ICIC Express Letters
Volume **6**, Number **3**, March **2012**

ICIC International ©2012 ISSN 1881-803X
pp. 705–710

INVESTIGATION ON DIFFERENT LEARNING TECHNIQUES FOR WEIGHTED KERNEL REGRESSION IN SOLVING SMALL SAMPLE PROBLEM

MOHD IBRAHIM SHAPIAI[1], ZUWAIRIE IBRAHIM[1], SHAHDAN SUDIN[1]
MOHD SABERI MOHAMAD[2] AND MARZUKI KHALID[1]

[1]*Centre of Artificial Intelligent and Robotics*
Faculty of Electrical Engineering
[2]Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
Skudai 81310, Malaysia
{ ibrahim; zuwairie; shahdan }@fke.utm.my; { marzuki; saberi }@utm.my

Received June 2011; accepted September 2011

ABSTRACT. *Previously, weighted kernel regression (WKR) for solving small sample problems has been reported. The proposed WKR has been successfully employed to solve rational functions with very few samples. The design and development of WKR is important in order to extend the capability of the technique with various learning techniques. Based on WKR, a simple iteration technique is employed to estimate the weight parameters before WKR can be used in predicting the unseen test samples. In this paper, we investigate two learning techniques in estimating the weight parameters. For this purpose, a Ridge Regression (RR) and a guided search based on Particle Swarm Optimization (PSO) are used to investigate the capability of WKR in solving small sample problems. It is found that RR and PSO are better than iteration technique in terms of computational time and flexibility of defining the objective function to estimate weight parameters, respectively, without sacrificing the quality of prediction, as supported by the conducted experiments.*
Keywords: Weighted kernel regression (WKR), Ridge regression (RR), Particle swarm optimization (PSO), Small samples

1. **Introduction.** In general, the kernel based regression aims at regressing the unknown function based on the available training samples. In real world applications, to obtain sufficient training samples is too expensive when dangerous real measurements have to be performed [1]. The application of learning from small samples has gained increasing attention in many fields, such as in semiconductor manufacturing [2], and engine control simulation [3]. There are numerous techniques [4] in machine learning for regression. However, all the available techniques mainly focus on solving sufficient training samples problem. As most existing techniques perform well under sufficiently large training samples, the performance of those techniques degrades as the size of samples decreases.

WKR has proved to solve small samples with high accuracy for theoretical functions [5] and application in semiconductor problem [2]. Basically, WKR framework is based on the Nadaraya-Watson Kernel Regression (NWKR). To design a WKR, one must estimate the weight parameters, W, before it can be used to predict unseen samples. In the existing WKR, the parameter estimation is simply based on the primitive iteration technique. In this study, we focus on introducing alternative estimation techniques for WKR in order to extend and to investigate the capability of the WKR in solving small sample problems.

In this study, the capability of Ridge Regression (RR) and Particle Swarm Optimization (PSO) is investigated in estimating the weight parameters. RR is initially introduced to address the numerical instability of the matrix inversion and to ensure a lower variances model. This technique adds a positive constant to the inverse matrix term to make the

matrix non-singular [6]. On the other hand, PSO is inspired by the social behavior of birds in nature [7]. PSO is a very popular choice to solve optimization problem, easy to implement, and computational efficient.

2. **Weighted Kernel Regression Review.** The concept of the WKR is introduced in the following. Given training samples, $\{x_i, y_i\}_{i=1}^{n}$, where n is the number of training samples, $x_i \in \Re^d$ is the input and $y_i \in \Re$ is the target output. WKR is the technique to regress the output space by mapping the input space $\Re^d$ to $\Re$. In general WKR is a modified Nadaraya-Watson kernel regression (NWKR) by expressing the weight based on the observed samples through a kernel function. The existing WKR relies on the Gaussian kernel function as given in Equation (1).

$$K(X, X_i) = \frac{1}{\sqrt{2\pi}} \exp \frac{\left(-\|X - X_i\|^2\right)}{h} \tag{1}$$

where h is the smoothing parameter. As in NWKR, the selection of smoothing parameter, h, is important to compromise between smoothness and fitness [8]. As in existing WKR, Equation (2) is employed to determine the value of h.

$$h = \max\left(\|X_{k+1}\|^2 - \|X_k\|^2\right) \text{ where } 1 < k < n-1 \text{ and } \|X_{k+1}\|^2 > \|X_k\|^2 \tag{2}$$

The kernel matrix $K = [K_{ij}]$, where $i = j = 1, \ldots, n$, with a generalised kernel matrix based on the Gaussian kernel is given in Equation (3). The matrix K transforms the linear observed samples to non-linear problems by mapping the data into a higher dimensional feature space.

$$K_{ij} = \begin{cases} \dfrac{\prod_{p=1}^{d} K\left(X_i^p, X_j^p\right)}{\sum_{l=1}^{n}\left[\prod_{p=1}^{d} K\left(X_{i \vee j}^p, X_j^p\right)\right]} & i \neq j \\ \dfrac{1}{\sum_{l=1}^{n}\left[\prod_{p=1}^{d} K\left(X_{i \vee j}^p, X_j^p\right)\right]} & i = j \end{cases} \tag{3}$$

In WKR, the most popular function for regression problems is used to minimize the sum of squared error (SSE) in order to estimate the weight parameters, W.

$$\min f(W) \Leftrightarrow \min \|Kw - y\|^2 \tag{4}$$

Once the optimum weight is estimated, the model is ready to predict any unseen samples (test samples). The test samples can be predicted by using Equation (5).

$$\hat{y}\left(X, \hat{W}\right) = \frac{\sum_{i=1}^{n} \hat{w}_i \left(\prod_{p=1}^{d} K\left(X^p, X_i^p\right)\right)}{\sum_{i=1}^{n}\left(\prod_{p=1}^{d} K\left(X^p, X_i^p\right)\right)} \tag{5}$$

3. **Investigated Learning Techniques.** In this study, the investigation to estimate weight parameters will be based on the RR and PSO, which are subjected to the inverse matrix solution and guided search on the problem space, respectively.

3.1. **Ridge regression.** Ridge regression extends Equation (4) by adding the L_2 regularization term in order to avoid the singular matrix problem. This is to ensure a lower variance model by compromising between solving the equation and at the same time keeping the w small. In this investigation, function to be minimized is given in Equation (6).

$$f_{reg}(W) = \|Kw - y\|^2 + \lambda \|w\|^2 \tag{6}$$

where λ is a positive constant value. Differentiating Equation (6) with respect to w gives the closed form solution in estimating the weight parameter as given in Equation (7).

$$W = \left(K^T K + \lambda I\right)^{-1} K^T y \tag{7}$$

In this investigation, only small λ value is used as to avoid multicollinearity effect from the kernel matrix as the given training samples are small and true samples.

3.2. **Particle swarm optimization.** In PSO algorithm, every particle represents the possible solution in the problem space. Each particle flies over d-dimensional problem space for searching the optimum solution by updating its own velocity, $v_{i,d}(t)$, and position, $p_{i,d}(t)$ with respect to the fitness function. The current velocity of each particle is updated based on the personal best previous position by every i^{th} particle, $p_{i,d}^{pbest}$, and the global best previous position found so far by the swarm, p_d^{gbest}. The d^{th} dimensional of the velocity and position for i^{th} particle is updated using Equation (8) and Equation (9), respectively.

$$v_{i,d}(t+1) = kv_{i,d}(t) + c_1 r_1 \left(p_{i,d}^{pbest} - p_{i,d}(t)\right) + c_2 r_2 \left(p_d^{gbest} - p_{i,d}(t)\right) \tag{8}$$

$$p_{i,d}(t+1) = p_{i,d}(t) + v_{i,d}(t+1) \tag{9}$$

where t is the iteration value, c_1 and c_2 are the cognitive and social coefficients, r_1 and r_2 are random values in the range $[0, 1]$ and k is the inertia weight. The cognitive and social coefficients control the tendency of particles to move toward its own or the entire particles position. The random values provide randomness exploitation for particle in the problem space. Meanwhile, the inertia weight controls the exploration of particle in finding the optimum solution. In PSO, the inertia weight is decreased overtime with typically large initial value and the equation is given as follows:

$$k = k_{init} - \frac{k_{init} - k_{final}}{iteration} \times iteration_t \tag{10}$$

where k_{init} and k_{final} are the predefined initial and final value of the inertia weight respectively, *iteration* is the maximum number of iteration and $iteration_t$ is a current iteration. In this investigation, Equation (4) is used as the fitness function in estimating the weight parameters. The value of k_{init} is purposely chosen to be large and the k_{final} value is chosen not to be very small as each particle is allowed to explore in wider problem space. In general, the weight parameter to be estimated corresponds to the found value of p_d^{gbest}.

4. Experimental Results and Discussion.

4.1. **Setup experiment.** Several experiments were performed to validate the quality of every parameter estimation techniques. Three functions, which are defined as Test 1, Test 2 and Test 3, are given in Equations (11)-(13), respectively. These functions, which are taken from [1], are used for the validation purpose.

$$y = x^2, \quad x \in [0, 1] \tag{11}$$

$$y = 0.01x + 0.02x^2 + 0.9x^3, \quad x \in [0, 1] \tag{12}$$

$$y = 1 - \exp\left(-2x^4\right), \quad x \in [0, 1] \tag{13}$$

The given training samples are on the intervals of $0 \leq x \leq 1$. The experiment is repeated ten times, where in each run only ten randomly generated samples are used for training. Equation (14) is used to evaluate the performance for all test functions.

$$MSE = \frac{1}{l} \sum \left(f_{true} - f_{predict}(W)\right)^2 \tag{14}$$

where l is the number of the test data, f_{true} is the true value of the tested function and $f_{predict}(W)$ is the predicted value. A grid of 101 test samples ($l = 101$) is generated in the interval $[0, 1]$. The parameter settings for all investigation are summarised in Table 1.

TABLE 1. Parameter settings for every investigated learning technique

Technique	Parameter Settings
PSO	swarm size = 100, iteration = 500, $c_1 = c_2 = 1.4$, $k_{init} = 2.5$, $k_{final} = 0.4$
RR	λ = 1e-10
Iteration	iteration = 10000 or MSE < 1e-10

4.2. **Results.** The performances and the computational times are tabulated in Tables 2-4. The performances indices for all test functions are measured based on the MSE of ten experiments and the recorded computational time is the average time of ten experiments. The quality regression for all techniques is comparable where PSO recorded the highest MSE for all tests. Also, PSO has small uncertainty in finding the final solution as it is categorized as stochastic optimizer. The uncertainty can be traced through the fluctuation in MSE curve of PSO technique as shown in Figure 1. However, for other techniques the average MSE gradually decrease as the number of training samples is increased. However, PSO is able to estimate the weight parameters in non-closed form solution. It is also important to highlight that introducing the L_2 term in solving Equation (4) avoids the multicollinearity effect. The computational of RR is faster compared with PSO.

TABLE 2. Results of ten experiments to predict Test 1 function

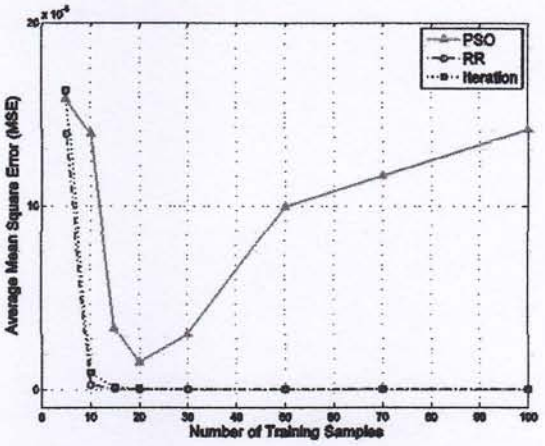
Technique	Average	Standard Deviation	Minimum	Maximum	Computation Time (s)
PSO	1.39E-04	3.18E-04	2.17E-07	1.15E-03	19.73
RR	2.44E-06	8.34E-06	2.08E-09	3.25E-05	4.37
Iteration [5]	9.17E-06	2.84E-05	1.50E-08	1.11E-04	6.02

TABLE 3. Results of ten experiments to predict Test 2 function

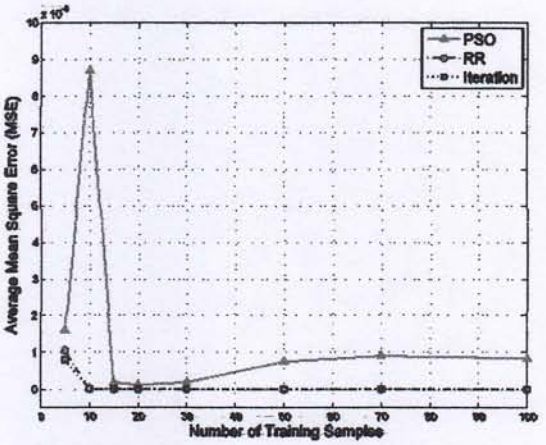
Technique	Average	Standard Deviation	Minimum	Maximum	Computation Time (s)
PSO	8.70E-06	3.06E-05	1.28E-09	1.19E-04	19.80
RR	2.71E-09	4.81E-09	5.92E-12	1.49E-08	4.32
Iteration [5]	7.81E-09	1.37E-08	4.99E-11	3.94E-08	5.98

TABLE 4. Results of ten experiments to predict Test 3 function

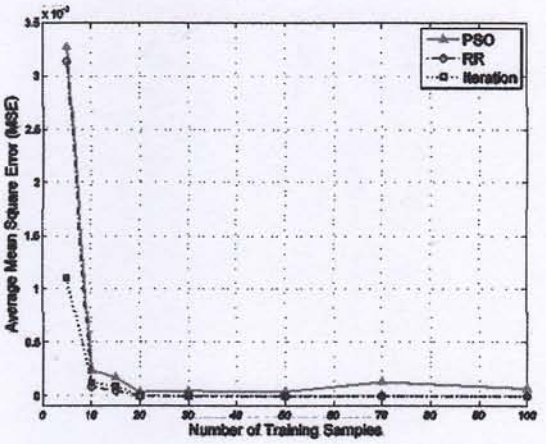
Technique	Average	Standard Deviation	Minimum	Maximum	Computation Time (s)
PSO	2.42E-04	3.82E-04	1.24E-05	1.56E-03	20.08
RR	8.49E-05	3.13E-04	6.15E-08	1.22E-03	4.52
Iteration [5]	1.26E-04	3.82E-04	1.05E-07	1.50E-03	6.27



(a)



(b)



(c)

FIGURE 1. The improved performance with increased training samples for each technique except for PSO. MSE curve for Test 1 (a), Test 2 (b), and Test 3 (c).

5. **Conclusions.** In this study, we investigate different types of learning techniques for WKR in solving small sample problems. RR and PSO are used for the investigation to compare with the iteration technique. Three experiments are conducted to show the effectiveness and practicability of WKR when using two different learning techniques. It is found that, all the investigated techniques give a comparable prediction quality in terms of the MSE value.

Although PSO is capable of solving non-closed form solution problem, it requires longer computational time and a small uncertainty exists in estimating the weight parameter. Meanwhile, RR is found to be fastest technique in estimating the weight parameter but it will fail to solve non-differentiable problem. As PSO is capable of solving non-closed form solution, the additional future work will include the investigation of different loss functions to be associated with WKR.

Acknowledgment. This work is financially supported by UTM-Intel Research Grant (VOTE 73332), Ministry of Higher Education Malaysia (MOHE) through Fundamental Research Grant Scheme (FRGS) (VOTE 78564), and UTM GUP Research Funds (VOTE Q.J130000.7107.01H29).

REFERENCES

[1] C. Huang and C. Moraga, A diffusion-neural-network for learning from small samples, *International Journal of Approximate Reasoning*, vol.35, pp.137-161, 2004.

[2] M. I. Shapiai et al., Recipe generation from small samples by weighted kernel regression, *International Conference on Modeling Simulation and Applied Optimization*, Kuala Lumpur, Malaysia, pp.1-4, 2011.

[3] G. Bloch et al., Support vector regression from simulation data and few experimental samples, *Information Sciences*, vol.178, pp.3813-3827, 2008.

[4] T. Su, J. Jhang and C. Hou, A hybrid artificial neural networks and particle swarm optimization for function approximation, *International Journal of Innovative Computing, Information and Control*, vol.4, no.9, pp.2363-2374, 2008.

[5] M. I. Shapiai, Z. Ibrahim, M. Khalid, L. W. Jau, V. Pavlovic and J. Watada, Function and surface approximation based on enhanced kernel regression for small sample sets, *International Journal of Innovative Computing, Information and Control*, vol.7, no.10, pp.5947-5960, 2011.

[6] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ Press, New York, NY, USA, 2004.

[7] J. Kennedy and R. Eberhart, Particle swarm optimization, *International Conference on Neural Networks*, Piscataway, New Jersey, pp.1942-1948, 1995.

[8] J. Zhang et al., An improved kernel regression method based on Taylor expansion, *Applied Mathematics and Computation*, vol.193, pp.419-429, 2007.