Journal of Korean
Data & Information Science Society
2003, Vol. 14, No.1 pp 103~111

On Predicting with Kernel Ridge Regression[1)]

Changha Hwang[2)]

Abstract

Kernel machines are used widely in real-world regression tasks. Kernel ridge regressions(KRR) and support vector machines(SVM) are typical kernel machines. Here, we focus on two types of KRR. One is inductive KRR. The other is transductive KRR. In this paper, we study how differently they work in the interpolation and extrapolation areas. Furthermore, we study prediction interval estimation method for KRR. This turns out to be a reliable and practical measure of prediction interval and is essential in real-world tasks.

Keywords : Kernel ridge regression, prediction interval, transductive inference.

1. Introduction

In the case of noisy learning data, the use of traditional neural networks due to its learning method often leads to poor generalization and overfitting. Kernel machines such as support vector machine(SVM) and kernel ridge regression(KRR) were designed to overcome these problems. Foundations of SVM and KRR were established by Vapnik(1995) and Saunders et al. (1998), respectively. Kernel machines are used widely in real-world regression tasks. Here, we focus on two types of KRRs. One is inductive KRR. The other is transductive KRR. In this paper, we study how differently they work in terms of interpolation and extrapolation. Furthermore, we study prediction interval estimation method for

1) This research was supported by the Catholic University of Daegu research grants in 2003.

2) Associate Professor, Dept. of Statistical Information, Catholic University of Daegu.
E-mail: chhwang@cataegu.ac.kr

KRR. This turns out to be a reliable and practical measure of confidence bound and is essential in real-world tasks.

To review KRRs, we need to take lots of materials from Chapelle et al.(1999). Suppose there exists a function $y^* = f_0(\boldsymbol{x})$ from which we observe the measurements corrupted with noise

$$\{(\boldsymbol{x}_i, y_i),\ i = 1, \cdots, l\},\ \ y_i = y_i^* + \varepsilon_i. \tag{1}$$

Find an algorithm A that using both the given set of training data (1) and the given set of test data

$$\{\boldsymbol{x}_{l+1}, \cdots, \boldsymbol{x}_{l+m}\} \tag{2}$$

selects from a set of functions $\{\boldsymbol{x} \mapsto f(\boldsymbol{x})\}$ a function

$$y = f(\boldsymbol{x}) = f_A(\boldsymbol{x} \mid \boldsymbol{x}_1, y_1, \cdots \boldsymbol{x}_l, y_l, \boldsymbol{x}_{l+1}, \cdots, \boldsymbol{x}_{l+m}) \tag{3}$$

and minimizes at the points of interest the functional

$$R(A) = E\left(\sum_{i=l+1}^{l+m} (y_i^* - f_A(\boldsymbol{x} \mid \boldsymbol{x}_1, y_1, \cdots \boldsymbol{x}_l, y_l, \boldsymbol{x}_{l+1}, \cdots, \boldsymbol{x}_{l+m}))^2 \right) \tag{4}$$

where expectation is taken over $\boldsymbol{x}$ and ε. For the training data we are given the vectors $\boldsymbol{x}$ and the value y, for the test data we are only given $\boldsymbol{x}$.

Usually, the problem of estimating values of a function at points of interest is solved in two steps: first in a given set of functions $f(\boldsymbol{x}, \boldsymbol{\alpha})$, $\boldsymbol{\alpha} \in \Lambda$ one estimates the regression function which minimizes the functional

$$R(\boldsymbol{\alpha}) = \int ((y - f(\boldsymbol{x}, \boldsymbol{\alpha}))^2 dF(\boldsymbol{x}, y) \tag{5}$$

(**the inductive step**) and then using the estimated function $y = f(\boldsymbol{x}, \boldsymbol{\alpha}_l)$ we calculate the values at points of interest

$$y_i^* = f(\boldsymbol{x}_i^*, \boldsymbol{\alpha}_l) \tag{6}$$

(**the deductive step**).

2. Kernel Ridge Regression and the Leave-One-Out procedure

For the discussion of the classical two-step (**inductive** plus **deductive**) KRR we consider the set of functions linear in their parameters

$$f(\boldsymbol{x}, \boldsymbol{\alpha}_l) = \sum_{i=1}^{H} \alpha_i \phi_i(\boldsymbol{x}). \tag{7}$$

In the case of KRR we use a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x})$ as basis function $\phi_i(\boldsymbol{x})$ and H equals to the sample size l. Notice that the first component of input vector $\boldsymbol{x}$ is 1 and the rest of components are actual input variables. To

minimize the expected loss (5), we minimize the following empirical functional

$$R_{\text{emp}}(\boldsymbol{\alpha}) = \frac{1}{l}\sum_{i=1}^{l}(y_i - f(\boldsymbol{x}_i, \boldsymbol{\alpha}))^2 + \gamma \| \boldsymbol{\alpha} \|^2 \tag{8}$$

where γ is a fixed positive constant, called the regularization parameter. The minimum is given by the vector of coefficients

$$\boldsymbol{\alpha}_l = \boldsymbol{\alpha}(\boldsymbol{x}_1, y_1, \cdots \boldsymbol{x}_l, y_l) = (K^tK + \gamma I_l)^{-1}K^t\boldsymbol{y} \tag{9}$$

where $\boldsymbol{y} = (y_1, \cdots, y_l)^t$ and K is a matrix with elements,

$$K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j), \; i=1,\cdots, l, \; j=1,\cdots, l. \tag{10}$$

Notice that the vector of coefficients in Saunders(1998) is given by

$$\boldsymbol{\alpha}_l = 2\gamma(K + \gamma I_l)^{-1}\boldsymbol{y}. \tag{11}$$

These are a little different. Now, the problem is to choose the value γ which provides small expected loss for training on a sample $S_l = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_l, y_l)\}$. For this purpose, we would like to choose γ such that f_γ minimizing (8) also minimizes

$$R = \int (y^* - f_\gamma(\boldsymbol{x}^* \mid S_l))^2 dF(\boldsymbol{x}^*, y^*) dF(S_l) \tag{12}$$

Since $F(\boldsymbol{x}, y)$ is unknown, we cannot estimate this minimum directly. To solve this problem we use the leave-one-out(LOO) procedure, which is an unbiased estimator of (12). The LOO error of an algorithm on the training sample S_l is

$$T_{loo}(\gamma) = \frac{1}{l}\sum_{i=1}^{l}(y_i - f_\gamma(\boldsymbol{x}_i \mid S_l \setminus (\boldsymbol{x}_i, y_i))^2 \tag{13}$$

The minimum over γ of (13) we consider as the minimum over γ of (12) since the expectation of (13) coincides with (12).

For KRR, we can derive a closed form expression for the LOO error. Denoting

$$A_\gamma^{-1} = (K^tK + \gamma I_l)^{-1} \tag{14}$$

the error incurred by the LOO procedure is

$$T_{loo}(\gamma) = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{y_i - \boldsymbol{k}_i^t A_\gamma^{-1}K^t\boldsymbol{y}}{1 - \boldsymbol{k}_i^t A_\gamma^{-1}\boldsymbol{k}_i}\right)^2 \tag{15}$$

where $\boldsymbol{k}_t = (K(\boldsymbol{x}_1, \boldsymbol{x}_t), \cdots, K(\boldsymbol{x}_l, \boldsymbol{x}_t))^t$. Let $\gamma = \gamma_0$ be the minimum of (15). Then the vector

$$\boldsymbol{y}_0 = \boldsymbol{K}^*(K^tK + \gamma_0 I_l)^{-1}K^t\boldsymbol{y} \tag{16}$$

where

$$K^* = \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_{l+1}) & \cdots & K(\boldsymbol{x}_l, \boldsymbol{x}_{l+1}) \\ \vdots & & \vdots \\ K(\boldsymbol{x}_1, \boldsymbol{x}_{l+m}) & \cdots & K(\boldsymbol{x}_l, \boldsymbol{x}_{l+m}) \end{pmatrix} \tag{17}$$

is the KRR estimate of the unknown values $(y^*_{l+1}, \cdots, y^*_{l+m})$.

3. Leave-One-Out Error for Transductive Inference

In transductive inference, our goal is to find an algorithm A which minimizes the functional (4) using both the training data (1) and the test data (2). We suggest the following method: predict $(y^*_{l+1}, \cdots, y^*_{l+m})$ by finding those values which minimize the LOO error of KRR training on the joint set

$$(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_l, y_l), (\boldsymbol{x}_{l+1}, y^*_{l+1}), \cdots, (\boldsymbol{x}_{l+m}, y^*_{l+m}) \tag{18}$$

This is achieved in the following way. Suppose we treat the unknown values $\boldsymbol{y}^*_m = (y^*_{l+1}, \cdots, y^*_{l+m})$ as variables and for some fixed value of these variables we minimize the following empirical functional

$$R_{\text{emp}}(\boldsymbol{\alpha} \mid \boldsymbol{y}^*_m) = \frac{1}{l}\left(\sum_{i=1}^{l}(y_i - f(\boldsymbol{x}_i, \boldsymbol{\alpha}))^2 + \sum_{i=l+1}^{l+m}(y^*_i - f(\boldsymbol{x}_i, \boldsymbol{\alpha}))^2\right) + \gamma \| \boldsymbol{\alpha} \|^2. \tag{19}$$

This functional differs only in the second term from the functional (8) and corresponds to performing KRR with the extra pairs $(\boldsymbol{x}_{l+1}, y^*_{l+1}), \cdots, (\boldsymbol{x}_{l+m}, y^*_{l+m})$.

Suppose that vector $\boldsymbol{y}^* = (y^*_{l+1}, \cdots, y^*_{l+m})^t$ is taken from some set $\boldsymbol{y}^* \in \mathcal{Y}$ such that the pairs $(\boldsymbol{x}_{l+1}, y^*_{l+1}), \cdots, (\boldsymbol{x}_{l+m}, y^*_{l+m})$ can be considered as a sample drawn from the distribution as the pairs $(\boldsymbol{x}_1, y^*_1), \cdots, (\boldsymbol{x}_l, y^*_l)$. In this case the LOO error of minimizing (19) over the set (18) approximates the functional (4). Using the closed form (15) one obtains

$$T_{loo}(\gamma \mid y^*_{l+1}, \cdots, y^*_{l+m}) = \frac{1}{l+m}\sum_{i=1}^{l+m}\left(\frac{\widehat{y}_i - \widehat{\boldsymbol{k}}_i^t \widehat{A_\gamma^{-1}} \widehat{K}^t \widehat{\boldsymbol{y}}}{1 - \widehat{\boldsymbol{k}}_i^t \widehat{A_\gamma^{-1}} \widehat{\boldsymbol{k}}_i}\right)^2 \tag{20}$$

where we denote $\widehat{\boldsymbol{x}} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{l+m})^t$ and $\widehat{\boldsymbol{y}} = (y_1, \cdots, y_l, y^*_{l+1}, \cdots, y^*_{l+m})^t$, and

$$\widehat{A_\gamma^{-1}} = (\widehat{K}^t \widehat{K} + \gamma I_{l+m})^{-1} \tag{21}$$

$$\widehat{K} = \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_{l+m}) \\ \vdots & & \vdots \\ K(\boldsymbol{x}_{l+m}, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_{l+m}, \boldsymbol{x}_{l+m}) \end{pmatrix} \tag{22}$$

$$\widehat{\boldsymbol{k}_t} = (K(\boldsymbol{x}_t, \boldsymbol{x}_1) \quad \cdots \quad K(\boldsymbol{x}_t, \boldsymbol{x}_{l+m})) \tag{23}$$

Now let us rewrite the expression (20) in an equivalent form to separate the terms with $\widehat{\boldsymbol{y}}$ from the terms with $\boldsymbol{x}$. Introducing

$$C = I_{l+m} - \hat{K}\hat{A}_{\gamma}^{-1}\hat{K}^t, \tag{24}$$

and the matrix M with elements

$$M_{ij} = \sum_{k=1}^{l+m} \frac{C_{ik}C_{kj}}{C_{kk}^2}. \tag{25}$$

We obtain the equivalent expression of (20)

$$T_{loo}(\gamma \mid y_{l+1}^*, \cdots, y_{l+m}^*) = \frac{1}{l+m}(\hat{\boldsymbol{y}}^t M \hat{\boldsymbol{y}}). \tag{26}$$

In order for the $\boldsymbol{y}^*$ which minimize the LOO procedure to be valid it is required that the pairs $(\boldsymbol{x}_{l+1}, y_{l+1}^*), \cdots, (\boldsymbol{x}_{l+m}, y_{l+m}^*)$ are drawn from the same distribution as the pairs $(\boldsymbol{x}_1, y_1^*), \cdots, (\boldsymbol{x}_l, y_l^*)$. To satisfy this constraint we choose vectors $\boldsymbol{y}^*$ from the set

$$\mathcal{Y} = \{\boldsymbol{y}^* : \| \boldsymbol{y}^* - \boldsymbol{y}_0 \|^2 \le R\} \tag{27}$$

where the vector $\boldsymbol{y}_0$ is the solution obtained from classical ridge regression.

To minimize (26) under constraint (27) we use the functional

$$T_{loo}(\gamma \mid y_{l+1}^*, \cdots, y_{l+m}^*) = \hat{\boldsymbol{y}}^t M \hat{\boldsymbol{y}} + \gamma^* \| \boldsymbol{y}^* - \boldsymbol{y}_0 \|^2 \tag{28}$$

where γ^* is a constant depending on R.

Now, all that remains is to find the minimum of (28) in $\boldsymbol{y}^*$. Note that the matrix M is obtained using only the vectors $\hat{\boldsymbol{x}}$. Therefore, to find the minimum of this functional we rewrite (28) as

$$T_{loo}(\gamma) = \boldsymbol{y}^t M_0 \boldsymbol{y} + 2\boldsymbol{y}^{*t} M_1^t \boldsymbol{y} + \boldsymbol{y}^{*t} M_2 \boldsymbol{y}^* + \gamma^* \| \boldsymbol{y}^* - \boldsymbol{y}_0 \|^2 \tag{29}$$

where

$$M = \begin{pmatrix} M_0 & M_1 \\ M_1^t & M_2 \end{pmatrix} \tag{30}$$

and M_0 is a $l \times l$ matrix, M_1 is a $l \times m$ matrix and M_2 is a $m \times m$ matrix. Taking the derivative of (29) in $\boldsymbol{y}^*$ we obtain the condition for the solution

$$2M_1^t \boldsymbol{y} + 2M_2 \boldsymbol{y}^* - 2\gamma^* \boldsymbol{y}_0 + 2\gamma^* \boldsymbol{y}^* = \boldsymbol{0} \tag{31}$$

which gives the predictions

$$\boldsymbol{y}^* = (\gamma^* I_m + M_2)^{-1}(-M_1^t \boldsymbol{y} + \gamma^* \boldsymbol{y}_0) \tag{32}$$

In this transductive KRR we have two parameters to control: γ and γ^*. The choice of γ can be found using the LOO estimator (15) for KRR. This leaves γ^* as the only free parameter.

4. Prediction Interval Estimation for KRR

Standard methods for computing prediction intervals in nonlinear regression can be effectively applied to neural networks when the number of training points is large. However, De Veaux et al.(1998) presented an approach to estimating prediction intervals which uses weight decay to fit the network and show that this method is effective on a wide range of problems. Since KRR uses weight decay, we can estimate prediction intervals for KRR in the same way as De Veaux et al.(1998) did. By the way, Seok et al.(2002) considered to estimating prediction intervals for standard SVM in a different way.

It is commonly assumed that the KRR satisfies the nonlinear regression model

$$y = f(\boldsymbol{x}, \boldsymbol{\alpha}^*) + \varepsilon \tag{33}$$

where $\boldsymbol{x}$ represent the inputs, y the outputs, $\boldsymbol{\alpha}^*$ the true values of the set of parameters, and ε is the error associated with the function f in modeling the system. Let $\widehat{\boldsymbol{\alpha}}$ be the least squares estimate of $\boldsymbol{\alpha}^*$ obtained by minimizing the error function

$$S(\boldsymbol{\alpha}) = \sum_{i=1}^{l} (y_i - f(\boldsymbol{x}, \boldsymbol{\alpha}))^2 \tag{34}$$

for a training set $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_l, y_l)$. The predicted output of the input $\boldsymbol{x}_0$ is

$$\widehat{y_0} = f(\boldsymbol{x}_0, \widehat{\boldsymbol{\alpha}}). \tag{35}$$

Assume that ε is independently and normally distributed with zero means. The $100(1-\alpha)\%$ confidence interval for the predicted value $\widehat{y_0}$ is $\widehat{y_0} \pm c$, where c is

$$c = t^{\frac{\alpha}{2}}_{l-p^*} s (1 + \boldsymbol{f}_0^t (F^t F)^{-1} \boldsymbol{f}_0)^{\frac{1}{2}}. \tag{36}$$

Here, $t^{\frac{\alpha}{2}}_{l-p^*}$ is the inverse of the Student t cumulative distribution function with $l-p^*$ degrees of freedom, evaluated at $\alpha/2$, p^* is the effective number of parameters, and $s^2 = S(\boldsymbol{\alpha})/(l-p^*)$. In effect, the vector $\boldsymbol{f}_0$ is given by

$$\begin{aligned} \boldsymbol{f}_0 &= \left(\frac{\partial f(\boldsymbol{x}_0, \boldsymbol{\alpha}^*)}{\partial \alpha_1^*} \ \frac{\partial f(\boldsymbol{x}_0, \boldsymbol{\alpha}^*)}{\partial \alpha_2^*} \ \cdots \ \frac{\partial f(\boldsymbol{x}_0, \boldsymbol{\alpha}^*)}{\partial \alpha_l^*} \right)^t \\ &= (K(\boldsymbol{x}_1, \boldsymbol{x}_0) \ \cdots \ K(\boldsymbol{x}_l, \boldsymbol{x}_0)) \end{aligned} \tag{37}$$

and the Jacobian matrix F is given by

$$F = \begin{pmatrix} \frac{\partial f(\boldsymbol{x}_1, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_1} & \frac{\partial f(\boldsymbol{x}_1, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_2} & \cdots & \frac{\partial f(\boldsymbol{x}_1, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_l} \\ \frac{\partial f(\boldsymbol{x}_2, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_1} & \frac{\partial f(\boldsymbol{x}_2, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_2} & \cdots & \frac{\partial f(\boldsymbol{x}_2, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_l} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f(\boldsymbol{x}_l, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_1} & \frac{\partial f(\boldsymbol{x}_l, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_2} & \cdots & \frac{\partial f(\boldsymbol{x}_l, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\alpha}_l} \end{pmatrix} = K. \tag{38}$$

Notice that

$$p^* = \sum_{i=1}^{l} \frac{\lambda_i}{\lambda_i + \gamma}, \tag{39}$$

where λ_i is the eigenvalues of F. See for details Vapnik(1998). Replacing $\boldsymbol{\alpha}^*$ in (37) by $\widehat{\boldsymbol{\alpha}}$, we can estimate the prediction interval straightforwardly.

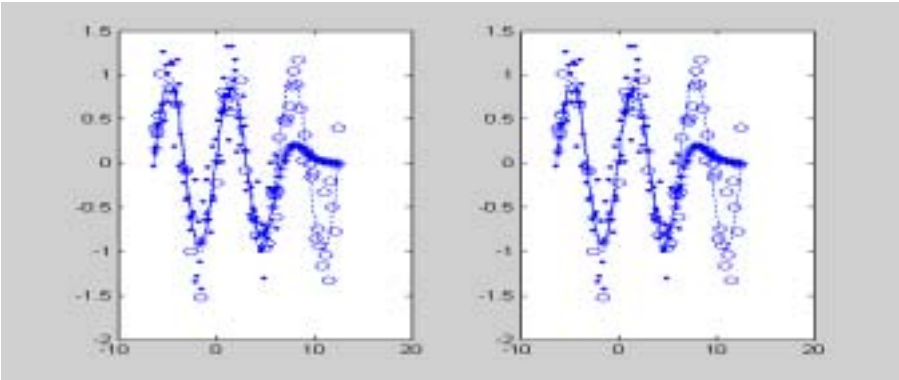
De Veaux et al.(1998) showed that the above method for the prediction interval works well when training data set is large. However, when the training data set is small and the network is trained to convergence, the matrix F^tF can be nearly singular. In this case, the estimated prediction intervals are unreliable. They suggested using the weight decay method, i.e minimizing the error function

$$S(\boldsymbol{\alpha}) + \gamma \| \boldsymbol{\alpha} \|^2 \tag{40}$$

instead of $S(\boldsymbol{\alpha})$. Following De Veaux et al.(1998), for KRR we get finally the prediction interval given by

$$c = t^{\frac{\alpha}{2}}_{n-p^*} s (1 + \boldsymbol{f}_0^t (F^tF + \gamma I)^{-1} F^tF (F^tF + \gamma I)^{-1} \boldsymbol{f}_0)^{\frac{1}{2}}. \tag{41}$$

5. Experiments



(a) The Result of Inductive KRR (b) The Result of Transductive KRR

Figure 1. A Comparison of Transductive KRR to Inductive KRR

We first examine the shape of the estimated regression functions to compare the one-step transductive KRR with the classical two-step inductive KRR in terms of interpolation and extrapolation. Artificial data is generated by a simple function $y = \sin x$ which is corrupted by Gaussian noise with variance 0.1. This function is well used in the papers, for example Gao et al.(2001) related to confidence bound. The training data points of size 100 are evenly distributed between -2π and 2π. The test data points of size 20 are unevenly distributed between -2π and 2π. The test data points of size 30 are evenly distributed between 2π and 4π. When $x > 2\pi$, two KRRs extrapolate the training data. Typical results are shown in Figure 1. Two KRRs are implemented with an Gaussian kernel function with $\sigma = 1.8$, $\gamma = 0.08$ for inductive KRR and $\gamma^* = 1.2$ for transductive KRR. The LOO estimator was used to determine these values.

In Figure 1, the dotted line indicates the true regression function, and the solid line indicates the estimated regression function based on 100 training data points between -2π and 2π. The points marked with dot and circle are the training and test data points, respectively. The points marked with asterisk are the estimated values of test data points. As seen from Figure 1, we notice that for training data points and test data points in the interpolation area two KRRs work very equally well. However, for test data points in the extrapolation area two KRRs do not work well. As a whole, two KRRs work in the exactly same way. Therefore, at least for this particular data set we can not argue that for test data set transductive KRR works better than inductive KRR does, even though transductive KRR is developed to improve estimation proficiency for test data. Actually, a series of experiments has been conducted, but only results for sine function are reported here.

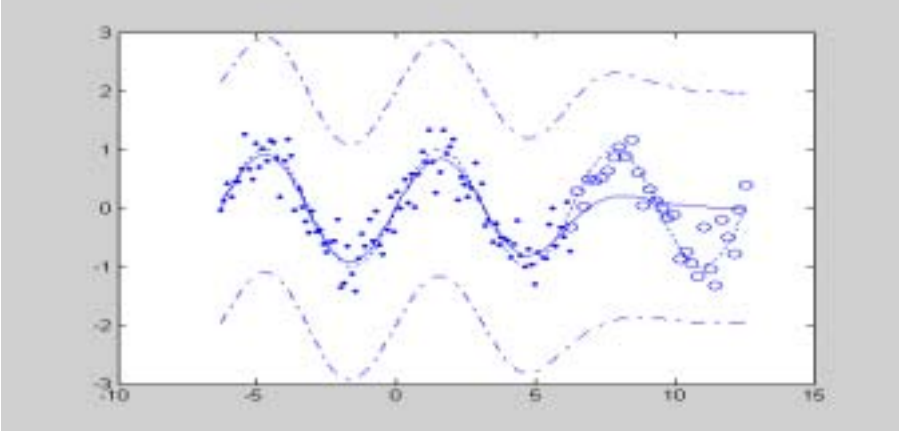


Figure 2 illustrates the prediction intervals for KRR. The points marked with dot and circle are the training and test data points, respectively. The test data

points are only in the extrapolations area. As seen from this figure, we notice that prediction intervals are somewhat wide and are wider in the extrapolation area. We see that prediction intervals are smooth. It is because KRR uses regularization method. This phenomenon happens to the neural networks using weigh dacay.

6. Conclusions

In this paper we perform transductive inference in the problem of estimating values of functions at the points of interest. We demonstrate that estimating the unknown values via a one-step transductive KRR is not necessarily more accurate than the traditional two-step (inductive plus deductive) KRR.

Like De Veaux et al.(1998), we also study an approach to estimating prediction intervals for KRR, which uses weight decay to fit the network and show that this method is especially simple and effective for kernel machines such as SVM and KRR.

References

1. Chapelle, O., Vapnik, V. and Weston, J. (1999). Transductive Inference for Estimating Values of Functions, Advances in Neural Information Processing Systems, 12.
2. DeVeaux, R., Schumi, J., Schweinsberg, J., Shellington, D. and Ungar, L.H. (1998). Prediction Intervals for Neural Networks via Nonlinear Regression, Technometrics, 40, 4, 273-282.
3. Gao, J. B., Gunn, S. R., Harris, C. J. and Brown, B. (2001). A Probabilistic Framework for SVM Regression and Error Bar Estimation, ISIS Technical Report, U. of Southampton.
4. Saunders, C., Gammerman, A. and Vo파, V. (1998). Ridge Regression Learning Algorithm in Dual Variables, Proceedings of the 15th International Conference on Machine Learning, 515-521.
5. Seok, K., Hwang, C. and Cho, D. (2002). Prediction Intervals for Support Vector Machine Regression, Communications in Statistics: Theory and Methods, 31, 10, 1887-1898.
6. Vapnik, V. (1995). The Nature of Statistical Learning Theory, Springer, New York.
7. Vapnik, V. (1998). Statistical Learning Theory, John Wiley & Sons, New York.

[received date : Dec. 2002, accepted date : Feb. 2003]