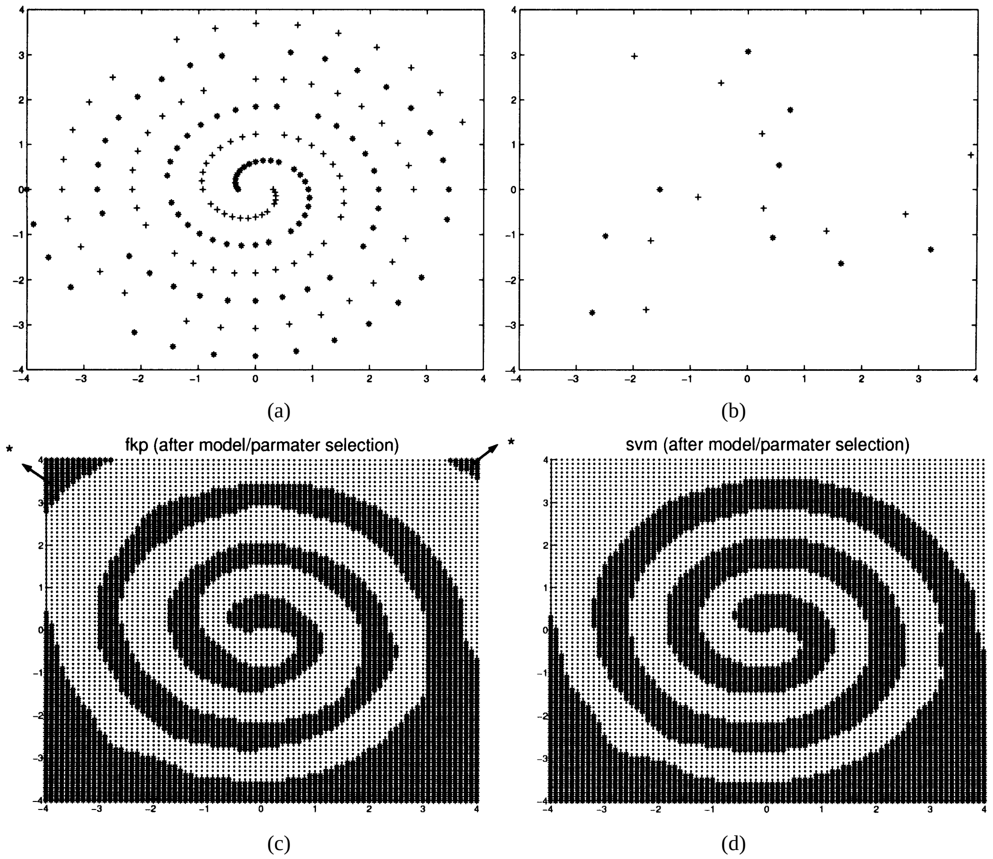


Fig. 6. (a) Example of a tenfold training set for the spiral problem. (b) Example of a tenfold testing set for the spiral problem. (c) Classifying boundary of the FKP. (d) Classifying boundary of the SVM. In particular, note the regions marked with outward arrows. These regions show that the FKP has learned a more suitable extrapolation from the same limited data set than the SVM did if the data are expected to be extrapolated as a spiral shape.

TABLE V
PERFORMANCE COMPARISON BETWEEN THE FKP AND THE SVM ON THE SPIRAL DATASET

	FKP	SVM
Average correct ratio for the training set	100%	99.9%
Average correct ratio for the testing set	97.9%	94.2%

the perceptron seems to be a good choice for solving such a linearly separable problem in the high-dimensional space because its learning rule is simple. However, it is also possible that the problem resulted via such a projection is still linearly nonseparable in the high-dimensional space, and the perceptron may fail to converge in this case. Therefore, the FP serves as a better choice for training a classifier in the high-dimensional space because the convergence problem is tackled more appropriately in FP by considering the fuzzy memberships of the training vectors. We have found that the operations involved in the FP can be replaced by some linear combinations of the inner products of the training vectors, which allows us to use the Mercer kernel to realize the FKP in the low-dimensional space. Such a generalization also allows it to deal with linearly nonseparable problems better.

Our experimental results show that the FKP has better average performance, both in convergence speed and in correct classification rate, than the KP. In addition, the FKP outperforms the FP in the average correct ratios for both the training and testing sets. Compared with the SVM, the FKP is also superior to the SVM for both the spiral and the real data sets tested in our experiments under the situation that appropriate models (or parameters) have been selected for both SVM and FKP.

How to speed up the training process of the FKP, particularly with a general-purpose computer, such that it can achieve real-time performance is important for on-line applications and merits further investigation.

ACKNOWLEDGMENT

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

REFERENCES

[1] C. L. Blake and C. J. Merz. (1998) UCI repository of machine learning databases. Univ. California, Dept. Inform. Comput. Sci., Irvine, CA. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[2] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 326–334, Mar. 1965.

[3] 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.

[4] S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems* Denver, CO, 1989, pp. 524–532.

[5] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Machine Learning*, vol. 37, pp. 277–296, 1999.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1990.

[7] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multi-class support vector machines," IEEE Trans. Neural Networks, vol. 13, pp. 415–425, Mar. 2001, to be published.

[8] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press, 1999.

[9] J. M. Keller and D. J. Hunt, "Incorporating fuzzy membership functions into the perceptron algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 693–699, June 1985.

[10] C.-J. Lin, "Formulations of support vector machines: A note from an optimization point of view," *Neural Comput.*, vol. 13, pp. 307–317, 2001.

[11] C. T. Lin and C. S. G. Lee, *Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems*. London, U.K.: Prentice-Hall, 1996.

[12] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Muller, "Fisher discriminant analysis with kernels," in *Proc. IEEE Neural Networks Signal Processing Workshop*, 1999, pp. 41–48.

[13] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, pp. 181–201, Jan. 2001.

[14] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. Comput. Vision Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 130–136.

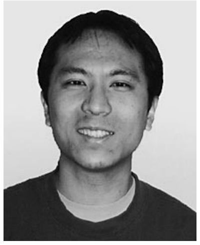
[15] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 185–208.

[16] M. Pontil and A. Verri, "Support vector machines for 3-d object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 637–646, June 1998.

[17] C. Schaffer, "Selecting a classification method by cross-validation," *Machine Learning*, vol. 13, pp. 135–141, 1993.

[18] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, pp. 1299–1319, 1998.

[19] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.



Jiun-Hung Chen received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1997 and 1999, respectively.

He is currently a Research Assistant in the Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C. His research interests include pattern recognition, image processing, and computer vision.

Chu-Song Chen (S'94–M'96) received the B.S. degree in control engineering from National Chiao-Tung University, Hsing-Chu, Taiwan, R.O.C., in 1989, and the M.S. and Ph.D. degrees in 1991 and 1996, respectively, both from the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

From 1997 to 1999, he was a Postdoctoral Fellow of the Institute of Information Science, Academia Sinica, and he became an Assistant Research Fellow in 1999. His research interests include pattern recognition, computer vision, signal/image processing, and computer graphics.

Dr. Chen received both the Outstanding Paper Award of the Image Processing and Pattern Recognition (IPPR) Society and the Best Paper Award of the Image Processing and Application Association (IPAA), Taiwan, R.O.C., in 1997. He received the Outstanding Paper Award in the field of applications of computers at ICS2000, Chiayi, Taiwan, R.O.C.