A New Composition Method of Admissible Support Vector Kernel Based on Reproducing Kernel

Wei Zhang, Xin Zhao, Yi-Fan Zhu, and Xin-Jian Zhang

Abstract—Kernel function, which allows the formulation of non-linear variants of any algorithm that can be cast in terms of dot products, makes the Support Vector Machines (SVM) have been successfully applied in many fields, e.g. classification and regression. The importance of kernel has motivated many studies on its composition. It's well-known that reproducing kernel (R.K) is a useful kernel function which possesses many properties, e.g. positive definiteness, reproducing property and composing complex R.K by simple operation. There are two popular ways to compute the R.K with explicit form. One is to construct and solve a specific differential equation with boundary value whose handicap is incapable of obtaining a unified form of R.K. The other is using a piecewise integral of the Green function associated with a differential operator $\mathcal{L}$. The latter benefits the computation of a R.K with a unified explicit form and theoretical analysis, whereas there are relatively later studies and fewer practical computations. In this paper, a new algorithm for computing a R.K is presented. It can obtain the unified explicit form of R.K in general reproducing kernel Hilbert space. It avoids constructing and solving the complex differential equations manually and benefits an automatic, flexible and rigorous computation for more general RKHS. In order to validate that the R.K computed by the algorithm can be used in SVM well, some illustrative examples and a comparison between R.K and Gaussian kernel (RBF) in support vector regression are presented. The result shows that the performance of R.K is close or slightly superior to that of RBF.

Keywords—admissible support vector kernel, reproducing kernel, reproducing kernel Hilbert space, Green function, support vector regression

I. INTRODUCTION

THE mathematical result underlying the kernel function, which was presented almost a century ago [1], has received relatively little attention in the machine learning community until support vector machines (SVM) was proposed [2]. SVM is one of excellent kernel methods, whose crucial ingredient is the kernel function. It's shown that SVM has many advantages, e.g. no local optima, good ability of generalization, intrinsic regularization and the sparseness of support vectors, etc. [3]. These advantages encourage researchers actively focus on applying SVM to various fields e.g. pattern recognition [4]–[6] and face detection [7], [8]etc. Moreover, it's also rapidly applied in the domain of regression, e.g. function approximation [3], [9], prediction [10], [11] and other applications [12]. The tutorial can be seen in [13], [14].

Wei Zhang is with the College of Information Systems and Management, National University of Defense Technology (NUDT), Changsha, Hunan, 410073, China (e-mail: the_ant@163.com).
Xin Zhao is with the College of Information Systems and Management, NUDT, Changsha, Hunan, China. (e-mail: zhx20mg@163.com).
Yi-Fan Zhu is with the College of Information Systems and Management, NUDT, Changsha,Hunan, China (e-mail: nudtzyf@hotmail.com).
Xin-Jian Zhang is with the College of Mathematic, NUDT, Changsha, Hunan, China, (e-mail: xjz_20075@163.com).

Kernel function is a crucial ingredient of SVM. It's known that a kernel is called admissible support vector kernel (SV kernel) [14] when it satisfies Mercer's condition [1]. Mercer's condition is one of popular methods to validate whether a prospective kernel is a positive definite function since any kernel should be capable of corresponding to a dot product in high dimensional feature space. Kernel function is regarded as a significant trick which can benefit the computation of dot products in high-dimensional feature space using simple function defined on pairs of input patterns [15], [16]. In other words, the computations depend only on the inner product defined in feature space, rather than an explicit nonlinear function φ. Additionally, the SV kernel implies the features of data in feature space since it contains all the information about the relative positions of the data, i.e. choosing different kernel will produce different SVMs.

However, it is usually time-consuming and demanding to validate or compose a SV kernel. It's well-known that almost all the methods, e.g. Mercer's method, only tell us whether or not a prospective kernel is actually a dot product in a given space, but it does not show how to construct the feature map and the images of the input data in the feature space and even what the feature space is. Although, there are plentiful studies on SVMs with some conventional kernels, e.g. polynomial kernel $K(x, x') = (< x, x' > +1)^d$, Gaussian kernel (RBF) $K(x, x') = \exp\left(-\|x - x'\|/(2\sigma^2)\right)$ and sigmoid kernel $K(x, x') = \tanh\left(\nu < x, x' > +c\right)$, the best choice of a kernel for a given problem is still an open research issue [6]. It's found that the polynomial kernel is usually inferior for higher nonlinearity problem and sigmoid kernel performs closely to RBF but with complex form, conditional satisfaction with Mercer's condition and unintelligibility [14]. Research has shown that RBF is not only theoretically well-founded but also superior in some practical classification applications [6], [17]. However, the performance of RBF is sensitive to the parameter σ [18], and there is no evidence to show that the RBF is the optimal choice for regression, especially dealing with multivariable complex function.

Therefore, many researches are devoted to an exploratory study on the composition method of SV kernels and related properties, e.g. hybrid composition method based on some operations of kernels, e.g. positive linear combinations, integrals and products, etc. [14], [19], [20], multi-scale kernel [21] especially based on wavelet functions [22], [23] and feature space [24], [25], such as reproducing kernel Hilbert space (RKHS) [26], [27], etc. Recently, the multi-scale kernel and RKHS are the research hotspots. Although the former adopts techniques from wavelet theory and shift invariant spaces to

construct a new class of kernels, it still bases on RKHS [21]. Furthermore, as a result of reproducing property of R.K, it can show us a feature map with explicit expression and the corresponding RKHS. Therefore, the attention is paid to the kernel function in RKHS.

Reproducing kernel(R.K) is a well-known kernel function in RKHS, which is regarded as a special SV kernel. Although the basis concept and principle [28], frames [29] and properties [30], and conceptual comparison of R.K to the other kernels, e.g. Mercer kernel, positive definite kernel(PDK) [25], etc., have been well studied, there are relatively little work on quantitative analysis and comparison in SVR based on some R.K with explicit expressions. There are two popular ways to compute R.K in RKHS. One is constructing and solving a specific differential equation with boundary value problem [31], [32] whose handicaps are the requirement of professional mathematic knowledge and incapable of extending to more general cases. The other is to compute a R.K with a unified explicit expression using a piecewise integral of Green function associated with a differential operator $\mathcal{L}$ [33] but with relatively later and less researches.

In this paper, an algorithm for composing R.K in Sobolev RKHS is proposed to obtain a unified explicit expression. It avoids constructing and solving the complex differential equations with boundary-value manually and benefits an automatic, flexible and rigorous computation with computer codes, e.g. Matlab. Finally, some examples and simulation comparisons are presented to validate the R.K computed. The result shows that the performance of R.K in SV algorithm is close or superior to that of conventional RBF.

II. Preliminary

A. Kernel Methods and SVM Formulation

Kernel methods (KMs) owe the name to the use of kernel functions, which handle the problems by mapping the data into a high dimensional feature space, where each coordinate corresponds to one feature of the data items. It is a significant trick that the learning machine can be operated in the feature space by simply computing the inner products between the images of all pairs of data in the feature space. Since the mapping can be quite general (not necessarily linear, for example), the relations found in this way are accordingly very general. There are several advantages in employing kernel function:

1) *Offering a computational shortcut.* The kernel function can implicitly map the input vectors into feature space to contain all the information about the relative positions rather than explicitly choose feature space and mapping functions. Therefore, the training data are only participated in the computation of the kernel matrix.

2) *Avoiding the curve of dimension.* The amount of computation required depends on the number of support vectors (SVs) rather than the dimensionality of primal input space.

3) *Allowing incorporating prior knowledge.* The kernel function allows one to incorporate prior knowledge of

problem domain since a similarity measure between two input data [16].

These attractive properties make the KMs widely applied in many fields, e.g. support vector machines [2] , kernel principle component analysis [34], kernel Fisher discriminate analysis [35] and kernel self-organizing map [36], etc.

In this section, a very brief introduction of SVMs will be presented as the application background of the composition approach of SV kernel.

Given a training set $\mathfrak{D} = \{(x_i, y_i), i = 1, ..., l\} \subset \Omega \times \mathcal{C}$, where Ω denotes the space of the input data (e.g. $\Omega = \mathbb{R}^d$, where d denotes dimensionality of input), $\mathcal{C}$ denotes the space of output (e.g. $\mathcal{C} = \{-1, +1\}$ for binary classification, $\mathcal{C} = \mathbb{R}$ for regression). SV algorithms aim at minimizing an upper bound of the generalization error through maximizing the margin between the separating hyperplane and the data, which is based on the structural risk minimization principle [14].

SV algorithm is to find the following curve in feature space both for classification or regression problems:

$$f(x) = \sum_{i \in SV} \beta_i K(x_i, x) + b \tag{1}$$

where $\beta_i = \alpha_i y_i$ for classification, $\beta_i = \alpha_i - \alpha_i^*$ for regression respectively, α_i and α_i^* are Lagrange multipliers, b denotes the bias, $i \in SV$ denotes the indices of support vectors (SVs), i.e. x_i with nonzero α_i or α_i^*, $K(\cdot, \cdot)$ is the kernel function.

It can be solved by the following constrained optimization problems:

(1) For classification problem [2], [13]:

$$\begin{aligned} &\max_{\alpha_i} \sum_{i=1}^{l} \alpha_i - \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ &s.t. \sum_{i=1}^{l} y_i \alpha_i = 0, 0 \le \alpha_i \le C, i = 1, ..., l \end{aligned} \tag{2}$$

(2) For regression problem [14]:

$$\begin{aligned} \max_{\alpha, \alpha^*} &-\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \\ &+ \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*) y_i - \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)\varepsilon \\ s.t. &\sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0, \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{3}$$

where $C > 0$ determines the upper bound, $\varepsilon \ge 0$ denotes a constant controlling the noise tolerances.

Obviously, the complexity of (1) only depends on the amount of SVs and SV kernel rather than the dimensionality of Ω. In practice, the SVs, which depend on the selection of kernel and parameters of SV algorithm [6], can be automatically extracted by SV algorithm. In other words, the major task of the SVM lies in the selection of its kernel [19].

B. Validating Approaches for SV Kernels

Kernel function is a crucial ingredient in SV algorithm, and a kernel function is called a SV kernel when it satisfies Mercer's condition [14]. As noted in precious section, the SV algorithm only depends on dot products between input

x_i, i.e. SV kernel essentially. In addition, the optimization problem in nonlinear case corresponds to finding the flattest function in feature space rather than in input space [14]. Therefore, different SV kernel implies different feature space, and consequently different reflection of the feature of the estimation function.

The question that raises now is which function $K(s,t)$ corresponds to a dot product in a feature space. There are large numbers of studies on this, e.g. [2], [14], [37], [38]. The following theorems including Mercer' and Bochner's theorem, represent these functions.

Theorem 1 (Mercer's Theorem): Let Ω be a closed subset of $\mathbb{R}^n$, $n \in \mathbb{N}$, μ be a Borel measure on Ω. Suppose $K \in L_\infty(\Omega^2)$ such that the integral operator $T_K : L_2(\Omega) \to L_2(\Omega)$ defined by

$$T_K f(\cdot) := \int_\Omega K(\cdot, x) f(x) d\mu(x) \qquad (4)$$

is semi-positive. Let $\psi_i \in L_2(\Omega)$ be the eigenfuntion of T_K associated with the eigenvalue $\lambda_i \neq 0$ and normalized such that $\|\psi_i\|_{L_2} = 1$ and let ψ_i denote its complex conjugate. Then

1) $(\lambda_i(T))_i \in l_1$
2) $\psi_i \in L_\infty(\Omega)$ and $\sup_i \|\psi_i\|_{L_\infty} < \infty$
3) $K(x,x') = \sum_{i \in \mathbb{N}} \lambda_i \psi_i(x)\psi_i(x')$ (referred to as Mercer kernel) holds for almost all (x, x'), where the series converges absolutely and uniformly for almost all (x, x').

Less formally speaking this theorem means that if

$$\int_{\Omega \times \Omega} K(x, x') f(x) f(x') dx dx' \geq 0, \text{for all } f \in L_2(\Omega) \quad (5)$$

holds, then $K(x, x')$ can be written as a dot product in some feature space, i.e. any function $K(x, x')$ which satisfies Mercer's condition is a SV kernel. Unfortunately, to validate some function in respect of the Mercer's condition is still of difficulty and intractability.

Theorem 2 (Bochner's Theorem [38]): Given a positive finite Borel measure μ on $\mathbb{R}$, the Fourier transform $\mathcal{Q}$ of μ, i.e. $\mathcal{Q}(t) = \int_\mathbb{R} \exp(-itx) d\mu(x)$ is a continuous function, then $\mathcal{Q}$ is a positive definite function and vice versa. In other words, every positive definite function is the Fourier transform of a positive finite Borel measure, i.e. the kernel takes the form $K(x, x') = \mathcal{Q}(x - x')$ is positive definite, and vice versa.

Here, the kernel function in theorem 2 is called translation invariant kernel, e.g. $K(x, x') = \exp\left(-\|x - x'\|^2/(2\sigma^2)\right)$. Smola *et al.* [37] presented the following method for validating a SV kernel based on the Bochner's theorem.

Theorem 3: A kernel $K(x, x') = K(x - x')$ is an admissible SV kernel if and only if the Fourier transform

$$\mathcal{F}[K](\omega) = \hat{K}(\omega) = (2\pi)^{-\frac{d}{2}} \int_\Omega e^{-i<\omega,x>} K(x) dx \qquad (6)$$

is nonnegative.

Moreover, for kernels $K(x, x') = K(< x, x' >)$ (dot-product kernel), there exists sufficient conditions for being admissible, for further details see [39].

III. Perspectives of Reproducing Kernel and SV Kernel

A. Definition of Reproducing Kernel

It is well-known that the reproducing kernel Hilbert space (henceforth abbr. as RKHS) is the foundation of the learning, sampling, filtering and scattered approximation problems. Some basic concepts will be introduced at first. For more details on RKHS see e.g. [26], [28], [30], [40].

Defintion 1 (RKHS): Let $\Omega \subseteq \mathbb{R}^d$ be an arbitrary nonempty set, $\mathcal{H}$ is a Hilbert space of function $f : \Omega \to \mathbb{R}$ (short for $f \in \mathbb{R}^\Omega$). The $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists $K : \Omega \times \Omega \to \mathbb{R}$, satisfies the following:

(i) For $\forall x$, $K_x(y) = K(y, x)$ as a function of y belongs to $\mathcal{H}$.

(ii) The reproducing property: $\forall x \in \Omega$, and $\forall f \in \mathcal{H}$,

$$f(x) = <f, K_x> \qquad (7)$$

(iii) $\mathcal{H}$ is spanned by K, that is, $\mathcal{H} = \overline{span\{K_x(\cdot)|x \in \Omega\}}$ And, $\mathcal{H}$ is called the native space of K [30].

Defintion 2 (R.K): $K : \Omega \times \Omega \to \mathbb{R}$ is called a R.K of $\mathcal{H}$, if it satisfies the conditions (i) and (ii) in Definition 1.

Here, some denotations are presented, i.e., the Hilbert space $\mathcal{H}$ with R.K K is denoted by $\mathcal{H}_K(\Omega)$, and correspondingly norm and inner product will be denoted by $\|\cdot\|_K$ and $<\cdot, \cdot>_K$ respectively.

The R.K possesses some basic properties, e.g. uniqueness, existence, positive definiteness, convergence, projection, and compositing complex R.K with simpler ones by sums, products and other operations, etc. [28].

As stated previously, the theorems presented above can only tell us whether or not a prospective kernel is actually a dot product in a given space, rather than show how to construct the feature map and the images of the input data in the feature space. Moreover, applying (7) to function K_x at y, then

$$K(y, x) = K_x(y) = <K_x, K_y>, \text{for } x, y \in \Omega \qquad (8)$$

which implies that the nonlinear map function takes the form of R.K with any fixed x or y. Whereas the conventional SV kernels are incapable of presenting the map functions explicitly. As a result, the features of images of data can be analyzed further.

B. Relations between SV Kernel and Reproducing Kernel

It's necessary to discuss the relations between various kernels to validate that the R.K can be used as a SV kernel. It is hoped that the discussion here would help to bridge the conceptual gap between some familiar kernels, e.g. positive (semi-)definite kernel (PDK), Mercer kernel and R.K, whereas some of the observations are not new or profound.

Defintion 3 (PDK): Let Ω be a subset of $\mathbb{R}^n$, $n \in \mathbb{N}$, $K : \Omega \times \Omega \to \mathbb{R}$ is symmetric and positive (semi-)definite (PD), if and only if for arbitrary finite sets $\{x_1, ..., x_m\} \subseteq \Omega$, the matrix $\overline{K} = (K(x_i, x_j))_{1 \leq i, j \leq m}$ is symmetric and positive definite, i.e. for $\forall m \in \mathbb{N}$, $\forall c_i \in \mathbb{R}$, and $\forall x_1, ..., x_m \in \Omega$, $i = 1, ..., m$, K satisfies the following inequation

$$\sum_{i,j=1}^m c_i c_j K(x_i, x_j) \geq 0 \qquad (9)$$

Theorem 4: $K : \Omega \times \Omega \to \mathbb{R}$ is a SV kernel iff K is a PDK. The proof is obvious. Refer to e.g. [41].

Theorem 5: $K : \Omega \times \Omega \to \mathbb{R}$ is a Mercer kernel iff K is a PDK.

Proof: if K is a Mercer kernel, i.e. there exists a map function Φ such that $K(t,s) = < \Phi(s), \Phi(t) >$. Then,

$$
\begin{aligned}
\sum_{i,j=1}^{m} c_i c_j K(x_i, x_j) &= \sum_{i,j=1}^{m} c_i c_j < \Phi(x_i), \Phi(x_j) > \\
&= \| \sum_{i=1}^{m} c_i \Phi(x_i) \|^2 \geq 0
\end{aligned}
$$

thus, K is a PDK according to (9).

For the converse, if K is a PDK, K is a Mercer kernel according to Theorem 4 and 1, which completes the proof. ∎

Theorem 6: $K : \Omega \times \Omega \to \mathbb{R}$ is a Mercer kernel iff there exists a RKHS $\mathcal{H}$ with R.K K, i.e. $\mathcal{H}_K(\Omega)$.

Proof: According to *Moore-Aronszajn Theorem* [28], any PDK K is associated with a space $\mathcal{H}_K(\Omega)$ and vice versa. Note that the Theorem 5 holds if K is a PDK, that is, K is a Mercer kernel, which completes the proof. ∎

IV. A NEW COMPOSITION APPROACH OF SV KERNEL

As stated in previous section, any R.K can be used as a SV kernel theoretically. Therefore, it's important to discuss how to compute a R.K with an explicit expression. In other words, it is a composition approach of SV kernel.

A. Existence and Construction Theorems

It's well-known that it is usually very time-consuming and demanding to compute the R.K for a given RKHS. There are two popular ways to compute the explicit form of a R.K. One is to construct and solve a specific differential equation with boundary value for a special case [31], [32]. It's found that it is incapable of obtaining a unified form of R.K for general RKHS since the differential equations have to be constructed for special case manually. For example, Cui *et al.* [42] defined the Inner products of RKHS $W_2^1[a,b]$ and $W_2^2[a,b]$ as follow:

$$
< u, v >= \int_{\Omega} [u(x)v(x) + u'(x)v'(x)]\, dx \quad (10)
$$

$$
< u, v >= \int_{\Omega} [u(x)v(x) + 2u'(x)v'(x) + u''(x)v''(x)]\, dx \quad (11)
$$

and constructed the following differential equations:

$$
\begin{aligned}
-d^2 F_x(y)/dy^2 + F_x(y) &= 0, y \neq x \\
dF_x(a)/dy = dF_x(b)/dy &= 0 \\
dF_x(x-0)/dy - dF_x(x+0)/dy &= 1
\end{aligned} \quad (12)
$$

and

$$
\begin{aligned}
d^4 F_x(y)/dy^4 - 2d^2 F_x(y)/dy^2 + F_x(y) &= 0, y \neq x \\
d^2 F_x(b)/dy^2 = d^2 F_x(a)/dy^2 &= 0 \\
d^3 F_x(b)/dy^3 - 2dF_x(b)/dy &= 0 \\
d^3 F_x(a)/dy^3 - 2dF_x(a)/dy &= 0 \\
dF_x(x+0)/dy = dF_x(x-0)/dy \\
F_x(x+0) = F_x(x-0) \\
d^2 F_x(x+0)/dy^2 = d^2 F_x(x-0)/dy^2 \\
d^3 F_x(x+0)/dy^3 - d^3 F_x(x-0)/dy^3 &= 1
\end{aligned} \quad (13)
$$

respectively to obtain the corresponding R.Ks for the first time, e.g. the R.K of $W_2^1[a,b]$ is as follows:

$$
K(x,y) = \begin{cases} \frac{e^{x+y}+e^{2a+2b-(x+y)}+e^{2a-(y-x)}+e^{2b+(y-x)}}{2(e^{2b}-e^{2a})}, y \leq x \\ \frac{e^{x+y}+e^{2a+2b+(x+y)}+e^{2a+(y-x)}+e^{2b-(y-x)}}{2(e^{2b}-e^{2a})}, \text{otherwise} \end{cases} \quad (14)
$$

It's noted that this method is too demanding to obtain an explicit form of R.K in more general RKHS, and consequently there are no solutions to $W_2^m(\Omega)$ for $m \geq 3$ so far (see [42], [43] for more details).

The other is based on a piecewise integral of Green function associated with a differential operator $\mathcal{L}$ [33], [44], [45]. It's noted that it can obtain a R.K with a unified explicit expression and benefits the theoretical analysis, whereas there are relatively later studies and fewer practical computations.

In this section, an algorithm for computing a R.K with a unified explicit form in Sobolev RKHS $W_2^m(\Omega)$ is presented. It avoids constructing and solving the differential equations with boundary values manually and benefits an automatic, flexible and rigorous computation with computer codes, e.g. Matlab.

Defintion 4 (Sobolev): A Sobolev space W_p^m is defined to be the subset of L^p such that function f and its weak derivatives up to some order m have a finite L^p norm, for given $p \geq 1$. In other words, a Sobolev space $W_p^m(\Omega)$ consists of function f, where f is m-times absolutely continuous differentiable on Ω, and $f^{(m)}(x) \in L^p(\Omega)$, $(1 \leq p \leq \infty)$.

1) For general differential operator: As an illustration, the computation in $W_2^m[a,b]$ is considered (henceforth W_2^m for short), whose inner product is adapted to the following form

$$
< u, v >= \sum_{i=1}^{m} (\lambda_i u)(\lambda_i v) + \int_a^b \mathcal{L}u(t) \cdot \mathcal{L}v(t) dt \quad (15)
$$

and give the following existence and construction theorems:

Theorem 7: $W_2^m[a,b]$ possesses a R.K as follows:

$$
K(t,s) = \sum_{i=1}^{m} \varphi_i(t)\varphi_i(s) + \int_a^b G(t,\tau) \cdot G(s,\tau) d\tau \quad (16)
$$

where $\mathcal{L} = D^m + a_{m-1}(t)D^{m-1} + a_1(t)D + a_0(t)$, $t \in [a,b]$ is any m-order linear differential operator, $D = d/dt$, $G(\cdot,\cdot)$ is a Green function and φ_i is the dual basis of λ_i for any $i = 1, ..., m$ in $\ker \mathcal{L}$ (the null space of $\mathcal{L}$), i.e.,

$$
\mathcal{L}\varphi_j(t) = 0, \lambda_i \varphi_j = \delta_{i,j}, i,j = 1, ..., m \quad (17)
$$

which can be computed by

$$
\Phi(t)^T = (\varphi_1(t), ..., \varphi_m(t)) = (u_1(t), ..., u_m(t)) M^{-1} \quad (18)
$$

where $\lambda_i \in \ker \mathcal{L}$ is a linearly independent functional, $\{u_i\}_{i=1}^{m}$ is any set of basis of $\ker \mathcal{L}$, $M = [\lambda_i u_j]_{m \times m}$ is a $m \times m$ matrix.

Proof: Firstly, given $\forall r_i \in \mathbb{R}$, $i = 1, ..., m$, $h \in L^2[a,b]$, suppose $u_f \in \ker \mathcal{L}$ satisfies the following equations:

$$
D^{j-1}u_f(a) = r_j, j = 1, ..., m \quad (19)
$$

Then, there exists a $f \in W_2^m$ as follows

$$
f(t) = u_f(t) + \int_a^t G_{\mathcal{L}}(t,\tau)\mathcal{L}f(\tau)d\tau \quad (20)
$$

is the solution of the following initial value problem

$$\mathcal{L}f(t) = h(t) \quad , \quad t \in [a, b]$$
$$D^{j-1}f(a) = D^{j-1}u_f(a) = r_j \quad , \quad j = 1, ..., m \tag{21}$$

where $G_{\mathcal{L}}(t, s)$ is the Green function associated with $\mathcal{L}$, i.e.

$$G_{\mathcal{L}}(t, s) = 0, t \in [a, s]; \mathcal{L}G_{\mathcal{L}}(t, s) = 0, t \in [s, b]$$
$$D^{j-1}G_{\mathcal{L}}(t, s)|_{t=s} = \delta_{j,m}, j = 1, ..., m \tag{22}$$

where δ denotes Kronecker Delta. It's obvious that substituting (22) and (20) into (21) can complete the verification.

For $\forall f \in W_2^m$, let $\Lambda : W_2^m \to \mathbb{R}^m | \Lambda f = (\lambda_1 f, ..., \lambda_m f)$ be a linear operator, $P : W_2^m \to \ker \mathcal{L}$ be a projection operator, then $\ker \mathcal{L} = (\ker \Lambda)^{\perp}$ and

$$(Pf)(t) = \sum_{i=1}^{m}(\lambda_i f)\varphi_i(t) = \Lambda f \cdot \Phi(t) \tag{23}$$

And let $G(t, s) = (I - P)G_{\mathcal{L}}(t, s)$, then

$$f(t) = \sum_{i=1}^{m}(\lambda_i f)\varphi_i(t) + \int_a^b G(t, \tau)\mathcal{L}f(\tau)d\tau \tag{24}$$

Since $u_f \in \ker \mathcal{L}$, then $(Pu_f)(t) = u_f(t)$, thus

$$Pf(t) = P\left(u_f(t) + \int_a^t G_{\mathcal{L}}(t, \tau)\mathcal{L}f(\tau)d\tau\right)$$
$$= u_f(t) + \int_a^t P_{(t)}G_{\mathcal{L}}(t, \tau)\mathcal{L}f(\tau)d\tau \tag{25}$$

where $P_{(t)}$ denotes the projection with respect to t, and thus

$$f = (I - P)f + Pf = \int_a^t \left(I - P_{(t)}\right)G_{\mathcal{L}}(t, \tau)\mathcal{L}f(\tau)d\tau$$
$$+ \quad Pf = \sum_{i=1}^{m}(\lambda_i f)\varphi_i(t) + \int_a^b G(t, \tau)\mathcal{L}f(\tau)d\tau \tag{26}$$

$$\mathcal{L}G(\cdot, \tau) = \delta(\cdot - \tau), \lambda_i G(\cdot, \tau) = 0, i = 1, ..., m \tag{27}$$

Secondly, it's easy to validate that (15) is the inner product of W_2^m. In addition, for any Cauchy sequence $\{f_n\}$ in W_2^m, $\exists f_0 \in W_2^m$ such that $\Lambda f_n \stackrel{n \to \infty}{\longrightarrow} \Lambda f_0$ since $\mathbb{R}^m$ is complete. Since $\{\mathcal{L}f_n\}$ is a Cauchy sequence in $L^2[a, b]$, then $\exists h \in L^2[a, b]$ such that $\mathcal{L}f_n \stackrel{n \to \infty}{\longrightarrow} h$ as $L^2[a, b]$ is complete too. Thus, for any Cauchy sequence $\{f_n\}$ in W_2^m, $\exists f_0 \in W_2^m$ is as follows:

$$f(t) = \Lambda f_0 \cdot \Phi(t) + \int_a^b G(t, \tau)h(\tau)d\tau \tag{28}$$

such that $f_n \stackrel{n \to \infty}{\longrightarrow} f_0$ in W_2^m, which proves that W_2^m is a Hilbert space.

Finally, $K(t, s)$ in (16) is a symmetric function, and $K(t, s)$ belongs to W_2^m as a function of either t or s. Additionally, the following equations hold

$$\mathcal{L}K(\cdot, \tau) = G(\tau, \cdot), \lambda_i K(\cdot, \tau) = \varphi_i(\tau), i = 1, ..., m \tag{29}$$

Thus, substituting (29) into (24)

$$f(t) = \sum_{i=1}^{m}(\lambda_i f)\varphi_i(t) + \int_a^b G(t, \tau)\mathcal{L}f(\tau)d\tau$$
$$= \sum_{i=1}^{m}(\lambda_i f)(\lambda_i K(\cdot, t)) + \int_a^b \mathcal{L}f(\tau)\mathcal{L}K(\tau, t)d\tau$$
$$= <f(\tau), K(\tau, t)> \tag{30}$$

which achieves our assertion. ∎

From the proof of theorem 7, the following corollary holds:

Corollary 1: W_2^m equipped with the inner product (15) can be decomposed as a direct sum of two subspaces as follows:

$$W_2^m[a, b] = \ker \mathcal{L} \oplus \ker \Lambda \triangleq \mathcal{H}_1 + \mathcal{H}_2 \tag{31}$$

where the subspaces $\mathcal{H}_1$ and $\mathcal{H}_2$ are RKHSs equipped with the following inner products:

$$<u, v>_{\mathcal{H}_1} = \sum_{i=1}^{m}(\lambda_i u)(\lambda_i v) \tag{32}$$

$$<u, v>_{\mathcal{H}_2} = \int_a^b \mathcal{L}u(t) \cdot \mathcal{L}v(t)dt \tag{33}$$

respectively, and the corresponding reproducing kernels are

$$K_1(t, s) = \sum_{i=1}^{m}\varphi_i(t)\varphi_i(s) \tag{34}$$

$$K_2(t, s) = \int_a^b G(t, \tau) \cdot G(s, \tau)d\tau \tag{35}$$

Recall (16), the computation of kernel consists of two parts, i.e. (i) computing the dual basis φ_i of λ_i for $\forall i = 1, ..., m$, and (ii) computing the Green function $G(t, s)$ and the integral (35). It's obvious that the computational cost is mainly in (ii) which is associated with m-order differential operator $\mathcal{L}$. Therefore, a basic green function was introduced [46], i.e.

$$g(t, s) = \sum_{i=1}^{m} u_i(t)\tilde{u}_i(s) \tag{36}$$

where $\{u_i\}_{i=1}^m$ is any set of basis of $\ker \mathcal{L}$ and $\{\tilde{u}_i\}_{i=1}^m$ are the corresponding adjoint functions, and then

$$G(t, s) = g(t, s) - \sum_{i=1}^{m} \lambda_i g(\cdot, s)\varphi_i(t) \tag{37}$$

which satisfies (27).

2) For differential operator with m different eigenvalues: Recall (16), it is computed for any m-order differential operator $\mathcal{L}$. When the following linear differential operator

$$\mathcal{L} = (D - r_1)(D - r_2) \cdots (D - r_m) \tag{38}$$

is chosen, where $r_i \neq r_j$, $(i \neq j)$ are the m different eigenvalues, the following theorem holds

Theorem 8: The RKHS $\ker \Lambda$ in (31) which is equipped with inner product (35) possesses a R.K as follows:

$$K_2(t, s) = b_m^T Q_m(t, s) b_m \tag{39}$$

where $b_m = (b_{m1}, ..., b_{mm})^T$ is the last column of M^{-1}, i.e.,

$$b_{mi} = (-1)^{m+i} \prod_{1 \leq k \leq i-1} \frac{1}{(r_i - r_k)} \prod_{i+1 \leq k \leq m} \frac{1}{(r_k - r_i)} \tag{40}$$

M denotes a Vandermonde matrix, i.e.

$$M_i = (1, r_i, ..., r_i^{m-1})^T \qquad (41)$$

where $r_i \neq r_j$, $(i \neq j)$ is the eigenvalue of $\mathcal{L}$, $Q_m(t,s) = (q_{ij}(t,s))_{m \times m}$ is as follows:

$$q_{ij}(t,s) = \begin{cases} \frac{e^{r_i(t-a)+r_j(s-a)}-e^{r_{\theta(i,j)}|t-s|}}{r_i+r_j}, r_i + r_j \neq 0 \\ \frac{e^{r_i(t-s)}(t+s-|t-s|-2a)}{2} \quad , otherwise \end{cases} \qquad (42)$$

where $\theta(i,j) = \begin{cases} j-1, t \leq s \\ i-1, otherwise \end{cases}$

Proof: let $\lambda_i^0 (1 \leq i \leq m)$ be a functional with an initial-value constrains, i.e. $\lambda_i^0 f = f^{(i-1)}(a)$, $W(u_1, ..., u_m)(t)$ be a Wronskian matrix as follows

$$W(u_1, ..., u_m)(t) = M \cdot diag(u_1(t), ..., u_m(t)) \qquad (43)$$

where $u_i(t) = \exp(r_i t)$ is the bases of $\ker \mathcal{L}$. Let $\nu_{ij} = M^{-1}(i,j)$. From the result in [47], we have

$$\nu_{ij} = (-1)^{i+j} \prod_{1 \leq k \leq i-1} \frac{1}{(r_i-r_k)} \prod_{i+1 \leq k \leq m} \frac{\sigma_{m-j}(i)}{(r_k-r_i)} \qquad (44)$$

where $\sigma_k(i) = \sum_{\substack{1 \leq j_1 \leq ... \leq j_k \leq m, \\ j_d \neq i, d=1, ..., k}} r_{j_1} r_{j_2} \cdots r_{j_k}$

Recall (18), the dual bases to $\{\lambda_i^0\}_1^m$ are as follows:

$$\varphi_i(t) = \sum_{k=1}^m \nu_{ki} \exp(r_k(t-a)) \qquad (45)$$

then

$$\begin{aligned} \tilde{u}_i(t) &= \nu_{im} e^{-r_i t} = b_{mi} \qquad (46) \\ &= (-1)^{m+i} \prod_{1 \leq k \leq i-1} \frac{1}{(r_i-r_k)} \prod_{i+1 \leq k \leq m} \frac{e^{-r_i t}}{(r_k-r_i)} \end{aligned}$$

Recall (36) and (37), the basic Green function can be obtained:

$$g(t,s) = \sum_{i=1}^m \nu_{im} \exp(r_i(t-s)) \cdot (t-s)_+^0 \qquad (47)$$

and substituting (44)-(47) to (37) and (35), will achieve our assertion. The verification will not be discussed further due to the limited space, for more details see e.g. [45]. ∎

3) For simpler differential operator: It's obvious that it is superior to the method using differential equation since the R.K can be obtained a uniform and concise form (16). It's found that the complexity is mainly concentrated on the computation of Green function which depends on the differential operator $\mathcal{L}$. In addition, if, according to Definition 4, the Sobolev spaces admit the following natural norm:

$$\|f\|_p^m = \sum_{i=0}^m \|f^{(i)}\|_p = \sum_{i=0}^m \left(\int |f^{(i)}(t)|^p dt \right)^{\frac{1}{p}} \qquad (48)$$

Then W_p^m equipped with the norm (48) is a Banach space. Note that it is enough to take only the first and last elements, i.e. the norm defined by $\|f^{(m)}\|_p + \|f\|_p$, is equivalent to the norm in (48). The fact described above motivates us to simplify the differential operator in Theorem 8 as a simpler one:

$$\mathcal{L} = D^m - 1 \qquad (49)$$

From Theorem 8, the following theorem holds, since the differential operator in (49) possesses m different eigenvalues.

Theorem 9: W_2^m equipped with the following inner product:

$$< u, v > = \sum_{i=1}^m u^{(i-1)}(a) v^{(i-1)}(a) + \int_a^b (u^{(m)} - u)(v^{(m)} - v) dt \qquad (50)$$

possesses a R.K as follows:

$$K(t,s) = \sum_{i=1}^m \varphi_i(t) \varphi_i(s) + V_m^T Q_m(t,s) V_m \qquad (51)$$

where $V_m = (\nu_{1m}, ..., \nu_{mm})^T = M^{-1}(i,m)$, and $Q_m(t,s)$ is a $m \times m$ matrix, whose element $q_{ij}(t,s)$ is as follows

$$q_{ij}(t,s) = \begin{cases} \frac{e^{r_{i-1}(t-a)+r_{j-1}(s-a)}-e^{r_{\theta(i,j)}|t-s|}}{r_{i-1}+r_{j-1}}, r_{i-1}+r_{j-1} \neq 0 \\ \frac{e^{r_{i-1}(t-s)}(t+s-|t-s|-2a)}{2} \quad , otherwise \end{cases} \qquad (52)$$

where $\theta(i,j)$ is the same as in Theorem 8.

Proof: It's found that the eigenvalues of (49) are

$$r_k = \exp(2k\pi i/m), k = 0, ..., m-1 \qquad (53)$$

Note that for $\forall k, j = 1, ..., m$,

$$r_k \cdot r_j = r_{k+j}, r_{-k} = r_k^{-1}, r_{m+k} = r_k, r_1^{m+k} = r_1^k \qquad (54)$$

It's obvious if substituting (53) and (54) into Theorem 7 and 8, which completes the proof. ∎

Additionally, two cases, according to the parity of m, should be considered in Theorem 9, there are the following corollaries:

Corollary 2: W_2^m, $(m = 2m_0)$ equipped with an inner product as follows:

$$\begin{aligned} < u, v > &= \sum_{i=1}^m u^{(i-1)}(a) v^{(i-1)}(a) \qquad (55) \\ &+ \sum_{i=1}^{m_0} (-1)^i \left[u^{(i-1)} v^{(m-i)} + v^{(i-1)} u^{(m-i)} \right]_a^b \\ &+ \int_a^b \left[u^{(m)} v^{(m)} - (-1)^{m_0} 2 u^{(m_0)} v^{(m_0)} + uv \right] dt \end{aligned}$$

possesses a R.K with form (51), where $Q_m(t,s)$ is as follows:

$$q_{ij}(t,s) = \begin{cases} \frac{e^{t-s}(t+s-|t-s|-2a)}{2} & , i=1; j=m_0+1 \\ \frac{e^{s-t}(t+s-|t-s|-2a)}{2} & , i=m_0+1; j=1 \\ \frac{e^{r_{i-1}(t-a)+r_{j-1}(s-a)}-e^{r_{\theta(i,j)}|t-s|}}{r_{i-1}+r_{j-1}}, otherwise \end{cases} \qquad (56)$$

Proof: Firstly, recall (50), then

$$\int_a^b (u^{(m)} - u)(v^{(m)} - v) dt = \int_a^b (u^{(m)} v^{(m)} - uv^{(m)} - u^{(m)} v + uv) dt \qquad (57)$$

and

$$\int_a^b uv^{(m)} + u^{(m)}v \, dt =$$

$$\sum_{i=1}^{m_0}(-1)^i \left[u^{(i-1)}v^{(m-i)} + v^{(i-1)}u^{(m-i)} \right]_a^b$$

$$+(-1)^{m_0}\int_a^b 2u^{(m_0)}v^{(m_0)}dt \qquad (58)$$

Then, substituting (57) and (58) into (50), it's found that the inner product in (55) is equivalent to (50).

Secondly, recall (53), then $r_{m_0} = \exp(i\pi) = -1$, i.e. $r_0 + r_{m_0} = 0$ since $m = 2m_0$. Then $r_{i-1} + r_{j-1} = 0$, $(1 \leq i \leq j \leq m)$ if and only if $i = 1; j = m_0 + 1$ or $i = m_0 + 1; j = 1$, and substitute into (52) can easily achieve the assertion. ∎

Corollary 3: W_2^m, $(m = 2m_0 + 1)$ equipped with an inner product as follows:

$$< u, v > = \sum_{i=1}^m u^{(i-1)}(a)v^{(i-1)}(a) \qquad (59)$$

$$+\sum_{i=1}^m (-1)^i \left[u^{(i-1)}v^{(m-i)} \right]_a^b + \int_a^b \left[u^{(m)}v^{(m)} + uv \right]dt$$

possesses a R.K with form (51), where $Q_m(t,s)$ is as follows:

$$q_{ij}(t,s) = \frac{e^{r_{i-1}(t-a)+r_{j-1}(s-a)} - e^{r_{\theta(i,j)}|t-s|}}{r_{i-1}+r_{j-1}} \qquad (60)$$

where $1 \leq i \leq j \leq m$

Proof: Similar to Corollary 2, the inner product (59) is equivalent to (50) when m is an odd number.

According to (53), there are no $1 \leq i \leq j \leq m$ such that $r_{i-1} + r_{j-1} = 0$ since m is odd number, thus (60) holds. ∎

B. An Algorithm for Computation Reproducing Kernel

In this section, the algorithm is proposed based on the theorems and corollaries in previous section. Fig.1 shows a brief flow chart of the algorithm for computing the R.K in $W_2^m[a,b]$. The algorithm can be summarized as follows:
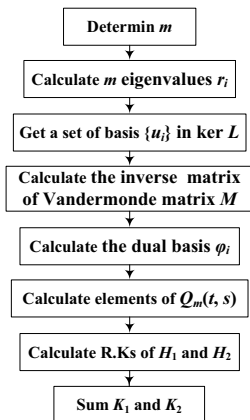

Fig. 1. Flow chart of the algorithm for the computation of reproducing kernel in $W_2^m[a,b]$

Step 1: Determine a positive integer m;

Step 2: Calculate the eigenvalues r_i e.g. in (38) or (53);

Step 3: Give any set of basis or calculate the corresponding set of basis of $\mathcal{H}_1 = \ker \mathcal{L}$ associated with r_i, e.g. in (43);

Step 4: Calculate the inverse matrix of Vandermonde matrix M, i.e. calculate all the elements ν_{ij}, e.g. in (44);

Step 5: Calculate the dual bases $\{\varphi_i\}_1^m$, e.g. in (45);

Step 6: Calculate $q_{ij}(t,s)$, e.g. in (42) or (52);

Step 7: Compute the reproducing kernel K_1 of $\mathcal{H}_1 = \ker \mathcal{L}$ according to Step 5 and (34)

Step 8: Compute the reproducing kernel K_2 of $\mathcal{H}_2 = \ker \Lambda$ according to Step 6 and (35) or Theorem 8, 9 and its corollaries

Step 9: Sum K_1 and K_2

Note that users are only asked to set an initial value of m to start the algorithm. Therefore, it will benefit an automatic, flexible and rigorous computation with computer codes, e.g. Matlab.

V. NUMERICAL EXAMPLES AND SIMULATION ANALYSIS

A. Numerical Examples

Example 1 (Even) Let $m = 2$, i.e. $m_0 = 1$, then $\mathcal{L} = D^2 - 1$, the eigenvalues are $r_0 = 1$, $r_1 = -1$, and the bases of $\ker \mathcal{L}$ are $u_1(t) = e^t$, $u_2(t) = e^{-t}$. According to (44), then $\nu_{11} = \nu_{12} = \nu_{21} = 1/2$, $\nu_{22} = -1/2$, then dual bases are $\varphi_1(t) = (e^{t-a} - e^{-t+a})/2$, $\varphi_2(t) = (e^{t-a} + e^{-t+a})/2$. Recall (55), it's obtained that

$$
\begin{aligned}
q_{11}(t,s) &= \left(\exp(t+s-2a) - \exp(|t-s|)\right)/2 \\
q_{12}(t,s) &= \left(\exp(t-s)[t+s-|t-s|-2a]\right)/2 \\
q_{21}(t,s) &= \left(\exp(s-t)[t+s-|t-s|-2a]\right)/2 \\
q_{22}(t,s) &= -\left(\exp(-t-s+2a) - \exp(-|t-s|)\right)/2
\end{aligned}
$$

Substituting the results calculated above into (51), and considering the equivalent inner product (55), thus

$$K(t,s) = \varphi_1(t)\varphi_1(s) + \varphi_2(t)\varphi_2(s) + \nu_{12}^2 q_{11}$$

$$+\nu_{12}\nu_{22}(q_{21} + q_{12}) + \nu_{22}^2 q_{22} = \frac{5e^{t+s-2a}}{8} + \frac{3e^{-(t+s-2a)}}{8}$$

$$-\frac{(t+s-|t-s|-2a)\cosh(t-s)}{4} - \frac{\sinh|t-s|}{4}$$

Example 2 (Odd) Let $m = 1$, then $\mathcal{L} = D-1$, the eigenvalue is $r_0 = 1$, $u_1(t) = e^t$ is a basis of $\ker \mathcal{L}$, and $\nu_{11} = r_0^{-1} = 1$, dual basis $\varphi_1(t) = e^{t-a}$, $q_{11}(t,s) = \frac{e^{t+s-2a} - e^{|t-s|}}{2}$. Finally, the R.K is

$$K(t,s) = \frac{3}{2}\left(\exp(t+s-2a) - \exp(|t-s|)\right) \qquad (61)$$

and the corresponding inner product is

$$< u, v > = 2u(a)v(a) - u(b)v(b) + \int_a^b [u'v' + uv]dt \qquad (62)$$

B. Simulation and Results

In order to validate that the R.K computed by the algorithm can be used as SV kernels well, a comparison between one of the kernels computed, i.e. R.K in (61), and classical RBF is presented. Consider a simple illustrative regression problem

involving one input and one output in which the target function $y(x)$ as follows is known for a target with additive noise.

$$y(x) = (1 - x + 2x^2) \exp(x^2/2) \qquad (63)$$

Take a data set of $N = 50$ training points in which the input data point x is picked uniformly from the interval $[-4, 4]$, and the target t is generated by an additive noise process $t_i = y(x_i) + \xi_i$, where ξ_i is additive noise as follows:

$$\xi_i = 0.2 \cdot \|y\|_\infty \cdot (2\varepsilon_i - 1)/2 \qquad (64)$$

where $\|y\|_\infty = \max\{y(x_1), ..., y(x_N)\}$, ε_i is uniform variable. For the experiment, a RBF $K(x, x') = e^{\left(-\|x - x'\|/(2\sigma^2)\right)}$ with width $\sigma = 0.6$, which is selected by cross validation, was used. In addition, three metrics was introduced for comparing the performance of SV kernels, i.e. (i) R square (R^2) as follows

$$R^2 = 1 - \frac{\sum_{i=1}^l (y_i - \hat{y}_i)^2}{\sum_{i=1}^l (y_i - \bar{y})^2} \qquad (65)$$

where $\hat{y}_i$ denotes the corresponding predicted value for observed value y_i; $\bar{y}$ denotes the mean of the observed values.

Generally speaking, (i) the larger the R^2, the more accurate the SVR; (ii) Training time implies the computational efficiency. The smaller the training time, the more efficient the SVR; (iii) Amount of support vectors shows the generalization ability. The smaller the amount of support vectors, the better the generalization ability.

The regression results (where the parameters in SVR, i.e. regularization constant $C = 100$ and tolerance error $\varepsilon = 0.15$) of RBF and R.K are illustrated in Fig. 2 and Fig. 3 respectively, and the comparison of R square, training time and amount of support vectors (SVs) are shown in Table I.
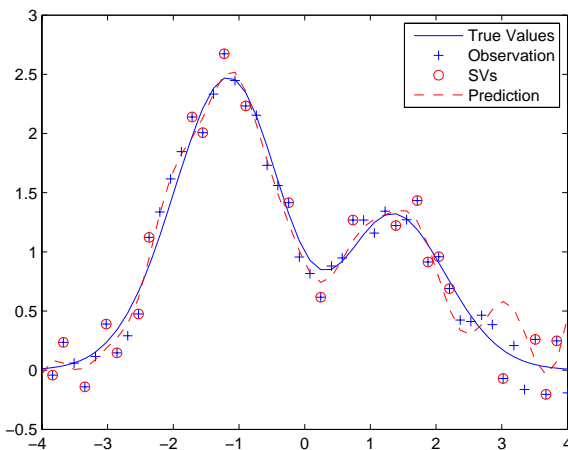

Fig. 2.   Regression results of RBF

It's noted that, the accuracy and training time of R.K is slightly superior to that of RBF, whereas the generalization ability is inferior. Overall, the performance of R.K is almost similar to that of RBF, which implies that the R.K computed by the algorithm can be used as SV kernel well. Consequently, it illuminates that the algorithm can be used to provide more alternatives of SV kernel for SVMs.
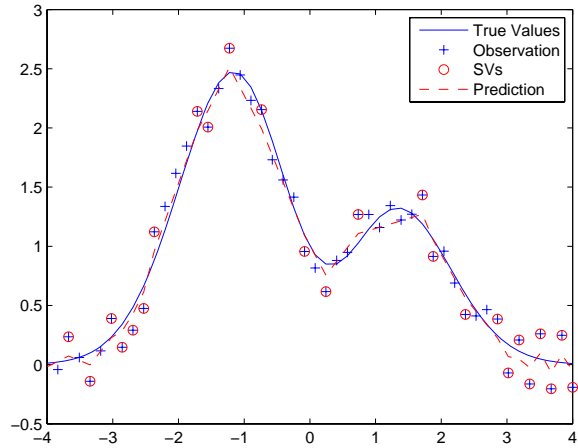

Fig. 3.   Regression results of R.K

TABLE I
COMPARISON OF R^2, TRAINING TIME AND AMOUNT OF SVs

	R^2	Training time	Amount of SVs
RBF	0.940292595217094	1.124502136444716	**24**
R.K	**0.975836312367347**	**1.043337910265132**	26
Best	**R.K**	**R.K**	**RBF**

VI.   Conclusions

This paper proposes an algorithm for computing the R.K in a general RKHS. Some concepts and relations analysis will help to elucidate that the proposed algorithm can be considered to be an approach for constructing a SV kernel. Meanwhile, the advantages of the algorithm, i.e. it avoids the complex and demanding task on constructing and solving the differential equation with boundary value problems in conventional method, and obtains a unified explicit expression of R.K for further analysis, are presented. In a word, it benefits an automatic, flexible and rigorous computation. Some numerical examples and experimental results are shown to illustrate the validity and effectiveness of the proposed method. However, the construction of the SV kernel function and the way to choose an appropriate kernel in a specific application are far from full-understood. As we know, the power of machine learning, especially SVMs, lies in the selection of kernel and its parameters which have to depend on the wit of the user and his "expert" understanding of the processing data. Therefore, the issue on choosing the right kernel and the optimal parameters either in kernel itself or in machine learning algorithms would be our future work.

REFERENCES

[1] J. Mercer, Ed., *Functions of Positive and Negative Type and Their Connection with The Theory of Integral Equations*, ser. Philosophical Transactions of the Royal Society, London, 1909, vol. A, 209.

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

[3] G. Bloch, F. Lauer, G. Colin, and Y. Chamaillard, "Support vector regression from simulation data and few experimental samples," *Information Sciences*, vol. 178, pp. 3813–3827, 2008.

[4] V. Blanz, B. Schölkopf, H. Bülthoff, and C. J. Burges, "Comparison of view-based object recognition algorithms using realistic 3d models," in *Artificial Neural NetworksICANN'96*, C. v. d. Malsburg, W. v. Seelen, J. C. Vorbrüggen, and B. Sendhoff, Eds., vol. 1112. Berlin: Springer Lecture Notes in Computer Science, 1996.

[5] C. J. Burges and B. Schölkopf, "Improving the accuracy and speed of support vector learning machines," in *Advances in Neural Information Processing Systems*, vol. 9. Cambridge, MA: MIT Press, 1997, pp. 375–381.

[6] B. Schölkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with gaussian kernels to radial basis function classifiers," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2758–2765, 1997.

[7] E. E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130–136.

[8] W. Kienzle, G. Bakir, M. Franz, and B. Schölkopf, "Face detectionefficient and rank deficient," in *Advances in Neural Information Processing Systems*, Y. Weiss, Ed., vol. 17. MIT Press, 2005, pp. 673–680.

[9] J.-B. Gao, S. R. Gunn, and C. J. Harris, "Mean field method for the support vector machine regression," *Neurocomputing*, vol. 50, pp. 391–405, 2003.

[10] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, "Support vector machines for wind speed prediction," *Renewable Energy*, vol. 29, no. 6, pp. 939–947, 2004.

[11] W.-W. He, Z.-Z. Wang, and H. Jiang, "Model optimizing and feature selecting for support vector regression in time series forecasting," *Neurocomputing*, vol. 73, no. 3, pp. 600–611, 2008.

[12] F. Pan, P. Zhu, and Y. Zhang, "Metamodel-based lightweight design of b-pillar structure via support vector regression," *Computers and Structures*, vol. 88, pp. 36–44, 2010.

[13] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[14] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[15] B. E. Boser, I. M. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed. ACM Press, 1992, pp. 144–152.

[16] B. Schölkopf, "The kernel trick for distances," *Neural Information Process. Systems (NIPS)*, vol. 13, 2000.

[17] D. Anguita and G. Bozza, "The effect of quantization on support vector machines with gaussian kernel," in *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, 2005, pp. 681–684.

[18] X.-Y. Zhang and Y.-C. Liu, "Performance analysis of support vector machines with gauss kernel," *Computer Engineering*, vol. 29, no. 8, pp. 22–25, 2003.

[19] Y. Tan and J. Wang, "A support vector machine with a hybrid kernel and minimal vapnik-chervonenkis dimension," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 385–395, 2004.

[20] J.-X. Liu, J. Li, and Y.-J. Tan, "An empirical assessment on the robustness of support vector regression with different kernels," in *Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, vol. 7, Guangzhou, China, 2005.

[21] R. Opfer, "Multiscale kernels," *Advances in Computational Mathematics*, vol. 25, pp. 357–380, 2006.

[22] L. Zhang, W.-D. Zhou, and L.-C. Jiao, "Wavelet support vector machine," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 34, pp. 34–39, 2004.

[23] X.-G. Zhang, D. Gao, X.-G. Zhang, and S.-J. Ren, "Robust wavelat support machines for regression estimation," *International Journal Information Technology*, vol. 11, no. 9, pp. 35–46, 2005.

[24] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, pp. 637–649, 1998.

[25] R. Schaback and H. Wendland, "Approximation by positive definite kernels," in *Advanced Problems in Constructive Approximation*, M. D. Buhmann and D. H. Mache, Eds. Birkhäuser, Basel: Verlag, 2002, pp. 203–221.

[26] G. Wahba, "Support vector machines, reproducing kernel hilbert spaces and randomized gacv," in *Advances in Kernel Methods-Support Vector Learning*, B. Schölkopf, C. J. Burges, and A. J. Smola, Eds. Cambridge, England: MIT Press, 1999, pp. 69–88.

[27] L.-M. Ma and Z.-M. Wu, "Kernel based approximation in sobolev spaces with radial basis functions," *Applied Mathematics and Computation*, vol. 215, p. 2229C2237, 2009.

[28] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[29] J. Gao, C. J. Harris, and S. R. Gunn, "Support vector kernel based on frames in function hilbert spaces," *Neural Computation*, vol. 13, pp. 1975–1994, 2001.

[30] R. Schaback, "A unified theory of radial basis functions native hilbert spaces for radial basis functions ii," *Journal of Computational and Applied Mathematics*, vol. 121, pp. 165–177, 2000.

[31] S. Bergman, "The approximation of functions satisfying a linear partial differential equation," *Duke Mathematics Journal*, vol. 6, pp. 537–561, 1940.

[32] M.-G. Cui and Z.-X. Deng, "On the best operator of interpolation," *Math. Numerica Sinica*, vol. 8, no. 2, pp. 209–216, 1986.

[33] J. Ling and Y.-S. Li, "A new method for computing reproducing kernels," *Northeast Math. J.*, vol. 14, no. 4, pp. 467–473, 1998.

[34] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[35] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminate analysis with kernels," in *Neural Networks for Signal Processing IX*, Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. IEEE Press, 1999, pp. 41–48.

[36] D. MacDonald and C. Fyfe, "The kernel self-organizing map," in *Proceedings of 4th International Conference on Knowledge-based Intelligent Engineering Systems and Allied Technologies, KES 2000*, R. J. Howlett and L. C. Jain, Eds., vol. 1, 2000, pp. 317–320.

[37] A. J. Smola, B. Schölkopf, and K.-R. Müller, "General cost functions for support vector regression," in *Proceedings of Ninth Australian Conf. on Neural Networks*, 1998, pp. 79–83.

[38] S. Bochner, *Lectures on Fourier Integral*. Princeton, New Jersey: Princeton University Press, 1959.

[39] A. J. Smola, Z. L. Óvári, and R. C. Williamson, "Regularization with dot-product kernels," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2001, pp. 308–314.

[40] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Boston, Dordrecht, London: Kluwer Academic Publishers Group, 2003.

[41] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge, U.K: Cambridge University Press, 2000.

[42] M.-G. Cui and Y.-Z. Lin, *Nonlinear Numerical Analysis in Reproducing Kernel Space*. Commack, NY: Nova Science Publishers, Inc., 2009.

[43] M.-G. Cui, M. Zhang, and Z.-X. Deng, "Two-dimensional reproducing kernel and surface interpolation," *Journal Computational Mathematic*, vol. 4, no. 2, pp. 177–181, 1986.

[44] W. Zhang, "The construction of reproducing kernel and some approximating problems in the reproducing kernel spaces," Ph.D. dissertation, National University of Defense Technology, Changsha, Hunan, China, 2005.

[45] X.-J. Zhang and H. Long, "Computing reproducing kernels for $w_2^m[a,b]$ (i)," *Mathematic Numerica Sinica*, vol. 30, no. 3, pp. 295–304, 2008.

[46] Y.-S. Li, "On the recurrence relations for b-splines defined by certain l-splines," *Journal of Approximation theory*, vol. 43, pp. 359–369, 1985.

[47] V. E. Neagoe, "Inversion of the van der monde matrix," *IEEE Signal Processing Letters*, vol. 3, no. 4, pp. 119–120, 1996.