

On a Class of Support Vector Kernels Based on Frames in function Hilbert Spaces

J.B. Gao, C.J. Harris and S. R. Gunn

Image, Speech and Intelligent Systems Research Group
Dept. of Electronics & Computer Science
University of Southampton, SO17 1BJ, U.K.
jg@ecs.soton.ac.uk, cjh@ecs.soton.ac.uk, S.R.Gunn@ecs.soton.ac.uk

Abstract

In recent years there has been an increasing interest in kernel-based techniques, such as Support Vector Techniques, Regularization Networks and Gaussian Processes. There are inner relationships among those techniques with the kernel function playing a central role. This paper discusses a new class of kernel functions derived from the so-called frames in a function Hilbert space.

Keywords: *Wavelet, Kernel, Frames, Support Vector Machines*

1 Introduction

The problem of empirical data modelling is germane to many applications of complex process modelling where there exists observational data and little or no phenomenological knowledge. In empirical data modelling, a process of induction is used to build up a model of the system from examples. Ultimately, the quantity and quality of the observations will govern the performance of a model. However, the choice of modelling approach will also influence the performance of a model. By its observational nature, data is finite and sampled; typically this sampling is non-uniform and due to the high dimensional nature of the problem, the data will form only a sparse distribution in the input space. Consequently, the problem is nearly always ill-posed ([Poggio et al. 1985](#)). To address the ill-posed nature of the problem it is necessary to convert the problem to one that is well-posed. For a problem to be well-posed, a unique solution must exist that varies continuously with the data. Conversion to a well-posed problem is typically achieved with some form of capacity control, which aims to balance the fitting of the data with constraints on the model flexibility, producing a robust model that generalises

successfully. One of the approaches to restoring the well posedness is the regularization method (Tikhonov and Arsenin 1977).

In recent years the number of different Support Vector algorithms (Vapnik 1998; Smola and Schölkopf 1998; Smola et al. 1998; Bennett 1999; Schölkopf et al. 2000) and other kernel based methods (Schölkopf et al. 1998) has grown rapidly. This is due to both the success of the method (Burgess and Schölkopf 1997) and the need to adapt it to particular problems. A Support vector machine (SVM) is a classification/approximation technique derived by Vapnik in the framework of Structural Risk Minimization, which aims at building parsimonious models, in the sense of statistical learning theory (Vapnik 1998). The formulation embodies the Structural Risk Minimization (SRM) principle, which has been shown to be superior (Vapnik 1998; Gunn et al. 1997) to traditional Empirical Risk Minimisation (ERM) principle, employed by many conventional neural networks. SRM minimises an upper bound on the expected risk, as opposed to ERM that minimises the error on the training data. It is this difference which equips SVM with a greater ability to generalise, which is the goal in statistical learning. Support vector machines have been proposed for pattern recognition, regression estimation, operator inversion, general cost functions, arbitrary kernel expansions, modified regularization methods, etc.

The main purpose of this paper is to investigate a new class of kernel functions for the support vector machine generated from a frame in a function Hilbert space. Section 2 is dedicated to introducing the concepts of both the support vector kernel and regularization operator; In section 3, some basic properties of the so-called frame in an abstract Hilbert space are reviewed; In section 4, construction methods for the kernel function are proposed using the Green's function of the frame operator. This kernel function satisfies the self-consistency condition with respect to the analysis operator of frames.

2 Support vector kernel and regularization operator

Support vector algorithms exploit the idea of mapping data into a high dimensional feature space where they can apply a linear algorithm. Usually, this map and many of its properties are unknown. Instead of evaluating this mapping explicitly, one uses an integral operator kernel $k(x, y)$ which corresponds to the dot product of the mapped data in a high dimensional space, (Aizerman et al. 1964; Boser et al. 1992), i.e.

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle \quad (2.1)$$

where $\Phi : \mathbb{R}^n \rightarrow \mathcal{F}$ denotes the map into feature space \mathcal{F} with a dot product $\langle \cdot, \cdot \rangle$. Thereby this algorithm can compute a nonlinear function in the space of the input data \mathbb{R}^n . These functions take the form

$$f(x) = \langle w, \Phi(x) \rangle + b$$

where $w \in \mathcal{F}$ is a vector in the feature space and $b \in \mathbb{R}$ is the bias of the “linear” model.

Given a training dataset,

$$\mathcal{D} = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \mathbb{R}, i = 1, 2, \dots, N\}$$

one tries to minimise the regularized risk functional

$$R_{\text{reg}}[f] = CR_{\text{emp}}[f] + \frac{1}{2}\|w\|^2 = \frac{C}{N} \sum_{i=1}^N c(f(x_i), y_i) + \frac{1}{2}\|w\|^2 \quad (2.2)$$

where $c(f(x_i), y_i)$ is, in general, a convex cost function determining the loss for deviation between the estimated value, $f(x_i)$, and the target value, y_i , where $C \in [0, +\infty)$ is a regularization constant controlling the amount of regularisation. For the class of convex cost functions, equation (2.2) can be solved using a convex mathematical programming algorithm. The solution of the SVM algorithm can be represented as a weighted linear combination of kernels,

$$f(x) = \sum_{i=1}^N \alpha_i k(x_i, x) + b, \quad (2.3)$$

where these kernels are ‘centered’ on the data points (Smola and Schölkopf 1998). That is, f can be expressed in terms of k alone (the mapping Φ appears only implicitly through the dot product in \mathcal{F}). If $k(x, y)$ is a function that can be computed easily, it is reasonable to use k instead of Φ . The obvious properties of kernel function $k(x, y)$ are that it is positive in the sense that $k(x, x) \geq 0$, and symmetric, i.e., $k(x, y) = k(y, x)$ due to the dot product relationship (2.1).

In the above argument, the kernel function defines the class of functions in the model space. Thus, the question is whether it is possible to reverse the way of reasoning for kernels, i.e., under which conditions a symmetric kernel $k(x, y)$ corresponds to a dot product in some feature space \mathcal{F} . In (Aizerman et al. 1964; Boser et al. 1992) an answer is given. If k is a symmetric positive definite function, i.e., if it satisfies Mercer’s condition, then the kernel k represents a dot product in some feature space \mathcal{F} (Aronszajn 1950; Girosi 1998; Wahba 1990). In fact we have, see (Smola 1998) page 38,

Theorem 1. *Suppose $\mu(\Omega) < \infty$ and $k \in L^2(\Omega \times \Omega)$ is a symmetric kernel such that the integral operator (from $L^2(\Omega)$ to $L^2(\Omega)$) T_k*

$$T_k f(\cdot) = \int_{\Omega} k(\cdot, y) f(y) dy$$

is positive. Then there exists a set of orthonormal eigenfunctions $(\psi_i(\cdot))_{i=1}^{\infty}$ and positive eigenvalues $(\lambda_i)_{i=1}^{\infty}$ of T_k such that k can be expanded into a uniformly convergent series

$$k(x, y) = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(y)$$

which holds for almost all (x, y) . In this case, the implicit mapping Φ can be represented as

$$\Phi(x) := \sum_{i=1}^{\infty} \sqrt{\lambda_i} \psi_i(x) e_i$$

where $\{e_i\}$ is an orthonormal basis of feature space \mathcal{F} .

The k satisfying the condition of Theorem 1 is called a Mercer kernel, and it satisfies

$$\int_{\Omega \times \Omega} g(x) k(x, y) g(y) dx dy \geq 0 \quad \forall g \in L^2(\Omega)$$

see (Smola 1998). However, defining Φ implicitly through k also creates some problems. This method does not give us any general information about which kernel would be better than another, or why mapping into a very high dimensional space often provides good results.

In (Smola et al. 1998), the authors showed how these kernels $k(x, y)$ correspond to regularization operators P with the link being that k is the Green's function of P^*P where P^* is the adjoint operator of P .

In regularization networks one minimizes the empirical risk functional $R_{\text{emp}}[f]$ plus a regularization term,

$$Q[f] := \frac{1}{2} \|Pf\|^2,$$

defined by a regularization operator P . The operator P is a positive semidefinite operator mapping from the Hilbert Space, H , of functions, f , into a dot product space, \mathcal{F} , such that the expression $\langle Pf, Pg \rangle$ is well defined. Similar to (2.2), the risk functional is given by,

$$R_{\text{reg}}[f] = CR_{\text{emp}}[f] + \frac{1}{2} Q[f] = \frac{C}{N} \sum_{i=1}^N c(f(x_i), y_i) + \frac{1}{2} \|Pf\|^2 \quad (2.4)$$

Thus by choosing a suitable operator that penalizes large variations of f one can reduce the well-known overfitting effect. For some choices of the regularization operator P , it is easy to prove (Girosi 1998) that the solution of the variational problem (2.4) always has the form (2.3) by using an expansion of f in terms of some symmetric function $k(x_i, x)$.

Unfortunately, this setting of the problem may not preserve sparsity. Thus it leads to the question if and under which conditions, given a suitable cost function, regularization networks might lead to a sparse decomposition, i.e., only a few of the expansion coefficients α_i in f would differ from zero. As shown in (Smola et al. 1998), a sufficient condition is

$$k(x, y) = \langle (Pk)(x, \cdot), (Pk)(y, \cdot) \rangle \quad (2.5)$$

which is called self-consistency condition.

Theorem 2. *Let P be a regularization operator, and G be the Green's function of P^*P . Then G is a Mercer kernel satisfying the self-consistency condition. SV machines using G minimize the risk functional (2.4) with regularization operator P .*

The Green's function of the operator P^*P is a function $G(x, y)$ satisfying

$$(P^*PG(\cdot, y))(x) = \delta_y(x) \quad \forall x, y \in \mathbb{R}^n$$

where $\delta_y(x)$ is the representer of the evaluation functional at y , see (Smola 1998) page 40. Theorem 2 says that using the Green's function of P^*P as the kernel, then SV machines and regularization networks are equivalent in the sense that the solutions of (2.2) and (2.4) are the same. In this case, the feature mapping Φ can be defined as $\Phi : x \rightarrow (PG)(\cdot, x)$. The goal here is to discuss those kernels which are generated by the regularization operators associated with some frames in a function Hilbert space.

3 Frame Concepts in Hilbert Space

The abstract notion of a frame (or, in other words, a stable representation system) in a Hilbert space was firstly introduced by Duffin and Schaeffer (1952). The first survey with an emphasis on frames was (Heil and Walnut 1989), see also (Chui (1992), Chapter 3) and (Daubechies (1992), Chapter 3). Let H be a Hilbert space generated by functions defined on Ω , for instance, the square integrable function space $L^2(\mathbb{R}^n)$ or $L^2(\Omega)$ with domain $\Omega \subset \mathbb{R}^n$. Denote by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ the inner product and the associated norm on H , respectively.

Let $F = \{f_i\} \subset H$ be an (at most) countable system of elements in H .

Definition 1. *If the system $F = \{f_i\}$ is a dense subset¹ of H and there exist two constants $0 < A \leq B < \infty$ such that*

$$A\|f\|^2 \leq \sum_k |\langle f, f_i \rangle|^2 \leq B\|f\|^2, \quad \forall f \in H \quad (3.1)$$

then the system F is called a frame in H .

Let F be a frame, furthermore, if F is also a Schauder basis² in H , then F is called a Riesz stable basis of H .

Orthonormal systems in H are obviously frames in H due to Parseval's equality. As F is dense in H , each element in H can be represented as a infinite linear combination of elements in F , but the representation form may not be unique. That means a frame F in H may not be a basis for H , since F may not be linearly independent.

¹Here it means that each element f in H is the limit of a sequence consisting of a finite linear combination of the elements in F .

²A Schauder basis in H is a sequence $(f_n)_n$ with the property that for each element f in H there exists a unique constant sequence $(c_n)_n$ such that $x = \sum_n c_n f_n$.

Let F be a frame in a Hilbert space H , then we can define the so-called synthesis operator R given by

$$c = \{c_i\} \in \ell^2 \longrightarrow Rc = \sum_i c_i f_i \in H \quad (3.2)$$

and the operator R is a bounded linear operator from ℓ^2 to H . Its adjoint operator $R^* : H \rightarrow \ell^2$ takes the form

$$f \in H \longrightarrow R^* f = \{\langle f, f_i \rangle\} \in \ell^2 \quad (3.3)$$

and is called an analysis operator associated with the frame F .

The two-sided inequality (3.1) can be expressed as:

$$A\langle f, f \rangle \leq \|R^* f\|_{\ell^2}^2 = \langle RR^* f, f \rangle \leq B\langle f, f \rangle, \quad \forall f \in H$$

which shows the symmetric operator $Q = RR^* : H \rightarrow H$ is boundedly invertible (Dahmen 1997; Lorentz and Oswald 2000). We call Q the frame operator with respect to F .

Theorem 3. (Benedetto and Frazier 1994; Lorentz and Oswald 2000) *Let F be a frame in a Hilbert space H and Q be the frame operator of F , then*

1. Q is a symmetric positive definite operator on H , and $\tilde{F} = \{\tilde{f}_i = Q^{-1} f_i\}$ is also a frame in H and is called a dual frame of F .
2. Each element $f \in H$ can be expanded in the following form

$$f = \sum_i \langle f, Q^{-1} f_i \rangle f_i = \sum_i \langle Q^{-1} f, f_i \rangle f_i \quad (3.4)$$

3. Furthermore if F is a Riesz stable basis, then F and \tilde{F} are dual to each other in the sense that

$$\langle f_j, \tilde{f}_i \rangle = \delta_{j,k}. \quad (3.5)$$

i.e., F and \tilde{F} are biorthogonal systems in H .

The frame decomposition (3.4) is optimal in a certain sense. In fact, given F , if there exists any decomposition $f = \sum_i c_i f_i$, then

$$\sum_i c_i^2 \geq \sum_i |\langle f, \tilde{f}_i \rangle|^2.$$

By the frame representation (3.4), the frame operator Q has following expansion, for any $f \in H$,

$$Qf = \sum_i \langle f, f_i \rangle f_i \quad (3.6)$$

Computing with frame representations requires the application of Q^{-1} on certain elements of H , or equivalently, to solve the frame operator equation

$$Qg = f \quad (3.7)$$

for a given f . It has been proposed in (Duffin and Schaeffer 1952) that a simple Richardson iteration

$$g^{(n+1)} = g^{(n)} - \omega(Qg^{(n)} - f) \quad n \geq 0 \quad (3.8)$$

with parameter $\omega = 2/(A + B)$ and arbitrary starting element $g^{(0)}$ could be used.

4 Kernel functions based on frames

In this section, we consider the function Hilbert space H , for instance the Sobolev space $H^s(\Omega)$, where $\Omega \subset \mathbb{R}^n$ or $\Omega = \mathbb{R}^n$, see (Adams 1975). There are many frames in such a function Hilbert space, e.g., the Gabor frame, Wavelet basis, multiscale frame (Dahmen 1997) etc. For our purpose, we further assume that $H^s(\Omega)$ can be included into a continuous function space, see (Adams 1975), so that $H^s(\Omega)$ can contain the linear evaluation functional $\delta_x(\cdot)$.

All of the above function Hilbert spaces H are separable. When there exists some frame in H , then one can define an analysis operator R^* by (3.3), which is the adjoint operator of synthesis operator R for that frame, see (3.2). In this case, we can take the infinite dimensional sequence space ℓ^2 as the feature space in the terminology of support vector machines. So the analysis operator R^* can be viewed as a regularization operator P in a class of regularization networks based on the frame. Thus, by using Theorem 2, the Green's function of the frame operator Q is one choice for the kernel for the support vector machines based on the frame decomposition of function Hilbert spaces. That is, we have

Theorem 4. *Let $F = \{f_i\}$ be a frame in function Hilbert space $H = H^s(\Omega)$ and Q be the frame operator associated with the frame F . Suppose the function $G(x, y) \in H^s(\Omega \times \Omega)$ is the Green's function of the frame operator Q , i.e.,*

$$(QG(\cdot, y))(x) = \delta_y(x). \quad (4.1)$$

Then $k = G$ satisfies the self-consistency condition (2.5) with respect to the analysis operator $P = R^$.*

Proof. Let $G(x, y)$ be the Green's function of Q . By the property of evaluation functional $\delta_y(x)$, one has

$$G(x, y) = \langle G(\cdot, y), \delta_x(\cdot) \rangle.$$

By the equality $(QG(\cdot, y))(x) = \delta_y(x)$ and $Q = RR^*$, the above equation can be rewritten as

$$\begin{aligned} G(x, y) &= \langle G(\cdot, y), \delta_x(\cdot) \rangle = \langle G(\cdot, y), QG(\cdot, x) \rangle \\ &= \langle G(\cdot, y), RR^*G(\cdot, x) \rangle = \langle R^*G(\cdot, y), R^*G(\cdot, x) \rangle_{\ell^2} \end{aligned}$$

Thus the self-consistency condition (2.5) is satisfied with $k = G$ and $P = R^*$. Furthermore, G is an admissible non-negative kernel, as it can be written as a dot product in the sequence Hilbert space $\mathcal{F} = \ell^2$. Q.E.D.

The key problem is the calculation of the Green's function. We will assume that there exist linear continuous evaluation functional on H , denoted by $\delta_x(\cdot)$ see (Smola 1998) page 40. The relationship between the Green's function and the dual frame \tilde{F} of F is formalized in the following theorem.

Theorem 5. *Let $H = H^s(\Omega)$ be a function Hilbert space such that evaluation functionals $\delta_x(\cdot)$ on it are continuous and $F = \{f_i\}$ be a frame in H with the dual frame $\tilde{F} = \{\tilde{f}_i\}$, and define a function $G(x, y)$ with \tilde{F} as*

$$G(x, y) = \sum_i \tilde{f}_i(x) \cdot \tilde{f}_i(y). \quad (4.2)$$

Then $Q(x, y)$ is the Green's function as defined in Theorem 4.

Proof. If F is a frame, then we can obtain equation (4.2) by definition. Now let $G(x, y)$ be a Green's function of Q ,

$$(QG(\cdot, y))(x) = \delta_y(x).$$

By Theorem 3 the frame operator Q is boundedly invertible operator on H , then

$$\begin{aligned} G(x, y) &= (Q^{-1}\delta_y(\cdot))(x) = (Q^{-1}\sum_i \tilde{f}_i(y) \cdot f_i(\cdot))(x) \\ &= \sum_i \tilde{f}_i(y) \cdot (Q^{-1}f_i(\cdot))(x) \\ &= \sum_i \tilde{f}_i(y) \cdot \tilde{f}_i(x) \end{aligned}$$

In the last step, we use the definition of a dual frame. Thus the function defined in equation (4.2) is the Green's function of Q in the sense of (4.1). Q.E.D.

Theorem 5 provides a means to calculate the Green's function, but this computation depends on knowing the dual frame. In most of cases, we cannot obtain the dual frame for a given frame. Here we will propose an approximation algorithm for the Green's function $G(x, y)$. In order to implement the support vector machines with the kernel function $k = G$ as defined in Theorem 4, one has to determine the Green's function G in the equation (4.1). For a given dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, the solution of the support vector machine takes the form, see (2.3),

$$f(x) = \sum_{i=1}^N \alpha_i G(x_i, x) + b$$

Thus we just need to determine the Green's function G for a particular dataset, i.e., a set of one-variable functions $G(x_i, x) \in H$ satisfying

$$QG(\cdot, x_i) = \delta_{x_i}$$

This is the frame operator equation (3.7) for $f = \delta_{x_i}$ which can be easily solved by the Richardson iteration (3.8).

The computational cost for determining kernels $G(x_i, x)$ ($i = 1, 2, \dots, N$) will increase rapidly with increasing training dataset size. We should note that Theorem 1 can be used as another method for constructing a non-negative kernel function. In the following we will discuss this issue based on the frame concepts.

Now let $F = \{f_i\} \subset L^2(\Omega)$ be a Riesz stable basis (of course, a frame) and a decreasing sequence of positive numbers $\{\lambda_i\}$ such that they define a function $k(x, y)$ in the following way

$$k(x, y) = \sum_i \lambda_i f_i(x) f_i(y) \quad (4.3)$$

where the series is well defined for all of x and y such that it converges uniformly. Then the function defined in equation (4.3) is positive semi-definite, satisfying the Mercer's condition, see Theorem 1. In fact, for any $f \in L^2(\Omega)$, one has, Schölkopf (1997),

$$\int f(x) k(x, y) f(y) dx dy = \sum_i \lambda_i \int f(x) f_i(x) dx \cdot \int f(y) f_i(y) dy \geq 0. \quad (4.4)$$

Thus the function defined by equation (4.3) can serve as a kernel function which can be used in a support vector machine.

Theorem 6. *Let F be a Riesz stable basis with a dual $\tilde{F} = (\tilde{f}_i)_i$ in H and the function k be defined as (4.3). Define an operator $P : H \rightarrow \ell^2$ satisfying*

$$P : f \in H \longrightarrow \left(\frac{1}{\sqrt{\lambda_i}} \langle f, \tilde{f}_i \rangle \right)_i \in \ell^2.$$

Then P is the regularization operator with respect to the kernel function k and the self-consistency condition (2.5) can be satisfied. Thus the support vector machines defined by k and regularization network defined by P are equivalent with the feature space $\mathcal{F} \cong \ell^2$.

Proof. First k is a support vector kernel due to (4.4). Since F is a Riesz stable basis, then by Theorem 3 F and its dual frame \tilde{F} are biorthogonal. In order to prove this theorem, it is enough to check that the self-consistency condition (2.5) is satisfied with respect to the operator P .

$$\begin{aligned} \langle Pk(\cdot, y), Pk(x, \cdot) \rangle_{\ell^2} &= \left\langle \left(\frac{1}{\sqrt{\lambda_n}} \langle k(\cdot, y), \tilde{f}_n \rangle \right)_n, \left(\frac{1}{\sqrt{\lambda_n}} \langle k(x, \cdot), \tilde{f}_n \rangle \right)_n \right\rangle_{\ell^2} \\ &= \sum_n \frac{1}{\lambda_n} \langle k(\cdot, y), \tilde{f}_n \rangle \cdot \langle k(x, \cdot), \tilde{f}_n \rangle \\ &= \sum_n \frac{1}{\lambda_n} \left\langle \sum_i \lambda_i f_i(\cdot) f_i(y), \tilde{f}_n \right\rangle \cdot \left\langle \sum_i \lambda_i f_i(x) f_i(\cdot), \tilde{f}_n \right\rangle \\ &= \sum_n \frac{1}{\lambda_n} \cdot \lambda_n f_n(y) \cdot \lambda_n f_n(x) = k(x, y) \end{aligned}$$

That is the self-consistency condition.

Q.E.D.

Using kernel k we can define a subspace H_1 of H whose elements take the form

$$f(x) = \sum_i c_i f_i$$

such that $\|f\|_{H_1} = \sum_i \frac{c_i^2}{\lambda_i} < \infty$. It is easy to check that H_1 is a Reproducing Kernel Hilbert Space (RKHS) with reproducing kernel given by k in which the dot product is defined as:

$$\langle \sum_i c_i f_i, \sum_i d_i f_i \rangle_{H_1} \equiv \sum_i \frac{c_i d_i}{\lambda_i}$$

In fact we have

$$\langle f(x), k(x, y) \rangle_{H_1} = \sum_k \frac{c_k \lambda_k f_k(y)}{\lambda_k} = \sum_i c_i f_i(y) = f(y).$$

Also see (Wahba 1999; Schölkopf 1997; Girosi 1998). Hence we can see that the sequence $\{\lambda_i\}$ can be used to constrain the subspace H_1 , or equivalently, the smoothness of function f .

5 Numerical Example

The dual frame plays a main role in the argument of this paper. Many frames, such as Gabor's, biorthogonal B-spline wavelet and RBF frames (Chui 1992; Dahmen 1997; Blatter 1998), can be used to construct new Green's function in this framework. The Green's function defined by equation (4.2) is an infinite series of functions in a frame. In practice, this sum of infinite terms should be truncated into a sum of finite terms. Denote the corresponding sum by

$$G_\Lambda(x, y) = \sum_{j \in \Lambda} \tilde{f}_j(x) \tilde{f}_j(y)$$

where Λ is a index subset of J and denote $J = |\Lambda|$ the number of elements in Λ .

In this paper, the spline prewavelet proposed by Chui and Wang (Chui and Wang 1991) is chosen to be the generator for Green's kernel function. Let $N_m(x)$ be the B-spline function of the m th order supported on $[0, m]$. Then the m th order pre-wavelet corresponding to $N_m(x)$ is defined as follows:

$$\psi_m(x) = \frac{1}{2^{m-1}} \sum_{j=0}^{2m-2} N_{2m}(j+1) N_{2m}^{(m)}(2x-j)$$

where $N_{2m}^{(m)}(x)$ is the m th derivative of $2m$ th order B-spline $N_{2m}(x)$, or equivalently,

$$\psi_m(x) = \frac{1}{2^{m-1}} \sum_{n=0}^{3m-2} (-1)^n \left[\sum_{j=0}^m \binom{m}{j} N_{2m}(n-j+1) \right] N_m(2x-n). \quad (5.1)$$

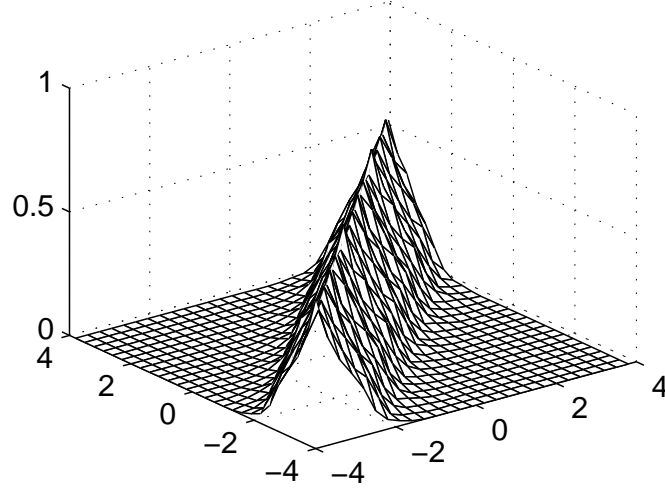


Figure 1: Pre-wavelet kernel function of order (4,3)

Define

$$\phi_{m;0,i} = N_m(x - i)$$

and

$$\psi_{m;j,i} = 2^{j/2}\psi_m(2^j x - i)$$

The union of $\phi_{m;0,k}$ and $\psi_{m;j,k}$ constitutes a frame for the function space $L^2(\mathbb{R})$, and a corresponding kernel can be constructed, see (4.3),

$$G_{m;J}(x, y) = \sum_i \phi_{m;0,i}(x)\phi_{m;0,i}(y) + \sum_{0 \leq j \leq J-1} \sum_i 2^{-2j}\psi_{m;j,i}(x)\psi_{m;j,i}(y). \quad (5.2)$$

where $G_{m;J}(x, y)$ is referred to as the pre-wavelet kernel of order m up to level J , that is, there are J different scales in the kernel function. Although the summation corresponding to index i in (5.2) is taken from $-\infty$ to $+\infty$, only finite terms are summed when both x and y are given because all of ϕ and ψ are compactly supported. The kernel function $G_{4,3}(x, y)$ is plotted in Figure 1. Multidimensional kernels can be formed in the usual way by forming a tensor product of univariate kernels or they can be constructed from certain multivariate frames (Dahmen 1997).

Although the kernel function defined by (5.2) provides J different scale/frequency components, the coarsest scale/lowest frequency is determined by the basic wavelet function (5.1). In order to enable the range of scales to match the data an extra scale factor can be introduced which is similar to the width parameter σ in the RBF kernel. In fact, the index $J = 0$ in the kernel definition corresponds to the coarsest scale/frequency, thus we can define the so-called scaled pre-wavelet kernel as follows:

$$G_{m;J,\sigma}(x, y) = \sum_i \phi_{m;0,i}\left(\frac{x}{\sigma}\right)\phi_{m;0,i}\left(\frac{y}{\sigma}\right) + \sum_{0 \leq j \leq J-1} \sum_i 2^{-2j}\psi_{m;j,i}\left(\frac{x}{\sigma}\right)\psi_{m;j,i}\left(\frac{y}{\sigma}\right). \quad (5.3)$$

where σ is called scale factor. This scale factor adjusts the starting range of the scales, but does not adjust the width of the scale range, which is controlled by J ; making σ smaller will produce a function scale with higher frequencies.

In this section an SVM will be implemented with the Vapnik's ϵ -insensitive loss function $c(f(x), y) = L_\epsilon(f(x) - y)$ which is defined as

$$L_\epsilon(u) = \begin{cases} 0 & |u| < \epsilon \\ |u| - \epsilon & \text{otherwise} \end{cases}$$

In this regression scenario an ϵ -insensitive region, in place of the classification margin, is introduced. The “tube” of $\pm\epsilon$ around the regression function within which errors are not penalised (Vapnik 1998) enables sparsity to be obtained within the support vectors. The support vectors lie on the edge, or outside, of this region. For simplicity, the regularisation problem (2.4) is re-written in the equivalent SVM formulation as

$$\min R_{\text{reg}}[f] = C \sum_{i=1}^N L_\epsilon(f(x_i) - y_i) + \frac{1}{2} \|Pf\|^2 \quad (5.4)$$

where C is the capacity control parameter. A larger value of C will increase penalisation of errors on the training data but will increase the potential for overfitting. The value of C should be related to the noise level in the targets. The determination of C with respect to a real problem is a difficult task. One approach is to use model selection criteria such as VC-theory (Vapnik 1995), Bayesian methods (MacKay 1991), AIC (Akaike 1974), NIC (Murata et al. 1994) and cross validation etc. In our experiments the data is generated from a known function and the capacity control parameter C was optimized by measuring the true generalisation error for a range of finely sampled values of C .

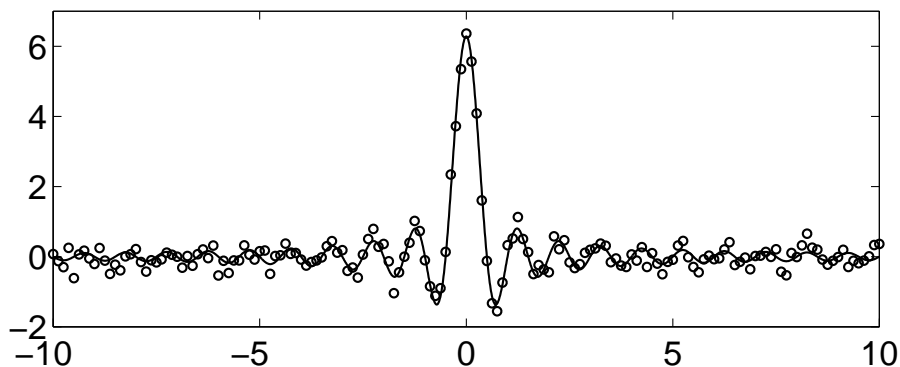


Figure 2: The true function f and the uniformly-spaced samples with a noise variance of $\sigma = 0.2$

Example 1: The function $f(x) = x^{-1} \sin(2\pi x)$ defined on $[-10, 10]$ is used to illustrate support vector regression with a pre-wavelet kernel (5.2). The approximation of f is computed by a support vector machine method using 160 uniformly-spaced samples.

The target values of the samples were corrupted by Gaussian noise with a standard noise deviation of 0.2. The learning data and the true function are shown in Figure 2. For comparison, we approximate the same function by the SVM method with $\epsilon = 0.1$ utilizing three different types of kernel functions:

1. Vapnik’s first order infinite spline kernel function (Vapnik 1998), given by

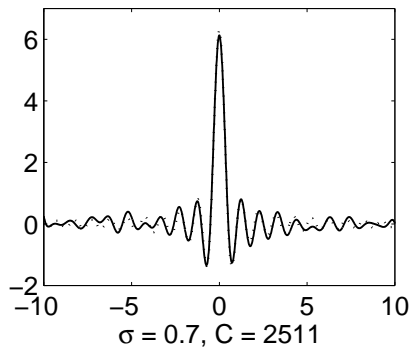
$$K(x, y) = 1 + xy + \frac{1}{2}xy \min(x, y) - \frac{1}{6}(\min(x, y))^3$$

2. RBF kernel function with width σ , see (Smola 1998);
3. The pre-wavelet kernel function (5.2) of order (m, J, σ) .

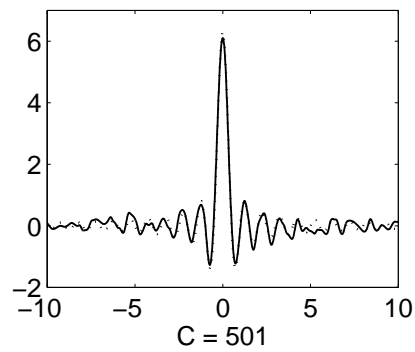
Figure 3 shows the SVM results for the optimal parameters. In plots (a), (b) and (c), the solid line is the curve of approximation function and the dotted line is the true function. From these plots, it can be seen that SVM regression with these three kernels can give good generalisation results. The optimal parameters were $\sigma = 0.7$, $C = 2511$ for the RBF kernel, $C = 501$ for the spline kernel and $m = 5$, $J = 1$, $\sigma = 0.7$, $C = 251188$ for the scaled pre-wavelet kernel. The mean square generalisation error is 0.016 for the RBF kernel (67% support vectors), 0.020 for the spline kernel (66% support vectors) and 0.015 for pre-wavelet kernel (67% support vectors). Plot (d) shows the mean square generalisation error (natural log) curves of the model versus the control factor C (\log_{10}). It is evident that the performance of the RBF kernel is superior to that of the spline kernel, and that the scaled pre-wavelet kernel is superior to both of the spline and RBF kernels, over a large range of C values. The best performance is given by the pre-wavelet kernel. The use of an ϵ insensitive loss function means that typically the generalisation error curve will become constant above a certain value of C ; for the spline kernel this occurs at $C \approx 10^4$.

The performance of the RBF kernel strongly depends on the values of the kernel width parameter σ and control factor C , resulting in poor generalisation when C is small and when σ is unsuitable. SVMs were also constructed for the pre-wavelet kernel (5.2) without a scale factor (i.e. $\sigma = 1$). In these cases ($\sigma = 1$), the kernel with best performance was the pre-wavelet kernel of order $(6, 2)$. The pre-wavelet of order $(6, 2)$ has two scale components and higher smoothness. The removal of the scale parameter fixes the coarsest scale/frequency of the function space and hence a larger number of scales (J) will typically be required to include the desired scale(s) in the function space. Since, better generalisation will be obtained if the function space is as ‘tight’ as possible, the scaled pre-wavelet kernel (5.3) should be used in preference to the pre-wavelet kernel (5.2), at the expense of an additional parameter to be determined. Example 1 demonstrates that the pre-wavelet kernel is competitive with commonly employed kernel functions.

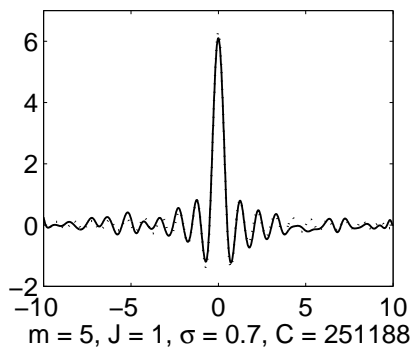
To illustrate how the parameters in the scaled pre-wavelet kernel control the function space, the SVM algorithm was implemented for different orders ($m = 3, 4, 5, 6$) and different bandwidths ($J = 1, 2, 3$). The results are shown in Figure 4 for several selected examples. From left to right, the plots correspond to increasing bandwidth and from top



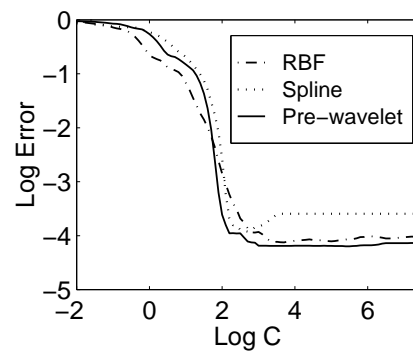
(a) Optimal result for the RBF kernel



(b) Optimal result for the spline kernel



(c) Optimal result for the scaled pre-wavelet kernel



(d) The generalisation error curves of the three kernels used in (a), (b) and (c) vs. control factor C .

Figure 3: The SVM approximation results for Example 1

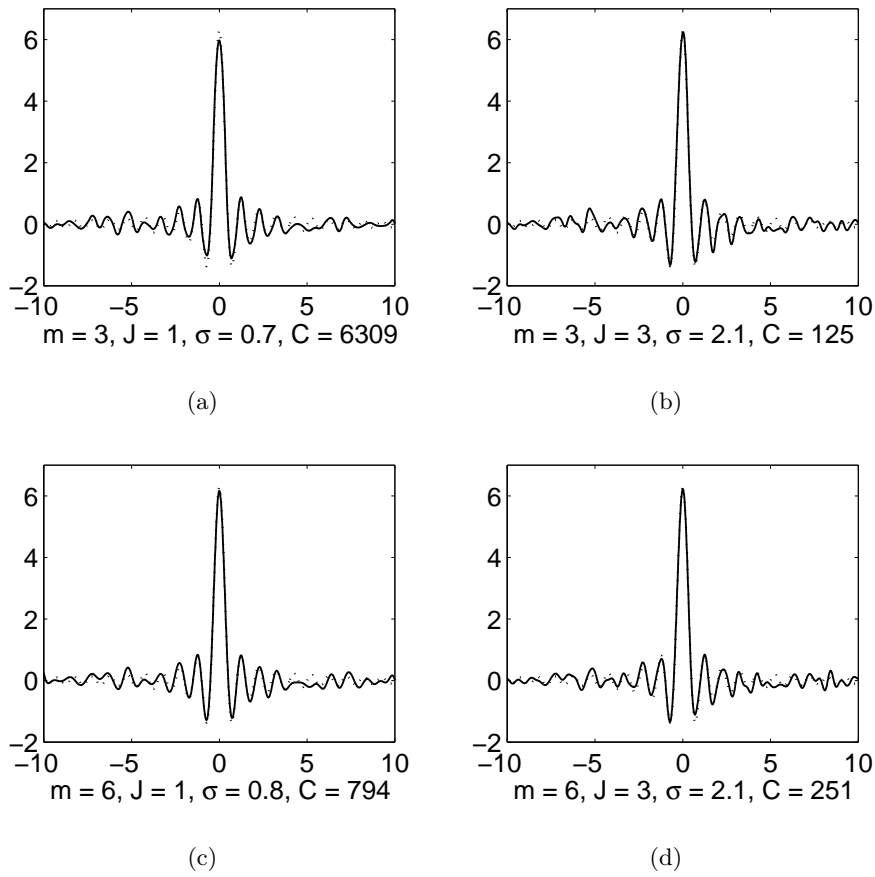


Figure 4: The SVM approximation with pre-wavelet kernels at the different orders and the optimal scaled factors

to bottom the plots correspond to increasing order. In the experiments the optimal scale factor was independent of the order (m), but dependent upon the bandwidth (J). It was found that the optimal value of σ was the one that placed the upper end of the wavelet bandwidth just above the sinc ‘frequency’, which is favourable from a generalisation perspective.

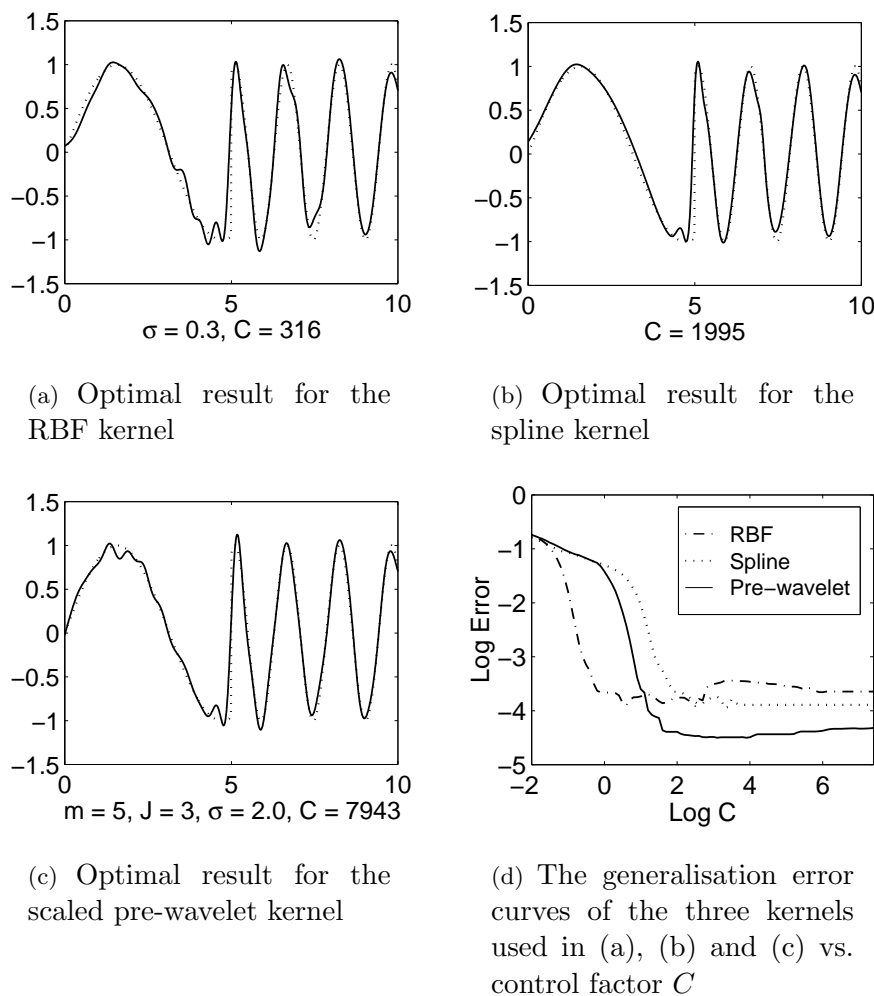


Figure 5: The SVM approximation results for Example 2.

Example 2: To illustrate a more realistic situation where the scaled pre-wavelet kernel may be used, a second example is now considered for a function containing multiple scales. As the wavelet function can be used to represent any multiscale components contained in the function, an SVM with a pre-wavelet kernel should easily capture the different scales in the data. In order to demonstrate this observation, consider the

following function:

$$f(x) = \begin{cases} \sin(x) & 0 \leq x < 5 \\ \sin(4x) & 5 \leq x \leq 10 \end{cases}$$

A dataset of 80 uniformly-spaced samples were generated in which Gaussian noise of standard deviation 0.1 was added to the targets. The SVM method with $\epsilon = 0.1$ was implemented for the three kernels utilised in Example 1. Figure 5 (a), (b) and (c) show the curves of the true function (dashed line) and the approximation function (solid line) by the SVM algorithm for these three kernels with their optimal parameters. Plot (d) shows the generalisation error curves (natural log) of the model versus the control factor C (\log_{10}). In this example, the performance superiority of the scaled pre-wavelet kernel to the other two kernels is more pronounced. The test mean square errors are 0.020 for the RBF kernel (51% support vectors), 0.019 for the spline kernel (45% support vectors) and 0.011 for scaled pre-wavelet kernel (45% support vectors).

This example highlights the poorer performance of the RBF kernel when multiple scales exist. The optimal values of the bandwidth for the scaled pre-wavelet kernel was $J = 3$, since the original data contains two different scales, one of which is 4 times the other and the pre-wavelet kernel scales are spaced at 2^j .

6 Conclusions

In this paper, we have developed kernel functions from frames in a function Hilbert space and shown their application within a SVM. The relationship between the analysis operator of a frame and the Green's function of a frame operator has been introduced. The dual frame plays the main role in the argument of this paper. Many frames, such as Gabor's, biorthogonal B-spline wavelet and RBF frames, can be used to construct new Green functions in this framework. Examples in this paper have demonstrated that the newly proposed kernels are competitive with the well-established kernel functions. The choice of an appropriate kernel within a kernel-based learning method is critical in obtaining good performance. When little prior knowledge exists, two approaches can be adopted: (1) try many different kernel functions, (2) try one kernel function with a flexible parameterisation. The wavelet kernels fall into the second category. This approach is more attractive when a Bayesian method is employed since the kernel parameters can then be treated as hyperparameters and re-estimated appropriately. Until now the kernel of choice for a dataset containing many scales was to employ a spline kernel (or similar) with no associated scale. In this paper we have shown that better results can be obtained when a 'tighter' function space is used by using a bandlimited kernel. In the limit as the bandwidth becomes small the kernel will induce a 'tight' function space (c.f. RBF) and in the limit as the the bandwidth becomes large the kernel will induce a 'loose' function space (c.f. spline). As such we provide the kernel-based modeller with a new tool for their armoury. Extensions to this work could consider the construction of other kernel functions based on the multilevel/multiscale Riesz basis and the development of effective algorithms for general support vector machines based on these multilevel/multiscale kernel functions.

Acknowledgements

The authors gratefully acknowledge the financial support of EPSRC and the reviewers for their constructive comments in improving this paper.

References

- Adams, R. (1975). *Sobolev Spaces*. New York: Academic Press.
- Aizerman, M., E. Braverman, and L. Rozonoér (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control* 25, 821–837.
- Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19(6), 716–723.
- Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions on the American Mathematical Society* 68, 337–404.
- Benedetto, J. and M. Frazier (Eds.) (1994). *Wavelets: Mathematics and Applications*. Boca Raton: CRC Press.
- Bennett, R. (1999). Combining support vector and mathematical programming methods for induction. In B. Schölkopf, C. Burges, and A. Smola (Eds.), *Advances in Kernel Methods — Support Vector Learning*, pp. 307–326. Cambridge, MA: MIT Press.
- Blatter, C. (1998). *Wavelets: A Primer*. Natick, Massachusetts: A.K. Peters, Ltd.
- Boser, B., I. Guyon, and V. Vapnik (1992). A training algorithm for optimal margin classifiers. In H. Haussler (Ed.), *5th Annual ACM Workshop on COLT*, Pittsburgh, PA, pp. 144–152. ACM Press.
- Burges, C. and B. Schölkopf (1997). Improving the accuracy and speed of support vector learning machines. In *Advances in Neural Information Processing Systems*, Volume 9, pp. 375–381. MA: MIT Press.
- Chui, C. (1992). *An Introduction to Wavelets*. Boston: Academic Press.
- Chui, C. and J. Wang (1991). A general framework for compactly supported splines and wavelets. *Proceedings of American Mathematical Society* 113, 785–793.
- Dahmen, W. (1997). Wavelet and multiscale methods for operator equations. *Acta Numerica* 6, 55–228.
- Daubechies, I. (1992). *Ten Lectures on Wavelets*, Volume 61 of *CBMS-NSF Reg. Conf. Ser. Appl. Math.* Philadelphia: SIAM Press.
- Duffin, R. and A. Schaeffer (1952). A class of nonharmonic fourier series. *Transactions of American Mathematical Society* 72, 341–366.
- Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Computation* 10(6), 1455–1480.

- Gunn, S., M. Brown, and K. Bossley (1997). Network performance assessment for neurofuzzy data modelling. In *Lecture Notes in Computer Science*, Volume 1280, pp. 313–323. Boston: Academic Press.
- Heil, C. and D. Walnut (1989). Continuous and discrete wavelet transform. *SIAM Review* 31, 628–666.
- Lorentz, R. and P. Oswald (2000). Criteria for hierarchical bases for Sobolev spaces. *Appl. Comput. Harm. Anal.* 8, 32–85.
- MacKay, D. (1991). *Bayesian Modelling and Neural Networks*. Ph. D. thesis, California Institute of Technology, Pasadena, CA.
- Murata, N., S. Yoshizawa, and S. Amari (1994). Network information criterion—determining the number of hidden units for artificial neural network models. *IEEE Transactions on Neural Networks* 5, 865–872.
- Poggio, T., V. Torre, and C. Koch (1985). Computational vision and regularization theory. *Nature* 317(26), 314–319.
- Schölkopf, B. (1997). *Support Vector Learning*. Ph. D. thesis, Technische Universität Berlin, Oldenbourg Verlag, Munich.
- Schölkopf, B., P. Bartlett, A. Smola, and R. Williamson (1998). Support vector regression with automatic accuracy control. In L. Niklasson, M. Bodeén, and T. Ziemke (Eds.), *Perspectives in Neural Computing – Proceedings of ICANN’98*, Berlin, pp. 111–116. Springer Verlag.
- Schölkopf, B., A. Smola, R. Williamson, and P. Bartlett (2000). New support vector algorithms. *Neural Computation* 12(5), 1207–1245.
- Smola, A. (1998). *Learning with Kernels*. Ph. D. thesis, Technischen Universität Berlin, Berlin, Germany.
- Smola, A. and B. Schölkopf (1998). On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica* 22, 211–231.
- Smola, A., B. Schölkopf, and K. Müller (1998). The connection between regularization operators and support vector kernels. *Neural Networks* 11, 637–649.
- Tikhonov, A. and V. Arsenin (1977). *Solution of Ill-posed Problems*. Washington, D.C.: W.H. Winston.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer.
- Vapnik, V. (1998). *Statistical Learning Theory*. New York: Wiley.
- Wahba, G. (1990). *Splines Models for Observational Data*, Volume 59 of *Series in Applied Mathematics*. Philadelphia: SIAM Press.
- Wahba, G. (1999). Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Schölkopf, C. Burges, and A. Smola (Eds.), *Advances in Kernel Methods — Support Vector Learning*, pp. 68–88. Cambridge, MA: MIT Press.