Hidden Space Support Vector Machines

Li Zhang, Weida Zhou, and Licheng Jiao, *Senior Member, IEEE*

***Abstract*—Hidden space support vector machines (HSSVMs) are presented in this paper. The input patterns are mapped into a high-dimensional hidden space by a set of hidden nonlinear functions and then the structural risk is introduced into the hidden space to construct HSSVMs. Moreover, the conditions for the nonlinear kernel function in HSSVMs are more relaxed, and even differentiability is not required. Compared with support vector machines (SVMs), HSSVMs can adopt more kinds of kernel functions because the positive definite property of the kernel function is not a necessary condition. The performance of HSSVMs for pattern recognition and regression estimation is also analyzed. Experiments on artificial and real-world domains confirm the feasibility and the validity of our algorithms.**

***Index Terms*—Artificial neural networks (ANNs), pattern recognition, regression estimation, structural risk, support vector machines.**

I. Introduction

SUPPORT VECTOR MACHINES (SVMs) based on the statistical learning theory (STL) are general and efficient learning machines [1]–[5]. In STL, the problem of consistency of learning procedure in machine learning is the one where the empirical risk converges uniformly to the actual risk. To obtain a small actual risk, i.e., a good generalization performance, the SLT shows that it is necessary to have a right balance between the empirical risk and the capacity of a learning machine. SVMs can do this, so they can obtain a good generalization performance. SVMs have other attractive properties, for example, SVMs have a unique global optimal solution and avoid the curse of dimensionality. The introduction of kernel methods has made SVMs have a nonlinear process ability. Presently, there are many Mercer kernels available such as Gaussian radial basis function kernel, sigmoidal kernel, polynomial kernel, spline kernels, and others. These kernels must satisfy Mercer's condition or they must be symmetric and positive semidefinite. Here we will extend the range of usable kernels that are not required to satisfy the condition of the positive definite property. As we know, the introduction of kernel functions is based on the view of nonlinear mapping. Before SVMs, the view of nonlinear mapping was embodied in many other fields. The hidden function mapping in forward neural networks (FNNs) and radial basis function networks (RBFNs) is a typical example.

Artificial neural networks (ANNs) were developed rapidly in the second half of the last century [9]–[11]. At present, ANNs have been successfully applied to signal processing, pattern recognition, regression estimation, predication, function interpolation, intelligent controlling, and other fields for their good nonlinear process ability [12], [13]. FNNs and RBFNs involved in this paper were presented in the middle and late 1980s [14], [15]. It has been shown that FNNs and RBFNs have a good nonlinear mapping ability and approximation performance. They can approximate arbitrary continuous function with any accuracy [16], [17]. However, they also have some shortages: 1) the resulting network training consists of solving a hard nonlinear optimization problem, with a possibility of getting trapped in local minima; 2) there is often a risk of getting overfitting; 3) ANNs are hard to interpret.

In 1999, Suykens *et al.* [18], [19] first introduced the structural risk into FNNs to obtain a good generalization. In [18], Suykens *et al.* presented a method for training a multilayer perceptron based on a modified SVMs (MLP-SVM), which introduced the capacity control in SLT into the object function of MLP directly. Thus, it can solve the overfitting problem in MLP to a certain extent and obtain a better generalization performance. This method directly uses the SVMs method to optimize multilayer perceptron including the output weight vector, the interconnection matrix for the hidden layer and the bias vector, so the algorithm is very complex. It includes two correlative quadratic programming with each programming similar to that of the support vector machines and constrained by a nonlinear equation simultaneously. Therefore, the computation complexity of the MLP-SVM method is at least two times that of the traditional SVMs. Moreover the optimization problem of the method is not convex, so there exist local minima in this method.

We use the idea of the nonlinear mapping in ANNs instead of introducing the structural risk into ANNs directly. We first map the data in the input space into a hidden space by a set of hidden functions and then introduce the structural risk in the hidden space to implement hidden space support vector machines (HSSVMs) for pattern recognition and regression estimation. Therefore, HSSVMs inherit not only the strong nonlinear processing ability of ANNs but also the good generalization performance of SVMs. On one hand, HSSVMs implement structural risk minimization in a hidden space, which makes HSSVMs have as good a generalization performance as SVMs and avoid the overfitting problem. On the other hand, the BP algorithm requires the derivativeness for the node functions, and SVMs require that kernel functions satisfy the rigorous Mercers condition. Here, we relax the constraints for nonlinear nodes or kernel functions in HSSVMs. It does not require that the kernel functions satisfy the rigorous Mercer's condition or derivativeness but it requires symmetry. Namely, in HSSVMs for the real-valued kernel the only condition is $k(\mathbf{x},\mathbf{y}) = k(\mathbf{y},\mathbf{x})$.

Manuscript received January 18, 2002; revised August 13, 2003 and December 17, 2003. This work was supported by the National Natural Science Foundation of China by Grants 60073053, 60133010, and 69831040.

The authors are with the Key Laboratory for Radar Signal Processing, Xidian University, Xi'an 710071, China.

Digital Object Identifier 10.1109/TNN.2004.831161

The training procedure of HSSVMs amounts to solving a constrained quadratic programming whose scale is the same as that of SVMs. So HSSVMs have the same computation complexity as SVMs. And the solution of HSSVMs is global one. Moreover, for the large scale problem, HSSVMs can adopt the decomposition methods used in the SVMs. The algorithm analysis and the experimental results confirm the feasibility and the validity of our algorithm.

II. Hidden Space Support Vector Machines (HSSVMs)

A. Hidden Space

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ denote the set of N independently and identical distributed (i.i.d.) patterns. Define a vector made up of a set of real-valued functions $\{\varphi_i(\mathbf{x}) | i = 1, 2, \cdots, d_1\}$, as shown by

$$\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \cdots, \varphi_{d_1}(\mathbf{x})]^T \tag{2.1}$$

where $\mathbf{x} \in X \subset \mathbb{R}^d$. The vector $\varphi(\mathbf{x})$ maps the points in the d-dimensional input space into a new space of dimension d_1. Namely

$$\mathbf{x} \xrightarrow{\varphi} \mathbf{z} = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \cdots, \varphi_{d_1}(\mathbf{x})]^T. \tag{2.2}$$

Since the set of functions $\{\varphi_i(\mathbf{x})\}$ plays a role similar to that of a hidden unit in FNNs, we refer to $\varphi_i(\mathbf{x}), i = 1, \cdots, d_1$ as hidden functions. Accordingly, the space $\mathcal{Z} = \{\mathbf{z} | \mathbf{z} = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \cdots, \varphi_{d_1}(\mathbf{x})]^T, \mathbf{x} \in X\}$ is called the hidden space or feature space.

Now consider a special kind of hidden function: the real symmetric kernel function $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and the kernel mapping be

$$\mathbf{x} \xrightarrow{k} \mathbf{z} = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \cdots, k(\mathbf{x}_N, \mathbf{x})]^T. \tag{2.3}$$

The corresponding hidden space based on X can be expressed as $\mathcal{Z} = \{\mathbf{z} | \mathbf{z} = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \cdots, k(\mathbf{x}_N, \mathbf{x})]^T, \mathbf{x} \in X\}$ whose dimension is N.

It is only the symmetry for kernel functions that is required, which will extend the set of usable kernel functions in HSSVMs while the rigorous Mercer's condition is required in SVMs. Some usual hidden functions are given as follows.

- **Sigmoidal kernel**

$$k(\mathbf{x}, \mathbf{y}) = \tanh(p_1 \cdot \mathbf{x}^T \mathbf{y} + p_2), p_1, p_2 \in \mathbb{R}. \tag{2.4}$$

It is noticeable that usually the sigmoidal kernel is not positive definite, which limits its application in SVMs. But it is not a problem here. If parameter $p_1 \to \infty$ in the sigmoidal kernel, we will obtain a sign function

$$k(\mathbf{x}, \mathbf{y}) = \operatorname{sgn}(p_1 \cdot \mathbf{x}^T \mathbf{y} + p_2), p_1, p_2 \in \mathbb{R}.$$

Although the differential of the sign function does not exist at some points, it also can be used in our algorithm.

- **Gaussian radial basis kernel**

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2p^2}\right), \quad p \in \mathbb{R} \tag{2.5}$$

which is a kernel in wide use and has been successfully applied to function approximation, density estimation, time-frequency analysis, neural networks and support vector machines.

- **Polynomial kernel**

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^p, \quad p \in \mathbb{N} \tag{2.6}$$

which is a positive definite kernel used in SVMs frequently.

- **Generalized multiquadrics kernel**

$$k(\mathbf{x}, \mathbf{y}) = \left(1 + \|\mathbf{x} - \mathbf{y}\|^2\right)^p, \quad p \notin \mathbb{Z} \tag{2.7}$$

where $\| \bullet \|$ denotes 2-norm. If $p < 0$, (2.7) is a Mercer admissible kernel.

- **Thin plate spline kernel**

$$k(\mathbf{x}, \mathbf{y}) = \begin{cases} \|\mathbf{x} - \mathbf{y}\|^{2p-d} \ln \|\mathbf{x} - \mathbf{y}\|, & \text{for } d \text{ even} \\ \|\mathbf{x} - \mathbf{y}\|^{2p-d}, & \text{for } d \text{ odd} \end{cases}$$
$$and\, p > \frac{d}{2}; \quad p, d \in \mathbb{N} \tag{2.8}$$

which is the solution of regularization functional if the smoothness factors or stabilizers take the p-order differential operator in regularization functional.

Other kernels [7], [20], [21] used in SVMs can be applied to our algorithm. We will not discuss them here.

B. HSSVMs for Pattern Recognition

Let a pattern set be $X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \cdots, (\mathbf{x}_N, y_N) | \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}\}$ and a kernel function be $k(\mathbf{x}, \mathbf{y})$. The mapped patterns in the hidden space $\mathcal{Z}$ can be expressed as $\{(\mathbf{z}_1, y_1), (\mathbf{z}_2, y_2), \cdots, (\mathbf{z}_N, y_N) | \mathbf{z}_i = [k(\mathbf{x}_1, \mathbf{x}_i), k(\mathbf{x}_2, \mathbf{x}_i), \cdots, k(\mathbf{x}_N, \mathbf{x}_i)]^T\}$. By analogy with the linear SVMs for pattern recognition, we introduce a structural risk for a set of linear functions $\{f(\mathbf{z}) = \mathbf{w}^T \mathbf{z} + b, \mathbf{w} \in \mathbb{R}^N\}$ in the hidden space $\mathcal{Z}$ and adopt the Vapnik's ε-insensitive loss function [1], [3], [4]. Thus, we have

$$\begin{aligned} \min \quad & R(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{N} \xi_i\right) \\ s.t. \quad & y_i(\mathbf{w}^T \mathbf{z}_i + b) \leq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \cdots, N \end{aligned} \tag{2.9}$$

where $C > 0$ is a penalty factor that adjusts the balance between the empirical risk and the capacity control of the learning machine. As it stands, the programming (2.9) is a convex programming in a convex area. So its optimum solution is unique. Its Wolfe dual problem becomes

$$\begin{aligned}\min\quad & \frac{1}{2}\sum_{i,j,m=1}^{N}\alpha_i y_i \alpha_j y_j k(\mathbf{x}_i,\mathbf{x}_m)\cdot k(\mathbf{x}_m,\mathbf{x}_j)-\sum_{i=1}^{N}\alpha_i\\ s.t.\quad & \sum_{i=1}^{N} y_i\alpha_i = 0\\ & 0\le \alpha_i\le C,\quad i=1,\cdots,N \end{aligned}\tag{2.10}$$

where α_i is a positive Lagrange multiplier. The training procedure of HSSVMs for pattern recognition is to solve the Wolfe dual programming (2.10). In the process of transforming the primal programming to the dual one, we have

$$\mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i \mathbf{z}_i=\sum_{i=1}^{N}\alpha_i y_i [k(\mathbf{x}_1,\mathbf{x}_i),k(\mathbf{x}_2,\mathbf{x}_i),\cdots k(\mathbf{x}_N,\mathbf{x}_i)]^T.\tag{2.11}$$

The threshold b can be obtained by the KKT conditions [4], which is similar to the situation for SVMs. The decision function of HSSVMs takes the following form

$$y=\mathrm{sgn}(\mathbf{w}^T\mathbf{z}+b)=\mathrm{sgn}\left[\sum_{i,j=1}^{N}\alpha_i y_i k(\mathbf{x}_i,\mathbf{x}_j)k(\mathbf{x}_j,\mathbf{x})+b\right].\tag{2.12}$$

C. HSSVMs for Regression Estimation

Let a set of i.i.d. patterns be

$$X=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2)\cdots,(\mathbf{x}_N,y_N)|\mathbf{x}_i\in\mathbb{R}^d,y_i\in\mathbb{R}\}$$

and a symmetric kernel function be $k(\mathbf{x},\mathbf{y})$. The mapped patterns in the hidden space $\mathcal{Z}$ can be expressed as $\{(\mathbf{z}_1,y_1),(\mathbf{z}_2,y_2),\cdots,(\mathbf{z}_N,y_N)|\mathbf{z}_i=[k(\mathbf{x}_1,\mathbf{x}_i),k(\mathbf{x}_2,\mathbf{x}_i),\cdots,k(\mathbf{x}_N,\mathbf{x}_i)]^T\}$. By analogy with the linear SVMs for regression estimation, we introduce a structural risk for a set of linear functions $\{f(\mathbf{z})=\mathbf{w}^T\mathbf{z}+b,\mathbf{w}\in\mathbb{R}^N\}$ in the hidden space.

$$\begin{aligned}\min\quad & R(\mathbf{w},b)=\frac{1}{2}\|\mathbf{w}\|^2+C\left(\sum_{i=1}^{N}\xi(\gamma_i)+\xi(\gamma_i^*)\right)\\ s.t.\quad & y_i-(\mathbf{w}^T\mathbf{z}_i+b)\le\gamma_i+\varepsilon\\ & (\mathbf{w}^T\mathbf{z}_i+b)-y_i\le\gamma_i^*+\varepsilon\\ & \gamma_i,\gamma_i^*\ge 0,\quad i=1,\cdots,N\end{aligned}\tag{2.13}$$

where $C>0$ is a penalty factor and $\xi(\bullet)$ is a loss function which can be chosen according to the distribution of patterns under the mean of maximum likelihood estimation [5]. The Wolfe dual programming of the primal programming (2.13) can be written as

$$\begin{aligned}\min\quad & \frac{1}{2}\sum_{i,j,m=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(\mathbf{x}_i,\mathbf{x}_m)\cdot k(\mathbf{x}_m,\mathbf{x}_j)\\ & -\sum_{i=1}^{N}y_i(\alpha_i-\alpha_i^*)+\varepsilon\cdot\sum_{i=1}^{N}(\alpha_i+\alpha_i^*)\\ & +C\cdot\sum_{i=1}^{N}[\xi(\gamma_i)-\gamma_i\xi'(\gamma_i)+\xi(\gamma_i^*)-\gamma_i\xi'(\gamma_i^*)]\\ s.t.\quad & \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0\\ & C\xi'(\gamma_i)-\alpha_i-\beta_i=0\\ & C\xi'(\gamma_i^*)-\alpha_i^*-\beta_i^*=0\\ & \alpha_i^{(*)},\beta_i^{(*)},\gamma_i^{(*)}\ge 0,\quad i=1,\cdots,N\end{aligned}\tag{2.14}$$

where $^{(*)}$ is a shorthand implying both the variables with and without asterisks, α_i, α_i^*, β_i and β_i^* are Lagrange multipliers and $\xi'(\gamma_i)$ denotes the one-order differential of $\xi(\gamma_i)$ with respect to γ_i. The training procedure of HSSVMs for regression estimation is to solve the Wolfe dual programming (2.14). In the process of transforming the primal programming to the dual one, we can have

$$\begin{aligned}\mathbf{w}&=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)\mathbf{z}_i\\ &=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)\\ &\quad\times[k(\mathbf{x}_1,\mathbf{x}_i),k(\mathbf{x}_2,\mathbf{x}_i),\cdots k(\mathbf{x}_N,\mathbf{x}_i)]^T.\end{aligned}\tag{2.15}$$

The Lagrange multipliers α_i and α_i^* obtained by solving (2.14) and the threshold b computed by the KKT conditions can express the regression estimation function of HSSVMs as

$$y=\mathbf{w}^T\mathbf{z}+b=\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)k(\mathbf{x}_i,\mathbf{x}_j)k(\mathbf{x}_j,\mathbf{x})+b.\tag{2.16}$$

If the loss function is the Vapnik's ε-insensitive loss function [1], [5], HSSVMs for regression estimation can be written as

$$\begin{aligned}\min\quad & \frac{1}{2}\sum_{i,j,m=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(\mathbf{x}_i,\mathbf{x}_m)\cdot k(\mathbf{x}_m,\mathbf{x}_j)\\ & -\sum_{i=1}^{N}y_i(\alpha_i-\alpha_i^*)+\varepsilon\cdot\sum_{i=1}^{N}(\alpha_i+\alpha_i^*)\\ s.t.\quad & \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0\\ & 0\le\alpha_i^{(*)}\le C,\quad i=1,\cdots,N.\end{aligned}\tag{2.17}$$

III. Performance Analysis of HSSVMs

Let us review the decision function (2.12) of HSSVMs for pattern recognition that can be rewritten as

$$y = \operatorname{sgn}\left[\sum_{j=1}^{N} \lambda_j k(\mathbf{x}_j, \mathbf{x}) + b\right] \tag{3.1}$$

where $\lambda_j = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}$ is a linear parameter. In the same way, the regression estimation function (2.16) can be rewritten as

$$y = \sum_{j=1}^{N} \lambda_j k(\mathbf{x}_j, \mathbf{x}) + b \tag{3.2}$$

where $\lambda_j = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}$. Namely, the output function of HSSVMs can be expressed as the summation of the linear combination of parametrical hidden functions and a threshold, which is completely identical to the output functions of single-layer FNNs and RBFNs whose output nodes use linear functions. In other words, the hypothesis space of HSSVMs is completely identical to that of the single-layer FNNs and RBFNs. The only difference is that the parameters in ANNs can be optimized but the parameters in HSSVMs are prior chosen. Because of the equivalence property of the hypothesis space between FNNs and HSSVMs, we will use available conclusions in FNNs and RBFNs to analyze the performance of HSSVMs.

A. Analysis of HSSVMs for Pattern Recognition

Let us consider the separability of patterns in FNNs and RBFNs to analyze the performance of HSSVMs for pattern recognition. When an FNN or an RBFN is used to perform a complex pattern recognition problem, the problem is basically solved by transforming it into a high-dimensional feature space in a nonlinear manner. In this space, the problem will become linear. Cover's theorem on the separability of patterns indicates that higher the dimension of the feature space is, the more linearly separable this problem is. For the detail description on this theorem the reader is referred to [22].

Intuitively, it is possible to shatter two points by any linear manner in the one-dimensional (1-D) space or real axis [1] and three points in the two-dimensional (2-D) space. By analogy, it is possible to shatter $N + 1$ points in the N-dimensional space with probability 1. In what follows, we will discuss the general case simply.

Consider a family of surfaces where each naturally divides an input space into two regions. Let X denote a set of N i.i.d. patterns or $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N | \mathbf{x}_i \in \mathbb{R}^d\}$. Each of the patterns is assigned to one of two classes X_1 or X_2. This dichotomy (binary partition) of the points is said to be separable with respect to the family of surfaces if a surface exists in the family that separates the points in the class X_1 from those in the class X_2. As mentioned above, the input patterns can be mapped into a hidden space by hidden functions $\varphi(\mathbf{x})$ [see (2.2)]. The hidden space has a dimension of d_1. If there exits a d_1-dimensional vector $\mathbf{w} \in \mathbb{R}^{d_1}$ such that the following inequalities

$$\begin{aligned} \mathbf{w}^T\mathbf{z} > 0, &\quad \mathbf{z} \in X_1 \\ \mathbf{w}^T\mathbf{z} < 0, &\quad \mathbf{z} \in X_2 \end{aligned} \tag{3.3}$$

hold true, a dichotomy $\{X_1, X_2\}$ of X is said to be φ-separable. The hyperplane defined by the equation

$$\mathbf{w}^T\mathbf{z} = 0$$

which describes the separating hyper-plane in the hidden space. The equation

$$\mathbf{x} : \mathbf{w}^T\varphi(\mathbf{x}) = 0 \tag{3.4}$$

defines the separating surface in the input space. In fact, because of the linear dependence among the mapped patterns in the hidden space and the intercept errors of computation, the rank of the matrix $[\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N]$ should be less than d_1.

In view of a probabilistic experiment, the separability of a set of patterns becomes a random event that depends on the dichotomy chosen and the distribution of the patterns in the input space. Assume that the activation patterns $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ are chosen independently in the input space. Suppose also that all the possible dichotomies of $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ are equiprobable. Let $P(N, d_1)$ be the probability that a particular dichotomy picked at random is φ-separable, where the class of separating surfaces chosen has a d_1 degree of freedom. Following [20], we have

$$P(N, d_1) = \left(\frac{1}{2}\right)^{N-1} \sum_{m=0}^{d_1 - 1} \binom{N-1}{m} \tag{3.5}$$

where $\binom{N-1}{m}$ are the binomial coefficients. Equation (3.5) embodies the essence of Cover's separability theorem for random patterns, and it is a statement of the fact that the cumulative binomial distribution corresponds to the probability that $N - 1$ flips of a fair coin will result in $(d_1 - 1)$ or fewer heads.

Although the hidden-unit surfaces envisioned in the derivation of (3.5) are of a polynomial form and therefore different from those commonly used in FNNs and RBFNs, the essential content of (3.5) has general applicability. Specially, the higher the dimension d_1 of the hidden space, the closer $P(N, d_1)$ will be to unity. In other words, the complex pattern recognition problem in a high-dimensional space is more likely to be linearly separable than in a lower dimensional space. However, in some cases the use of nonlinear mapping may be sufficient to produce linear separability without having to increase the dimensionality of the hidden space.

Equation (3.5) makes it possible to compute the expected number of patterns that are linearly separable in a high-dimensional space. To explore this issue, let $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ be a sequence of random patterns as previously described. Let N be a random variable defined as the largest integer such that this

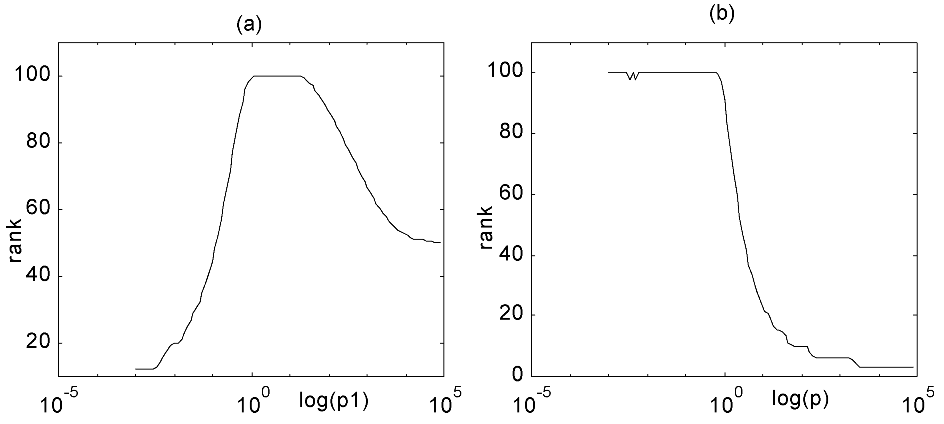


Fig. 1. The graph of the rank of the mapped pattern matrix versus the parameter (a) p_1 in $k_{\mathbf{x},\mathbf{y}} = \tanh(p_1 \cdot \mathbf{x}^T\mathbf{y} + p_2)$ where $p_2 = 0$ and (b) p in $k(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2/2p^2)$. 100 i.i.d. 2-D patterns are normal distributions with zero mean and unity variance. The curves are the average over 50 runs.

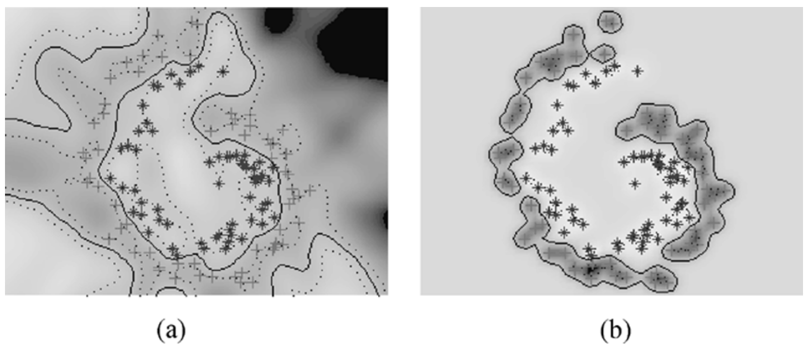


Fig. 2. The overfitting case for the two noised spirals classification problem. (a) was obtained by single-layer FNNs and (b) was obtained by RBFNs, where the real lines are the decision surfaces. Let $X_1 = \{(x,y)|\left\{\begin{smallmatrix} x=(2\theta+1)\sin\theta+n_x, n_x\sim N(0,1) \\ y=(2\theta+1)\cos\theta+n_y, n_y\sim N(0,1)\end{smallmatrix}\right\}$ and $X_2 = \{(x,y)|\left\{\begin{smallmatrix} x=(2\theta+5)\sin\theta+n_x, n_x\sim N(0,1) \\ y=(2\theta+5)\cos\theta+n_y, n_y\sim N(0,1)\end{smallmatrix}\right\}$ Each class has 63 patterns. The number of hidden nodes in both networks is 126.

sequence is φ-separable, where φ has a d_1 degree of freedom. Then from (3.5), the probability that $N = n$ is given by

$$\Pr(N = n) = P(n, d_1) - P(n+1, d_1) = \left(\frac{1}{2}\right)^n \binom{n-1}{d_1-1}. \tag{3.6}$$

The expectation and the median of the random variable N can be expressed as

$$E[N] = 2d_1 \tag{3.7}$$

and

$$\text{Median}[N] = 2d_1 \tag{3.8}$$

respectively, which means that the expected maximum number of randomly assigned patterns that are linearly separable in a d_1-dimensional space is equal to $2d_1$.

Now consider two nonlinear mappings generated by corresponding kernel functions. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. From (2.3), we know that the dimension of the hidden space is N. But the rank of a pattern matrix $[\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N]$ may be less than d_1 because of the linear dependence among the mapped patterns and the intercept errors of computation. Fig. 1 shows the order of a pattern matrix obtained by sigmoidal and Gaussian kernel mappings versus the parameters in the sigmoidal kernel and the Gaussian radial basis kernel, respectively. Note that the sigmoidal kernel is not positive definite in most cases, so it is rarely used in SVMs.

In conclusion, the dimension of a hidden space is equal to the number of patterns. Although there may exist the loss of rank, it is sufficient or even more than sufficient for the dimension of a hidden space to perform the linear classification on the patterns. It is the redundancy or too large a hidden space that makes ANNs result in overfitting and hence a bad generalization performance if the hidden space in ANNs is generated by the kernel function mapping. The overfitting case for the two noised spirals classification problem is shown in Fig. 2, where (a) was obtained by FNNs in which kernel mapping (2.4) was used, and (b) by RBFNs in which kernel mapping (2.5) was used. Obviously, the overfitting both in Fig. 2(a) and in Fig. 2(b) occurs, and that in Fig. 2(b) is worse. To avoid the overfitting problem, we introduce not only the hidden space generated by kernel mapping, but also the structural risk to control the set of hypothesis functions or the hidden space. In other words, a better generalization performance will be achieved if the right balance is struck between the empirical risk and the capacity control of a set of hypothesis functions, which HSSVMs can implement.

B. Analysis of HSSVMs for Regression Estimation

To analyze the performance of HSSVMs for regression estimation, let us first consider the approximation ability of FNNs and RBFNs. As early as the 1990s, researchers studied the approximation ability of ANNs especially for FNNs and RBFNs [16], [17], [23]–[28]. They obtained a general approximation theorem: FNNs and RBFNs whose scales are unlimited or

TABLE I
RECOGNITION RESULTS OBTAINED BY FIVE ALGORITHMS OF THE TWO-SPIRAL CLASSIFICATION PROBLEM

Learning model	**Sigmoidal FNNs**	**RBFNs**	**SVMs with Gaussian kernel**	**HSSVMs with Gaussian kernel**	**HSSVMs with sigmoidal kernel**
# Training samples	63+63	63+63	63+63	63+63	63+63
# Test samples	158+158	158+158	158+158	158+158	158+158
Parameter	Optimized by algorithm	Optimized by algorithm	p=3.0	p=6.8	p_1=0.033 p_2=0
# SVs or hidden neurons	126	126	29.1	14.9	32.8
Average recognition error over 30 runs (%)	6.5	7.0	6.0	4.6	5.3

whose numbers of the hidden nodes can be increased arbitrarily can approximate any continuous function with any accuracy [16], [17]. In the following, we cite a theorem to explain the approximation ability of FNNs and RBFNs.

Theorem 3.1 [28]: Let $1 \le h \le d, r \ge 1, n \ge 1, 1 \le p < \infty$ and h, d, r, $n \in \mathbb{N}$, where d is the dimension of the input patterns and h is the dimension of the hidden function. Let also $\phi : \mathbb{R}^h \to \mathbb{R}$ be continuously differentiable an infinite number of times in some open sphere in $\mathbb{R}^h$. Suppose that there exists $\mathbf{b}$ in this open sphere such that

$$D^{\mathbf{m}}\phi(\mathbf{b}) \neq 0, \quad \mathbf{m} \in \mathbb{Z}^h, \quad \mathbf{k} \ge 0 \tag{3.9}$$

where $D^{\mathbf{m}} f = ((\partial^{|\mathbf{m}|} f)/(\partial x_1^{m_1} \partial x_2^{m_2} \cdots \partial x_h^{m_h}))$, $\mathbf{m} = [m_1, m_2, \cdots, m_h] \in \mathbb{N}^h$, $0 \le \mathbf{m} \le \mathbf{r} = (\underbrace{r, \cdots, r}_{h})^T$ and $|\mathbf{m}| := \sum_{j=1}^{h} |m_j|$. Then there exist $d \times h$ matrices $\{A_j\}_{j=1}^{n}$ with the following property. For any r-order and d-dimensional function $f \in W_p^{r,d}$ in the Sobolev space, there exist coefficients α_j such that

$$\left\| f - \sum_{j=1}^{n} \alpha_j \phi\left[A_j(\bullet) + \mathbf{b}\right] \right\|_p \le c n^{-\frac{r}{d}} \|f\|_{W_p^{r,d}} \tag{3.10}$$

where

$$\|f\|_{W_p^{r,d}} = \sum_{0 \le \mathbf{m} \le \mathbf{r}} \|D^{\mathbf{m}} f\|_p$$
$$:= \begin{cases} \sum\limits_{0 \le \mathbf{m} \le \mathbf{r}} \left(\int |D^{\mathbf{m}} f(\mathbf{x})|^p d\mathbf{x}\right)^{\frac{1}{p}} \|f\|_p, & for\ 1 \le p < \infty \\ \sum\limits_{0 \le \mathbf{m} \le \mathbf{r}} \text{ess sup}\, |D^{\mathbf{m}} f(\mathbf{x})|, & for\ p = \infty \end{cases}$$

and $\| \bullet \|_p$ denotes the p-norm.

In what follows, we give some typical hidden functions that satisfies (3.9) in Theorem 3.1.

- **Sigmoidal function** $\phi(x) := (1 + e^{-x})^{-1}$, for $h = 1$;
- **Generalized multiquadrics function** $\phi(x) := (1 + \|\mathbf{x}\|^2)^p$, for $h \ge 1, p \in \mathbb{Z}$;
- **Thin plate spline function**

$$\phi(x) := \begin{cases} \|\mathbf{x}\|^{2p-h} \log \|\mathbf{x}\| & \text{for } d \text{ even} \\ \|\mathbf{x}\|^{2p-h}, & \text{for } d \text{ odd} \end{cases}$$
$$and\, h \ge 1, \quad p > \frac{h}{2}, \quad p \in \mathbb{Z};$$

- **Gaussian radial basis function** $\phi(x) := \exp(-\|\mathbf{x}\|^2/p^2)$, $h \ge 1, p \in \mathbb{R}$.

Theorem 3.1 implies that if only the scale of single-layer FNNs is sufficiently large, these ANNs whose node function satisfies (3.9) can approximate any differentiable function in the Sobolev space with any accuracy. Moreover the lower bound on the approximation error is less than $cn^{-r/d}$, where c is a constant, r is the order of the Sobolev space and d is the dimension of input data.

In regression estimation there exists not only the approximation error generated by the improper hypothesis space but also the estimation error resulting from a finite number of patterns, noised patterns, and others. Generally the larger the hypothesis space is, the smaller the approximation error and the larger the estimation error and vice versa. From the above theorem, when the number of patterns is proper, it is sufficient or more than sufficient for the hypothesis space generated by hidden functions whose number is equal to that of patterns to approximate a differential function in the Sobolev space. Therefore, if the common algorithms of ANNs are adopted an overfitting problem will result. By analogy with SVMs, we introduce the structural risk in the hidden space and can obtain a better generalization performance by implementing the right balance between the empirical risk and the capacity control of a set of hypothesis functions, which can be implemented by HSSVMs for regression estimation. It is worth noticing that the hypothesis space of HSSVMs for regression estimation is slightly different

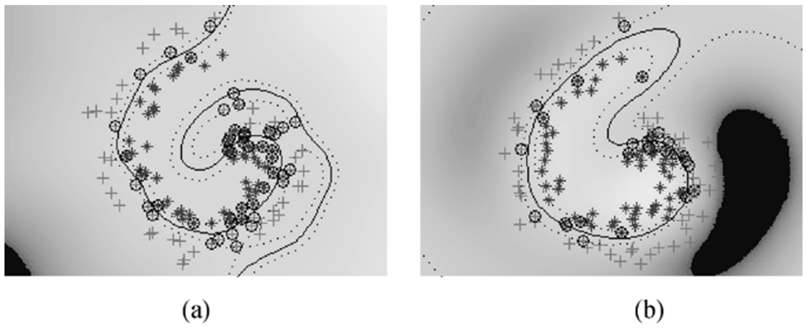


Fig. 3. Decision surfaces obtained by (a) HSSVMs with the sigmoidal kernel and (b) HSSVMs with the Gaussian kernel for two-spiral classification problem. Symbol $*$ denotes the positive sample and $+$ the negative one. The support vectors are circled. The real lines denote the decision surfaces and the dotted lines $f(\mathbf{x}) = \pm 1$.

from that of FNNs. The hypothesis space of HSSVMs is identical to that of FNNs if the weight matrixes between the input layer and the hidden layer are all the unit matrixes and the parameters in hidden functions are fixed in FNNs. It is still an open problem to give an exact conclusion on how the constraints described above affect the approximation error in principle. The effect of fixing the parameter in sigmoidal hidden function on the approximation error was studied in [24] and an asymptotical bound of the approximation error with respect to the number of hidden nodes and the dimension of the input pattern was given. The asymptotical bound shows that even if the parameter of the sigmoidal hidden function is fixed, FNNs still have a better approximation performance. The experiments in this paper also validate this.

C. Comparison of HSSVMs and SVMs

From Section II, we know that the feature spaces of the traditional SVMs and HSSVMs are different. The traditional nonlinear SVMs are constructed in the feature space mapped by the Mercer kernel function. If the kernel function does not satisfy the Mercer condition, the feature space that SVMs lives in does not exist. So the traditional nonlinear SVMs are impossible without the Mercer kernel function. Here we will show that the unique condition for the kernel function in HSSVMs is symmetry. No other conditions are required.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N | \mathbf{x}_i \in \mathbb{R}^d\}$. Compare the output functions of HSSVMs (2.12) and (2.16) with those of SVMs [4], [5], we define the following symmetric kernel function depending on the input patterns

$$K(\mathbf{x},\mathbf{y}) := \sum_{j=1}^{N} k^T(\mathbf{x}_j,\mathbf{x})k(\mathbf{x}_j,\mathbf{y}) = \sum_{j=1}^{N} k(\mathbf{x},\mathbf{x}_j)k(\mathbf{x}_j,\mathbf{y}). \tag{3.11}$$

The defined kernel function can be expressed as the product of two kernel functions, so it is called the square kernel.

Theorem 3.2: Let a set of i.i.d. patterns be $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N | \mathbf{x}_i \in \mathbb{R}^d\}$ and the square kernel depending on patterns be $K(\mathbf{x},\mathbf{y}) := \sum_{j=1}^{N} k^T(\mathbf{x}_j,\mathbf{x})k(\mathbf{x}_j,\mathbf{y}) = \sum_{j=1}^{N} k(\mathbf{x},\mathbf{x}_j)k(\mathbf{x}_j,\mathbf{y})$. Then the square kernel $K(\mathbf{x},\mathbf{y})$ is an admissible Mercer kernel whatever the kernel function $k(\mathbf{x},\mathbf{y})$ is.

The proof of Theorem 3.2 is shown in the Appendix. We can simply conclude that the SVMs that use the square kernel $K(\mathbf{x},\mathbf{y})$ are completely identical to HSSVMs that use the hidden function $k(\mathbf{x},\mathbf{y})$. Thus, HSSVMs implement the structural risk minimization in a hidden space, which makes that HSSVMs have as good a generalization performance as SVMs and avoid the overfitting problem. In other aspects, the BP algorithm requires derivativeness for the node functions, and SVMs requires that kernel functions satisfy the rigorous Mercer's condition. Here we relax the constraints for nonlinear nodes or kernel functions in HSSVMs. According to Theorem 3, we know that the square kernel $K(\mathbf{x},\mathbf{y})$ is certainly positive definite, whatever $k(\mathbf{x},\mathbf{y})$ is. It only requires that the kernel functions satisfy symmetry that is the essential condition for a kernel function. Hence the range of nonlinear mapping (kernel) functions used in HSSVMs becomes larger than that for the traditional SVMs. The training procedure of HSSVMs amounts to solving a constrained quadratic programming whose scale is the same as that for SVMs. So HSSVMs have the same computation complexity as SVMs. And the solution of HSSVMs is a global optimum. Moreover for the large-scale problem, HSSVMs can adopt the decomposition methods used in SVMs.

IV. Simulations

To evaluate the performance of HSSVMs for pattern recognition and regression estimation, we performed four kinds of experiments: two-spiral classification problem, handwritten digit recognition problem, UCI data sets recognition, and function estimation problem. For the sake of comparison, different algorithms used the same input sequence. These algorithms include FNNs with the sigmoidal function, RBFNs, SVMs with the Gaussian kernel, HSSVMs with the sigmoidal kernel (2.4), and the Gaussian radial basis kernel (2.5).

A. Two-Spiral Classification problem

The two-spiral classification problem is referred to as the "touchstone" to test the ability of the classification learning algorithm [29]. Two classes of samples are defined by

$$X_1 = \left\{(x_1,x_2) \middle| \begin{cases} x_1 = (2\theta+1)\sin\theta + n_1, n_1 \sim N(0,1) \\ x_2 = (2\theta+1)\cos\theta + n_2, n_2 \sim N(0,1) \end{cases}\right\}$$

and

$$X_2 = \left\{(x_1,x_2) \middle| \begin{cases} x_1 = (2\theta+5)\sin\theta + n_1, n_1 \sim N(0,1) \\ x_2 = (2\theta+5)\cos\theta + n_2, n_2 \sim N(0,1) \end{cases}\right\}$$

TABLE II
RECOGNITION RESULTS OBTAINED BY FIVE ALGORITHMS FOR THE HANDWRITTEN DIGIT RECOGNITION PROBLEM

Learning model	**Sigmoidal FNNs**	**RBFNs**	**SVMs with Gaussian kernel**	**HSSVMs with Gaussian kernel**	**HSSVMs with sigmoidal kernel**
# Training samples "7" + "9"	65+64	65+64	65+64	65+64	65+64
# Test samples	1028+1009	1028+1009	1028+1009	1028+1009	1028+1009
Parameter	Optimized by algorithm	Optimized by algorithm	$p=1.5$	$p=0.5$	$p_1=150$ $p_2=0$
# SVs and hidden neurons	129	129	56	49	24
Recognition error (%)	7.0	9.3	4.3	5.0	6.6

TABLE III
RESULTS OF SVMS AND HSSVMS FOR THREE UCI DATA SETS

UCI data sets	**SVMs**			**HSSVMs**		
	Average optimum p	# Average SVs	Cross Validation error (%)	Average optimum p	# Average SVs	Cross Validation error (%)
Wisconsin Breast Cancer	2.8	247.1	2.15	12.1	55.8	2.72
Ionosphere	3.1	117.7	6.11	2.4	93.8	5.56
Diabetes	42.0	392.6	23.75	37.0	398.2	23.83

Each class has 63 training samples. Table I shows the average results obtained by these five algorithms over 30 runs. The decision surfaces obtained by HSSVMs with the sigmoidal kernel and the Gaussian kernel are shown in Fig. 3. For the results obtained by ANNs the reader is referred to Fig. 2. The introduction of structural risk is to avoid overfitting problem. Compared with Fig. 2, the decision surfaces obtained by HSSVMs are a smoother or have a smaller oscillation frequency. In terms of the regularization theory, it has a small smoothness degree.

B. Handwritten Digit Recognition Problem

The experimental data is the MNIST database of 60 000 training and 10 000 testing handwritten digits from the AT&T Research Labs,[1] which have been taken as the experimental data in [30]. Since the database is so large, we only have obtained two-class examples belonging to classes "7" and "9," respectively, and normalized these examples. The results obtained by five algorithms are shown in Table II.

[1]URL: http://www.research.att.com/~yann/ocr/mnist

C. UCI Data Sets Recognition

To evaluate the performance of HSSVMs, we make a comparison between the traditional classical SVMs and HSSVMs on the three real world datasets from the UCI Machine Learning Repository (available at the UCI Machine Learning Repository [31]). All involve only 2-way classification. These sets are Wisconsin Breast Cancer data set including 458 examples of "benign" and 241 examples of "malignant," nine attributes for each example; Ionosphere data set including 225 examples of "good" and 126 examples of "bad," nine attributes for each example; PIMA Indians Diabetes data set including 500 plus examples and 268 minus examples, eight attributes for each example. In our experiment, 10-block cross validation is employed for each dataset. Each original dataset is divided into 10 blocks. Each block is used for test, while the other ones are used to train the algorithms. The 10 test blocks form the entire

TABLE IV
ESTIMATION RESULTS OBTAINED BY FIVE ALGORITHMS FOR THE FUNCTION REGRESSION PROBLEM

Learning model	**Sigmoidal FNNs**	**RBFNs**	**SVMs with Gaussian kernel**	**HSSVMs with Gaussian kernel**	**HSSVMs with sigmoidal kernel**
# Training samples	101	101	101	101	101
# Test samples	501	501	501	501	501
$\sigma = 0$, SVMs & HSSVMs: C=100, ε=0.01					
Parameter	Optimized by algorithm	Optimized by algorithm	$p = 0.12$	$p = 0.08$	$p_1 = 15$ $p_2 = 1.3$
# SVs or hidden neurons	101	101	21.8	23.2	78.9
Average mean-squared errors over 30 runs (10^{-5})	940	8.05	5.14	5.14	130
$\sigma = 0.1$, SVMs & HSSVMs: C=100, ε=0.01					
Parameter	Optimized by algorithm	Optimized by algorithm	$p = 0.1$	$p = 0.1$	$p_1 = 15$ $p_2 = 1.3$
# SVs or hidden neurons	101	101	91.4	89.8	90.5
Average mean-squared errors over 30 runs (10^{-5})	8.9	20	2.9	2.2	2.9

original dataset. Table III shows the cross validation error and the average number of the support vectors for each of the four data sets. The RBF kernel is adopted and the near-optimum parameters are selected for every training data block.

D. Function Regression Estimation Problem

Let the training samples $(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)$ be defined by the nonlinear noised sinc function $y_i = \sin(30x_i)/(30x_i) + \sigma n_i$, where $x_i \sim U[-1, 1]$ (or x has a uniformly distribution in the interval [−1, 1]), $y_i \in \mathbb{R}$, $n_i \sim N(0, 1)$ (or noise has a normal distribution with zero mean and unity variance) and random variable x_i is independent of n_i. This regression problem was often used to test the validity of learning algorithms [1]. The experimental results are shown in Table IV and Fig. 4.

V. CONCLUSION

Hidden space support vector machines for pattern recognition and regression estimation are presented based on the concepts of hidden function mapping and hidden space in FNNs and RBFNs. The input patterns are mapped into a high-dimensional hidden space by a nonlinear hidden function and then a structural risk is introduced into the hidden space to construct HSSVMs. Moreover, it is not important for the differentiability of the nonlinear function in HSSVMs. Compared to SVMs, HSSVMs can adopt more kinds of kernel functions because the positive definite property of the kernel function is not a necessary condition. The research on the classification and the approximation ability of FNNs and RBFNs in the 1990s has indicated that they can magnificently classify complex patterns and approximate functions in the Sobolev space. According to the statistical learning theory, SVMs introduce a structural risk so as to have s good generalization. An analysis of our algorithms implies that HSSVMs inherit the merits of ANNs and SVMs. Experiments on artificial and real-world domains confirm the feasibility and the validity of HSSVMs.

APPENDIX
THE PROOF OF THEOREM 3.2

Proof: Given a set of i.i.d. patterns $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N | \mathbf{x}_i \in \mathbb{R}^d\}$ and a symmetric kernel function

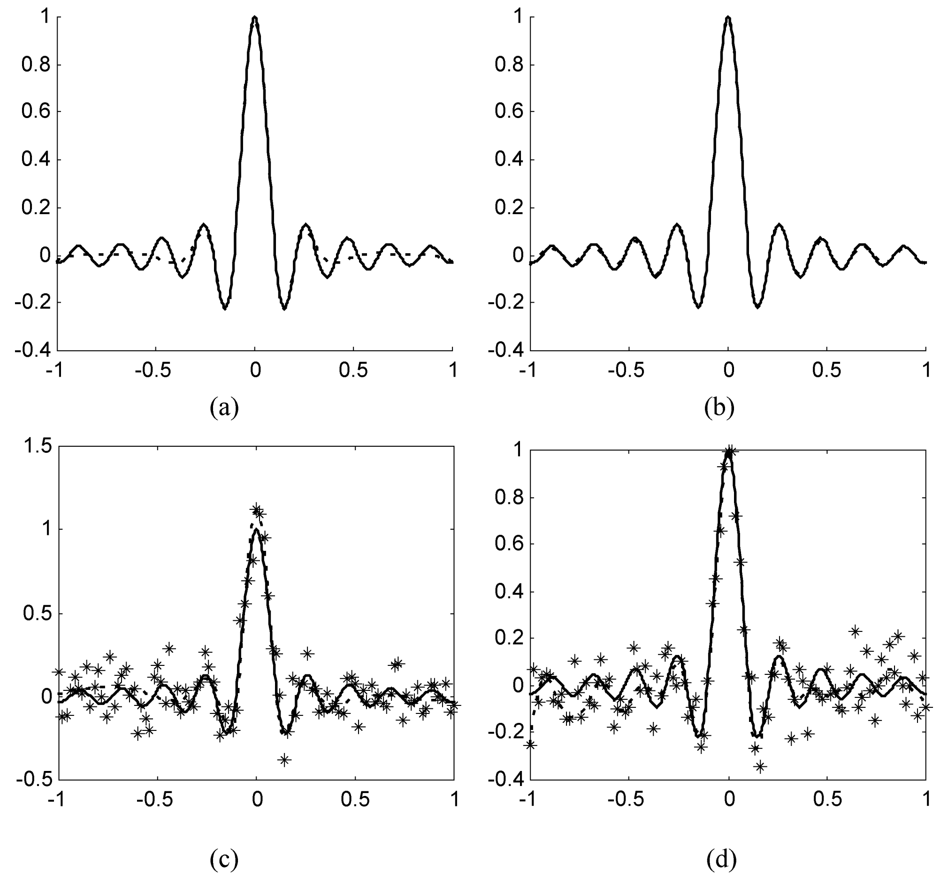


Fig. 4. Approximation results obtained by HSSVMs with the sigmoidal kernel (a) and (c) and the Gaussian kernel (b) and (d), where (a) and (b) are obtained under noise level $\sigma = 0$ and (c) and (d) are obtained under noise level $\sigma = 0.1$. In these figures, $*$ denotes training samples, the real line denotes the original function and the dotted line the estimated function.

depending on patterns $K(\mathbf{x},\mathbf{y}) := \sum_{j=1}^{N} k^H(\mathbf{x}_j,\mathbf{x})k(\mathbf{x}_j,\mathbf{y}) = \sum_{j=1}^{N} k(\mathbf{x},\mathbf{x}_j)k(\mathbf{x}_j,\mathbf{y})$, it is sufficient for us to prove that this symmetric kernel satisfies the Mercer's condition [6]. Namely

$$\begin{aligned}\int\int K(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} &= \int\int \sum_{j=1}^{N} k(\mathbf{x},\mathbf{x}_j)k(\mathbf{x}_j,\mathbf{y})d\mathbf{x}d\mathbf{y}\\ &= \int\int \sum_{j=1}^{N} k^H(\mathbf{x}_j,\mathbf{x})k(\mathbf{x}_j,\mathbf{y})d\mathbf{x}d\mathbf{y}\\ &= \sum_{j=1}^{N}\int k^H(\mathbf{x}_j,\mathbf{x})d\mathbf{x}\int k(\mathbf{x}_j,\mathbf{y})d\mathbf{y}\\ &= \sum_{j=1}^{N}\left[\int k(\mathbf{x}_j,\mathbf{x})d\mathbf{x}\right]^H\int k(\mathbf{x}_j,\mathbf{y})d\mathbf{y}\\ &= \sum_{j=1}^{N}\left[\int k(\mathbf{x}_j,\mathbf{x})d\mathbf{x}\right]^2 \geq 0\end{aligned}$$

which indicates that $K(\mathbf{x},\mathbf{y})$ is positive definite and therefore it is an admissible Mercer kernel.

This completes the proof of Theorem 3.2.

Acknowledgment

The data sets of Wisconsin breast cancer, Diabetes and Ionosphere come from the University of Wisconsin Hospitals, Space Physics Group, Applied Physics Laboratory, Johns Hopkins University and National Institute of Diabetes and Digestive and Kidney Disease, respectively. Thanks go to Dr. W. H. Wolberg, V. Sigillito, etc. for providing the data. Thanks are also owed to those who contributed to the data sets used in this paper. The author would like to thank Prof. D. Qing for helping to make this paper more clear and readable.

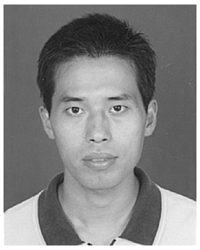
References

[1] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[2] ——, "An overview of statistical learning theory," *IEEE Trans. Neural Networks*, vol. 10, pp. 988–999, 1999.
[3] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press, 2000.
[4] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
[5] A. Smola, "Regression estimation with support vector learning machines," Master's thesis, Technische Universität München, Germany, 1996.
[6] J. Mercer, "Function of positive and negative type and their connection with the theory of integral equations," *Philos. Trans. Roy. Soc. London*, vol. A 209, pp. 415–446, 1909.
[7] S. Saitoh, *Theory of Reproducing Kernels and Its Applications*. Harlow, UK: Longman Scientific & Technical, 1988.
[8] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 686, pp. 337–404, 1950.
[9] S. Haykin, *Neural Networks, a Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.
[10] K. Gurney, *Introduction to Neural Networks*. London, UK: UCL, 1997.

[11] L. C. Jiao, *Neural Networks System Theory (in Chinese)*. Xi'an, China: Xidian University Press, 1996.
[12] *Neural Networks Theory Technology and Application, IEEE Technology Updates Series*, P. K. Simpson, Ed., The Inst. Elect. Electron. Eng.., New York, 1996.
[13] F. Luo, *Applied Neural Networks for Signal Processing*. Cambridge, UK: Cambridge, 1997.
[14] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986, vol. 1–2.
[15] M. J. D. Powell, "Radial basis functions for multivariable interpolation: a review," in *Algorithms for Approximation*, J. C. Mason and M. G. Cox, Eds. Oxford: Clarendon, 1987.
[16] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
[17] H. N. Mhaskar and C. A. Micchelli, "Approximation by superposition of a sigmoidal function and radial basis functions," *Advances in Applied Mathematics*, vol. 13, pp. 350–373, 1992.
[18] J. A. K. Suykens and J. Vandewalle, "Training multiplayer perceptron classifiers based on a modified support vector method," *IEEE Trans. Neural Networks*, vol. 10, July 1999.
[19] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, Singapore: World Scientific, 2002.
[20] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A. Smola, "Support Vector Machine—Reference Manual," Dept. Computer Science, Royal Holloway, Univ. London, Egham, UK, Technical Report CSD-TR-98-03, 1998.
[21] G. Wahba, "Support vector machines, reproducing kernel hilbert spaces and the randomized GACV," 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. 69–88.
[22] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comp.*, vol. EC-14, pp. 326–334, 1965.
[23] L. K. Jones, "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training," *The Annals of Statistics*, vol. 20, pp. 608–613, Mar. 1992.
[24] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Math. Contr., Signal and Syst.*, vol. 2, pp. 303–314, 1989.
[25] A. R. Barron, "Universal approximation bounds for superposition of a sigmoidal function," *IEEE Trans. Inform. Theory*, vol. 39, pp. 930–945, 1993.
[26] J. Park and I. W. Sandberg, "Universal approximation using radial basis function networks," *Neural Comput.*, vol. 3, pp. 246–257, 1991.
[27] M. J. D. Powell, "The theory of radial basis function approximation," in *Advances in Numerical Analysis III, Wavelets, Subdivision Algorithms and Radial Basis Functions*, W. A. Light, Ed. Oxford, UK: Clarendon, 1992, pp. 105–210.
[28] H. N. Mhaskar, "Neural networks for optimal approximation of smooth and analytic functions," *Neural Comput.*, vol. 8, pp. 164–177, 1996.
[29] K. J. Lang and M. J. Witbrock, "Learning to tell two spirals apart," in *Proc. 1989 Connectionis Models Summer School*, 1989, pp. 52–61.
[30] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," in *The Proc. First Int. Conf. Knowledge Discovery & Data Mining*, M. Fayyad and R. Uthurusamy, Eds. Menlo Park, CA: AAAI, 1995.
[31] UCI Repository of Machine Learning Databases, C. J. Merz and P. M. Murphy. (1996). *http://www.ics.uci.edu/~mlearn/MLPepository.html* [Online]

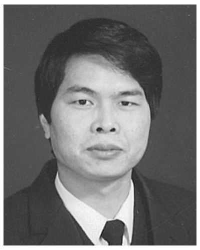
Li Zhang was born in Guizhou, China, on April 9, 1975. She received the B. S. degree in 1997 and the Ph.D. degree in 2002 in electronic engineering from Xidian University, Xi'an, China.

She is currently a Postdoctoal researcher with the Institute of Automation, Shanghai Jiao Tong University, Shanghai, China. Her research interests have been in the areas of pattern recognition, machine learning, and data mining.



Weida Zhou was born in Zhejiang, China, on November 21, 1974. He received the B.S. degree in 1996 and the Ph.D. degree in 2003 in engineering from Xidian University, Xi'an, China.

He is currently a faculty member with the School of Electronic Engineering, Xidian University. His research interests include machine learning, learning theory, and data mining.



Licheng Jiao (M'87–SM'91) was born in Shaanxi, China, on October 15, 1959. He received the B.S. degree from Shanghai Jiaotong University, China, in 1982 and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

From 1984 to 1986, he was an Assistant Professor with the Civil Aviation Institute of China, Tianjing, China. During 1990 and 1991, he was a Postdoctoral Fellow with the Key Lab for Radar Signal Processing, Xidian University, Xi'an, China. He is currently Professor and Dean of the School of Electronic Engineering in Xidian University. His current research interests include signal and image processing, nonlinear circuits and systems theory, learning theory and algorithms, optimization problems, wavelet theory, and data mining. He is the author of four books: *Theory of Neural Network Systems*, *Theory and Application on Nonlinear Transformation Functions*, *Neural Computing*, and *Applications and Implementations of Neural Networks* (1990, 1992, 1993, and 1996, respectively, Xidian University Press: Xi'an). He is the author or coauthor of more than 150 scientific papers.

Dr. Jiao is a member of several international scientific committees.