Applied Mathematics and Computation 245 (2014) 509–519

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Support vector machine adapted Tikhonov regularization method to solve Dirichlet problem

Yan Mo, Tao Qian *

Department of Mathematics, Faculty of Science and Technology, University of Macau, Taipa, Macau, China

ARTICLE INFO

Keywords:
Dirichlet problem
Support vector machine
Tikhonov regularization
Reproducing kernel
Sobolev space
Gaussian RKHS

ABSTRACT

Numerical solutions of partial differential equations are traditional topics that have been studied by many researchers. During the last decade, support vector machine (SVM) has been widely used for approximation problems. The contribution of this paper is two folds. One is to combine the reproducing kernel-SVM method with the Tikhonov regularization method, called the SVM-Tik methods, in which the kernels K_λ and K_λ^σ (see below) are newly developed. In the paper they are respectively phrased as the SVM-Tik-K_λ and SVM-Tik-K_λ^σ methods. The second contribution is to use the two models, SVM-Tik-K_λ and SVM-Tik-K_λ^σ, to solve the Dirichlet problem. The methods are meshless. They produce sparse representations in the linear combination form of specific functions (the K_λ and K_λ^σ kernels). The generalization bound result in learning theory is used to give an estimation of the approximation errors. With the illustrative examples the sparseness and robustness properties, as well as the effectiveness of the methods are presented. The proposed methods are compared with currently the most commonly used finite difference method (FDM) showing promising results.

1. Introduction

Many methods have been developed so far for solving partial differential equations. Some methods such as the finite difference method (FDM) [15] and the finite element method (FEM) [3] require the definition of a mesh (domain discretization) where the functions are approximated locally. The construction of a mesh in two or more dimensions is a non-trivial problem. A major disadvantage of those methods, however, is their mesh-dependent characteristics which normally require enormous computational effort and induce numerical instability when large number of grids or elements are required.

Another approach for solving partial differential equations is to use artificial neural networks (ANNs) [4–6]. The approach using ANNs to solve partial differential equations relies on the functional approximation capability of feedforward neural networks and results in construction of a solution written in a differentiable and closed analytic form. This form employs feedforward neural network as the basic approximation element, whose parameters (weights and biases) are adjusted to minimize an appropriate error function. The solution in terms of artificial neural networks posses several attractive features. One of the features is that the solution is infinitely differentiable and closed analytic form which can be easily used in any subsequent calculation. Another is that they posses smaller number of parameters compared to other solution technique

* Corresponding author.
E-mail addresses: marvel-2008@163.com (Y. Mo), fsttq@umac.mo (T. Qian).

http://dx.doi.org/10.1016/j.amc.2014.07.089
0096-3003/© 2014 Elsevier Inc. All rights reserved.

[4,5]. However, ANNs suffer from their theoretical weakness. For example, back-propagation may not converge to a optimal global solution.

SVM, developed by Vapnik and his coworkers in 1995 [16], is based on statistical learning theory which seeks to minimize an upper bound of the generalization error consisting of the sum of the training error and a confidence interval. This principle is different from the commonly used empirical risk minimization (ERM) principle which only minimizes the training error. Based on this, SVMs usually achieve higher generalization performance than ANNs which implement ERM principle. As consequence, SVMs can be used wherever that ANNs can, and usually achieve better results. Another key characteristic of SVM is that training SVM is equivalent to solving a linearly constrained quadratic programming problem so that the solution of SVM is unique and global, unlike ANNs' training which requires nonlinear optimization with the possibility of getting stuck into local minima.

In this paper, we combine SVM with the Tikhonov regularization method which is called the SVM-Tik methods to solve Dirichlet problem numerically. Two kernels K_λ and K_λ^σ which are newly developed will be used in our algorithms, the corresponding algorithms are phrased as SVM-Tik-K_λ and SVM-Tik-K_λ^σ algorithms. The solutions are sparse representations in the linear combination form of specific functions (the K_λ and K_λ^σ kernels).

Experiments are done for testing the proposed approach. It shows good performance in noise-free data case and in Gaussian noise-corrupted data case. The comparisons are between three types of algorithms in Example 1. (1) The proposed SVM-Tik with K_λ kernel algorithm (SVM-Tik-K_λ). (2) The proposed SVM-Tik with K_λ^σ kernel algorithm (SVM-Tik-K_λ^σ). (3) The FDM. Although the FDM can achieve better performance than the two SVM based algorithms for noise-free data, the two SVM based algorithms behave better than the FDM in presence of Gaussian noise. With the two SVM based algorithms, sparse representations in linear combination form of specific functions (the K_λ and K_λ^σ kernel) are obtained. On the other hand, in the FDM case, the solution is not expressed in any closed analytical form as in our case, additional interpolation computations are required in order to find the value of the solution at particular points in the domain. As for comparison between the two SVM based algorithms, SVM-Tik-K_λ performs better than SVM-Tik-K_λ^σ.

2. Preliminary

We shall propose a new approach for constructing approximate solutions for the Dirichlet problem

$$\begin{cases} \Delta u = 0, & \text{in } D, \\ u = g, & \text{on } \partial D, \end{cases} \tag{2.1}$$

on an appropriate domain D in $\mathbf{R}^n$ with boundary ∂D, where $\mathbf{R}^n$ is the Euclidean space.

In our case, we shall first consider this problem in the Sobolev space $H^s(\mathbf{R}^n)$ with $n \geqslant 1,\ s \geqslant 2,\ s > n/2$ from the viewpoint of numerical analysis. In the sequel we abbreviate $H^s(\mathbf{R}^n)$ as H^s. The Sobolev Hilbert space H^s comprises functions F on $\mathbf{R}^n$ with the norm

$$\|F\|_{H^s}^2 = \int_{\mathbf{R}^n} |\hat{F}(\xi)|^2 (1 + |\xi|^2)^s d\xi,$$

which admits a reproducing kernel

$$K_s(x, y) = \frac{1}{(2\pi)^n} \int_{\mathbf{R}^n} \frac{1}{(1 + |\xi|^2)^s} e^{i(x-y)\cdot\xi} d\xi, \tag{2.2}$$

where $\hat{F}$ is the Fourier transform of F,

$$\hat{F}(\xi) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbf{R}^n} e^{-i\xi\cdot x} F(x) dx$$

(See [7]).

The Dirichlet principle asserts that the solution of the Eq. (2.1) is the extremal function minimizing the Dirichlet integral under the boundary condition $F(x) = g(x)$ on ∂D.

We want to obtain some good representations of the extremal functions when they exist.

In [7] by applying the theory of Tikhonov regularization and the reproducing kernel based methods of Saitoh et al. in [2,8,10–13] the authors first formulated the problem as follows:

For a fixed $\lambda > 0$ and a given $g \in L_2(\partial D)$ find a solution for

$$\inf_{F \in H^s} \left\{ \lambda \|F\|_{H^s}^2 + \|\Delta F\|_{L_2(\mathbf{R}^n)}^2 + \|F - g\|_{L_2(\partial D)}^2 \right\}. \tag{2.3}$$

The authors are considering the approximation by the Sobolev functions over the whole space.

The strategy is to first represent for each $\lambda > 0$ the extremal function $F = F_{s,\lambda,g}^*(x)$ in (2.3); and, secondly, to obtain the solution u of the problem (2.1) by taking the limit $\lambda \to 0$ on the corresponding extremal functions [7]. The theory of Tikhonov regularization guarantees that such limit exists.

Theorem 2.1 [10]. *Let H_λ be a Hilbert space admitting the reproducing kernel $K(p,q)$ on a set E. Let $L : H_K \to H$ be a bounded linear operator on H_K into H. For $\lambda > 0$ introduce the inner product in H_K and call it H_{K_λ} as*

$$\langle f_1, f_2\rangle_{H_{K_\lambda}} = \lambda\langle f_1, f_2\rangle_{H_K} + \langle Lf_1, Lf_2\rangle_H,$$

then H_{K_λ} is a Hilbert space with the reproducing kernel $K_\lambda(p,q)$ on E satisfying the equation

$$K(\cdot,q) = (\lambda I + L^*L)K_\lambda(\cdot,q),$$

where L^* is the adjoint of $L : H_K \to H$.

According to Theorem 2.1, the reproducing kernel Hilbert space (RKHS) H_{K_λ} that we will be working on has the norm

$$\left(\lambda\|F\|^2_{H^s} + \|\Delta F\|^2_{L_2(\mathbf{R}^n)}\right)^{\frac{1}{2}}, \tag{2.4}$$

that is a modification of the classical Sobolev space. The reproducing kernel of the space is

$$K_\lambda(x,y) = \frac{1}{(2\pi)^n}\int_{\mathbf{R}^n} \frac{e^{ip\cdot(x-y)}}{\lambda(|p|^2+1)^s + |p|^4}\,dp, \tag{2.5}$$

([8]).

The extremal function of (2.3) is represented by g as

$$F^*_{s,\lambda,g}(x) = (g, LK_{s,\lambda,\Delta}(\cdot,x))_{L_2(\partial D)}, \tag{2.6}$$

(see, [2,8,7]), where $K_{s,\lambda,\Delta}(\cdot,x)$ is determined through the relation

$$K_\lambda(x,y) = (\lambda I + L^*L)K_{s,\lambda,\Delta}(x,y). \tag{2.7}$$

However, as stated in [7], the Eq. (2.7) cannot be solved effectively. The first problem is that we are unable to apply the Neumann expansion to the equation due to the requirement $\|L^*L\| < \lambda$. If $\lambda \to 0$, then this requirement cannot be met. The second reason is that when the operator L is compact, in order to solve the Eq. (2.7) by applying the spectral theory, one has to look for singular values and singular functions of the operator L^*L. But, for a general domain D, the singular values and singular functions are abstract in a sense. Therefore, in [7], the authors propose a new approach. They modify (2.3) as follows:

For any fixed points $\{x_j\}_{j=1}^N$ of the boundary ∂D, any given values $\{A_j\}_{j=1}^N$ and any fixed $\{\lambda_j\}_{j=1}^N, \lambda_j > 0$, find

$$\inf_{F\in H^s}\left\{\lambda\|F\|^2_{H^s} + \|\Delta F\|^2_{L_2(\mathbf{R}^n)} + \sum_{j=1}^N \lambda_j|F(x_j) - A_j|^2\right\}. \tag{2.8}$$

Then the problem (2.8) is solved through combining the generalized inverses, Tikhonov's regularization and the theory of reproducing kernels. Their algorithm requires to compute N reproducing kernels through iterations. The effectiveness of their algorithm is heavily dependent of the power of the computer in use.

The contribution of this work is to adapt the Tikhonov regularization principle into the SVM formulation, then we solve the newly formulated problem in reproducing kernel spaces following the recent work of Saitoh.

The algorithm used in [7] can fit into a more general frame work. Consider the following functional in learning theory: For any fixed $C_j > 0$, find

$$\min_{F\in H^s}\left\{\lambda\|F\|^2_{H^s} + \|\Delta F\|^2_{L_2(\mathbf{R}^n)} + \sum_{j=1}^N C_j V(F(x_j), A_j)\right\}, \tag{2.9}$$

for F in H^s, where $V(\cdot,\cdot)$ is a so-called *loss function*. The functional (2.8) can be regarded as the particular case $V(F(x_j),A_j) = |F(x_j) - A_j|^2$. Our philosophy and treatment are suggested by learning theory. We, in particular, take

$$V(F(x_j),A_j) = |F(x_j) - A_j|_\varepsilon = \max\{|F(x_j) - A_j| - \varepsilon, 0\} \tag{2.10}$$

which is called *Vapnik's ε-insensitive loss function* [16]. Such loss functions are often adopted in learning theory and solved via SVMs. Different loss functions can derive quite different types of solutions. The ε-insensitive loss function can provide sparse solutions and deal with noise-corrupted data. Such a procedure produces sparse solutions, which can dramatically reduce the computational burden of the solution in its application stage.

The Gaussian kernel is widely used with good performance. We will consider the solution in the RKHS induced by Gaussian kernel, called Gaussian-RKHS. The terminology Gaussian-RKHS denoted by H_{K^σ}, is to be compared with the Sobolev RKHS formulation given in (2.9) that can be called Sobolev-RKHS. In some literature [9], the Gaussian-RKHS is regarded as an infinite-order Sobolev space. In the Gaussian-RKHS setting, we are to find the solution for

$$\min_{F\in H_{K^\sigma}}\left\{\lambda\|F\|^2_{H_{K^\sigma}} + \|\Delta F\|^2_{L_2(\mathbf{R}^n)} + \sum_{j=1}^N C_j V(F(x_j), A_j)\right\}, \tag{2.11}$$

in H_{K^σ}, and finally, by letting $\lambda \to 0$, to get solution u for (2.1).

3. SVM based approach

In this section, we will use SVM based methods to solve the problem. We will focus on developing the algorithms that find solutions in $H^s(\mathbf{R}^n)$, the extension of the algorithms to Gaussian RKHS $H_{K^\sigma}(\mathbf{R}^n)$ being straightforward.

3.1. Solution in $H^s(\mathbf{R}^n)$

For any fixed points $\{x_j\}_{j=1}^N$ on the boundary ∂D and for any given values $\{A_j\}_{j=1}^N$, for certain constants $C_j > 0$, we want to find the solution of

$$\min_{F\in H^s}\left\{\lambda\|F\|_{H^s}^2 + \|\Delta F\|_{L_2(\mathbf{R}^n)}^2 + \sum_{j=1}^N C_j|F(x_j) - A_j|_\varepsilon\right\}, \tag{3.13}$$

where $|F(x_j) - A_j|_\varepsilon$ is defined in (2.10). By recalling the definition of the Hilbert space H_{K_λ}, with a multiple on the constants C_j, the above is equivalent to solving

$$\min_{F\in H^s}\left\{\frac{1}{2}\|F\|_{H_{K_\lambda}}^2 + \sum_{j=1}^N C_j|F(x_j) - A_j|_\varepsilon\right\}. \tag{3.14}$$

We like the factor $\frac{1}{2}$ there as it induces a better looking solution in the end.

Remark 3.1. The form (3.14) makes it easy to use the SVM method. Although SVM usually deals with identical C_j, the algorithm, allows different values of C_j to be used to more general cases.

To deal with the function $|\cdot|_\varepsilon$, by introducing some new variables, called *slack variables*, considering that $F(x_j) = \langle F(\cdot), K_\lambda(\cdot, x_j)\rangle$, we replace the problem (3.14) by the following equivalent one (by equivalent" we mean that they have the same solutions, see [16]):

$$\min_{F\in H^s,\xi,\xi^*}\left\{\frac{1}{2}\|F\|_{H_{K_\lambda}}^2 + \sum_{j=1}^N C_j(\xi_j + \xi_j^*)\right\}, \tag{3.15}$$

where

$$\xi_j = \max\{\langle F(\cdot), K_\lambda(\cdot, x_j)\rangle - A_j - \varepsilon, 0\}, \quad \text{and} \quad \xi_j^* = \max\{A_j - \langle F(\cdot), K_\lambda(\cdot, x_j)\rangle - \varepsilon, 0\},$$

and

$$\xi^{(*)} = [\xi_1^{(*)}, \ldots, \xi_N^{(*)}] \; (\xi_j^{(*)} \text{ stands for } \xi_j \text{ and } \xi_j^*).$$

It is easy to verify that

$$\xi_j + \xi_j^* = \max\{|\langle F(\cdot), K_\lambda(\cdot, x_j)\rangle - A_j| - \varepsilon, 0\} = |F(x_j) - A_j|_\varepsilon,$$

and

$$\begin{aligned}
&\langle F(\cdot), K_\lambda(\cdot, x_j)\rangle - A_j \leqslant \varepsilon + \xi_j, \quad j = 1, \ldots, N,\\
&A_j - \langle F(\cdot), K_\lambda(\cdot, x_j)\rangle \leqslant \varepsilon + \xi_j^*, \quad j = 1, \ldots, N,\\
&\xi_j, \xi_j^* \geqslant 0,
\end{aligned} \tag{3.16}$$

for $j = 1, \ldots, N$.

This problem can be solved through the dualization method utilizing Lagrange multipliers [16].

First, we introduce the constraints into the primal functional by means of Lagrange multipliers, obtaining the following Lagrange function:

$$L = \frac{1}{2}\|F\|_{H_{K_\lambda}}^2 + \sum_{j=1}^N C_j(\xi_j + \xi_j^*) + \sum_{j=1}^N \alpha_j[\langle F(\cdot), K_\lambda(\cdot, x_j)\rangle - A_j - \varepsilon - \xi_j] + \sum_{j=1}^N \alpha_j^*\left[-\langle F(\cdot), K_\lambda(\cdot, x_j)\rangle + A_j - \varepsilon - \xi_j^*\right] - \sum_{j=1}^N (\beta_j\xi_j + \beta_j^*\xi_j^*). \tag{3.17}$$

It can be seen that a Lagrange multiplier (or dual variable) has been introduced for each constraint of the primal problem. Lagrange multipliers are constrained to be $\alpha_j^{(*)}, \beta_j^{(*)} \geqslant 0$.

Denote $\alpha^{(*)}$ as $[\alpha_1^{(*)}, \ldots, \alpha_N^{(*)}]$ and $\beta^{(*)}$ as $[\beta_1^{(*)}, \ldots, \beta_N^{(*)}]$. The primal problem (3.15) is equivalent to $\min_{F,\xi,\xi^*}\max_{\alpha^{(*)},\beta^{(*)}} L$. Its dual problem is $\max_{\alpha^{(*)},\beta^{(*)}}\min_{F,\xi,\xi^*} L$. According to [1], if the primal problem is convex, a point is optimal of primal problem and dual problem if and only if it satisfies the Karush–Kuhn–Tucker (KKT) conditions.

The problem (3.15) is convex. The corresponding KKT conditions consist of the equations:

$$\partial_F L = F - \sum_{j=1}^{N}(\alpha_j^* - \alpha_j)K_\lambda(x_j) = 0, \tag{3.18}$$

$$\partial_{\xi_j^{(*)}} L = C_j - \alpha_j^{(*)} - \beta_j^{(*)} = 0, \tag{3.19}$$

and

$$\alpha_j[\langle F(\cdot), K_\lambda(\cdot, x_j)\rangle - A_j - \varepsilon - \xi_j] = 0,$$

$$\alpha_j^*[-\langle F(\cdot), K_\lambda(\cdot, x_j)\rangle + A_j - \varepsilon - \xi_j^*] = 0,$$

$$\beta_j \xi_j = 0,$$

$$\beta_j^* \xi_j^* = 0,$$

for $j = 1, \ldots, N$.

Under those conditions, the primal problem (3.15) can be solved through solving its dual problem:

$$\max_{\alpha,\alpha^*} \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K_\lambda(x_i, x_j) + \varepsilon\sum_{j=1}^{N}(\alpha_j^* + \alpha_j) - \sum_{j=1}^{N}A_j(\alpha_j^* - \alpha_j),$$

subject to the constraints

$$0 \leqslant \alpha_j^*, \alpha_j \leqslant C_j, \quad j = 1, \ldots, N.$$

The solution can be written as

$$F(x) = \sum_{j=1}^{N}(\alpha_j^* - \alpha_j)K_\lambda(x_j, x). \tag{3.20}$$

As in the usual SVM framework, by setting appropriate $\varepsilon > 0$, we obtain that only a subset of the Lagrange multipliers will be nonzero. This, therefore, leads to a sparse solution. The input data points x_i for which α_i or α_i^* are different from zero are called support vectors. Hence, our solution can be written as

$$F(x) = \sum_{j\in J}(\alpha_j^* - \alpha_j)K_\lambda(x_j, x), \tag{3.21}$$

where J is the index set of the support vectors with $|J| \leqslant N$.

By taking a small λ, we will be able to obtain the approximate solution of the problem:

$$\Delta u \sim 0,$$

and

$$u(x_j) \sim A_j, \quad j = 1, \ldots, N.$$

Letting λ tend to zero, we obtain, mathematically, the solution u of the problem:

$$\Delta u = 0 \tag{3.22}$$

and

$$u(x_j) = A_j \quad j = 1, 2, 3, \ldots, N, \tag{3.23}$$

for any finite points $\{x_j\}_{j=1}^N$ and for any values $\{A_j\}_{j=1}^N$.

Remark 3.2. The parameters C_j are adjustable in order to get better solutions for the problem. A number of methods were proposed to set the parameter values, such as genetic algorithm (GA), particle swarm optimization (PSO), cross-validation. In our algorithm, we will adopt genetic algorithm to set the parameter values.

3.2. Solution in Gaussian-RKHS

The Gaussian-RKHS, which is defined as [9]

$$H_{K^\sigma(\mathbf{R}^n)} = \{f \in L^2(\mathbf{R}^n) : \|f\|_{H_{K^\sigma}}^2 = \frac{\int_{\mathbf{R}^n} e^{\frac{\sigma^2|\xi|^2}{4}}|\hat{f}(\xi)|^2 d\xi}{(2\pi)^n(\sigma\sqrt{\pi})^n} < \infty\}.$$

The reproducing kernel is Gaussian kernel

$$K^{\sigma}(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}},$$

where $x, y \in \mathbf{R}^n$, and $\sigma > 0$.

Our problem is formulated as follows.

For any fixed points $\{x_j\}_{j=1}^N$ of the boundary ∂D, any given values $\{A_j\}_{j=1}^N$ and any fixed $\{\lambda_j\}_{j=1}^N, \lambda_j > 0$, find

$$\min_{F \in H_{K^{\sigma}}} \left\{ \lambda \|F\|^2_{L_2(\mathbf{R}^n)} + \|\Delta F\|^2_{L_2(\mathbf{R}^n)} + \sum_{j=1}^N C_j |F(x_j) - A_j|_{\varepsilon} \right\}. \tag{3.24}$$

According to Theorem 2.1, the reproducing kernel of Hilbert space $H_{K^{\sigma}_{\lambda}}$ with the norm

$$\left(\lambda \|F\|_{H_{K^{\sigma}}} + \|\Delta F\|^2_{L_2(\mathbf{R}^n)} \right)^{\frac{1}{2}} \tag{3.25}$$

is

$$K^{\sigma}_{\lambda}(x,y) = (\lambda I + \triangle^* \triangle)^{-1} K^{\sigma}(x,y) = \int_{\mathbf{R}^n} \frac{(2\pi)^{n/2} \sigma^2 e^{\frac{-\sigma^2 |\xi|^2}{2} + i\xi(x+y)}}{\lambda + |\xi|^4} d\xi. \tag{3.26}$$

The next steps are the same as in the $H^s(\mathbf{R}^n)$ case.

4. Generalization ability analysis

In learning from a set of examples, the key property of a learning algorithm is generalization, namely the ability of an algorithm to perform accurately on new examples after having trained on a set of training data. The training examples come from some generally unknown probability distribution, while the learner has to extract from them something more general that allows him to produce useful predictions in new cases. According to the result in [14], we can obtain a generalization bound for our algorithm.

A generalization result in [14] is as follows. We fix a target accuracy $\theta > 0$ and $0 < \gamma \leqslant \theta$. Consider a real-valued (hypothesis) function class $\mathcal{F}$ with domain X. For a function $F \in \mathcal{F}$ and a training point $(x_i, A_i) \in X \times \mathbf{R}$, we define

$$\xi((x_i, A_i), F, \theta, \gamma) = \xi_i = \max\{0, |F(x_i) - A_i| - (\theta - \gamma)\}.$$

This quantity is the amount by which $|F(x_i) - A_i|$ exceeds $\theta - \gamma$ on the point (x_i, A_i) or 0 if g is within $\theta - \gamma$ of the targeted value. This is, in fact, the ε insensitive loss function given by (2.10) with $\varepsilon = \theta - \gamma$.

For a training set $S = ((x_1, y_1), \ldots, (x_l, y_l))$, define the vector valued

$$\xi = \xi(S, F, \theta, \gamma) = (\xi_1, \ldots, \xi_l).$$

Note that $\xi_i > \gamma$ means the error of g on (x_i, A_i) is larger than θ.

Proposition 4.1. [14] *Let $\mathcal{F}$ be the set of real-valued linear functionals on a real-Hilbert space X that, in accordance with the Riesz representation theorem, is identical with the space X itself. Fix $\theta \in \mathbf{R}, \theta > 0$, and a probability distribution P on the space $X \times \mathbf{R}$. If we restrict the inputs to the ball $B(0,R) = \{x \in X : \|x\| \leqslant R\}$, then there is a constant c such that with probability at least $1 - \delta$ over randomly drawn training sets S of size l and for all $\gamma, 0 < \gamma \leqslant \theta$, the probability that a function $F \in \mathcal{F}$ has error larger than θ on a randomly chosen input is bounded by*

$$\epsilon(l, \delta, \gamma) = \frac{c}{l} \left(\frac{\|F\|_2^2 R^2 + \|\xi\|_1^2 \log(1/\gamma)}{\gamma^2} \log^2 l + \log \frac{1}{\delta} \right). \tag{4.27}$$

In other words, with the notation

$$\mathrm{err}_P(F, \theta) = P(\{(x, A) \in X \times \mathbf{R} : \ |F(x) - A| \geqslant \theta\}),$$

there holds

$$P^l(\{S : \mathrm{err}_P(F, \theta)) \leqslant \epsilon(l, \delta, \gamma)\}) \geqslant 1 - \delta, \tag{4.28}$$

where P^l is the product probability induced by P over $(X \times \mathbf{R})^l$.

Our algorithms are equivalent to first map the fixed points $\{x_j\}_{j=1}^N$ to the space $H_{K_{\lambda}}$ (or space $H^{\sigma}_{K_{\lambda}}$) by $x_j \to K_{\lambda}(x_j, \cdot)$ (or $x_j \to K^{\sigma}_{\lambda}(x_j, \cdot)$), then find solution in $\mathcal{F}$ (is $\{\langle F, K_{\lambda}(x, \cdot)\rangle : F \in H_{K_{\lambda}}\}or\{\langle F, K^{\sigma}_{\lambda}(x, \cdot)\rangle : F \in H_{K^{\sigma}_{\lambda}}\}$). Therefore, based on Proposition 4.1, we can obtain the generalization ability of our algorithms, that is, the probability that the solution has error larger than a fixed target accuracy θ on a randomly chosen input is bounded by (4.27).

5. Numerical experiments

In this section, we evaluate the performance of the two SVM based algorithms, SVM-Tik-K_λ algorithm, SVM-Tik-K_λ^σ algorithm, and they are compared with standard numerical method in Example 1. Specifically, the following algorithms are considered:

(1) SVM-Tik-K_λ^σ.
(2) SVM-Tik-K_λ.
(3) Five-point difference method.

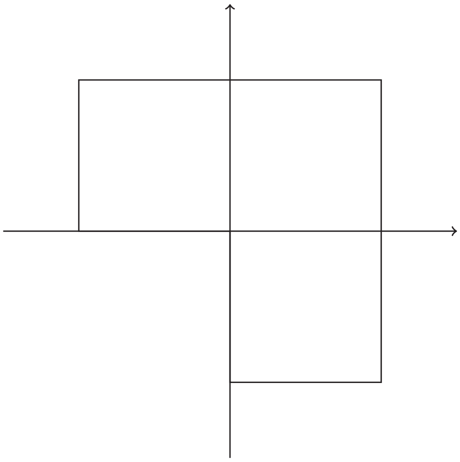
5.1. Training and test data

In the following examples, we consider $n = 2$ and $s = 2$, we take $N = 80$ or $N = 160$ points on the boundary of a L-shaped domain Ω with 0.025 span (80 points on the boundary) or 0.0125 span (160 points on the boundary) without noise and with different signal-to-noise (SNR) Gaussian noise. The L-shaped domain is a subset of $[-0.25, 0.25] \times [-0.25, 0.25]$ with interior corner at $(0,0)$ and $g(x,y) = xy$ in Example 1 and $g(x,y) = \ln(\sqrt{(x+0.01)^2 + (y+0.01)^2})$ in Example 2, where $(x,y) \in \partial\Omega$.

Let u be the original function, f be our approximated function. Two measurements are used to estimate the performance of our results. One is mean squared error $(\text{MSE}) = \frac{1}{N}\sum_{i=1}^N (u(p_i) - f(p_i))^2$, the other is squared correlation coefficient (SCC), defined by

$$\frac{\left(N\sum_{i=1}^N f(p_i)u(p_i) - \sum_{i=1}^N f(p_i)\sum_{i=1}^N u(p_i)\right)^2}{\left(N\sum_{i=1}^N u(p_i)^2 - \left(\sum_{i=1}^N u(p_i)\right)^2\right)\left(N\sum_{i=1}^N f(p_i)^2 - \left(\sum_{i=1}^N f(p_i)\right)^2\right)}.$$

The following area is the L-shaped domain we consider.



For the five-point difference method, the grid is constructed based on the boundary points, the time and space interval are equal to 0.0125 (160 points on the boundary) or 0.025 (80 points on the boundary). The performance of the five-point difference method is measured by approximated values of grid points.

5.2. Tuning the free parameters

Two free parameters have to be tuned in SVM algorithms, which are the insensitivity zone ε and the penalty parameter C. These free parameters need to be a priori fixed. In this paper, the optimal free parameters are searched using genetic algorithm.

Example 1. Fig. 1 shows the performance of the five-point difference method, SVM-Tik-K_λ and SVM-Tik-K_λ^σ given 160 noise-free data on the boundary. Fig. 2 shows the performance of five-point difference method, SVM-Tik-K_λ and SVM-Tik-K_λ^σ algorithms in the presence of additive, Gaussian noise with SNR = 30. Tables 3 and 4 show the performance of the proposed algorithms and five-point difference method, in the presence of additive, Gaussian noise, as a function of SNR. The five-point difference method can achieve better performance than SVM-Tik-K_λ and SVM-Tik-K_λ^σ for noise free case. However, with the decrease of SNR, the five-point difference method is not as robust as SVM based algorithms to noise.

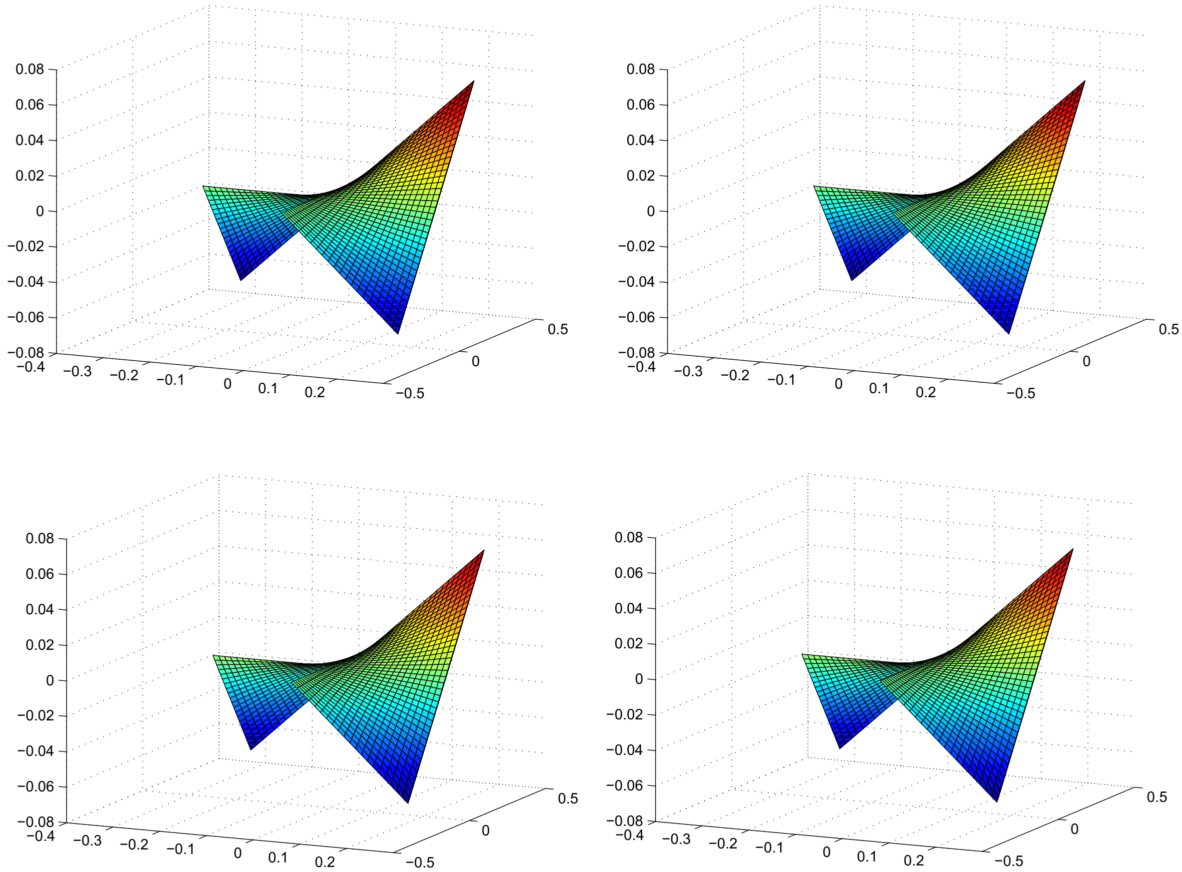


Fig. 1. The exact graph of $u(x,y) = xy$ (top left). Given 160 noise-free points on the boundary. The approximated graph by five-point difference method (top right). The approximated graph by SVM-Tik-K_λ (left bottom) for $\lambda = 10^{-5}$. The approximated graph by SVM-Tik-K_λ^σ (right bottom) for $\lambda = 10^{-5}$.

5.3. Number of support vectors

Sparseness in SVM-Tik-K_λ and SVM-Tik-K_λ^σ have been studied as a function of both SNR and sample length (in the presence of Gaussian noise SNR = 20), as shown in Tables 1 and 2. The sparseness is defined as

$$sparseness = \frac{\text{number of support vectors}|J|}{N} \times 100\%.$$

The SVM methods clearly tend to yield less sparse solutions (in average) with decreasing SNR and with increasing samples.

5.4. Execution time

The execution time of the three algorithms has been studied as a function of SNR which is shown in Table 5. In order to work out the kernels and appropriate parameters, the proposed methods require more computing time. However, as compensation, we obtain differentiable closed forms for the approximation. The finite difference method is a discrete method that can only obtain discrete data approximation.

Example 2. Tables 8 and 9 show the performance of the proposed algorithms, in the presence of additive, Gaussian noise, as a function of SNR. We can see that SVM-Tik-K_λ and SVM-Tik-K_λ^σ can achieve a good performance.

Sparseness in SVM based regression has been studied as a function of both SNR and sample length (in the presence of Gaussian noise SNR = 20), as shown in Tables 6 and 7. The SVM methods clearly tend to yield less sparse solutions (in average) with decreasing SNR and with increasing samples. The execution time of the two algorithms has been studied as a function of SNR which is shown in Table 10. The computing time are reasonable to compute the kernels and appropriate parameters.

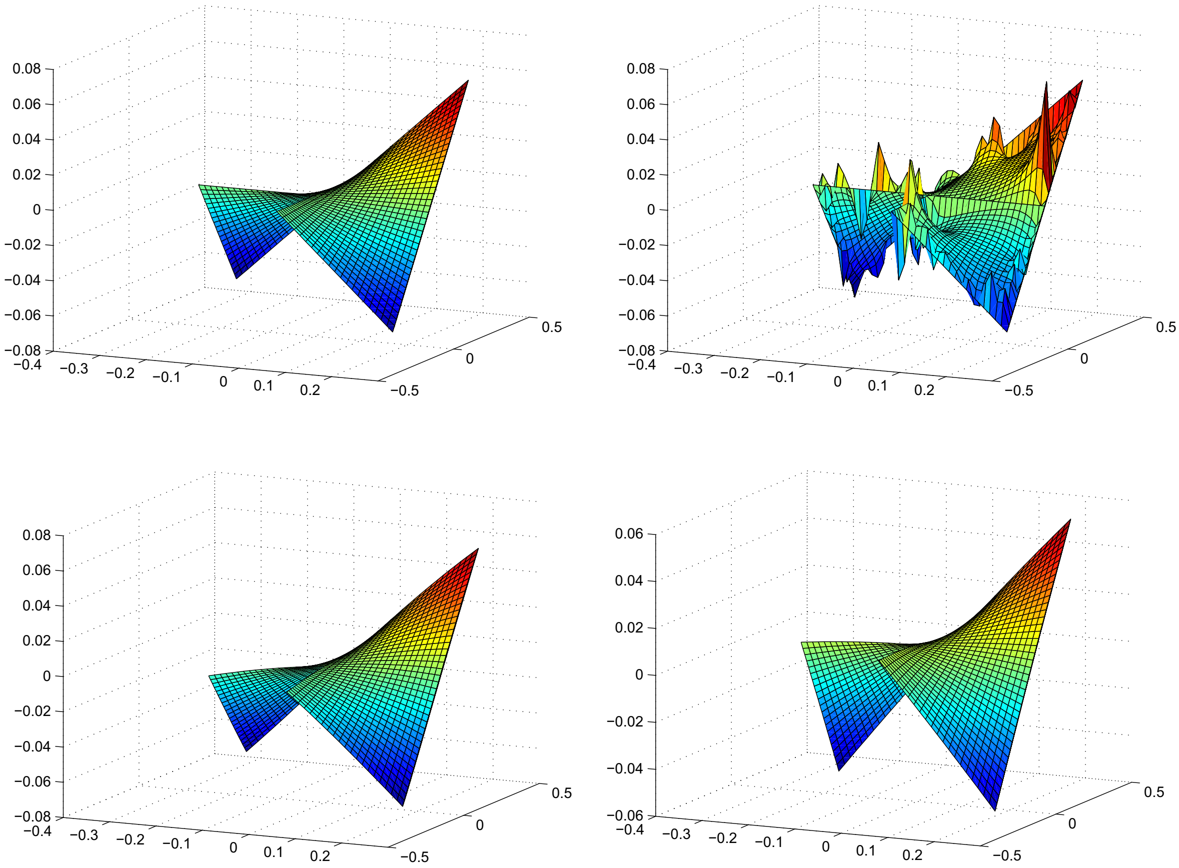


Fig. 2. The exact graph of $u(x,y) = xy$ (top left). Given 160 Gaussian noise (SNR = 30) corrupted points on the boundary. The approximated graph by five-point difference method (top right). The approximated graph by SVM-Tik-K_λ (left bottom) for $\lambda = 10^{-5}$. The approximated graph by SVM-Tik-K_λ^σ (right bottom) for $\lambda = 10^{-5}$.

Table 1
Sparseness with SNR (Gaussian noise).

Method	Noise			
	No noise (%)	40 dB (%)	30 dB (%)	20 dB (%)
SVM-Tik-K_λ	6.88	1.87	3.75	23.13
SVM-Tik-K_λ^σ	6.25	2.5	21.88	33.38

Table 2
Sparseness with number of samples.

Method	Number of samples	
	80	160
SVM-Tik-K_λ	20%	23.13%
SVM-Tik-K_λ^σ	20.62%	25%

Table 3
MSE with SNR (Gaussian noise).

Method	Noise			
	No noise	40 dB	30 dB	20 dB
SVM-Tik-K_λ	2.1444e−007	2.1075e−006	1.6271e−005	1.8360e−004
SVM-Tik-K_λ^σ	3.4562e−005	4.2987e−005	3.3182e−005	8.5398e−004
Five-point difference method	7.0266e−034	5.3191e−006	3.6362e−005	4.0248e−004

Table 4
SCC with SNR (Gaussian noise).

Method	Noise			
	No noise	40 dB	30 dB	20 dB
SVM-Tik-K_λ	0.9998	0.9983	0.9957	0.9348
SVM-Tik-K_λ^σ	0.9992	0.9994	0.9945	0.9934
Five-point difference method	1	0.9957	0.9578	0.5758

Table 5
Computation time (seconds) with SNR (Gaussian noise).

Method	Noise			
	No noise	40 dB	30 dB	20 dB
SVM-Tik-K_λ	35.6	53.4	56.8	83.4
SVM-Tik-K_λ^σ	29.7	58.4	65.3	86.8
Five-point difference method	0.58	0.51	0.54	0.53

Table 6
Sparseness with SNR (Gaussian noise).

Method	Noise			
	No noise (%)	40 dB (%)	30 dB (%)	20 dB (%)
SVM-Tik-K_λ	25	36.25	37.38	44.38
SVM-Tik-K_λ^σ	25	36.25	37.5	46.88

Table 7
Sparseness with number of samples.

Method	Number of samples	
	80	160
SVM-Tik-K_λ	21.88%	44.38%
SVM-Tik-K_λ^σ	25%	46.88%

Table 8
MSE with SNR (Gaussian noise).

Method	Noise			
	No noise	40 dB	30 dB	20 dB
SVM-Tik-K_λ	1.477e−004	1.584e−004	3.187e−004	3.706e−003
SVM-Tik-K_λ^σ	3.985e−004	2.349e−004	4.546e−004	5.721e−003

Table 9
SCC with SNR (Gaussian noise).

Method	Noise			
	No noise	40 dB	30 dB	20 dB
SVM-Tik-K_λ	0.9971	0.9968	0.9936	0.9906
SVM-Tik-K_λ^σ	0.9937	0.9936	0.9923	0.9902

Table 10
Computation time (seconds) with SNR (Gaussian noise).

Method	Noise			
	No noise	40 dB	30 dB	20 dB
SVM-Tik-K_λ	21.73	56.34	109.88	115.24
SVM-Tik-K_λ^σ	34.76	55.87	112.67	113.89

6. Conclusion

A new approach based on the SVM framework and the Tikhonov regularization numerically solving the Dirichlet problem is presented. With the proposed approach, in each Dirichlet problem, sparse representations in linear combinations of specific functions are obtained to approximate the desired precise solution. The solutions of the proposed algorithms are robust to noise. The proposed algorithms are compared with the existing popular algorithm to solve Dirichlet problem. The proposed approach of combining SVM and Tikhonov regularization gives promising results.

The proposed approach can be easily adopted to deal with domains in higher dimensions (three or more). As dimension increases, the number of training points becomes large. This turns to be crucial for the methods being based on particular functions on local such as splines around each grid point. In such case the required number of parameters becomes excessively large and, therefore, both memory and computation time become extremely large. In the case of the SVM based method, the number of training parameters remains almost the same as the dimension increases. The other great advantage is that proposed approach can be used to irregular domains.

Acknowledgements

The work was partially supported by Research Grant of University of Macau MYRG116(Y1-L3)-FST13-QT, Macao Science and Technology Fund FDCT/098/2012/A3. The authors would like to thank sincerely Prof. Saburou Saitoh and Dr. Guanghui Hu for their precious helps. The authors thank the reviewers for their comments, which helped improve the paper significantly.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.amc.2014.07.089.

References

[1] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[2] M. Asaduzzaman, T. Matsuura, S. Saitoh, Constructions of approximate solutions for linear differential equations by reproducing kernels and inverse problems, in: 4th ISAAC Toronto Congress Proceedings, World Scientific, 2005, pp. 335–343.
[3] T.J.R. Hughes, The Finite Element Method, Prentice Hall, New Jersey, 1987.
[4] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Networks 9 (1998) 987–1000.
[5] I.E. Lagaris, A. Likas, D.I. Fotiadis, Neural-network methods for boundary value problems with irregular boundaries, IEEE Trans. Neural Networks 11 (2000) 1041–1049.
[6] J.Y. Li, S.W. Luo, Y.J. Qi, Y.P. Huang, Numerical solution of elliptic partial differential equation using radial basis function neural networks, Neural Networks 16 (2003) 729–734.
[7] T. Matsuura, S. Saitoh, Dirichlet principle using computers, Appl. Anal. 84 (2005) 989–1003.
[8] T. Matsuura, S. Saitoh, D.D. Trong, Numerical solutions of the Poisson equation, Appl. Anal. 83 (2004) 1037–1051.
[9] Ha Quang Minh, Some properties of gaussian reproducing kernel Hilbert spaces and their implications for function approximation and learning theory, Constr. Approx. 32 (2) (2010) 307–338.
[10] S. Saitoh, Integral transforms, reproducing Kernels and their applications, in: Chapman & Hall/CRC Research Notes in Mathematics Series 369, CRC Press, 1997.
[11] S. Saitoh, Approximate real inversion formulas of the Gaussian convolution, Appl. Anal. 83 (2004) 727–733.
[12] S. Saitoh, Applications of reproducing kernels to best approximations, Tikhonov regularizations and inverse problems, in: 4th ISAAC Toronto Congress Proceedings, World Scientific, 2005, pp. 439–445.
[13] S. Saitoh, Best approximation, Tikhonov regularization and reproducing kernels, Kodai Math. J. 28 (2005) 359–367.
[14] J. Shawe-Taylor, N. Cristianini, On the Generalization of Soft Margin Algorithms, IEEE Trans. Inf. Theory 48 (2002) 2721–2735.
[15] G.D. Smith, Numerical Solution of Partial Differential Equations: Finite Difference Methods, Clarendon Press, Oxford, 1978.
[16] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.