A COMPLETE INVESTIGATION OF USING WEIGHTED KERNEL REGRESSION FOR THE CASE OF SMALL SAMPLE PROBLEM WITH NOISE

Zuwairie Ibrahim[1], Mohd Ibrahim Shapiai[2], Siti Nurzulaikha Satiman[1], Mohd Saberi Mohamad[3], and Nurul Wahidah Arshad[1]
[1]Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, Pekan, Malaysia
[2]Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
[3]Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
E-Mail: zuwairie@ump.edu.my

ABSTRACT

Weighted kernel regression (WKR) is a kernel-based regression approach for small sample problems. Previously, for the case of small sample problems with noise, we have done preliminary studies which investigated different learning techniques and different learning functions, separately. In this paper, a complete investigation of using WKR for the case of noisy and small training samples is presented. Analysis and discussion are provided in detail.

Keywords: weighted kernel regression, small sample problem, noise, genetic algorithm, ridge regression, and LOOCV.

INTRODUCTION

There are numerous algorithms for regression problems, which perform well, provided the number of training samples is sufficient. However, the performance of those algorithms degrades as the size of samples decreases.

Weighted kernel regression (WKR) has been introduced [1] to solve small sample regression problems. The WKR is based on Nadaraya-Watson kernel regression (NWKR). To do regression using WKR, one must estimate the weight parameters, *W*, before it can be used to predict unseen samples. The estimation of the weight parameters depends on the learning functions and learning techniques.

Even though WKR has a good ability to perform accurate regression for the case of small training samples, its performance degrades when noisy training samples are considered. To extend the capability of WKR when noisy and small training samples, previously, we have investigated different learning techniques [2] and different learning functions [3], separately. In this paper, those preliminary studies are combined in order to have a complete algorithm of using WKR for the case noisy and small training samples. To obtain noisy sample, Gaussian noise is added to the training samples as illustrated in Figure-1. In this figure, it is shown that the entire noisy training samples deviate from the trajectory of the true function, which make accurate regression difficult to obtain.

WEIGHTED KERNEL REGRESSION

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 $\Re$. The existing WKR relies on the Gaussian kernel function as given in Equation (1).

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

where *h* is the smoothing parameter. The selection of smoothing parameter, *h*, is important to compromise between smoothness and fitness [2]. The Equation (2) is employed to determine the value of *h*.

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

The kernel matrix *K* = [*K*$_{ij}$], where *i* = *j* = 1,..., *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.

$$K_{ij}=\begin{cases}\dfrac{\prod_{p=1}^{d}K\left(X_i^p,X_j^p\right)}{\sum_{l=1}^{n}\left[\prod_{p=1}^{d}K\left(X_{i\vee j}^p,X_j^p\right)\right]} & i\neq j\\[2ex]\dfrac{1}{\sum_{l=1}^{n}\left[\prod_{p=1}^{d}K\left(X_{i\vee j}^p,X_j^p\right)\right]} & i=j\end{cases} \tag{3}$$

In WKR, the most popular function for regression problems is used which to minimize the RSS to estimate the weight parameters, *W*, as follows:

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

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).

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

EXTENSION OF WKR – LEARNING FUNCTIONS

Minimizing the error term only may lead to numerical instabilities and bad generalization performance. The instability yields a high variance model which potentially produces large differences of weight parameter values given different training samples, even minor perturbation of the same training samples. In general, this instability can be addressed by restricting the class of permissible solution by introducing the regularization term to the formulated learning function. Therefore, the learning functions should comprise not only the error term but also the regularization term as given in Equation (6).

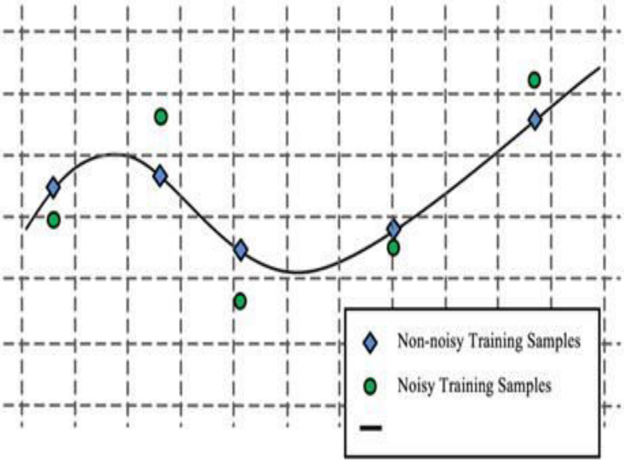


Figure-1. Illustration of non-noisy and noisy training samples. Note that for the case of noisy training samples, Gaussian noise has been added.

$$\text{learning function} = \text{error term} + \text{regularization term} \qquad (6)$$

The addition of regularization term is to avoid the magnitude of estimated weight parameters to be very large, thereby avoiding over-fitting of the regression. Hence, the addition of regularization term gives advantages to the regression quality. In general, the error term and regularization term can be formulated either with L$_1$-norm or L$_2$-norm function. Therefore, we formulated the four learning functions with combination of L$_1$-norm and L$_2$-norm as error term and regularization term based on the WKR concept.

In general, the formulated learning functions can be categorized into two types; (1) closed form solution function and (2) non-closed form solution function. Closed form solution function can be derived analytically as compared to non-closed form solution function when estimating the weight parameters. For non-closed form function, there is no analytical solution can be obtained as the function is non-differentiable. As evolutionary computing offers an effective way to estimate the weight parameters for non-differentiable function, we employ GA as a learning technique for non-differentiable function. The formulated learning function with L$_1$-norm term either in error term or regularization term is considered as non-closed form solution function. The formulated learning functions which based on learning function in the existing WKR are given in Equation (7) to Equation (10).

$$f_{L_2R_2}(W) = \arg\min_W \left(\|KW - Y\|^2 + \lambda \|W\|^2 \right) \qquad (7)$$

$$f_{L_2R_1}(W) = \arg\min_W \left(\|KW - Y\|^2 + \lambda \|W\|_1 \right) \qquad (8)$$

$$f_{L_1R_2}(W) = \arg\min_W \left(\|KW - Y\|_1 + \lambda \|W\|^2 \right) \qquad (9)$$

$$f_{L_1R_1}(W) = \arg\min_W \left(\|KW - Y\|_1 + \lambda \|W\|_1 \right) \qquad (10)$$

where K is kernel matrix, W is weight parameter to be estimated, Y is observed output domain values, $\|.\|_1$ is L$_1$-norm function, $\|.\|^2$ L$_2$-norm function, and λ is a free parameter that control the generalization of the regressed function.

Cross-validation is a technique to evaluate model in order to generalize the predictive performance when predicting unseen samples. The need of cross-validation is important in model selection as some model's parameters has to be estimated. In general, cross-validation separates the available training samples into two sets, called the training set and validation set. Training set is used to build the model and validation set is used to evaluate the model based on the selected model's parameter with respect to the cross-validation error. Typically, the cross-validation error is measured based on MSE performance criterion. The model with the lowest cross-validation error is then used as a final model which possibly offers a better generalization performance.

There are various cross-validation techniques available in literatures such as hold-out method, K-fold cross-validation and leave-one-out cross-validation (LOOCV). In general, LOOCV is very expensive to compute but it is able to retrieve a lot of information from the available training samples [5]. As the focus of the study is to solve small and limited training samples problem, LOOCV is found to offer several advantages in terms of information retrieval and computational time. In general, LOOCV separates the available training samples into a training set of size -1 and a validation of size 1. For every selected model's parameter, there are different combinations of training and validation set. The lowest cross-validation on the validation set is used as an indicator to select the final model.

However, since the number of weight parameters in WKR is determined based on number of available training samples, the found model from LOOCV cannot be used in the final model as there are only n-1 weight parameters. LOOCV is introduced only to determine the λ value in regularization term of the employed learning function for final model. Hence, the h value is kept to be the same either for λ estimation or for used in the final model. It is to ensure the relationship information of the available training samples is retained when estimating λ value as the estimated λ value will be used the final model. As mentioned previously, the estimated λ value is chosen after the evaluation phase based on the lowest cross-validation error from several initialized λ values. The same employed learning function and learning technique in estimating the λ value will be used in the final model. The estimated λ value is derived based on the Equation (11).

$$\hat{\lambda} = \arg\min_{|\lambda|}\left(\frac{1}{n}\sum_{i=1}^{n} f_{-i}(W_{-i};\lambda) - y_{-i}\right)^2 \qquad (11)$$

where $f_{-i}(W_{-i};\lambda) = K_{-i}W_{-i}$, $\hat{\lambda}$ is the estimated λ value to be used in final model, $|\lambda|$ refers to a set of initialized λ values, n refers to number of training samples, $K_{-i}, K_{-i} \in \Re^{1\times(n-1)}$ is the kernel matrix (row vector) of input space of ith left-out training set with the defined h, $W_{-i}, W_{-i} \in \Re^{(n-1)\times 1}$ is the estimated weight parameters of the corresponding training set with initialized λ value, and $y_{-i}, y_{-i} \in \Re$ is the output domain of ith left-out training set.

EXTENSION OF WKR – LEARNING TECHNIQUES

Previously, iteration, ridge regression (RR), and genetic algorithm (GA) have been investigated in using WKR for the case of noise and small training samples [2]. The RR only can be used to solve closed-form solution function by differentiating Equation. (7) with respect to W. Prior to the differentiation, Equation. (7) is expanded to be Equation (12).

$$f_{L_2R_2}(W) = W^TK^TKW - 2Y^TKW + Y^TY + \lambda W^TW \qquad (12)$$

The analytic solution for the estimated weight parameter is given in Equation (13).

Table-1. Setting parameters.

Genetic algorithm	
Generation	300
Population size	100
Probability of cross-over	0.7
Probability of mutation	0.001
Ridge regression	
λ	1e-10
Iteration	
Iteration	10000

$$\begin{aligned}\frac{\partial f_{L_2R_2}(W)}{\partial W} &= 0 \Rightarrow 2K^TKW - 2K^TY + 0 + 2\lambda W = 0\\ &\Rightarrow \left(\mathrm{K}^TK + \lambda I\right)W = K^TY\\ &\Rightarrow \hat{\mathrm{W}} = \left(\mathrm{K}^TK + \lambda I\right)^{-1}K^TY\end{aligned} \qquad (13)$$

where I is an identity matrix of size $n\times n$ and $\widehat{W}$ is the estimated weight parameter. Examining Equation (13), the addition of L$_2$-norm regularization term is simply adds a positive constant to the diagonals of K^TK, to make the matrix non-singular.

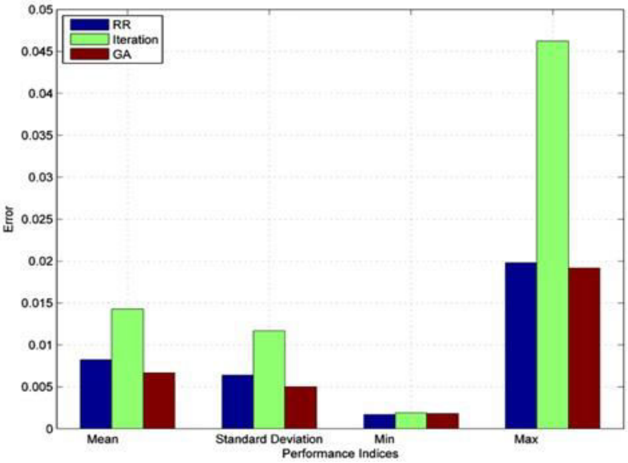


Figure-2. Graphs show the result of 100 experiments to predict $v = u^2$ with n=5.

GA was also employed in estimating weight parameter. As all the formulated learning functions are categorized as continuous search space, therefore, real-coded GA [5] was employed in finding the optimal solution.

EXPERIMENTS AND RESULTS

The experiments were carried out in two sub-problems; (1) investigation on learning techniques and (2) investigation on formulated learning functions. Prior to these two investigations, free parameter, λ has firstly to be estimated using LOOCV and a grid of 101 test samples (l = 101) is generated in the interval [0,1]. In this study, three non-linear functions as formulated in Equation (14), Equation (15), and Equation (16), were used.

$$\text{Test 1}: y = x^2, \quad x \in [0,1] \qquad (14)$$

$$\text{Test 2}: y = 0.01x + 0.02x^2 + 0.9x^3, \quad x \in [0,1] \qquad (15)$$

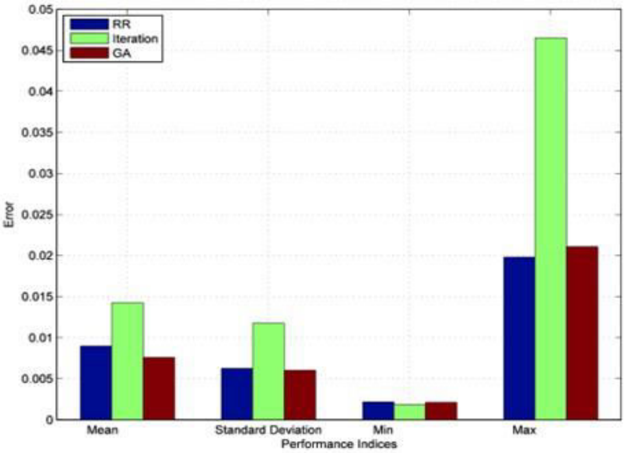


Figure-3. Graphs show the result of 100 experiments to predict v = 0.01u + 0.02u^2 + 0.9u^3 with n=5.

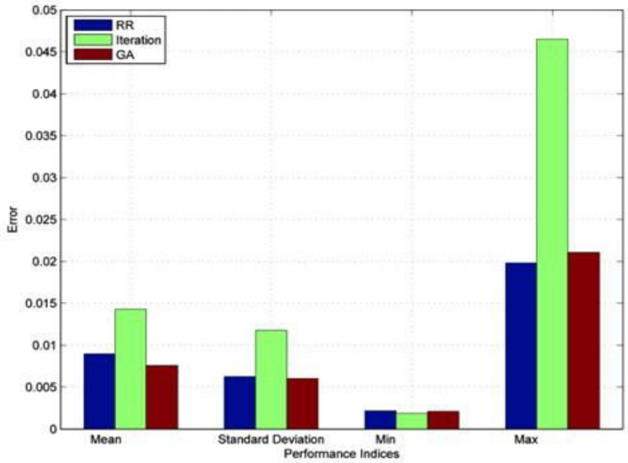
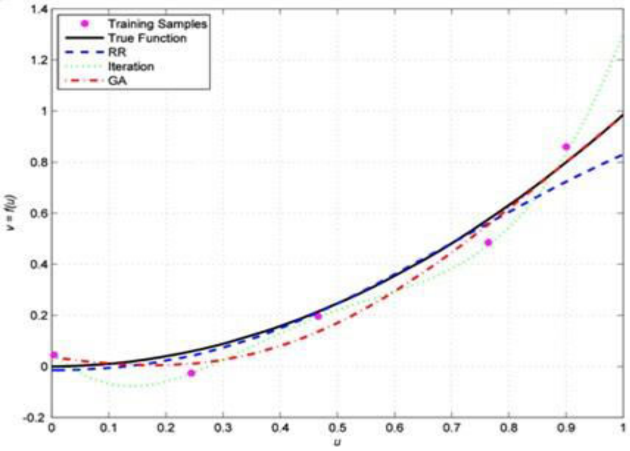


Figure-4. Graphs show the result of 100 experiments to predict v = 1-exp (-2u^4) with n= 5.

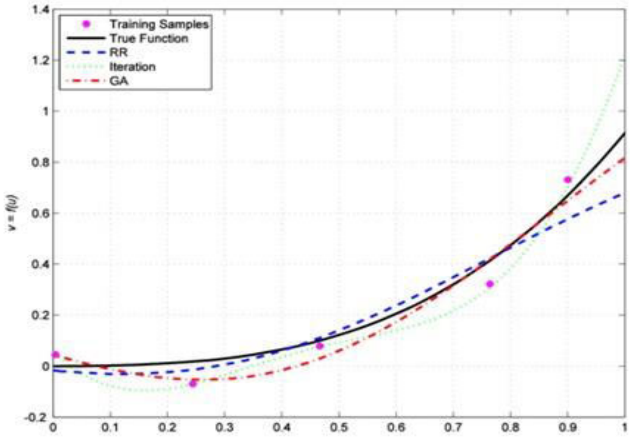
$$\text{Test 3: } y = 1-\exp\left(-2x^4\right), \quad x \in [0,1] \qquad (16)$$

where the output space, y, is contaminated with Gaussian noise $N \sim (0,0.1)$. Initially, all the parameter settings for every investigated learning technique were predefined except for RR technique as it is based on analytical solution which derived from inversion of matrix. The parameter settings are summarized in Table-1. The experiment was repeated 100 times for every problem. In each run only five randomly generated training samples, S, were used. The input space, X, is pre-defined in the range of domain value with the corresponding output space, y. Performance criterion based on MSE was used to validate the quality of prediction. To investigate the best learning technique, we limit the investigation to only one formulated learning function which is given in Equation (7) as it can be solved analytically.

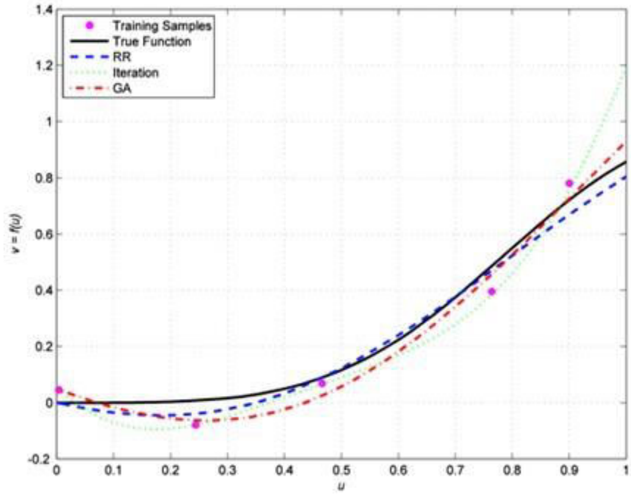
The quality of prediction for every technique for three different problems is shown in Figure-2, Figure-3, and Figure-4. In general, GA offers the lowest average MSE for all problems and the existing WKR with iteration technique fails to capture a good regression curve in all problems by recording the highest average MSE for all problems.



(a)



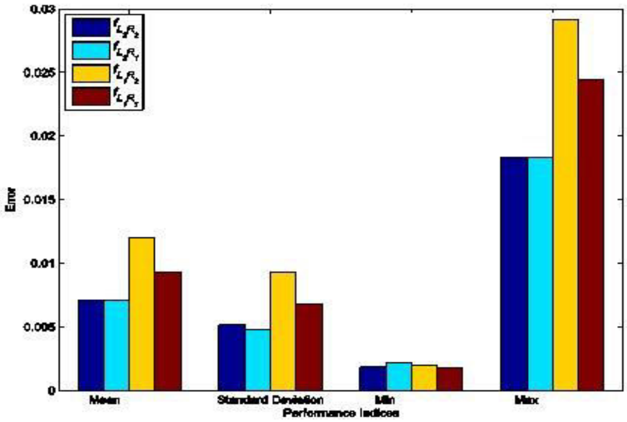
(b)



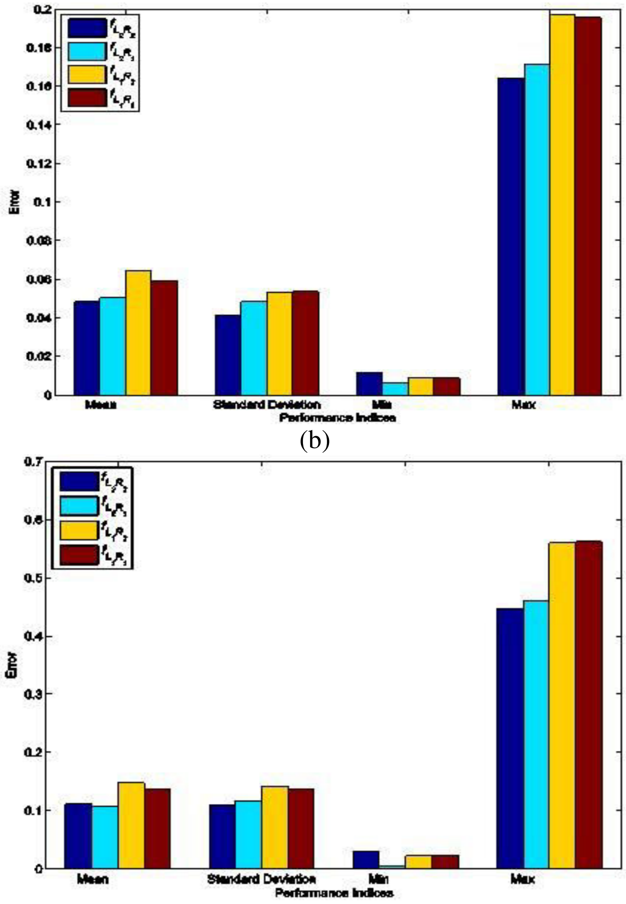
(c)

Figure-5. Quality of prediction (a) Type 1 (b) Type 2 (c) Type 3.

We also demonstrate one example of regression quality for all non-linear function types for all techniques in Figure-5. The plotting results show that the existing WKR is trapped into over-fitting problem in all problems. Meanwhile, the other techniques at least can capture the trajectory of the non-linearity of the curve in all problems.



(a)



(b)

(c)

Figure-6. Graphs show the result of 100 experiments to predict v = 1-exp(-2u^4) with n= 5 and contaminated by Gaussian noise (a) N~(0,0.1) (b) N~(0,0.3), and (c) N~(0,0.5).

To investigate the best learning function, Equation (7) to Equation (10) are considered. The learning function with L_1-norm term is non-differentiable function. Hence, an analytic form solution when minimizing the corresponding learning function in estimating the weight parameter cannot be obtained. This drawback has led to the use of GA.

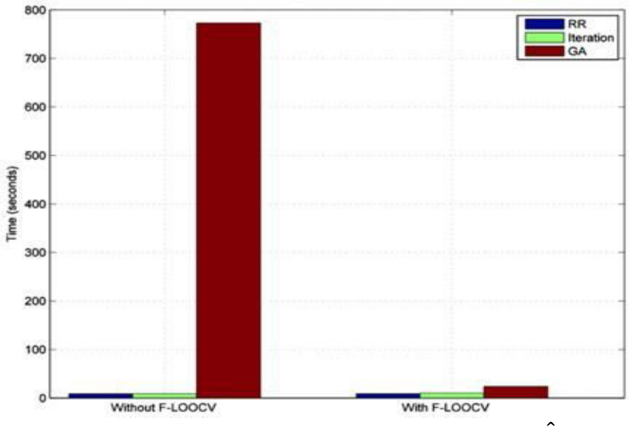


Figure-7. Computational time without estimated $\hat{\lambda}$ (left) and with estimated $\hat{\lambda}$ (right).

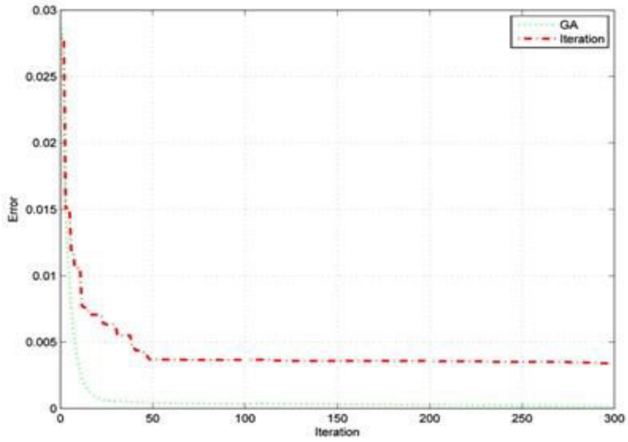


Figure-8. Example of convergence curve for Type 1 function with noise at 0.1.

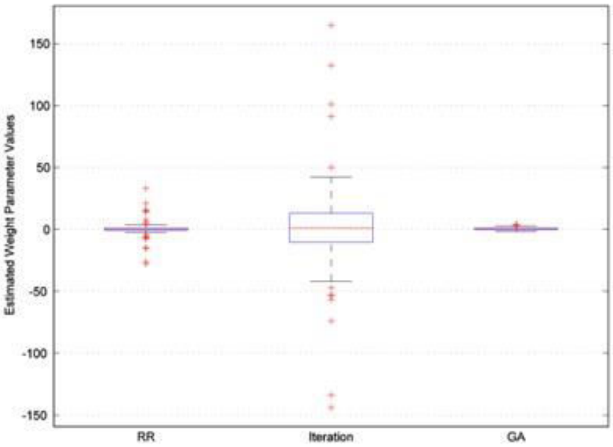


Figure-9. Distribution of weight parameter for the three learning techniques in solving Type 1 function with noise at 0.1.

DISCUSSION

Generally, the results of the investigation show that the capability of WKR can be improved with the investigated learning techniques and formulated learning functions. The quality of the predictions for all investigations for all problems is significantly improved as compared to the existing WKR.

There are three main features to be considered when selecting the learning technique; (1) simplicity, (2) computational time and (3) flexibility. As the name implies, simplicity relates to how easy the technique in estimating the weight parameter. Computational time is a measurement of how fast the technique estimates the weight parameters in time. Finally, the term flexibility refers to the capability of the technique in solving closed-form solution and non-closed form solution functions.

The iteration technique can be considered as simple technique with fast computational time. RR has fast computational time and slightly flexible as it only limits to solve closed-form solution function. Meanwhile, GA offers a good flexibility as it can solve closed-form solution function and non-closed form solution function with longer computational time. The recorded computational times of the three techniques in regressing Type 1 function with noise at 0.1 are shown in Figure-7.

The bar graph on the right side shows the time consumes by the three weight estimation techniques in finding the model by assuming the value is available. It shows GA slightly requires longer computational time as compared to the RR and iteration technique. Meanwhile, the bar graph on the left side refers to the computational time of the three techniques in finding the final model without having value. Therefore, in finding the final model has first to be estimated. In practice, longer computational time is required for GA as is not possibly available. It is shown that GA is absolutely requires longer time as compared to the RR and iteration technique. However, the computational time for GA and iteration technique completely depends on the iteration number, population size, and stopping condition as compared to the RR technique regardless to the size of the training samples.
We also show the convergence curve at training phase for GA and iteration technique in estimating the weight parameters, as shown in Figure-8. As iteration technique does not offer a mechanism to minimize the regularization term, therefore, the convergence curve of the iteration technique reaches to very small error.

Minimizing the error term only will cause large variance model. Implicitly, the large variance model refers to large magnitude of weight parameters values. The regularization term can avoid this type of problem by compromising between minimizing the error term and minimizing the magnitude of weight parameter values. In Figure. 9, the distribution of weight parameters value of the three techniques when regressing Type 1 with noise level at 0.1 is shown. In general, iteration technique has the largest variance model and GA has the lowest variance model. As iteration technique does not provide a mechanism to minimize the regularization term, the estimated weight parameter value varies from very small negative value to very large positive value as compared to RR and GA. The implication of large variances of the estimated weight parameter values for iteration technique is the over-fitting problem. Meanwhile, the lowest variance model of GA produces the lowest average MSE as compared to RR and iteration technique. However, the addition of regularization term may also cause an under-fitting problem when the average of estimated weight parameter value is too small.

In general, the quality prediction of learning function with L2-norm error term is better than L1-norm error term. The learning function with L2-norm error term and L1-norm regularization term gives a good quality of prediction for noise level at 0.1 and 0.5. Meanwhile, learning function with L2-norm error term and L2-norm regularization term gives a good quality of prediction for noise level at 0.3. However, the difference of the quality prediction error between the two above-mentioned learning functions is small.

In Figure-10, the distribution of weight parameter values for all investigated learning functions when regressing Type 3 function with noise level at 0.1 is shown. The learning function with L1-norm regularization term shows sparseness solution as the median value of the distribution centralized at value close to zero as compared to learning function with L2-norm regularization term. The sparseness solution offers slightly smaller model as the estimated weight parameter value close to zero. However, we found that this feature is not substantially important when addressing small training samples problem.

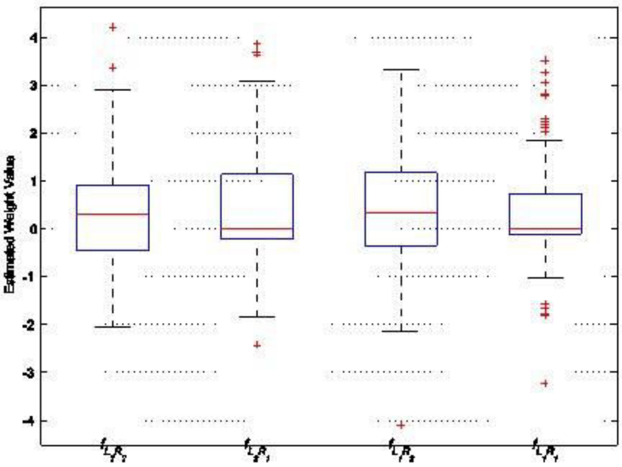


Figure-10. Distribution of weight parameter values for the learning functions in solving Type 3 function with noise at 0.1.

CONCLUSIONS

A complete extension of WKR is proposed to address regression problem with noisy training samples. The investigation branches into two parts (1) investigation on weight parameters techniques and (2) investigation on formulated learning functions. Prior to these two investigations, the free parameter value has firstly to be estimated. The improvement, in terms of quality of prediction is experimented and presented. We found that in general, GA is flexible but requires more time to obtain weight parameters. In addition, for learning functions, the L_2-norm error term is better than L_1-norm error term.

ACKNOWLEDGEMENTS

This work is financially supported by the RAGS (RDU131416) awarded by the Ministry of Higher Education (MOHE) to Universiti Malaysia Pahang.

REFERENCES

[1] Shapiai M. I., Ibrahim Z., Khalid M., Lee W. J., Pavlovic V. and Watada J. 2012. 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.

[2] Shapiai M. I., Sudin S., Arshad N. W., and Ibrahim Z. 2015. Investigation on different learning techniques for weighted kernel regression in solving small sample problem with noise. ICIC Express Letters. Vol. 9. No. 4, pp. 965-971.

[3] Ibrahim Z., Arshad N. W., Shapiai M. I., and Mokhtar N. Different learning functions for weighted kernel regression in solving small sample problem with noise. Proceedings of the 2015 International Conference on Artificial Life and Robotics, 2015.

[4] Moore A. W., Hill D. J., and Johnson M. P. 1992. An empirical investigation of brute force to choose features, smoothers and function approximators. Computational Learning Theory and Natural Learning Systems. Vol. 3, pp. 361-379.

[5] Goldberg D. E. 1989. Genetic algorithm in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc. Boston, USA.