Parameter Optimization for Nadaraya-Watson Kernel Regression Method with Small Samples

Li Fengping
College of Mechanical & Electrical Engineering
Wenzhou University
Wenzhou, China

Zhou Yuqing*
College of Mechanical & Electrical Engineering
Wenzhou University
Wenzhou, China

Xue Wei
College of Mechanical & Electrical Engineering
Wenzhou University
Wenzhou, China

***Abstract*—Many current regression algorithms have unsatisfactory prediction accuracy with small samples. To solve this problem, a regression algorithm based on Nadaraya-Watson kernel regression (NWKR) is proposed. The proposed method advocates parameter selection directly from the standard deviation of training data, optimized with leave-one-out cross-validation (LOO-CV). Good generalization performance of the proposed parameter selection is demonstrated empirically using small sample regression problems with Gaussian noise. The results show that proposed parameter optimization method is more robust and accurate than other methods for different noise levels and different sample sizes, and indicate the importance of Vapnik's ε-insensitive loss for regression problems with small samples.**

Keywords—small samples regression; Nadaraya-Watson kernel regression; parameter optimization; loss function; cross validation

I. Introduction

This template, modified in MS Word 2007 and saved as a "Word 97-2003 Document" for the PC, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

Regression is one of the most fundamental and useful statistical techniques and is widely used to model practical problems arising from such fields as economics, psychology, management, signal processing, product design and medicine. It helps to relate explanatory variable(s) with a response variable and build predictive models. Given a set of independent observations $D = \{(x_1,y_1),\ldots,(x_n,y_n)\}$ from a population (X,Y), where X and Y are called the explanatory variable(s) and the response variable respectively, we want to find a function $f(x)$, assumed to be smooth, such that $y_i = f(x_i) + \delta_i$, $(i = 1,2,\cdots,n)$, where δ_i are independent, identically distributed random noises, so that $E(\delta_i) = 0$ for each i. The function $f(x)$ is called a regression function of Y on X.

At present, there are many available regression analysis models. In general, these regression models can be divided into the classes of parametric regression models and nonparametric regression models (Wand & Jones, 1995). Parametric regression models can be specified by a finite number of parameters, which implies that the regression function $f(x)$ is known except for the values of the parameters. Linear regression models and polynomial regression models are typical of the parametric models usually applied. Parametric regression models have a distinct interpretation of the relationship between X and Y, but the choice of parametric model depends on the situation. Restricting $f(x)$ to belong to a parametric family means that $f(x)$ can sometimes be too rigid (Zhang, Huang, et al, 2007). Once a parametric family is chosen, the mathematical form is fixed regardless of whether it is appropriate in reality, which could result in incorrect conclusions in the regression analysis. Non-parametric regression is proposed to overcome the rigidity of parametric regression. It only assumes that the regression function belongs to a smooth family of functions, and offers a way of estimating the regression function without specifying a parametric model. When the regression function between X and Y is complex, it is hard to deal with the observations using a parametric model, while a nonparametric model can analyze such situations effectively.

In nonparametric regression, ANNs (artificial neural networks) and k-nearest neighbor are widely used, and have good performance in many applications (Maxwell & Stinchcombe, 1995; Su, Jing, et al, 2008; Cho, Ishida, et al, 2011; La, Guo, et al, 2012). However, these methods need sufficiently large samples. When the size of samples is insufficient the quality of the results can decrease. In real world applications, obtaining sufficient training samples is often too expensive when dangerous measurements or complex technical experiments have to be performed, such as fault diagnosis for expensive equipment(Huang & Moraga, 2004), semi-conductor manufacturing (Li, Wu, et al, 2006), engine control simulation (Andonie, 2009), and biological studies (Lee & Ong, 2010). Therefore, designing a regression approach that performs well with small samples is a significant problem. Support vector regression (SVR) is motivated by the growing popularity of support vector machines (SVM) for regression with small samples (Smola & Scholkopf, 2004; Chu & Keerthi, 2007; Bloch, 2008; Huang, Zheng, et al, 2009). However, the quality of SVR models depends on proper settings of the SVR hyperparameters, and the main issue for practitioners trying to apply SVR is determining these parameter values for a given data set. Cherkassky and Ma have proved an effective approach to selecting SVR parameters, based on noise variance estimation in the observed data (Cherkassky & Ma, 2004a). In practice, with small samples, the noise variance cannot be precisely estimated by any well-known approach (such as polynomial or *k*-nearest-neighbor regression). Nadaraya-Watson kernel regression (NWKR) is a nonparametric technique in statistics for estimating the conditional expectation of a random variable, and allows interpolation and approximation a little beyond the samples (Shapiai, Ibrahim, et al, 2010). However, there is no appropriate approach for the selection of its parameter. This paper describes a practical analytical approach to selecting the parameter for NWKR directly from training data. The practical validity of the proposed approach is demonstrated using synthetic data sets.

This paper is organized as follows. Section 2 gives a brief introduction to NWKR regression. Section 3 describes the proposed approach to selecting the NWKR parameter using cross-validation (CV). Section 4 describes experimental tests for regression problems with Gaussian noise; these tests indicate that the proposed approach provides better generalization performance than other approaches. Finally, a conclusion is given in Section 5.

II. Nadaraya-Watson Kernel Regression

Nadaraya-Watson kernel regression (NWKR) estimates the regression function $f(x)$ corresponding to any arbitrary x value using Eq. (1):

$$\hat{y} = f(x, D, h) = \frac{\sum_{i=1}^{n} y_i K_h(x, x_i)}{\sum_{j=1}^{n} K_h(x, x_j)} \tag{1}$$

where D denotes the training set, $K_h(x, x_i)$ denotes a kernel function which fulfills some properties and *h* is the bandwidth parameter of the kernel function. Several types of kernel functions are commonly used, such as the Gaussian, uniform, triangle and Epanechnikov functions.

According to Eq. (1), we can see that NWKR is a weighted average technique that matches the given samples using a kernel function as weighting values. This method allows accurate interpolation and approximation in the vicinity of training samples. Kernels assign weights to arbitrary samples based on their distance from the given samples.

In NWKR, a Gaussian kernel function is found to have a better prediction accuracy than the other kernel functions (Shapiai, Sudin, et al, 2011). The expression of the Gaussian kernel is as follows:

$$K_h(x, x') = \frac{1}{\sqrt{2\pi}} \exp(-\frac{(x - x')^2}{2h^2}) \tag{2}$$

However, several articles found that the performance of NWKR mainly depends on the choice of the bandwidth parameter *h* rather than the kernel function. Figure 1 shows the relationship between the bandwidth *h* and the root mean squared error (RMSE, shown in Eq. (10)). In Fig. 1, the bandwidth *h* distinctly influences RMSE. Choosing *h* based on experience may result in poor prediction accuracy, especially when knowledge of the bandwidth *h* is insufficient. Thus, finding the optimal value of *h* is crucial for the prediction quality of NWKR.

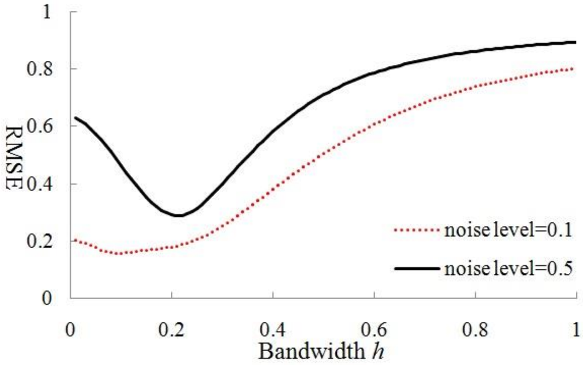

Fig. 1. Scatter diagram of bandwidth h and RMSE

(Note: The samples are generated from $y = f(x) + \delta$, and the target function $f(x)$ is shown in Eq. (9). The *x*-values for the training data are sampled from a uniform distribution in the input space [0, 3], and the *y*-values are corrupted using an additive Gaussian noise δ with zero mean, with the noise levels 0.1 and 0.5 denoting the standard deviation of the noise. The sample size is *n*=20.)

III. Parameter Optimization with Cross-Validation

Because the number of samples is small, CV is used to optimize the value of *h*. CV is a standard resampling technique used in many applications, such as model selection, and selecting variables and the number of components (Browne, 2000; Arlot & Celisse, et al, 2011). Under CV, the available data are divided into *v* disjoint sets, and the *v*-fold CV is then run *v* times using (*v*-1) groups as training sets and the remaining group as the validation set. This is done in turn until each group is left out once. Clearly, if *v*=*n* then *v*-fold CV is leave-one-out CV (LOO-CV), since exactly one object is left out at a time. The sample reuse technique of CV can help us optimize the parameter *h* where the amount of available data is small. Therefore, the CV error is taken as the objective function, and the optimal value of the bandwidth *h* is that which minimizes CV error. Owing to the fact that LOO-CV provides an almost unbiased estimate (Cawley & Talbot, et al, 2003), LOO-CV is chosen and the objective function is given by:

$$Min\ \ m(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i - \hat{y}_i^{-i}) \tag{3}$$

where $L(y_i - \widehat{y}_i^{-i})$ denotes the loss function of the LOO-CV estimator when (x_i,y_i) is the validation set, and $\widehat{y}_i^{-i}$ denotes the prediction value corresponding to x_i by using $D^{(-i)} := D \setminus (x_i,y_i)$ for training.

$$\widehat{y}_i^{-i} := f(x_i, D^{(-i)}, h) = \frac{\sum_{j=1, j\neq i}^{n} y_j K_h(x_i, x_j)}{\sum_{s=1, s\neq i}^{n} K_h(x_i, x_s)} \tag{4}$$

$L(y_i - \widehat{y}_i^{-i})$ represents the quality of estimation. In practice, different optimization results can be obtained by using different loss functions, which significantly influences the performance of the regression model. For such problems, we consider three representative loss functions, namely square loss, Huber's loss, and Vapnik's ε-insensitive loss function.

The square loss function is the following:

$$L_s(y_i - \widehat{y}_i^{-i}) = (y_i - \widehat{y}_i^{-i})^2 \tag{5}$$

Huber's loss function, which is also called the L_1-loss function, is:

$$L_l(y_i - \widehat{y}_i^{-i}) = \left| y_i - \widehat{y}_i^{-i} \right| \tag{6}$$

Vapnik's ε-insenstive loss function is defined as:

$$L_\varepsilon(y_i - \widehat{y}_i^{-i}) = \begin{cases} 0 & if \left| y_i - \widehat{y}_i^{-i} \right| \leq \varepsilon \\ \left| y_i - \widehat{y}_i^{-i} \right| - \varepsilon & otherwise \end{cases} \tag{7}$$

In SVR, it has been demonstrated that for small sample regression problems Vapnik's ε -insensitive loss (with a properly chosen ε -parameter) yields better generalization than other loss functions (Cherkassky & Ma, 2004b). Cherkassky and Ma proposed a practical method for selecting the value of ε for SVR directly from the training data:

$$\varepsilon = 3\sigma_{noise}\sqrt{\ln(n)/n} \tag{8}$$

where σ_{noise} is the standard deviation of the additive noise and n is the number of training samples.

Vapnik's loss function coincides with a special form of Huber's loss (with ε=0). From the viewpoint of traditional robust statistics, there is a well-known correspondence between the noise model and the optimal loss function. However, this connection between the noise model and the loss function is based on (asymptotic) maximum likelihood arguments, which are not suitable with small samples. Therefore, we compare the generalization performance of Vapnik's ε -insensitive loss in NWKR (with different values for ε) with other loss functions in the next section.

IV. Prepare Your Paper Before Styling Experimeental Results with Gaussian Noise

A. Comparison with Three Loss Function

First we describe the experimental procedure used for comparisons, and then we present the experimental results.

Training data: The simulated training data is (x_i,y_i), (i= *1,2,…,n*), where the x-values are sampled from a uniform distribution on the input space, and the y-values are generated according to $y = f(x) + \delta$. The target function *f(x)* is shown in Eq. (9). The y-values of the training data are corrupted by additive Gaussian noise. For each training data set, we generate five data sets using a small sample size (n=20) with additive Gaussian noise (the different noise levels are shown in Table 1).

$$y = (x^2 - 3x + 1)^4, \quad x \in [0,3] \tag{9}$$

Test data: 150 samples are used for the testing data set, generated sequentially with step-size $\Delta x = (b-a)/150$ from the lower bound a=0 to the upper bound b=3.

Kernel function: Gaussian kernel functions (2) are used in all experiments.

Performance metric: Since the goal is optimal selection of the NWKR parameter in the sense of generalization, the main performance metric is Pred_accuracy (prediction accuracy):

$$RMSE(h) = \sqrt{\frac{1}{150-1}\sum_{i}^{150}(y_i^{test} - f(x_i^{test}, h))^2} \tag{10}$$

$$\Pr ed\_accuracy = 1 - \frac{RMSE(h) - RMSE(h_{best})}{RMSE(h_{best})} \times 100\% \tag{11}$$

where RMSE(h) defines the root mean squared error between NWKR estimates and the true values of the target function for the inputs. h_{best} is the approximate optimal (minimum RMSE) bandwidth h obtained by calculating the RMSE for a range value of h with step t=0.01 in the domain [0,1].

TABLE I. Experimental Results for Different Parameter Selection Methods and Several Noise Levels

Noise level(σ)	***Loss function***	***Pred_accuracy***		
		Min	***Max***	***Average***
0.01	Square	99.44%	99.95%	99.67%
	Huber	78.42%	100.00%	95.58%
	Vapnik(c-m)	78.44%	100.00%	95.66%
	Vapnik(opt)	99.85%	100.00%	99.97%
0.03	Square	61.97%	99.52%	91.42%
	Huber	61.22%	99.96%	87.60%
	Vapnik(c-m)	62.56%	99.61%	87.76%
	Vapnik(opt)	99.92%	100.00%	99.98%
0.05	Square	94.78%	100.02%	97.21%
	Huber	45.40%	99.40%	87.41%
	Vapnik(c-m)	45.77%	99.02%	87.69%
	Vapnik(opt)	96.74%	100.00%	99.24%
0.08	Square	80.38%	94.35%	88.36%
	Huber	65.91%	99.96%	83.43%
	Vapnik(c-m)	72.20%	96.82%	83.08%
	Vapnik(opt)	98.40%	99.99%	99.61%
0.1	Square	98.33%	99.61%	98.97%
	Huber	86.46%	99.76%	96.82%
	Vapnik(c-m)	97.48%	99.68%	98.57%
	Vapnik(opt)	99.73%	100.01%	99.85%
0.2	Square	88.78%	99.94%	97.11%
	Huber	86.67%	99.30%	93.95%
	Vapnik(c-m)	94.96%	99.80%	97.90%
	Vapnik(opt)	99.47%	100.00%	99.78%
0.3	Square	80.48%	98.99%	86.96%
	Huber	58.03%	100.00%	82.82%
	Vapnik(c-m)	74.29%	99.66%	87.66%
	Vapnik(opt)	96.01%	99.99%	99.15%

0.5	Square	61.91%	99.52%	88.15%
	Huber	52.94%	95.00%	73.64%
	Vapnik(c-m)	84.01%	99.98%	94.79%
	Vapnik(opt)	87.41%	100.00%	96.85%
0.8	Square	72.92%	98.81%	90.22%
	Huber	65.71%	98.77%	81.43%
	Vapnik(c-m)	77.17%	96.60%	86.94%
	Vapnik(opt)	96.99%	100.00%	99.40%
1.0	Square	61.77%	99.77%	89.47%
	Huber	58.50%	99.95%	85.56%
	Vapnik(c-m)	85.68%	98.44%	93.21%
	Vapnik(opt)	92.59%	100.00%	97.96%

In Table 1, we present experimental comparisons for regression estimation using three representative loss functions: squared loss, Huber's loss (ε=0), and Vapnik's ε- insensitive loss with ε given according to Eq. (8). The noise level (σ) column indicates the standard deviation of the Gaussian noise with zero mean. In the column for loss function (and ε -selection), Vapnik(c-m) denotes the value of ε from Eq.(8), and Vapnik(opt) denotes the optimal value of ε for Vapnik's ε-insensitive loss function whose corresponding h is closest to h_{best}. Pred_accuracy shows the minimal, maximal and average values for Pred_accuracy in five training data sets for each parameter selection method.

It can be seen in Fig. 2 that:

1) the NWKR approach has good performance for small sample regression problem (approximately 90%) with different noise levels;

2) the prediction accuracy for the square loss function is better than Huber's and Vapnik's (c-m) loss functions when the noise level is smaller than 0.1, but when the noise level is larger than 0.1 Vapnik's (c-m) loss function is the best of the three loss functions;

3) the robustness of the three loss functions is not strong, and weakens as the noise level increases;

4) we can obtain very good prediction accuracy using Vapnik's loss function with an appropriate choice of ε (Vapnik(opt)).

Selecting an appropriate value for the parameter ε for better prediction accuracy is important, and is studied in the next section.

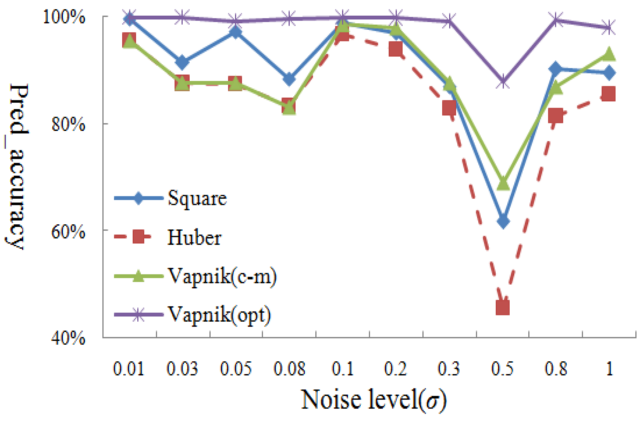

Fig. 2. Scatter diagram of bandwidth h and RMSE The Average Prediction Accuracy for Different Loss Function

B. *Parameter estimation with regression model*

As mentioned before, the NWKR regression approach with Vapnik's loss function could provide excellent prediction accuracy when the parameter ε is set appropriately. However, selecting ε from Eq. (8) is not the best choice because of its unsatisfactory prediction accuracy and robustness. In practice, it is not possible to know in advance the noise level, and the deviation in estimating the noise level using some well-known approach with small samples is unacceptable. It could be feasible to estimate the value of ε according to the dispersion of sample data. In this section, we attempt to estimate an appropriate value of ε depending on the standard deviation of the sample output data in Section 4.1.

Fig. 3 shows a scatter chart between the standard deviation of Y and the optimal value of ε with Vapnik's ε- insensitive loss function. Theoretically, when the standard deviation of Y is 0, the parameter ε should also be 0, and the parameter ε should increase with an increase in the standard deviation of Y. However, when the standard deviation of Y is large enough, the parameter ε should remain invariant, for otherwise some loss value with the Vapnik's ε- insensitive loss function could be 0 leading to the parameter optimization not obtaining the optimum solution. Thus, the logistic regression model was used to establish the relationship between the standard deviation of Y and the value of ε. The logistic regression model chosen in this paper is: $y=\frac{1}{c+e^{a+bx}}$. The parameters are estimated using Matlab 12.0, and the fitting equation is shown in Eq. (12) and the fitting curve in Fig. 3:

$$\varepsilon=\frac{1}{0.82+e^{7.9-8.22\times std\_y}} \tag{12}$$

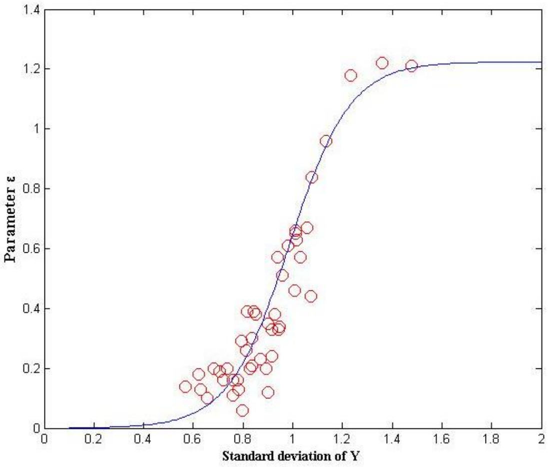

Fig. 3. Experimental results between the standard deviation of Y and Vapnik(opt), and the fitting logistic model

C. *Comparisons with other parameter selection methods*

In this section, the performance of the proposed method (parameter selection with Eq. (12)) is demonstrated in two ways. First the standard deviation of Y is changed while the sample size is unchanged, and second the sample size is changed and the standard deviation of Y is unchanged.

1) Standard deviation changed and sample size unchanged

The sample size n is set to 20, and the standard deviation of Y takes the values 0.01, 0.05, 0.1 and 0.5. For each training data set, we generate ten data sets, with the x-values sampled from a uniform distribution on the input space [0, 3], and the y-values generated from Eq. (9) and corrupted by an additive Gaussian noise with zero mean and specified standard deviation. The test data set, kernel function and performance metric are the same as in Section 4.1. The comparison results for the four parameter methods are shown in Table 2. The robustness of the proposed method is stronger than the methods using square loss, Huber's loss and Vapnik's (c-m) loss. Fig. 4 shows the average prediction accuracy of the different methods for ten data sets, where we can see that the proposed method performs better than the other three methods. An increase in the noise level has little effect on the proposed method, while the performances of the other three methods are weakened. Meanwhile, the gaps between the average prediction accuracy between the proposed method and Vapnik(opt) are less than 5%.

TABLE II. COMPARISON RESULTS FOR SEVERAL PARAMETER METHODS AT DIFFERENT NOISE LEVELS

Noise level(σ)	***ε – selection method***	***Pred_accuracy***		
		Min	***Max***	***Average***
0.01	Square	85.42%	100.0%	95.82%
	Huber	81.09%	100.0%	96.08%
	Vapnik(c-m)	81.09%	99.98%	96.09%
	Proposed	85.31%	100.0%	**97.15%**
0.05	Square	82.67%	99.51%	92.19%
	Huber	81.33%	99.18%	91.06%
	Vapnik(c-m)	86.26%	97.46%	90.96%
	Proposed	89.51%	99.07%	**93.84%**
0.1	Square	58.78%	99.94%	92.57%
	Huber	57.37%	99.98%	88.65%
	Vapnik(c-m)	57.56%	100.0%	90.29%
	Proposed	90.77%	100.0%	**96.72%**
0.5	Square	33.47%	99.87%	85.08%
	Huber	33.47%	99.05%	83.99%
	Vapnik(c-m)	66.12%	99.01%	88.43%
	Proposed	82.20%	98.08%	**91.95%**

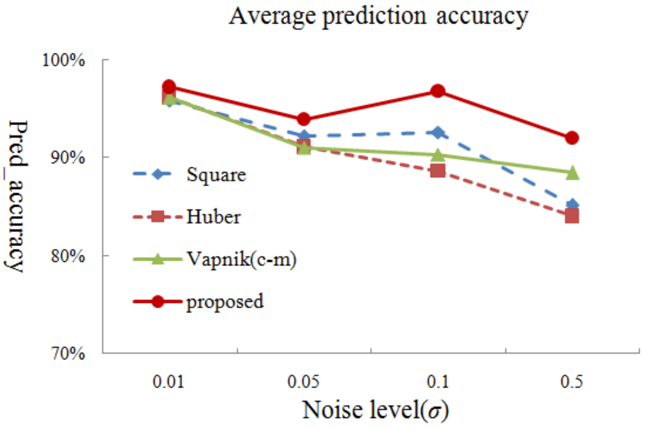

Fig. 4. Average prediction accuracy for several parameter selection methods in different noise levels

2) Sample size changed and standard deviation unchanged

The standard deviation of Y is set to 0.1, and the sample size n is between 10, 15, 20, 25, 30 and 50. For each training data size, we generate ten data sets, with the x-values sampled from a uniform distribution on the input space [0, 3], and the y-values generated from Eq. (9) and corrupted by the additive Gaussian noise *N(0, 0.01)*. The test data set, kernel function and performance metric are the same as in Section 4.1. The comparison results for the four parameter methods are shown in Table 3. Fig. 5 shows the average prediction accuracy of the different methods for ten data sets, where we can see that the proposed method outperforms the other three methods (square loss, Huber's loss and Vapnik (c-m) loss), and the number of samples has little effect on the proposed method. However, the computation time is lengthened with the increased sample size, and the advantages of the proposed method are weakened because the standard deviation of Y is reduced when the sample size increases. It is suggested that the proposed method is suitable when the sample size is less than 30.

TABLE III. COMPARISON RESULTS FOR SEVERAL PARAMETER METHODS WITH DIFFERENT SAMPLE SIZES

The sample size(n)	***ε – selection method***	***Pred_accuracy***		
		Min	***Max***	***Average***
10	Square	63.59%	100.00%	92.76%
	Huber	63.32%	100.00%	92.66%
	Vapnik(c-m)	63.51%	100.00%	92.17%
	Proposed	**64.19%**	**100.00%**	**94.97%**
15	Square	49.01%	98.50%	91.30%
	Huber	88.83%	99.68%	93.93%
	Vapnik(c-m)	54.64%	99.00%	91.54%
	Proposed	**82.95%**	**99.90%**	**96.41%**
20	Square	58.78%	99.94%	92.57%
	Huber	57.37%	99.98%	88.65%
	Vapnik(c-m)	57.56%	100.00%	90.29%
	Proposed	**90.77%**	**100.00%**	**96.72%**
25	75.73%	100.00%	95.00%	75.73%
	87.00%	99.89%	95.25%	87.00%
	75.92%	100.00%	93.67%	75.92%
	90.67%	**100.00%**	**97.73%**	**90.67%**
30	Square	-30.92%	100.00%	81.68%
	Huber	22.97%	100.00%	86.47%
	Vapnik(c-m)	49.26%	100.00%	89.05%
	Proposed	**71.56%**	**100.00%**	**94.89%**
50	Square	83.41%	99.59%	95.53%
	Huber	85.35%	**99.91%**	**96.87%**
	Vapnik(c-m)	**89.09%**	99.55%	94.52%
	Proposed	78.71%	99.57%	95.05%

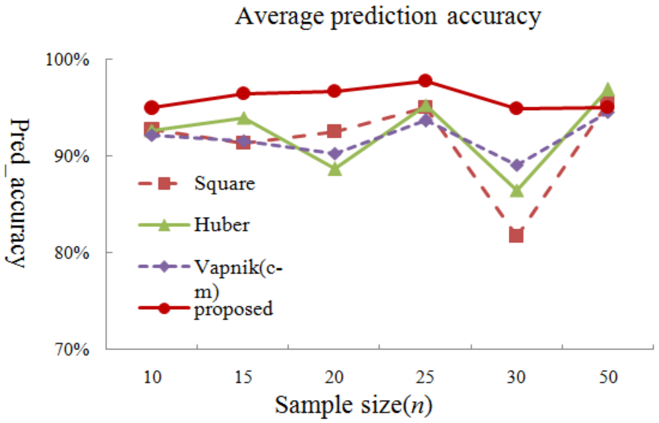

Fig. 5. Average prediction accuracy for several parameter methods with different sample sizes

V. Conclusions

This paper describes practical recommendations for setting meta-parameters for NWKR regression with small samples. Namely, the value of the ε parameter is obtained directly from the training data without estimating the noise level. Empirical comparisons suggest that the proposed parameter selection method (Eq. (12)) yields good generalization performance for NWKR estimates under different noise levels and sample sizes. Hence, the proposed approach for NWKR parameter selection can be used by practitioners interested in applying NWKR to various application domains in which the sample size is small.

In this paper, the proposed value of ε is derived for one target function, with Gaussian noise and an RBF kernel, but it is not clear whether such optimal selection is appropriate for other target functions, noise distributions and kernel types. Future related research may be concerned with investigating the optimal selection of ε for different target functions, noise distributions and kernel types.

Acknowledgment

This work was supported by National Nature Science Foundation of China (No. 51405346 & 71271156) and Key S&T Innovation Team Project of Wenzhou City in China (No. C20120002-02).

References

[1] J. Smola and B. Scholkopf. "A tutorial on support vector regression," Statistics and Computing, vol. 14, pp. 199-222, March 2004.

[2] Arlot S, Celisse A. "A survey of cross-validation procedures for model selection," Statistics Surveys, vol. 4, pp. 40-79, 2010.

[3] Browne M. "Cross-validation methods. Journal of Mathematical Psychology," vol. 44, pp. 108-132, January 2000.

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

[5] Cawley G C, Talbot N L C. "Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers," Pattern Recognition, vol. 36, pp. 2585-2592, November 2003.

[6] Cherkassky V, Ma Y. "Practical selection of SVM parameters and noise estimation for SVM regression," Neural networks, vol. 17, pp. 113-126, January 2004(a).

[7] Cherkassky V, Ma Y. "Comparison of loss functions for linear regression," IEEE International Joint Conference on Neural Networks, vol. 17, pp. 395-400, 2004(b).

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

[9] Jin Seo Cho, Isao Ishida, Halbert White. "Revisiting Tests for Neglected Nonlinearity Using Artificial Neural Networks," Neural Computation, vol. 23, pp. 1133-1186, May 2011.

[10] JS Zhang, XF Huang, CH Zhou. "An Improved Kernel Regression Method Based on Taylor Expansion," Applied Mathematics and Computation, vol. 193, pp. 419-429, 2007.

[11] Kaizhu Huang, Danian Zheng, Irwin King, Michael R. Lyu. "Arbitrary Norm Support Vector Machines. Neural Computation," vol. 21, pp. 560-582, February 2009.

[12] Kar-Ann Toh. "Deterministic Neural Classification. Neural Computation," vol. 20, pp. 1565-1595, July 2008.

[13] Li, D.C., Wu, C.S., Tsai, T.I., et al.. "Using megafuzzification and data trend estimation in small data set learning for early FMS scheduling knowledge," Computers and Operations Research, vol. 33, pp. 1857-1869, 2006.

[14] Maxwell B. Stinchcombe. "Precision and Approximate Flatness in Artificial Neural Networks," Neural Computation, vol. 7, pp. 1021-1039, May 1995.

[15] M.P. Wand M.C. Jones. "Kernel Smoothing," Chapman & Hall, 1995.

[16] R. Andonie. "Fuzzy ARTMAP prediction of biological activities for potential HIV-1 protease inhibitors using a small molecular dataset," IEEE Transactions on Computational Biology and Bioinformatics, vol. 8, pp. 80-93, January 2009.

[17] Shapiai M I, Ibrahim Z, Khalid M, et al.. "A non-linear function approximation from small samples based on Nadaraya-Watson kernel regression," 2nd Conference on Computational Intelligence, Communication Systems and Networks, pp. 28-32, 2010.

[18] Shapiai M I, Sudin S, Ibrahim Z, et al.. "Investigation on Different Kernel Functions for Weighted Kernel Regression in Solving Small Sample Problems," Fifth UKSim European Symposium on. IEEE, pp. 64-69, 2011.

[19] T Su, J Jing, C Hou. "A Hybrid Artificial Neural Networks and Particle Swarm Optimization for Function Approximation," International Journal of Innovative Computing, Information and Control, vol. 4, pp. 2363-2374, September 2008.

[20] W. Lee and S. Ong. "Learning from small data sets to improve assembly semiconductor manufacturing processes," 2nd ICCAE, pp. 50-54, 2010.

[21] Wei Chu, S. Sathiya Keerthi. Support Vector Ordinal Regression. Neural Computation, vol. 19, pp. 792-815, March 2007.

[22] Yali Wang and Brahim Chaib-draa. "A KNN Based Kalman Filter Gaussian Process Regression," In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 1771-1777, 2013.