

Research Article

Knowledge-Based Green's Kernel for Support Vector Regression

Tahir Farooq,¹ Aziz Guergachi,² and Sridhar Krishnan¹

¹ Department of Electrical and Computer Engineering, Ryerson University, 350 Victoria Street, Toronto, ON, Canada M5B 2K3

² School of Information Technology Management, Ryerson University, Toronto, ON, Canada M5B 2K3

Correspondence should be addressed to Tahir Farooq, tahir510@gmail.com

Received 19 January 2010; Accepted 19 May 2010

Academic Editor: Cristian Toma

Copyright © 2010 Tahir Farooq et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a novel prior knowledge-based Green's kernel for support vector regression (SVR). After reviewing the correspondence between support vector kernels used in support vector machines (SVMs) and regularization operators used in regularization networks and the use of Green's function of their corresponding regularization operators to construct support vector kernels, a mathematical framework is presented to obtain the domain knowledge about magnitude of the Fourier transform of the function to be predicted and design a prior knowledge-based Green's kernel that exhibits optimal regularization properties by using the concept of matched filters. The matched filter behavior of the proposed kernel function makes it suitable for signals corrupted with noise that includes many real world systems. We conduct several experiments mostly using benchmark datasets to compare the performance of our proposed technique with the results already published in literature for other existing support vector kernel over a variety of settings including different noise levels, noise models, loss functions, and SVM variations. Experimental results indicate that knowledge-based Green's kernel could be seen as a good choice among the other candidate kernel functions.

1. Introduction

Over the last decade support vector machines (SVMs) have been reported by several studies [1–4] to perform equal or better than other learning machines such as neural networks for the problem of learning from finite dataset and approximating a given function from sparse data. Vapnik [1, 2, 5, 6] has laid down the theoretical foundations of the structural risk minimization (SRM) principle to comprehend the problem of learning from a finite set of data in the context of regularization theory given by Tikhonov [7, 8]. SRM principle provides a connection between capacity of the hypothesis space that contains the learning

models to approximate the given function and the size of the training set. Generally, the smaller the size of the training set is, the lower the capacity of the hypothesis space should be to avoid overfitting [1, 2, 6, 9]. This motivates one to understand SVM in the context of regularization theory and find a linear solution in the kernel space that minimizes a certain loss function while keeping capacity of the hypothesis space as small as possible. The kernel function in SVM provides a nonlinear mapping from input space to a higher-dimensional feature space. Research studies [10, 11] have affirmed that SVM's regularization properties are associated with the choice of kernel function used for mapping. In the literature, Babu et al. [12] proposed local kernel based color modeling for visual tracking. Maclin et al. [13] presented a method for incorporating and refining domain knowledge for support vector machines via successive linear programming. Yen et al. [14] used kernel-based clustering methods for detecting clusters in weighted, undirected graphs. Toma [15] proposed nonlinear differential equations capable of generating continuous functions similar to pulse sequence for modeling time series. M. Li and J.-Y. Li [16] introduced a generalized mean-square error (MSE) to address the predictability of long-range dependent (LRD) series. Bakhoum and Toma [17] presented an extension to the Fourier/Laplace transform for the analysis of signals that are represented by traveling wave equations and offered a mathematical technique for the simulation of the behavior of large systems of optical oscillators. Liu [18] gave analysis of chaotic, dynamic time series events. Poggio and Girosi [19, 20] described a general learning approach using regularization theory. Girosi et al. [21, 22] have provided a unified framework for regularization networks and learning machines. Evgeniou et al. [9], Smola and Schölkopf [23], and Williamson et al. [24] demonstrated a correspondence between regularization networks (RNs) and support vector machines (SVMs). Smola et al. [11] and Schölkopf and Smola [10] have shown a connection between regularization operators used in regularization networks and support vector kernels and presented a method of using Green's functions of their corresponding regularization operators to construct support vector kernels with equivalent regularization properties. However, the problem of choosing the optimal regularization operator to construct the corresponding SV kernel for a given training set still remains unanswered. The work presented herein is focused on using prior knowledge about the magnitude spectrum of the function to be predicted to design the support vector kernels from Green's functions having suitable regularization properties by utilizing the concept of matched filters, an idea inspired by Schölkopf and Smola [10]. The intuition of matching Green's kernel comes from the fact that most real world systems are inevitably contaminated with noise in addition to their intrinsic dynamics [25, 26] and matched filters are known to be the optimal choice to recover signals in the presence of additive white noise [27, 28]. However, no mathematical justification is given in the literature for the use of matched filter theorem to obtain the matching Green's kernel. No experimental results are so far available in the literature to compare the performance of knowledge-based Green's kernel with existing support vector kernels. In this paper, we provide a mathematical framework for utilizing the matched filter theorem to design knowledge-based Green's kernel and conduct experiments on different datasets (mostly benchmarks) with different levels and models (Gaussian and Uniform) of additive white noise to evaluate the performance of our proposed kernel function. Although the assumption of additive white noise will not exactly hold in all real world cases, we keep up with the time-honored tradition [2–4, 25, 26, 29, 30] of using benchmark datasets with additive white noise assumption to evaluate the performance of our proposed method. The focus is on support vector regression (SVR). The rest of the paper is organized as follows. Section 2 reviews the theory of support vector regression, SV kernels, regularization networks, and the connection between support vector method and the theory

of regularization networks. Section 3 provides the theory of Green's functions and how they can be used to construct SV kernels. Section 4 describes the theory of matched filters and lays down the mathematical foundation for building knowledge-based Green's kernel. Section 5 presents the experimental results and Section 6 concludes the paper.

2. Support Vector Machines and Regularization Networks

Support vector machines introduced by Vapnik and coworkers for pattern recognition and regression estimation tasks have been reported to be an effective method during the last decade [31–33]. Initially developed for classification problems, a generalization of support vector (SV) algorithm known as ϵ -insensitive SV regression [1, 33] was derived to solve the problems where the function to be estimated belongs to the set of real numbers.

2.1. ϵ -Insensitive Support Vector Regression

Suppose that we have $\{(x_1, y_1), \dots, (x_N, y_N)\}$ as the training set with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, where y_i are the training targets. The problem of calculating an estimate $f(x_i)$ of y_i for training data $\{x_i, y_i\}_{i=1, \dots, N}$ can be formulated as

$$f(x) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b. \quad (2.1)$$

The goal of ϵ -insensitive SV algorithm is to calculate an estimate $f(x_i)$ of y_i by selecting the optimal hyperplane \mathbf{w} and bias b such that $f(x_i)$ is at the most ϵ distance from y_i while keeping the norm $\|\mathbf{w}\|^2$ of the hyperplane minimum. The corresponding quadratic optimization problem can be written in terms of regularized risk functional as described by [10, 11], that is, to minimize

$$\mathcal{R}[f] = \frac{\gamma}{2} \|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^N |y_i - f(x_i)|_{\epsilon}, \quad (2.2)$$

where \mathcal{R} is the regularized risk functional, γ is the regularization constant such that $\gamma \geq 0$, and the second term on the right-hand side of (2.2) is the empirical risk functional with Vapnik's ϵ -insensitive loss function [1, 2, 34]. By introducing the slack variables, in the sense of [1, 2, 34] and rewriting the problem in (2.2), we get, that is, to minimize

$$\mathcal{R}[f] = \frac{\gamma}{2} \|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^N (\zeta_i + \zeta_i^*) \quad (2.3)$$

subject to

$$\begin{aligned} y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b &\leq \epsilon + \zeta_i, \\ \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b - y_i &\leq \epsilon + \zeta_i^*, \\ \zeta_i, \zeta_i^* &\geq 0. \end{aligned} \quad (2.4)$$

In order to obtain the SV expansion of the function $f(\mathbf{x})$, we use the standard [1, 2, 34, 35] Lagrangian technique to form the objective function and the corresponding constraints. The well known formulation of the quadratic optimization problem can be reached by taking the partial derivatives of the objective function, putting them equal to zero for optimal solution and substituting the values obtained into the objective function. We follow the lines of [11, 23] and write the quadratic optimization problem as

$$\text{minimize} \begin{cases} \frac{1}{2} \sum_{i,j=1}^N (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle, \\ +\epsilon \sum_{i=1}^N (\alpha_i - \alpha_i^*) - \sum_{i=1}^N (\alpha_i^* - \alpha_i) y_i \end{cases} \quad (2.5)$$

subject to

$$\begin{aligned} \sum_{i=1}^N (\alpha_i - \alpha_i^*) &= 0, \\ 0 \leq \alpha_i, \quad \alpha_i^* &\leq \frac{1}{\gamma N}. \end{aligned} \quad (2.6)$$

This leads to the well-known formulation of SV regression, that is,

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i^* - \alpha_i) \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b. \quad (2.7)$$

Comparing (2.7) and (2.1), it shows that the training examples that lie inside the ϵ -tube contribute to a sparse expansion of \mathbf{w} because the corresponding Lagrange multipliers α_i, α_i^* are zero [1, 2, 10, 34, 35].

The expression given by (2.7) corresponds to linear SV regression. In order to obtain nonlinearity, SV algorithms can be quipped with nonlinear operators $\varphi(\cdot)$ mapping from input space into a high-dimensional feature space, $\varphi : \mathcal{X} \rightarrow \mathcal{F}$ as described in [36, 37]. The kernel function is defined as

$$k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}) \cdot \varphi(\mathbf{x}') \rangle, \quad (2.8)$$

$$\int_{\mathcal{X}^2} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}') dx dx' \geq \quad \forall f \in L_2(\mathcal{X}). \quad (2.9)$$

According to Mercer theorem, the kernel is any continuous and symmetric function that satisfies the positivity condition given by (2.9). Such a function $k(\mathbf{x}, \mathbf{x}')$ defines a dot product in the feature space given by (2.8) [36].

Hence, by making use of (2.8), (2.7) can be written as

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i^* - \alpha_i) \langle \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}) \rangle + b = \sum_{i=1}^N (\alpha_i^* - \alpha_i) k(\mathbf{x}_i, \mathbf{x}) + b. \quad (2.10)$$

2.2. Regularization Networks

The idea of regularization method was first given by Tikhonov [8] and Tikhonov and Arsenin [7] for the solution of ill-posed problems. Assume that we have a finite dataset $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, independently and identically drawn from a probability distribution $p(\mathbf{x}, y)$ in the presence of noise. Assume that the probability distribution $p(\mathbf{x}, y)$ is unknown. One way of approaching the problem is to estimate the function f by minimizing a certain empirical risk functional:

$$\mathcal{R}_{\text{emp}}[f] = \frac{1}{N} \sum_{i=1}^N |y_i - f(\mathbf{x}_i)|_{\epsilon}. \quad (2.11)$$

The problem of approaching the solution through minimizing (2.11) is ill-posed because the solution is unstable [2]. Hence, the solution is to utilize the idea proposed by [7, 8] and add a capacity control or stabilizer [22] term to (2.11) and minimize a regularized risk functional:

$$\mathcal{R}_{\text{RN}}[f] = \frac{1}{N} \sum_{i=1}^N |y_i - f(\mathbf{x}_i)|_{\epsilon} + \frac{\gamma}{2} \|\lambda f\|^2, \quad (2.12)$$

where λ is a linear, positive semidefinite regularization operator. The first term in (2.12) corresponds to finding a function that is as close to the data examples as possible in terms of Vapnik's ϵ -insensitive loss function whereas the second term is the smoothness functional and its purpose is to restrict the size of the functional space and to reduce the complexity of the solution [23]. The filter properties of λ are given by $\lambda^* \lambda$, where $*$ represents the complex conjugate. Following the lines of [11, 23], the problem of minimizing the regularized risk functional given by (2.12) can be transformed into constrained optimization problem by utilizing the standard Lagrange multipliers technique and a formulation similar to (2.5) can be obtained where a direct relationship between SV technique and the RN method can be observed. In other words, training an SVM with a kernel function obtained from the regularization operator λ is equivalent to implementing RN to minimize the regularized risk functional given by (2.12) with λ as the regularization operator. We refer the reader to [2, 11, 23] for the detailed discussion on the relationship between the two methods, that is, SV and RN.

3. Green's Functions and Support Vector Kernels

The idea of Green's functions was introduced in the context of solving inhomogeneous differential equations with boundary conditions. However, Green's functions of their corresponding regularization operators can be used to design kernel functions that exhibit the regularization properties given by their corresponding regularization operators, satisfy the Mercer condition, and qualify to be SV kernels [11, 23, 35]. Green's kernel of a discrete regularization operator $\lambda[n]$ can be written as [38]

$$G(x, x') = \sum_{i=1}^N \lambda[n] \phi_n(x) \phi_n(x'), \quad (3.1)$$

where $\phi_n \{n = 1, \dots, N\}$ are the basis of the orthonormal eigenvectors of G corresponding to nonzero eigenvalues $\lambda[n]$ such that $\lambda[n]$ confers the spectrum of G . The expression given by (3.1) assumes one-dimensional case. A generalization to multidimension is straight forward and will be discussed later. From [38], it can be easily shown that G satisfies the condition of positive definiteness and the series converges absolutely uniformly since all the eigenvalues of G are positive. As $G(x_i, x_j) = G(x_j, x_i)$, it also satisfies the symmetry property and Mercer theorem can be applied to prove that G is an admissible support vector kernel [39] and it can be written as a dot product in feature space, that is,

$$G(x_i, x_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle. \quad (3.2)$$

At this point, we refer the reader to literature [10, 11, 23, 35] for useful discussion on regularization properties of commonly used SV kernels. We can also utilize (3.1) to obtain periodic kernels for given regularization operators. For example [10], by taking $\lambda[n]$ as eigenvalues of the given discrete regularization operator and Fourier basis $\{1/2\pi, \sin(nx), \cos(nx), n \in N\}$ as corresponding eigenvectors, we get Green's kernel:

$$k(x, x') = \sum_{n=1}^M \lambda[n] (\sin(nx) \sin(nx') + \cos(nx) \cos(nx')), \quad (3.3)$$

$$k(x, x') = \sum_{n=1}^M \lambda[n] \cos(n(x - x')).$$

Capacity control can be achieved by restricting the summation to different eigensubspaces with different values of M . Excluding the eigenfunctions that correspond to high frequencies would result in increased smoothness thereby decreasing the system capacity and vice versa. A general lowpass smoothing functional is a good choice if there is no prior information available about the frequency distribution of the signal to be predicted. However, (3.1) can be seen as a reasonable choice for building kernels if there is some prior information available about the magnitude spectrum of the signal that we would like to approximate by utilizing the concept of matched filters [10].

4. Matched Filter and Knowledge-Based Matched Green's Kernel

Matched filter [40, 41] is the optimum time invariant filter among all linear or nonlinear filters to recover a known signal from additive white noise [42]. Assume the input signal $f(x)$ in the presence of additive white noise $n(x)$ passing through the matched filter with impulse response $h(x)$. The output of the filter is given by

$$y(x) = h(x) \otimes (f(x) + n(x)), \quad (4.1)$$

where \otimes denotes the convolution operation. Our aim is to obtain the conditions for which signal-to-noise ratio (SNR) at the filter output takes its maximum value since it is

understandable that the probability of recovering a signal from noise is high when SNR is maximum [27]. From [27, 42, 43] the impulse response of the matched filter is given by

$$h(x) = Af(x_1 - x), \quad (4.2)$$

where A is the filter gain constant and x_1 is the point at which the output power of the filter takes its maximum value. The maximum SNR is given by

$$(\text{SNR})_{\max} = \frac{2}{N_0} \int_{-\infty}^{\infty} f^2(x_1 - \alpha) d\alpha, \quad (4.3)$$

where N_0 is the noise power density. We refer the reader to original literature [27, 42, 43] for details and the proof of matched filter theorem. For simplicity we will assume unity gain, that is, $A = 1$. It is noteworthy in (4.2) that the filter impulse response is independent of noise power density N_0 with the prior assumption of white noise. Secondly, maximum SNR (4.3) is a function of signal energy and is independent of signal shape [42].

In order to design a matching kernel based on prior knowledge, it is sufficient to have an estimate of the magnitude spectrum of the signal to be predicted as prior knowledge about the signal as opposed to the theory of matched filters where complete knowledge of the signal is required to recover the signal from noise. From (4.2) it can be seen that the impulse response of the optimum filter is time reversed signal $f(x)$ with x_1 delay. Nevertheless, in order to obtain matching kernel we are only interested in magnitude spectrum of matched filter which can be obtained by taking the Fourier transform of $h(x)$ in (4.2) and multiplying it with its complex conjugate:

$$\begin{aligned} H(e^{j\omega}) &= \int_{-\infty}^{\infty} h(x)e^{-j\omega x} dx = \int_{-\infty}^{\infty} f(x_1 - x)e^{-j\omega x} dx \\ &= e^{-j\omega x_1} \int_{-\infty}^{\infty} f(x_1 - x)e^{j\omega(x_1 - x)} dx = F^*(e^{j\omega})e^{-j\omega x_1}, \end{aligned} \quad (4.4)$$

$$|H(\omega)|^2 = H(e^{j\omega})H^*(e^{j\omega}) = |F(\omega)|^2, \quad (4.5)$$

$$|H(\omega)| = |F(\omega)|, \quad (4.6)$$

where $H(e^{j\omega})$ is the frequency response of matched filter, $|H(\omega)|$ is the magnitude response and $|F(\omega)|$ is the magnitude spectrum of the matched filter and the signal $f(x)$, respectively. An important note at this point is that (4.6) does not depend on delay x_1 whereas in the case of matched filters it is necessary to have a delay to make the impulse response realizable. Hence the matching kernel can be obtained by simply calculating the magnitude spectrum of $f(x)$ and utilizing (3.1). As $f(x)$ is the signal to be predicted, we assume that its magnitude spectrum does not significantly change from the training targets $y(x)$ in (2.2) and this is the prior knowledge that we acquire from $y(x)$ about $f(x)$ to obtain Green's kernel. This is a weak condition since many signals with completely different characterization in time domain share the similar magnitude spectrum.

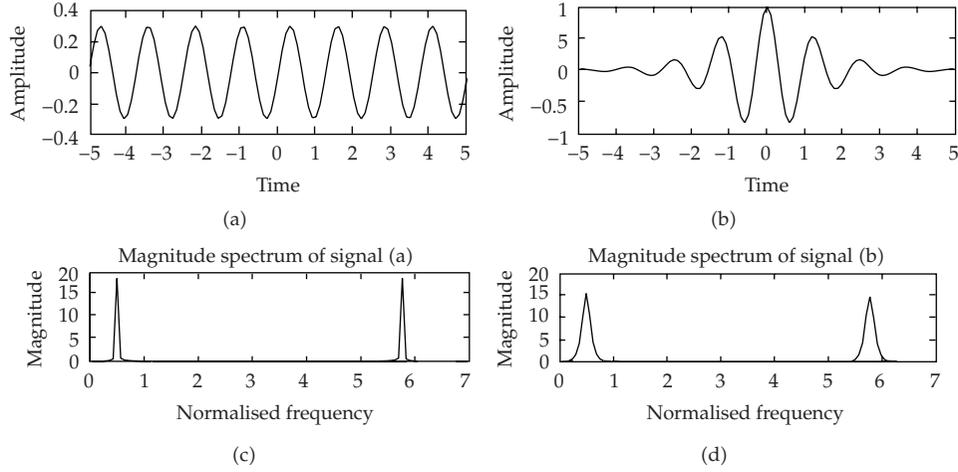


Figure 1: Time and frequency domain representation of two different signals.

Figure 1 shows the time and the frequency domain representation of two different signals. Signal in Figure 1(a) is a sinusoid whereas signal in Figure 1(b) is the modified Morlet wavelet. Despite their completely different time domain characterization they share similar frequency localization given by Figures 1(c) and 1(d), respectively.

In order to be capable of using (3.1) to obtain our desired kernel function we need its eigenvalues and we will use Fourier basis $\{1/2\pi, \sin(nx), \cos(nx), n \in N\}$ as the corresponding eigenfunctions since complex exponentials are the eigenfunctions of any linear time-invariant (LTI) system that includes matched filters and sinusoids can be expressed as linear combination of complex exponentials using Eulers formula [44]. The eigenvalues of and LTI system are given by frequency response $H(e^{j\omega})$ which is a complex-valued quantity [45]. Frequency response can however be written as

$$H(\omega) = |H(\omega)|e^{j\theta(\omega)}, \quad (4.7)$$

namely as a product of magnitude response $|H(\omega)|$ and phase response $\theta(\omega)$ [3]. Since we are only interested in smoothness properties of the kernel and not the phase response, it is adequate to take the magnitude response $|H(\omega)|$ as eigenvalues of the system. Another reason for this is that in order to have positive definite Green's kernel function the eigenvalues need to be strictly positive [38]. Hence, matching Green's kernel function can be obtained by using (3.1):

$$\begin{aligned} G(x, x') &= \sum_{n=1}^{N-1} |H(\omega_n)| (\sin(\omega_n x) \sin(\omega_n x') + \cos(\omega_n x) \cos(\omega_n x')) \\ &= \sum_{n=1}^{N-1} |H(\omega_n)| \cos(\omega_n (x - x')), \end{aligned} \quad (4.8)$$

where ω_n is the discrete time counterpart of continuous frequency variable ω , such that $\omega_n = 2\pi n/N, 0 \leq n \leq N-1$, that is, normalized to have a range of $0 \leq \omega_n \leq 2\pi$. By making use of (4.6) and ignoring the constant eigenfunction with $n = 0$, we get

$$G(x, x') = \sum_{n=1}^{N-1} |F(\omega_n)| \cos(\omega_n(x - x')), \quad (4.9)$$

which is a positive definite SV kernel that exhibits matched filter regularization properties given by $|F(\omega)|$. From the algorithmic point of view, we only need to compute magnitude of the discrete Fourier transform of the training targets with the assumption that the function $f(x)$ to be predicted takes a similar magnitude spectrum with additive noise. To control the model complexity of the system we introduce two variables to restrict the summation calculation to desired eigensubspaces and write (4.9) as

$$G(x, x') = \sum_{n=i}^j |F(\omega_n)| \cos(\omega_n(x - x')), \quad (4.10)$$

where i and j are the kernel parameters for Green's kernel similar to the kernel parameters of other SV kernel such as kernel width σ in the case of Gaussian RBF kernel or degree of the kernel d in the case of polynomial kernel. Similar to other SV kernels an optimal value for i and j is required to achieve the best results.

Analogous to the conventional Gaussian kernel that exhibits Gaussian lowpass filter behavior, that is, $\lambda(\omega) = \exp[-\sigma^2 \|\omega\|^2 / 2]$ [10, 11] (recall that the Fourier transform of a Gaussian function is also a Gaussian function) the knowledge-based Green's kernel obtained from the eigenvalues of the matched filter exhibits the matched filter properties. This property makes the knowledge-based Green's kernel an optimal choice for noise regime since matched filters are the optimal filters for noise-corrupted data regardless of the signal shape and the noise level. Since most of the real world systems are unavoidably contaminated with noise in addition to their intrinsic dynamics [25, 26, 30], we keep up with the long-established tradition [2-4, 25, 26, 29, 30] of using benchmark datasets with additive white noise to evaluate the performance of the proposed techniques and conduct several experiments on mostly benchmark datasets ranging from simple regression models to chaotic and nonlinear time series with additive white noise in order to compare the performance of our technique with that of existing support vector (SV) kernels. Nevertheless, the advantage of knowledge-based Green's kernel comes at the cost of slightly increased computational complexity. However for most of the practical signals only a small portion of the whole eigensubspace turns out to be nonzero thereby lessening the computational load. Another way to overcome this problem is the efficient algorithmic implementation.

A generalization of the kernel function given by (4.10) to N dimensions can be easily made by

$$K(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^d k_i(x^i, y^i) \quad (4.11)$$

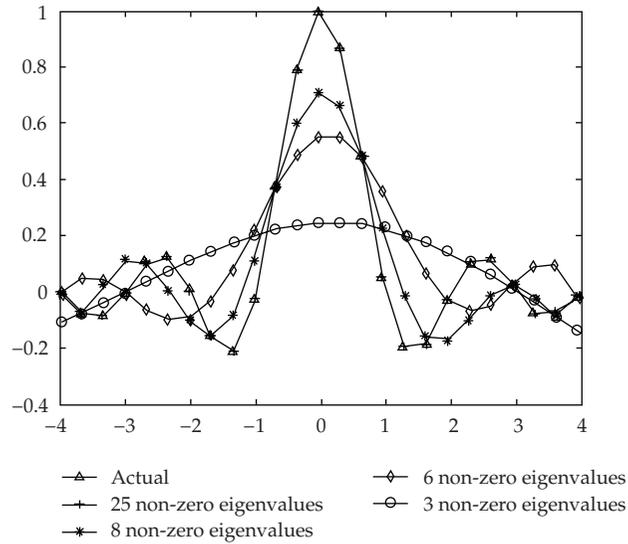


Figure 2: SV regression using Green's kernel with the value of $j = 25, 8, 6, 3$.

(see [2] for proof of the theorem). Alternatively,

$$K(\mathbf{x}, \mathbf{y}) = k(\|\mathbf{x} - \mathbf{y}\|) \quad (4.12)$$

can also be used [10].

5. Experimental Results

5.1. Model Complexity Control

The purpose of this experiment is to examine the ability of Green's kernel to control the complexity of an SV model trained with Green's kernel. Sinc function is used as training and testing data. The training data is approximated with different models built only by reducing the size of eigensubspace in kernel matrix computation, that is, by reducing the value of kernel parameter j while keeping the SV regularization parameter C , and kernel parameter i constant throughout the experiment. In other words, the complexity of the model is reduced by reducing the number of nonzero eigenvalues, that is, reducing the value of j thereby removing the high capacity eigenfunctions to obtain a smoother approximation. The value of $i = 1$ was used for all the models.

Figure 2 shows the regression results obtained for different values of j . It is evident from the figure that reducing the size of eigensubspace produces smoother approximations which highlights the ability of Green's kernel as a regularizer. No GOF criteria are used in this experiment since the point of interest is to produce a smoother approximation not necessarily a good approximation.

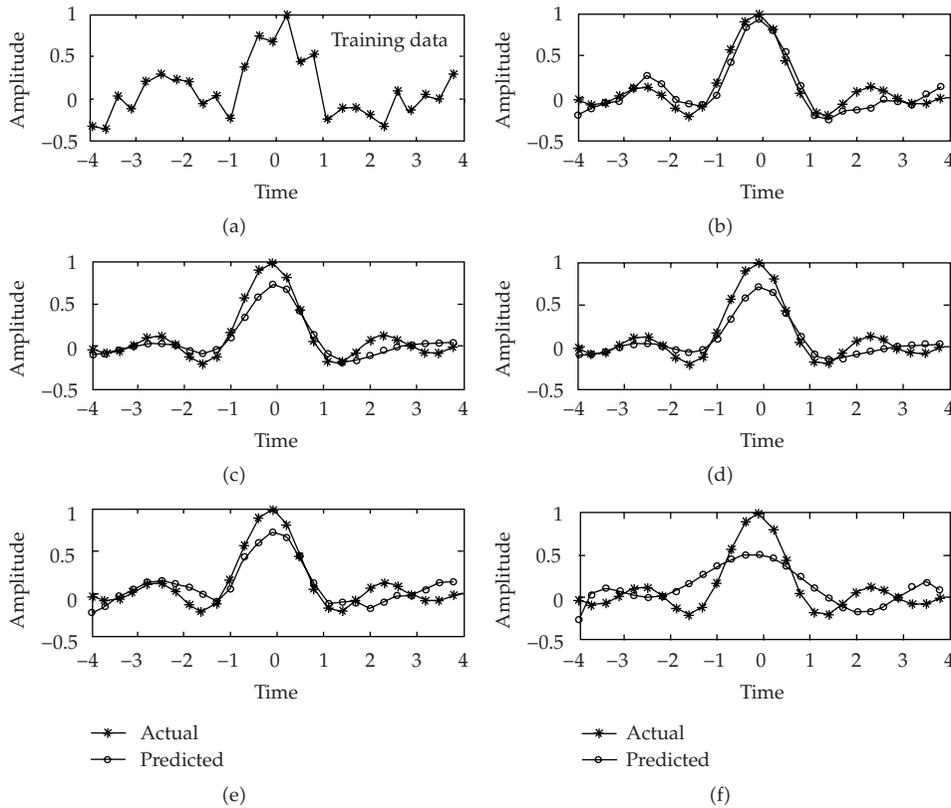


Figure 3: Regression results obtained by (b) Green’s kernel, (c) Gaussian RBF kernel, (d) Bspline kernel, and (e) Exponential RBF kernel, and (f) Polynomial kernel.

5.2. Regression on Sinc Function

Sinc function given by (5.1) has become a benchmark to validate the results of SV regression [2, 3, 10, 34, 35, 46]:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \tag{5.1}$$

The training data is 27 points with zero mean, 0.2 variance additive Gaussian white noise. Mean square error was used as the figure of merit. Figure 3 shows the regression results obtained by Green’s kernel and other commonly used SV kernels. Although the results obtained by Gaussian RBF and Bspline kernel are very similar, we prefer to use Gaussian RBF because only Bspline of odd order n is admissible support vector kernels [10] and this restricts the model complexity control.

Figure 4 shows the magnitude spectrum of the training signal and the actual sinc function. Magnitude spectrum of the training signal is used as the prior knowledge about the actual signal, that is, the signal to be predicted and used to construct the matching Green’s kernel. Table 1 shows the regression results obtained with different kernel functions. Results indicate that Green’s kernel achieved better performance than any other support

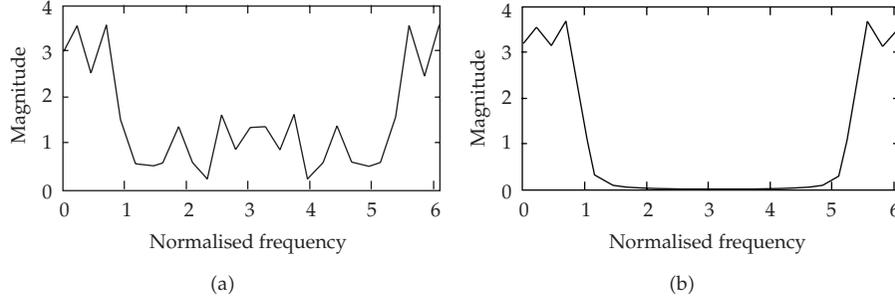


Figure 4: Magnitude spectrum of (a) training signal and (b) actual sinc function.

Table 1: Performance comparison of different kernels for sinc function.

	Kernel function	MSE	No. of SV	CPU Time (Sec.)
1	Green's kernel	0.0126	22	0.007
2	Gaussian RBF	0.0152	22	0.034
3	Bspline	0.0163	23	0.17
4	Exponential RBF	0.0214	24	0.035
5	Polynomial	0.0559	24	0.033

vector kernel for the given function. The CPU time is the kernel matrix computation time in seconds on an Intel (R) 2.8 GHz, 2 GB Memory system using Matlab 7. The CPU time for other kernel functions was computed using [47]. The lesser computational time of knowledge-based Green's kernel is owed to efficient algorithmic implementation which only includes nonzero eigenvalues in kernel matrix computation. The (near) optimal values of SVM hyperparameters for each kernel function were selected after several hundred trials.

5.3. Regression on Modified Morlet Wavelet Function

Modified Morlet wavelet function is described by

$$\text{Modified Morlet Wavelet Function, } \psi(x) = \frac{\cos(\omega_0 x)}{\cosh(x)}. \quad (5.2)$$

This function was selected because of its complex model. A signal of 101 data points with zero mean, 0.3 variance white noise was used the training set. The (near) optimal values of SVM hyperparameters for each kernel function were selected after several hundred trials. Figure 5 shows the performance of the different SV kernels for modified Morlet wavelet function and the magnitude spectrum of training and actual signals. Although the training function is heavily corrupted with noise, there is still some similarity between the magnitude spectrum of two functions and this similarity is used as the prior knowledge about the problem. As shown in Table 2, again, Green's kernel performed better than any other kernel for heavily noise corrupted data.

The purpose of next two experiments is to evaluate the performance of the proposed kernel function against the conventional Gaussian kernel in a broader perspective, that is, across different noise models, noise levels, prediction steps (short-term and long-term

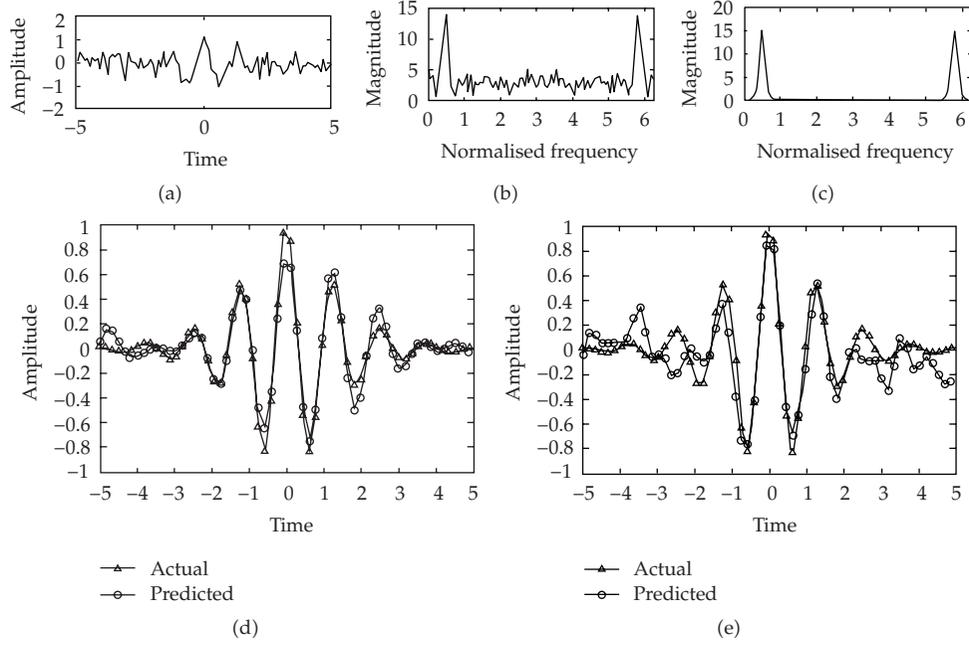


Figure 5: (a) Training signal, magnitude spectrum of (b) training signal, (c) actual (modified Morlet wavelet function) signal, regression results (d) Green's kernel, (e) Gaussian RBF kernel.

Table 2: Performance comparison of different kernels for modified Morlet wavelet function.

	Kernel function	MSE	No. of SV	CPU Time (Sec.)
1	Green's kernel	0.0087	53	0.095
2	Gaussian RBF	0.0272	54	0.45
3	Bspline	1.0173	59	2.37
4	Polynomial	1.0358	59	0.447

prediction for time series), and different variations of SVM that use different loss functions and optimization schemes. To perform a faithful comparison, we use the results already published in literature as our reference point and use the same datasets, noise model, noise level, and loss function as suggested by the corresponding authors. For the next two experiments, long-term and short-term prediction of chaotic time series is considered as a special case of regression. We use Mackey-Glass, a high-dimensional chaotic benchmark time series, originally introduced as a model of blood cell regulation [48]. Mackey Glass is generated by the following delay differential equation [3]:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \quad (5.3)$$

with $a = 0.2$, $b = 0.1$, and $\tau = 17$.

Table 3: Mackey-Glass time series prediction results using SVM and LS-SVM.

Noise	Normal				Uniform					
	22.15%		44.3%		6.2%		12.4%		18.6%	
SNR	1S	100S	1S	100S	1S	100S	1S	100S	1S	100S
Prediction Step	1S	100S	1S	100S	1S	100S	1S	100S	1S	100S
RBF kernel SVM [3]	0.017	0.218	0.040	0.335	0.006	0.028	0.012	0.070	0.017	0.142
Green's kernel SVM	0.00051	0.0989	0.0019	0.0665	0.00017	0.0623	0.00033	0.0661	0.00052	0.115
RBF kernel LS-SVM [4]	0.016	0.165	0.032	0.302	0.005	0.026	0.010	0.064	0.018	0.136
Green's kernel LS-SVM	0.00051	0.0973	0.0016	0.067	0.00011	0.0631	0.00029	0.0645	0.00059	0.0748

5.4. Chaotic Time Series Prediction Using SVM and LS-SVM

For comparison purposes, we use Muller et al. [3] that employs SVM with Gaussian RBF kernel and Zhu et al. [4] that utilizes LS-SVM with Gaussian RBF kernel for short-term (1 step) and long-term (100 step) prediction of Mackey-Glass system for different noise models and noise levels. Table 3 shows the mean square error obtained by Green's kernel using SVM and LS-SVM over different noise settings in comparison to the results reported by [3, 4]. 1S and 100S denote the 1 step and 100 step prediction of time series. We use the same definition of SNR as used by the corresponding authors, that is, ratio between the standard deviation of the respective noise and the underlying time series. Experimental results indicate that knowledge-based Green's kernel should be considered as a good kernel choice for noise-corrupted data.

6. Conclusion

This paper provides a mathematical framework for using Green's functions to construct problem specific admissible support vector kernel functions based on the prior knowledge about smoothness properties of the function to be predicted. Matched filter theorem is used to incorporate domain knowledge of the magnitude spectrum of the signal to be predicted into support vector kernels to achieve desired regularization properties. It has been shown that the knowledge-based matching Green's kernel is a positive definite SV kernel that exhibits matched filter behavior. Since matched filters are known to be the optimal choice for noise corrupted data, the key contribution of the proposed technique is its noise robustness (see Figure 5) which makes it suitable for many real world system. Experimental results show that the knowledge-based Green's kernel has the ability to control the model complexity (see Figure 2) of the system and shows good generalization performance compared to other existing support vector kernels (see Tables 1, 2, and 3). Future research would include implementation of Green's kernel on real world problems such as speech synthesis and ultra sound image analysis.

References

- [1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [2] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [3] K. R. Miller, A. J. Smola, G. Rtsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik, "Using support vector machines for time series prediction," in *Advances in Kernel Methods: Support Vector Machine*, B. Scholkopf, J. Burges, and A. Smola, Eds., MIT Press, Cambridge, Mass, USA, 1999.

- [4] J.-Y. Zhu, B. Ren, H.-X. Zhang, and Z.-T. Deng, "Time series prediction via new support vector machines," in *Proceedings of International Conference on Machine Learning and Cybernetics (ICMLC '02)*, pp. 364–366, November 2002.
- [5] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [6] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Nauka, Moscow, Russia, 1979, English translation: Springer, New York, NY, USA, 1982.
- [7] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, John Wiley & Sons, New York, NY, USA, 1977.
- [8] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Mathematics Doklady*, vol. 4, pp. 1035–1038, 1963.
- [9] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics*, vol. 13, no. 1, pp. 1–50, 2000.
- [10] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, Mass, USA, 2001.
- [11] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, no. 4, pp. 637–649, 1998.
- [12] R. V. Babu, P. Pérez, and P. Bouthemy, "Robust tracking with motion estimation and local Kernel-based color modeling," *Image and Vision Computing*, vol. 25, no. 8, pp. 1205–1216, 2007.
- [13] R. Maclin, E. Wild, J. Shavlik, L. Torrey, and T. Walker, "Refining rules incorporated into knowledge-based support vector learners via successive linear programming," in *Proceedings of the 22nd National Conference on Artificial Intelligence*, vol. 1, pp. 584–589, July 2007.
- [14] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens, "Graph nodes clustering with the sigmoid commute-time kernel: a comparative study," *Data and Knowledge Engineering*, vol. 68, no. 3, pp. 338–361, 2009.
- [15] G. Toma, "Specific differential equations for generating pulse sequences," *Mathematical Problems in Engineering*, vol. 2010, Article ID 324818, 11 pages, 2010.
- [16] M. Li and J.-Y. Li, "On the predictability of long-range dependent series," *Mathematical Problems in Engineering*, vol. 2010, Article ID 397454, 9 pages, 2010.
- [17] E. G. Bakhoun and C. Toma, "Mathematical transform of traveling-wave equations and phase aspects of quantum interaction," *Mathematical Problems in Engineering*, vol. 2010, Article ID 695208, 15 pages, 2010.
- [18] Z. Liu, "Chaotic time series analysis," *Mathematical Problems in Engineering*, vol. 2010, Article ID 720190, 31 pages, 2010.
- [19] T. Poggio and F. Girosi, *A Theory on Networks for Approximation and Learning*, MIT Press, Cambridge, Mass, USA, 1989.
- [20] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.
- [21] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, pp. 219–269, 1995.
- [22] F. Girosi, M. Jones, and T. Poggio, "Priors, stabilizers and basis functions: from regularization to radial, tensor and additive splines," AI Memo no. 1430, MIT AI Lab, 1993.
- [23] A. J. Smola and B. Schölkopf, "On a kernel-based method for pattern recognition, regression, approximation and operator inversion," *Algorithmica*, vol. 22, no. 1-2, pp. 211–231, 1998.
- [24] R. C. Williamson, A. J. Smola, and B. Schölkopf, "Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2516–2532, 2001.
- [25] R. Metzler, Y. Bar-Yam, and M. Kardar, "Information flowsystem chaotic channel: prediction and ostdiction at finite resolution," *Physical Review E*, vol. 70, Article ID 026205, 2004.
- [26] G. Nolte, A. Ziehe, and K.-R. Müller, "Noise robust estimates of correlation dimension and K_2 entropy," *Physical Review E*, vol. 64, no. 1, Article ID 016112, 10 pages, 2001.
- [27] V. Olshevsky and L. Sakhnovich, "Matched filtering for generalized stationary processes," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3308–3313, 2005.
- [28] G. L. Turin, "An introduction to matched filters," *IRE Transactions on Information Theory*, vol. 6, pp. 311–329, 1960.
- [29] E. S. Chng, S. Chen, and B. Mulgrew, "Gradient radial basis function networks for nonlinear and nonstationary time series prediction," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 190–194, 1996.

- [30] D. Lowe and A. R. Webb, "Time series prediction by adaptive networks. A dynamical systems perspective," *IEEE Proceedings F*, vol. 138, no. 1, pp. 17–24, 1991.
- [31] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "Training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed., pp. 144–152, July 1992.
- [32] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," in *Advances in Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds., vol. 5, pp. 147–155, Morgan Kaufmann, San Mateo, Calif, USA, 1993.
- [33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [34] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Advances in Neural Information Processing Systems*, M. Mozer, M. Jordan, and T. Petsche, Eds., vol. 9, MIT Press, Cambridge, Mass, USA, 1997.
- [35] A. J. Smola and B. Schlkopf, "A tutorial on support vector regression," Neuro COLT Tech. Rep. TR-1998-030, Royal Holloway College, 1998.
- [36] M. A. Aizerman, È. M. Braverman, and L. I. Rozonoër, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.
- [37] N. J. Nilsson, *Learning Machines: Foundations of Trainable Pattern Classifying Systems*, McGraw-Hill, New York, NY, USA, 1965.
- [38] G. F. Roach, *Green's Functions Introductory Theory with Applications*, Van Nostrand Reinhold, London, UK, 1970.
- [39] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equation," *Philosophical Transactions of the Royal Society A*, vol. 209, pp. 415–446, 1909.
- [40] D. O. North, "An analysis of the factors which determine signal/noise discrimination in pulsed-carrier systems," Tech. Rep. PTR-6c, RCA Lab, June 1943.
- [41] D. O. North, "An analysis of the factors which determine signal/noise discrimination in pulsed-carrier systems," *Proceedings of the IEEE*, vol. 51, no. 1, pp. 1016–1027, 1963.
- [42] H. J. Blinichikoff and A. I. Zverev, *Filtering in the Time and Frequency Domains*, John Wiley & Sons, New York, NY, USA, 1976.
- [43] B. V. Kumar, A. Mahalanobis, and R. D. Juday, *Correlation Pattern Recognition*, Cambridge University Press, New York, NY, USA, 2005.
- [44] J. O. Smith, "Introduction to digital filters with audio applications," August 2006, <https://ccrma.stanford.edu/~jos/filters/>.
- [45] O. Akay, "Linear fractional shift invariant (LFSI) systems," in *Proceedings of the 7th International Symposium on Signal Processing and Its Applications*, pp. 585–588, 2003.
- [46] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.
- [47] S. Gunn, "Support Vector Machines for Classification and Regression," Tech. Rep., Image Speech and Intelligent Systems Research Group, University of Southampton, 1997.
- [48] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

