An Overview of Online Learning in Reproducing Kernel Hilbert Spaces

Supawan Ponpitakchai
Department of Electrical and Computer Engineering, Naresuan University, Thailand
supawanph@nu.ac.th

***Abstract*—Learning System is a method to approximate an underlying function from a finite observation data. Since batch learning has a disadvantage in dealing with large data set, online learning is proposed to prevent the computational expensive. Iterative method called Stochastic Gradient Descent (SGD) is applied to solve for the underlying function on reproducing kernel Hilbert spaces (RKHSs). RKHS is widely used in many applications such as kernel method, radial basis function neural networks, Volterra filers and estimation of bandlimited functions. This approach has advantages that there is no local minima problem and convergence is also guaranteed because of using convex optimisation.This paper aims to provide background and theory of learning in RKHS which online kernel method is our main interest. The experiments show the results of learning from 3 test sets and some important parameters are also discussed.**

***Index Terms*—Reproducing kernel Hilbert spaces, online learning, stochastic gradient descent, kernel methods.**

I. Introduction

The goal of learning is to approximate a function from data samples, perhaps perturbed with noise [1]. In this approach, learning is considered as finding an approximation function which estimates an unknown mapping between known input/output samples. Once such unknown mapping has been accurately estimated, it can be extended to predict a future outputs from the known input value [2].

Reproducing kernel Hilbert spaces (RKHSs) have been applied to a number of well-known problems in signal processing, control, machine learning and function approximation as this approach performs advantages in solving these problems. The strong theoretical foundations of RKHS have been developed [3], [4] and can be appreciated in comparison to neural networks that the solutions are global and usually unique since kernel methods use the idea of convex optimisation in training method which does not suffer from the problem of local minima [5], [6]. Recent interest in kernel methods has been dominated by the machine learning community as it allows to obtain nonlinear algorithms from linear ones in a simple and elegant manner, for example, support vector machines, Gaussian process and regularisation networks [3], [7], [8].

Generally, the batch learning is assumed that the learner is given all examples simultaneously and allowed to use them as often as desired. However, it is very inefficient in the case of training with large data set. Then online learning is employed instead [9] as it has a better performance in handling a huge data set. In this framework, only one example is given at a time and then discarded after learning. Therefore, the learner receives new information at every moment and adapts to it, without having a large memory for storing old data [9]. Apart from easier feasibility and data handling the most important advantage of online learning is its ability to adapt to changing environment [10].

Solving for the approximated function with online kernel method was recently developed under the iterative calculation called stochastic gradient descent (SGD) [11]–[14] that the objective function is estimated sequentially. In each data coming, a new approximation function is produced by making use of gradient information. Under some mild conditions this approach is consistent, meaning that as the number of data observations becomes large, the error between the approximation function and the true function converges to the minimum possible risk [2]. SGD method has the parameter called learning rate, η, or step size which determines the length of algorithm movement in the direction of gradient. An improper value of learning rate can cause the algorithm to diverge. Moreover, finding a solution of the approximation function from a finite given sample is ill-posed as the output from measurements is contaminated by small errors (noises) such that our function fits very well on the training data but cannot generalise to future data. The so-called classical regularisation technique is applied for solving such ill-conditioned problems. Therefore, the approximation function can be investigated by minimising the risk functional together with a stabiliser (penalty or regularisation) term [15]. Regularisation parameter, ρ, needs to be considered in learning method that it must be chosen at an appropriate value for each problem.

The characteristic of estimating the function in RKHS is that the size of model, number of kernels and model parameters, increased without a limit. The method to bound the growth of the approximation function called sparse solution is introduced in [16], [17]. By making use of orthogonal projection, we can neglect the kernel which contributes an insignificant error. Then, the size of model and time of calculation can be decreased.

The convergence rate of online kernel method has been analysed in a number of research studies with variety setting of learning rate and regularisation parameter. The SGD algorithm called naive online R_{reg} minimisation algorithm (NORMA) is proposed as an online algorithm to find the minimiser of risk functional. This technique uses decayed learning rate and the convergence of the algorithm is investgated without probabilistic assumptions in [14], [18]. The convergence of online learning using SGD with probabilistic assumption of observation data is presented in [19]. This paper uses a decaying learning rate whereas the regularisation parameter is assigned as a constant. In setting regularisation parameter equal to zero, the convergence rate can be competitive when the value of learning rate is chosen appropriately [20].

This paper presents a general framework of online kernel method in RKHSs used in function approximation (regression) problem. The background theory starts from least squares solution in batch learning and then extended to sequential method called gradient descent and online gradient descent (stochastic gradient descent-SGD). In online method which is the main interest of this paper, the important parameters and selected reproducing kernel are discussed in terms of effects to learning performance. Data using in experiments is generated from 3 functions added with noise. The simulation results illustrate characteristics of approximated function which is the output of kernel method comparing with the given training data and the 3 functions.

In the next section, the general framework for function approximation in RKHS is described and followed by properties of reproducing kernel. Least square solution which is the method to approximate function from finite given data in batch learning is explained. Characteristics of gradient descent and their extension to SGD including with the important parameters are explained in Section III. Finally, simulation results of online kernel method which training data is generated from 3 test sets are shown.

II. Preliminaries and Notation

In general learning we need to find an unknown characteristic known as underlying function, $f : X \times Z \rightarrow R$ [7]. A set of samples is given in the form of observation data $\{x_i, z_i\}_{i=1}^N \in X \times Z$. The unknown f is assumed to belong to some RKHS, $\mathcal{H}_k$, defined on a subset of Euclidian space $X \subseteq \mathbb{R}^n$. We assume that the space of all possible observations is Z. Neglecting errors, the observations arise as follows

$$z_i = L_i f \tag{1}$$

where $\{L_i\}_{i=1}^N$ is a set of linear evaluation functionals defined on $\mathcal{H}_k$ which have a unique correspondence to z_i. The complete set of z_i can be expressed by

$$\mathbf{z} = \sum_{i=1}^{N} (L_i f) e_i = Lf \tag{2}$$

where $e_i \in R^N$ is the ith standard basis vector such that every value of e_i is equal to zero except position i which is equal to one [21].

A RKHS can be defined as a Hilbert space of functions on X, with the property that, for each $x \in X$, the evaluation functional, L_i, which associates f with $f(x_i)$, $L_i f \rightarrow f(x_i)$, is a bounded linear functional. The boundedness means that there exists a scalar M such that

$$|L_i f| = |f(x_i)| \leq M \|f\|_{\mathcal{H}_k} \quad \text{for all } f \text{ in the RKHS} \tag{3}$$

where $\| \cdot \|_{\mathcal{H}_k}$ is the norm in RKHS [22]. Following from the Riesz representation theorem, the observations can be expressed as [23]

$$L_i f = \langle k_i, f \rangle = f(x_i) \qquad f \in \mathcal{H}_k \tag{4}$$

where k_i depends only on x_i. Therefore the learning problem can be stated as follows: given the RKHS of functions ($\mathcal{H}_k$), the set of functions $\{k_i\}_{i=1}^N \in \mathcal{H}_k$, and the observations $\{x_i, z_i\}_{i=1}^N$, find a function $f \in \mathcal{H}_k$, such that (4) is satisfied [24].

The k_i is a positive definite function called the reproducing kernel (r.k.) of $\mathcal{H}_k$ and the unique representation of evaluation at x_i [25]. The function $k : X \times X$ can be defined by the Riesz representation theorem such that

$$\langle k(x_i, \cdot), k(x_j, \cdot) \rangle_{\mathcal{H}_k} = k(x_i, x_j) \tag{5}$$

where $k(x_i, \cdot)$ is considered as a function of X centered on x_i and we can write $k(x_i, \cdot) = k_i(\cdot)$, or more simply k_i. There are several examples of kernel functions. One common kernel used is the polynomial [26],

$$k(x_i, x_j) = (1 + x_i x_j)^d \qquad \text{where } d \in \mathbb{N}. \tag{6}$$

Another choice is the Gaussian kernel which is used in Gaussian radial basis function neural networks

$$k(x_i, x_j) = \exp(-\sigma \|x_i - x_j\|^2), \tag{7}$$

where σ defines the width of the kernel function. Besides, sigmoid kernels

$$k(x_i, x_j) = \tanh(\kappa(x_i, x_j) + \Theta) \tag{8}$$

are used with suitable value of gain κ and threshold Θ [27].

We can then represent any function in RKHS with reproducing kernel k in the form

$$f(\cdot) = \sum_{i=1}^{N} \alpha_i k(x_i, \cdot) \tag{9}$$

for $N \in \mathbb{Z}^+$ and $\alpha_i \in \mathbb{R}$ where this expression defines a finite dimensional subspace of $\mathcal{H}_k$ and the reproducing kernels are a basis for RKHS. The learning problem then reduces to that of estimating the parameters, α_i, in (9) using the information contained in the observation pairs, $\{x_i, z_i\}_{i=1}^N$.

III. Function approximation and gradient descent for kernel methods

In order to find a function approximation in a form of RKHS function, the problem is reduced to estimate appropriate values for the parameters, α_i. Using the available information contained in the sample data pairs, $\{x_i, z_i\}_{i=1}^N$. This section summaries the least squares solution to this problem with a batch gradient descent method and then extends to online learning case through stochastic gradient descent method (SGD).

A. Least squares solution

The approximation problem can be investigated for a best possible solution, u. Since $\mathcal{Z}$ is a subspace of $\mathbb{R}^N$, the range of $R(L)$ is closed [28]. It always has a unique solution $z = L^{-1} f$ when the operator L has an inverse. The following theorem presents a least squares solution of u.

Theorem III.1. *[28] Suppose $R(L)$ is closed and $z \in \mathcal{Z}$, then the following conditions on $u \in \mathcal{H}_k$ are equivalent:*

1) $\|Lu - z\|_{\mathcal{Z}} = \inf\{\|Lf - z\|_{\mathcal{Z}}$ *for any* $f \in \mathcal{H}_k\}$
2) $L^* Lu = L^* z$
3) $Lu = Pz.$

Note that, P is the projection of z onto $R(L)$ and L^* is the adjoint operator to L. However, we may have the case that the null space of L, $N(L) \neq 0$. The set of least squares solutions is given by [29]

$$S_g = \{u \in \mathcal{H}_k \ \mid \ u = u_0 + v, Lv = 0\} \tag{10}$$

where $v = N(L)$. This set is a closed convex set which contains a unique vector of minimal norm called the generalised solution [28] and denoted by $f^\dagger$:

$$\|f^\dagger\|_{\mathcal{H}_k} = \inf\{\|u\|_{\mathcal{H}_k} \quad | \quad u \in S_g\}. \tag{11}$$

The generalised inverse of L is defined by the mapping $L^\dagger : \mathcal{Z} \to \mathcal{H}_k$ such that $L^\dagger z = u$ and it has the relationship presented in the following theorem [29].

$$L^\dagger = (L^*L)^\dagger L^* = L^*(LL^*)^\dagger \tag{12}$$

Then [28]

$$f^\dagger = L^\dagger z = L^*(LL^*)^\dagger z. \tag{13}$$

In the case of RKHS, there exist expressions for the operators L^* and LL^* [21]

$$\begin{aligned} L^*c &= \sum_{i=1}^{p} k(x_i, \cdot)c_i \\ LL^* &= \sum_{j=1}^{p}\sum_{i=1}^{p} k(x_i, x_j)e_j e_i^T \end{aligned} \tag{14}$$

for any $c \in \mathbb{R}^p$. Let matrix K be a representation of LL^* that $[K]_{ij} = k(x_i, x_j)$. Assuming the inverse of the matrix K exists, we have $(LL^*)^\dagger = (LL^*)^{-1} = K^{-1}$. From (13), the prediction at a future data, x, is given as the following

$$f^\dagger(x) = \langle f^\dagger(\cdot), k(x, \cdot)\rangle \tag{15}$$

which can be written as

$$\begin{aligned} f^\dagger(x) &= \langle L^*(LL^*)\dagger z, k(x, \cdot)\rangle \\ &= \langle (LL^*)\dagger z, Lk(x, \cdot)\rangle \\ &= \langle k(x, \cdot), (LL^*)\dagger z\rangle \end{aligned} \tag{16}$$

Using the definition of K^{-1}, we then have

$$f^\dagger(x) = k^T K^{-1} z \tag{17}$$

where k is the vector

$$k = Lk(x, \cdot) = [k(x_1, x), \ldots, k(x_p, x)]^T. \tag{18}$$

However, the problem of determining the generalised solution can sometimes be ill-conditioned. In this situation, it is necessary to use the technique which applies to the case that $R(L)$ is not closed.

When $R(L)$ is not closed, the generalised solution $f^\dagger$ does not exist for any z as the generalised inverse is not continuous. The problem of determining the generalised solution is ill-posed. Hence, it is necessary to use the so-called regularisation theory. Roughly speaking, this theory performs the continuous approximations to the discontinuous inverse, generalised inverse, of operator L [29]. Then, the solution of the problem is considered as finding function f_{reg} which is a least squares solution.

Lemma III.2. *[29] For any value of regularisation parameter $\rho > 0$, there exists a unique solution $f_{reg} \in \mathcal{F}$ which minimises the functional*

$$g_{reg}(f) = \|Lf - z\|^2_{\mathcal{Z}} + \rho\|f\|^2_{\mathcal{F}} \tag{19}$$

and equivalent to the unique solution of the Euler equation

$$(L^*L + \rho I)f_{reg} = L^*z. \tag{20}$$

Then

$$f_{reg} = (L^*L + \rho I)^{-1}L^*z. \tag{21}$$

This lemma is induced using Tikhonov regularisation which is a powerful tool for finding stable approximations for inverse problem. This method imposes well-posedness on ill-posed problems by making use of prior information like smoothness constraints [30]. In the framework of Tikhonov's regularisation, smoothness constraint is represented in terms of penalisation which is revealed by the term $\rho\|f\|^2$ in (19). Introducing ρ as a smoothness constraint can also overcome the problem of overfitting that the approximation function fits very well on observation data but cannot perform prediction of an unforseen data. From (21), the expression is equivalent to

$$f_{reg} = L^*(\rho I + LL^*)^{-1}z. \tag{22}$$

To compute the prediction value at point x, we have

$$f_{reg}(x) = \langle f_{reg}(\cdot), k(x, \cdot)\rangle \tag{23}$$

and can be written as

$$\begin{aligned} f_{reg}(x) &= \langle (\rho I + LL^*)^{-1}z, Lk(x, \cdot)\rangle \\ &= \langle Lk(x, \cdot), (\rho I + LL^*)^{-1}z\rangle \\ &= k^T(\rho I + K)^{-1}z \end{aligned} \tag{24}$$

where k is the vector presented in (18). The regularised solution, (24), is same as the generalised solution, (17), where K is replaced by $\rho I + K$.

B. Batch gradient descent

In this section, we present the gradient descent method to investigate the regularised solution from given observation data. This method differs from the previous technique that the result is produced at each step of iterations. As a number of iteration increased, we will have more accuracy of approximated function. Let the non negative cost functional $g_{reg} : Z \to \mathbb{R}$ be defined by

$$g_{reg}(x) = \frac{1}{2}\|Lf - z\|^2 + \frac{\rho}{2}\|f\|^2 \tag{25}$$

where $\rho > 0$ known as the regularisation parameter. Solution of the function approximation problem is a minimiser of g_{reg}. Given an initial approximation, f_0, the gradient descent method for minimising g_{reg} is given by

$$f_{n+1} = f_n - \eta_n \nabla g_{reg}(f_n) \tag{26}$$

where $\nabla g_{reg}(f_n)$ is the gradient of g_{reg} at f_n and η_n is a learning rate or step size for moving to a direction of decreasing g_{reg}. [28] presented that

$$\nabla g_{reg}(f_n) = L^*Lf_n - L^*z + \rho f_n \tag{27}$$

and we then have

$$\begin{aligned} f_{n+1} &= f_n - \eta_n(L^*Lf_n - L^*z + \rho f_n) \\ &= (1 - \eta_n\rho)f_n - \eta_n L^*(Lf_n - z). \end{aligned} \tag{28}$$

Using (9), the solution can be given by

$$f_{n+1} = L^*\alpha_{n+1} = \sum_{i=1}^{n+1} \alpha_{n+1}^i k_i. \tag{29}$$

Substitution the above expression to (28), the update for α is as the follows

$$\alpha_{n+1} = (1-\eta_n\rho)\alpha_n - \eta_n(K\alpha_n - z). \tag{30}$$

IV. Online Stochastic Gradient Descent

This method uses only a pair of data to update the parameters at each iteration. After that, the pair of data is discarded and wait for a new pair of data for the next iteration. Suppose that, at each iteration, we observe only a part of Z, denoted z_n (typically the nth observation). The associated linear evaluation functional at each iteration is then [13]

$$L_n f = z_n. \tag{31}$$

Generally, the function approximation of f can be achieved by minimising a risk functional, $\hat{g}(f_n) = \frac{1}{2}\|L_{n+1}f_n - z_{n+1}\|^2$ where f_n is a function approximation at time n. However, the problem of estimating the function f_n is ill-posed [15] in the sense that the estimation function fits very well on the observation data but cannot fit the unforeseen data. To overcome this problem, Tikhonov regularisation method [30] is applied for solving this problem. We now define the instantaneous non-negative functional

$$\hat{g}_{reg}(f_n) = \frac{1}{2}\|L_{n+1}f_n - z_{n+1}\|^2 + \frac{\rho}{2}\|f_n\|^2 \tag{32}$$

where $\frac{\rho}{2}\|f_n\|^2$ is a regularisation term and $\rho \geq 0$ is a regularisation parameter [11].

The objective of online learning is then to evaluate function f_n which minimises the risk functional $\hat{g}_{reg}$. Using SGD, the function update, f_{n+1}, will be calculated at each incoming observations, (x_n, z_n), by

$$f_{n+1} = f_n - \eta_n \nabla\hat{g}_{reg}(f_n) \tag{33}$$

where $\nabla\hat{g}_{reg}(f_n)$ is the instantaneous gradient of $\hat{g}_{reg}$ or the direction of gradient descent at f_n and η_n is the learning rate. Hence, we obtain the function update

$$f_{n+1} = (1-\eta_n\rho_n)f_n - \eta_n L_{n+1}^*(L_{n+1}f_n - z_{n+1}) \tag{34}$$

where L^* is the adjoint operator of L defined by $\langle Lf, z\rangle = \langle f, L^*z\rangle$. The instantaneous gradient of $\hat{g}_{reg}$ at f_n is given by

$$\begin{aligned}\nabla\hat{g}_{reg}(f_n) &= \frac{\partial\hat{g}_{reg}(f_n)}{\partial f_n} \\ &= L_{n+1}^*(L_{n+1}f_n - z_{n+1}) + \rho_n f_n. \end{aligned} \tag{35}$$

Here, for some constants, $L_{n+1}^* a = k_{n+1}a$ and also $L_{n+1}^* f_n = f_n(x_{n+1})$, therefore

$$f_{n+1} = (1-\eta_n\rho_n)f_n - \eta_n k_{n+1}(f_n(x_{n+1}) - z_{n+1}). \tag{36}$$

From (9), the function update which is the approximation of the unknown function we need to find, can be written in the form of kernel function. Then, the function update, f_{n+1}, is [31]

$$\begin{aligned} f_{n+1}(x) &= (1-\eta_n\rho_n)\sum_{i=1}^{n}\alpha_n^i k_i(x) - \eta_n e_{n+1}k_{n+1}(x) \\ &= \sum_{i=1}^{n+1}\alpha_{n+1}^i k_i(x) \end{aligned} \tag{37}$$

where the prediction error $e_{n+1} = L_{n+1}f_n - z_{n+1}$ and $L_{n+1}^* e_{n+1} = k_{n+1}e_{n+1}$. Considering (37), the parameter α_{n+1}^{n+1} equals to $-\eta_n e_{n+1}$, the function update at time $n+1$ is the previous function update added with kernel function which is weighted by prediction error and learning rate. Hence, the update parameters α_{n+1}^i are calculated from [32]

$$\alpha_{n+1}^i = \begin{cases} (1-\eta_n\rho)\alpha_n^i & \text{for } i \leq n \\ -\eta_n e_{n+1} & \text{for } i = n+1. \end{cases} \tag{38}$$

In SGD, the learning rate, η_n, plays an important role as the step size for moving in the direction such that $\hat{g}_{reg}(f_n)$ is minimised. The value of the learning rate to be used in algorithm is critical. Small values can lead to a slow minimisation process whereas high values may cause divergence.

Convergence of online learning has been guaranteed in a number of papers with variety setting of learning rate and regularisation parameter. The SGD algorithm called naive online R_{reg} minimisation algorithm (NORMA) is proposed as an online algorithm to find the minimiser of risk functional. This technique uses decayed learning rate and the convergence of the algorithm is investgated without probabilistic assumptions in [14], [18]. The convergence of online learning using SGD with probabilistic assumption of observation data is presented in [19]. Also, the convergence of SGD is guaranteed by using the decaying learning rate and regularisation parameter as given in [33],

V. Experimental design and results

In this section, kernel method are applied to 3 test problems which are function approximation problems. The training data (200 data points) were randomly generated from sinc function, exponential function and Paley-Wiener (PW) RKHS [34], [35]. These 3 test functions are selected due to their variety of complexity. Exponential function is the simplest and PW is the most complex. The corresponding r.k. for PW RKHS is

$$k(x, x') = \prod_{i=1}^{d}\frac{\sigma_i}{\pi} sinc[\frac{\sigma_i}{\pi}(x_i - x_i')] \tag{39}$$

where $sinc(u) = sin(\pi u)/\pi u$. Observation data was generated from a known function belonging in the PW RKHS

$$f(\cdot) = \sum_{i=1}^{M} m_i k_i(\cdot) \tag{40}$$

where M was chosen to be 20, m_i were randomly selected and kernel centres were uniformly distributed from the interval $[-10, 10]$. The data generated from function in (40), with $d = 1$ and $\sigma_1 = 3$ was used as the training data. Also, the data was added with noise such that

$$\hat{f}(x_n) = f(x_n) + N(0, \sigma^2) \tag{41}$$

where $N(0, \sigma^2)$ is a Gaussian distribution with variance $\sigma^2 = 0.2$ throughout. The performance of the algorithms was assessed using the usual MSE

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - z_i)^2 \tag{42}$$

which is calculated on the test set (50 data points). The algorithm for kernel method is summarised as follows:

1) Choose an appropriate reproducing kernel function and define their parameters. This study used kernel function in (7).
2) Choose a value for regularisation parameter $\rho \geq 0$ and learning rate $\eta \geq 0$.
3) For a whole set of training data, calculate the update of α from (30) and the prediction from (29) where the kernel function in each term has a centre at each data point.
4) Calculate MSE and repeat step (3)-(4) until it reaches the iteration number.

TABLE I
RESULTS OF KERNEL METHOD FROM 3 TEST SETS

Test set	η	ρ	kernel width	average MSE
sinc	0.2	0.005	50	0.0223
exponential	0.8	0.005	30	0.4495
PW	0.8	0.001	1	0.1094

Table I represents the results from online learning with kernel method. Parameters using in the table were the best set (minimum MSE) for each test set. In these experiments, Gaussian kernel was selected as reproducing kernel and kernel width was the corresponding parameter. The kernel width is a width of gaussian radial basis function which the small value means narrow shape. The proper values of kernel width in 3 test sets can be investigated from trial and error and the idea that complex data set needs a small value of kernel width.

The regularisation paremeter, ρ, is an important parameter to prevent overfitting problem. This parameter also means a level of believing in data set. Small value of regularisation parameter is a high level of believing. Then this value should be set as a small positive value (less than 0.005) as we trust in data set in high level but not in a hundred percent level. The other parameter is a step size parameter, η, which is a length of step to go down to minimum of error bound. The large value can cause divergence whereas the small value can be a very slow convergence.

From the table, the average MSE of sinc function is the smallest whereas the MSE of exponential function is the highest. This is an effect of using Gaussian kernel because the shape of Gaussian kernel is similar to sinc function then online kernel method performs better approximation than using in exponential data. In case of learning from PW test set, MSE is higher than sinc test set due to the high complexity of PW function. Instead of using Gaussian kernel, we can use other reproducing kernel as presented in Section II.

The results of approximation function generated from online kernel method using Gaussian kernel after 200th iteration for 3 test sets are shown with solid line in Figures 1- 3. The approximated data (solid line) in 3 test sets is closed to their true function (dash line) which means that our kernel method performs as an acceptable approximator. Moreover the approximated data does not have the overfitting problem that the solid line pass through every data points.

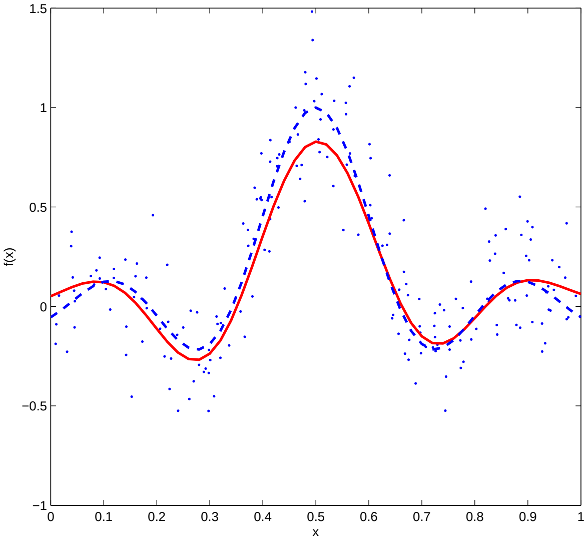

Fig. 1. Result of learning from sinc test set. Shown are approximated data (‘—–’), training data (‘··’) and sinc function (‘— —’).

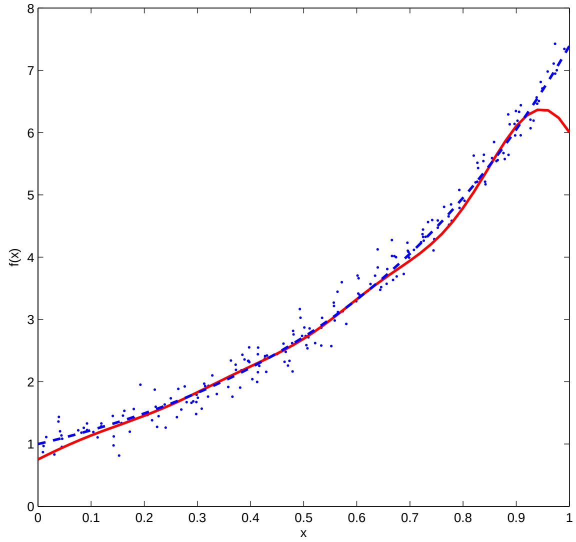

Fig. 2. Result of learning from exponential test set. Shown are approximated data (‘—–’), training data (‘··’) and exponential function (‘— —’).

MSE of 3 test sets are illustrated in Figure 4 in logscale. From the results, online kernel method has an ability to learn non-linear test sets because kernel method is a non-linear approximator. Moreover, kernel method uses the idea of convex optimisation then it does not have a problem of local minima. Then, initial value of α can be any value (always set to zero) and the

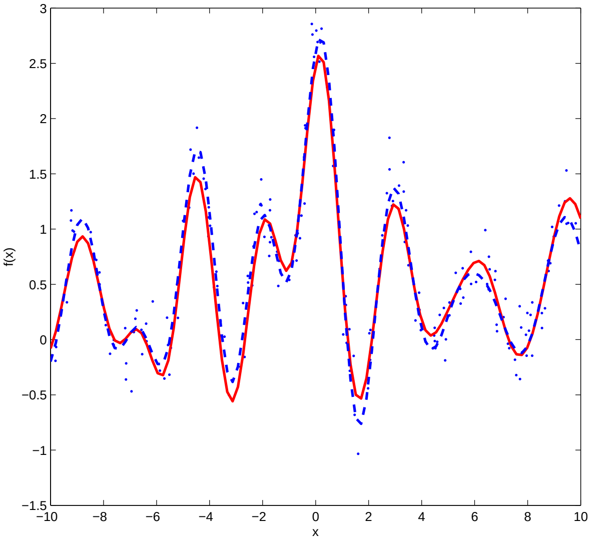

Fig. 3. Result of learning from PW test set. Shown are approximated data (‘—’), training data (‘··’) and PW function (‘– –’).

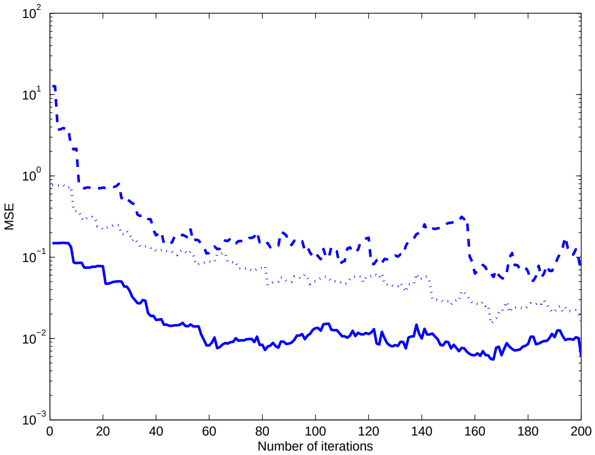

Fig. 4. MSE of learning from 3 test sets. Shown are sinc (‘—’), PW (‘··’) and exponential test set (‘– –’).

convergence are not effected.

VI. Conclusions

This work presents an overview of learning in RKHS which is mainly interested in solving the approximation function from finite given data. A sequential learning method called the online kernel method, is introduced including with characteristics of important parameters: step size, regularisation parameter and kenel width according to selected reproducing kernel. Performance of learning is investigated by using 3 sets of data generated from sinc, exponential and PW function. From experimental results, kernel method can be use as approximator for non-linear training data and does not have a problem of local minima. Convergence of the method also guaranteed corresponding to the conditions on parameters setting. However, calculation in online kernel method using SGD can be very slow due to a large number of data set. Technique to increase a speed of calculation should be investigated in future research.

References

[1] F. Cucker and D. X. Zhou, *Learning Theory: An Approximation Theory Viewpoint*. Campridge University Press, 2007.

[2] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, ser. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, 1998.

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

[4] N. Christianini, “Support vector machines and kernel methods: the new generation of learning machines,” *AI Magazine*, vol. 23, no. 3, pp. 31–41, 2002.

[5] C. Campbell, *Radial Basis Function Networks: Design and Applications*. Springer Verlag, 2000.

[6] N. Cristianini, C. Campbell, and C. Burges, Eds., *Special Issue: Support Vector Machines and Kernel Methods*, ser. Machine Learning, vol. 46. Kluwer Academic, 2002.

[7] O. Bousquet and F. Perez-Cruz, “Kernel methods on their applications to signal processing,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03)*, vol. 4, April 2003, pp. IV– 860–3.

[8] D. MacKay, “Gaussian Processes: A replacement for supervised neural networks?” 1997, lecture notes for a tutorial at NIPS 1997.

[9] K. Müller, A. Ziehe, N. Murata, and S. Amari, “Online learning in switching and drifting environments with application to blind source separation,” in *Online learning in neural networks*. Cambridge University Press, 1998, pp. 93–110.

[10] N. Murata, M. Kawanabe, A. Ziehe, and K. Müller, “On-line learning in changing environments with applications in supervised and unsupervised learning,” *Neural Networks*, vol. 15, no. 6, pp. 743–760, 2002.

[11] T. Dodd and R. Harrison, “Gradient descent approach to approximation in reproducing kernel Hilbert spaces,” Department of Automatic Control and Systems Engineering, University of Sheffield, UK, Tech. Rep. 821, 2002.

[12] T. Dodd, V. Kadirkamanathan, and R. Harrison, “Function estimation in Hilbert space using sequential projections,” in *Proceedings of the IFAC International Conference on Intelligent Control Systems and Signal Processing*, 2003.

[13] T. Dodd, B. Mitchinson, and R. Harrison, “Sparse stochastic gradient descent learning in kernel models,” in *Proceedings of The Second International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore*, 2003.

[14] J. Kivinen, A. Smola, and R. Williamson, “Online learning with kernels,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[15] 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.

[16] S. Phonphitakchai and T. J. Dodd, “Sparse learning and adaptation in online kernel methods,” in *The Proceeding of the Second Mahasarakham International Workshop on AI (MIWAI’08)*, R. Booth and C. Sombattheera, Eds., 2008, pp. 20–29.

[17] ——, “Stochastic meta descent in online kernel methods,” in *Sixth annual international conference organized by Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association*, 2009.

[18] J. Kivinen, A. Smola, and R. Williamson, “Online learning with kernels,” in *Advances in Neural Processing Systems*, 2001.

[19] S. Smale and Y. Yao, “Online learning algorithms,” October 2004, unpublished.

[20] Y. Ying and M. Pontil, “Online gradient descent learning algorithms,” *Foundations of Computational Mathematics*, vol. Online First, 2007, http://www.springerlink.com/content/106038.

[21] T. Dodd and R. Harrison, “Some lemmas on reproducing kernel Hilbert spaces,” Department of Automatic Control and Systems Engineering, University of Sheffield, UK, Tech. Rep. 819, 2002.

[22] N. Young, *An Introduction to Hilbert Space*. Cambridge University Press, 1988.

[23] N. Akhiezer and I. Glazman, *Theory of Linear Operators in Hilbert Space*. Pitman, 1981, vol. I.

[24] M. Bertero, C. De Mol, and E. Pike, “Linear inverse problems with discrete data. I: General formulation and singular system analysis,” *Inverse Problems*, vol. 1, pp. 301–330, 1985.

[25] G. Wahba, *Spline Models for Observational Data*, ser. Series in Applied Mathematics. Philadelphia: SIAM, 1990, vol. 50.

[26] B. Schölkopf and A. Smola, "Kernel methods and support vector machines," in *Encyclopedia of Biostatistics*, P. Armitage and T. Colton, Eds. John Wiley and Sons, 2003.

[27] B. Schölkopf, C. Burges, and A. Smola, *Advances in Kernel Methods:Support Vector Learning*. The MIT Press, 1999.

[28] C. Groetsch, *Generalized Inverses of Linear Operators*, ser. Monographs and Textbooks in Pure and Applied Mathematics. Marcel Dekker, 1977.

[29] M. Bertero, "Regularization methods for linear inverse problems," in *Inverse Problems*, ser. Lecture Notes in Mathematics. Springer Berlin, 1986, vol. 1225, pp. 52–112.

[30] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, ser. Scripta Series in Mathematics. John Wiley & Sons, 1977.

[31] T. Dodd and R. Harrison, "The method of successive approximations for reproducing kernel Hilbert spaces," Department of Automatic Control and Systems Engineering, University of Sheffield, UK, Tech. Rep. 805, 2001.

[32] ——, "Steepest descent for generalised and regularised solution of linear operator equations," Department of Automatic Control and Systems Engineering, University of Sheffield, UK, Tech. Rep. 825, 2002.

[33] S. Phonphitakchai, "Convergence and adaptation in online kernel methods," Ph.D. dissertation, The University of Sheffield, 2008.

[34] J. Partington, *Interpolation, Identification, and Sampling*, ser. London Mathematical Society Monographs New Series. Clarendon Press, 1997, vol. 17.

[35] S. Vijayakumar and H. Ogawa, "RKHS-based functional analysis for exact incremental learning," *Neurocomputing*, vol. 29, pp. 85–113, 1999.