ON STRUCTURE SELECTION OF RADIAL BASIS FUNCTION NETWORKS

C. W. Chan and K. Y. Choy

Department of Mechanical Engineering, The University of Hong Kong
Hong Kong, China, Email: mechan@hkucc.hku.hk

Abstract: The orthogonal least squares algorithm (*OLS*) and the support vector regression (*SVR*) are two popular approaches to choose the structure of the Radial Basis Function Network (*RBFN*). The former is derived based only on the modelling errors, whilst the latter also on the model complexity. A comparison of the generalization results of networks selected from the *OLS* and the *SVR* is presented here using a simulated nonlinear system, and river discharges and rainfall data of Fuji River. The *RBFN* based on the *SVR* is shown to perform better than that based on the *OLS*.

Keywords: Support vector regression, orthogonal least squares, radial basis function networks, nonlinear systems

1. INTRODUCTION

The lattice and the scattering methods are the two most popular approaches to select the structure of a Radial Basis Function Network (*RBFN*). In the former, the input space of the network is partitioned by lattice with the radial basis functions (*RBF*s) positioned at the nodes (Brown and Harris, 1994), whilst in the latter, the *RBF*s are scattered over the input space (Jang, *et al*., 1997). The well known problem in the lattice method is the "curse of dimensionality", since the number of *RBF*s in the network increases exponentially as the dimension of the input space increases. With the *RBF*s scattered over the input space in the scattered method, fewer *RBF*s are used, hence reducing this problem to a more manageable one. However, the main problem in the scattering method is the selection of the centres of the *RBF*s, so as to reduce complexity of network, whilst maintaining its generalization ability.

There are two popular approaches to select the structure of the networks in the scattering method. The first one is to minimize a cost function consisting of the model complexity and the variance of the modeling errors, and the second one, just to minimize the variance of the modeling errors. The Support Vector Regression (*SVR*) derived from statistical learning theory and structural risk minimization principle (Vapnik, 1998) is an example of the first approach. The Support Vector Radial Basis Function Network (*SVRBFN*) is derived from *SVR* with the Support Vectors (*SV*s) as the centres of the *RBF*s of the network (Chan, *et al*. (2001). In the second approach, the Orthogonal Least Squares (*OLS*) learning algorithm (Chen *et al.*, 1989) is proposed for selecting the centres of the *RBF*s. Both approaches have their own strength. However, if they are compared based on the variance of the modeling errors, better results will generally be obtained from the second approach, as it minimizes only the variance of the modeling errors. However, the generalization result may not necessary be better than the lattice method, as the latter may over-fit the data. In this paper, the generalization ability of both approaches is compared using a simulated nonlinear dynamic system with additive white noise, and the river discharges and rainfall data of Fuji River from January 1990 to December 1993.

2. RADIAL BASIS FUNCTION NETWORKS

Consider a nonlinear system with a white noise,

$$y(k) = f(y(k-1),\cdots,y(k-m_y), u(k-1),\cdots,u(k-m_u)) + e(k) \quad (1)$$

where $y(k)$ and $u(k)$ are the output and the input; m_y and m_u are known orders of the system; $e(k)$ is a normally distributed zero mean white noise, $e(k) \sim N(0,\sigma^2)$; $f(.)$ is a well-defined but unknown nonlinear function. The nonlinear system (1) can be rewritten as,

$$y(k) = f(X(k)) + e(k) \quad (2)$$

where $X(k) = [y(k–1),\ldots,y(k–m_y),\ldots,u(k–1),\ldots,y(k–m_u)]^T$. Let $m = m_y + m_u$. For convenience, denote $X(k) = [x_1(k)\ ,\ \ldots\ ,\ x_m(k)]^T$. Since $f(.)$ is a well-defined nonlinear function, and *RBF*s are able to interpolate in high-dimensional input space (Powell, 1985), the following *RBFN* can provide a good approximation of $f(.)$,

$$y(k) = \sum_{i=1}^{n} \theta_i N_i(X(k)) \quad (3)$$

where $N_i(.)$ is the i^{th} multivariate basis function and θ_i is the corresponding weight. In (3), the *RBF* is a nonlinear function of the Euclidean distance between the network input $X(k)$ and the centre $C(i)$ of the basis function, where $C(i) = [c_1(i)\ ,\ \ldots\ ,\ c_m(i)]^T$. The Gaussian function $\mu_i(X(k))$ is used here for their good localization property,

$$\mu_i(\mathrm{X}(k)) = \exp\left(-\left\|X(k) - C(i)\right\|^2 / 2\gamma^2\right) \quad (4)$$

where γ is the spread (standard deviation) of the Gaussian function, which must be chosen to ensure a thorough coverage of the input space. As discussed previously, the generalization ability of *RBFN*s depend on the choice of the centres of the *RBF*s, which can be chosen by the scattering or the lattice methods (Jang *et al.*, 1997).

Since *RBF*s do not necessarily form a partition of unity, $N_i(.)$ is obtained after normalization, as given below.

$$N_i(\mathrm{X}(k)) = \mu_i(X(k)) \Big/ \sum_{i=1}^{n} \mu_i(X(k)) \quad (5)$$

3. ORTHOGONAL LEAST SQUARES ALGORITHM

The *OLS* learning algorithm is a technique for computing the centres of the *RBF* and the weights of the *RBFN* from input data (Chen, *et al.*, 1991). An input datum is selected as a centre, if it maximizes the cost function, defined below as the error reduction ratio (*ERR*) (Chen, *et al.*, 1989). For N samples, (3) is rewritten in matrix form as,

$$Y = \Phi\theta + \xi \quad (6)$$

where $Y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}$, $\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$, $\xi = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix}$,

$$\Phi = \begin{bmatrix} N_1(\mathrm{X}(1)) & \cdots & N_n(\mathrm{X}(1)) \\ \vdots & & \vdots \\ N_1(\mathrm{X}(N)) & \cdots & N_n(\mathrm{X}(N)) \end{bmatrix} \quad (7)$$

If all the input data are chosen as the centres of the *RBF*s, then

$$\Phi = \begin{bmatrix} N_1(X(1)) & \cdots & N_N(X(1)) \\ \vdots & & \vdots \\ N_1(X(N)) & \cdots & N_N(X(N)) \end{bmatrix} = \begin{bmatrix} \phi_1 & \cdots & \phi_N \end{bmatrix}$$

Consider an orthogonal decomposition of Φ,

$$\Phi = TQ \quad (8)$$

where Q is a N x N upper unit triangular matrix

$$Q = \begin{bmatrix} 1 & \lambda_{12} & \lambda_{13} & \cdots & \lambda_{1N} \\ & 1 & \lambda_{23} & \cdots & \lambda_{2N} \\ & & \ddots & & \\ 0 & & & & 1 \end{bmatrix} \quad (9)$$

and $T = [t_1\ ,\ \ldots\ ,\ t_N]$ is a N x N matrix, satisfying,

$$T^T T = \begin{bmatrix} \kappa_1 & & 0 \\ & \ddots & \\ 0 & & \kappa_N \end{bmatrix}; \qquad \kappa_i = t_i^T t_i \quad (10)$$

Applying the classical Gram-Schmidt (CGS) procedure to compute Q one column a time and orthogonalizing Φ gives,

$$\left.\begin{array}{l} t_1 = \phi_1 \\ \lambda_{ik} = < t_i, \phi_k > / < t_i, t_i >; \textit{ for } 1 \le i < k \\ t_k = \phi_k - \sum_{i=1}^{k-1} \lambda_{ik} t_i \end{array}\right\} k = 2,\ldots,N \quad (11)$$

where $<\cdot,\cdot>$ is the inner product. Rearranging (6),

$$Y = \Phi\theta + \xi = (\Phi Q^{-1})(Q\theta) + \xi = Tg + \xi \quad (12)$$

and the linear least squares estimate of g is given by,

$$\hat{g} = (T^T T)^{-1} T^T Y \quad \text{or} \quad \hat{g}_i = t_i^T Y / t_i^T t_i \quad (13)$$

The estimated weights $\hat{\theta}$ is obtained from $Q\theta = g$ by back substitution. Consider the cost function,

$$J = \xi^T \xi \quad (14)$$

From (12) and (13), (14) becomes,

$$J = Y^T Y - \sum_{i=1}^{N} g_i^2 (t_i^T t_i) \quad (15)$$

The ERR_i due to each input datum is given by,

$$ERR_i = g_i^2 t_i^T t_i / Y^T Y \quad (16)$$

From (16), a simple and effective forward-selection procedure can be derived for choosing the *RBF* centres. This can be considered as finding a subset of models with all the related variables denoted by the subscript s, e.g., T_s. At the i^{th} stage, by interchanging the i^{th} to N^{th} columns of Φ, a ϕ_i is selected that gives the largest ERR_i when orthogonalized into t_i. At the first stage, for $i = 1\ ,\ \ldots\ ,\ N$, denote $t_1^{(i} = \phi_i$ and compute

$$g_1^{(i} = \frac{< t_1^{(i}, Y >}{< t_1^{(i}, t_1^{(i} >}, \quad ERR_1^{(i} = \frac{(g_1^{(i})^2 < t_1^{(i}, t_1^{(i} >}{< Y, Y >} \quad (17)$$

Assume that $ERR_1^{(j} = \max\{ERR_1^{(i},\ 1 \le i \le N\}$, then $t_1 = t_1^{(j}$ is the first column of T_s; $g_1 = g_1^{(j}$ is the first

element of g_s; $ERR_1 = ERR_1^{(j}$. At the second stage, for $i = 1, \ldots, N$ and $i \neq j$, then compute

$$\left.\begin{aligned} \lambda_{12}^{(i} &= \frac{< t_1, \phi_i >}{< t_1, t_1 >}, \quad t_2^{(i} = \phi_i - \lambda_{12}^{(i} t_1 \\ g_2^{(i} &= \frac{< t_2^{(i}, Y >}{< t_2^{(i}, t_2^{(i} >}, \quad ERR_2^{(i} = \frac{(g_2^{(i})^2 < t_2^{(i}, t_2^{(i} >}{< Y, Y >} \end{aligned}\right\} \tag{18}$$

Assume that $ERR_2^{(k} = \max\{ERR_2^{(j}, 1 \leq i \leq N \text{ and } i \neq j\}$, then $t_2 = t_2^{(k}$ is the second column of T_s; $\lambda_{12} = \lambda_{12}^{(k}$ for Q_s; $g_2 = g_2^{(k}$ is the second element of g_s; $ERR_2 = ERR_2^{(k}$. The selection procedure continues until the N_s^{th} stage until

$$1 - \sum_{i=1}^{N_s} ERR_i < \rho \tag{19}$$

where ρ, $0 < \rho \leq 1$, is a desired tolerance. The estimated weights for the subset model $\hat{\theta}_s$ can be computed from $Q_s \theta_s = g_s$ by back substitution.

4. SUPPORT VECTOR BASIS FUNCTION NETWORK

Given N input-output data pairs $\{X(k), y(k)\}_{k=1}^{N}$, where $X(k)$ is defined in (2), the *SVR* is given by,

$$f(X) = < w, X > + b \tag{20}$$

where w is a set of weighting parameters, b is the bias, and <.,.> is defined in (11). The input space is transformed into a high-dimensional feature space involving both w and X. The *SVR* is obtained by minimizing the regularized risk functional for a precision level ε (Vapnik, 1998),

$$\frac{1}{2}\|w\|^2 + C V_{emp}^{\varepsilon}[f] \tag{21}$$

where $V_{emp}^{\varepsilon}[f]$ is the ε-insensitive loss function and C is the regularization constant determining the trade-off with the complexity cost $\|w\|^2$. The ε-insensitive loss function gives the *SVR* a sparseness property, since training errors with amplitude less than ε will not be penalized, i.e.,

$$\left|y - f(X)\right|_{\varepsilon} = \max\{0, \left|y - f(X)\right| - \varepsilon\} \tag{22}$$

The minimization problem of (21) is equivalent to the constrained optimization involving the Lagrange multipliers $\overline{\alpha}, \underline{\alpha}$ (Schölkopf and Smola, 2002):

Min.

$$\begin{aligned} L(\overline{\alpha}, \underline{\alpha}) = \frac{1}{2} \sum_{i,j=1}^{N} (\underline{\alpha}_i - \overline{\alpha}_i)(\underline{\alpha}_j - \overline{\alpha}_j)(X(i), X(j)) \\ - \varepsilon \sum_{i=1}^{N} (\underline{\alpha}_i + \overline{\alpha}_i) + \sum_{i=1}^{N} y(i)(\underline{\alpha}_i - \overline{\alpha}_i) \end{aligned} \tag{23}$$

subject to $\sum_{i=1}^{N} (\overline{\alpha}_i - \underline{\alpha}_i) = 0$, $\overline{\alpha}_i, \underline{\alpha}_i \in [0, C/N]$.

The regression (20) reduces to,

$$f(X) = \sum_{i=1}^{N} (\underline{\alpha}_i - \overline{\alpha}_i) < X(i), X > + b \tag{24}$$

From the Karush-Kuhn-Tucker conditions:

(1) For data lying outside the ε-insensitive tube,

if $y(i) - f(\mathrm{X}(i)) > \varepsilon$, $\underline{\xi}_i > 0$, then $\begin{cases} \underline{\alpha}_i = C/N \\ \overline{\alpha}_i = 0 \end{cases}$,

or

if $f(X(i)) - y(i) > \varepsilon$, $\overline{\xi}_i > 0$, then $\begin{cases} \underline{\alpha}_i = 0 \\ \overline{\alpha}_i = C/N \end{cases}$.

(2) For $\underline{\alpha}_i \in (0, C/N)$ such that $\underline{\xi}_i = 0$, $\overline{\alpha}_i = 0$, then $b = y(i) - < w, X(i) > + \varepsilon$, and for $\overline{\alpha}_i \in$ (0, C/C), such that $\overline{\xi}_i = 0$, $\underline{\alpha}_i = 0$, then $b = y(i) - < w, X(i) > - \varepsilon$. The bias b can be obtained from data lying on the boundary of the ε-insensitive tube, $|y(i) - f(X(i))| = \varepsilon$.

(3) For data within the ε-insensitive tube, $|y(i) - f(X(i))| < \varepsilon$, both Lagrange multipliers will be zero, and they are therefore ignored. For data lying outside or on the boundary of the ε-insensitive tube, they are retained as the *SV*s of (24), since one of their Lagrange multipliers is non-zero.

Let the kernel $K(X(i), X)$ be given by the inner product in the feature space, where X is the m-dimensional training data in the input space $\Re^m$. With kernels that have the *SV*s as their centres are retained, X now reduces to $X_{sv}(i)$, where n_{sv} is the number of *SV*s, and (24) reduces to the *SVR*,

$$f(X(k)) = \sum_{i=1}^{n_{sv}} (\underline{\alpha}_i - \overline{\alpha}_i) K(X_{\mathrm{sv}}(i), X(k)) + b \tag{25}$$

The main advantage of the *SVR* is that its structure, i.e., the number and the positions of the *SV*s, are determined objectively for a given precision level ε. As the output of the *SVR* is biased, the *SVRBFN* (Chan *et al.* 2001) is derived from the *SVR* to overcome this problem,

$$y(k) = \frac{\sum_{i=1}^{n_{sv}} \theta_i K_i(X(k))}{\sum_{i=1}^{n_{sv}} K_i(X(k))} = \sum_{i=1}^{n_{sv}} \theta_i N_i(X(k)) \tag{26}$$

where $N_i(\mathrm{X}(k))$ is the i^{th} normalized basis function, θ_i is the corresponding weight and the kernel is defined by the Gaussian function as given by (4),

$$K_i(X(k)) = \exp(-\left\|X(k) - X_{\mathrm{sv}}(i)\right\|^2 / 2\gamma^2) \tag{27}$$

The *SVRBFN* (28) is a kernel-based *RBFN* with clustered partitioning of the input space, where the centres of the *RBF* are given by the *SV*s. It is shown in (Chan *et al.* 2001) that the variance of the modelling errors of the *SVRBFN* is bounded by ε^2.

Since the *SVRBFN* is a linear-in-weight network, the weights can be computed by the linear least squares method. In vector notation, the *SVRBFN* (26) becomes,

$$y(k) = B^T(X(k))\theta \tag{28}$$

where $B(X(k)) = [N_1(X(k)), \ldots, N_{nsv}(X(k))]^T$ and $\theta = [\theta_1, \ldots, \theta_{nsv}]^T$. The estimate of $y(k)$ is given by,

$$\hat{y}(k) = B^T(X(k))\hat{\theta} \tag{29}$$

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y \tag{30}$$

where $$\Phi = \begin{bmatrix} N_1(X(1)) & \cdots & N_{n_{sv}}(X(1)) \\ \vdots & & \vdots \\ N_1(X(N)) & \cdots & N_{n_{sv}}(X(N)) \end{bmatrix} \tag{31}$$

$$Y = [y(1) \quad \cdots \quad y(N)]^T \qquad (32)$$

The training procedure of the *SVRBFN* is as follows.

Step 1 Normalize the input-output data to be within the range [0,1]. Choose the spread γ of the Gaussian function (29) to ensure a thorough coverage of the input space.

Step 2 Select ε and C, and obtain the *SV*s by minimizing $L(\overline{\alpha}, \underline{\alpha})$ subject to the constraints given by (23).

Step 3 Using the *SV*s as the centres of the normalized basis functions, compute $B(X(k)) = [N_1(X(k)),\ldots, N_{nsv}(X(k))]^T$ in (30). Obtain the linear least squares estimate of the weights $\hat{\theta}$ from (32).

Step 4 Evaluate the modelling errors of the *SVRBFN*. Choose another ε, if necessary, and repeat Steps 2 and 3.

Step 5 Compute the estimated output from (29), then use the scaling factors in Step 1 to re-scale the estimated output back to the original range.

5. EXAMPLES

To compare the generalization results of the *RBFN* obtained by the *OLS* algorithm and the *SVRBFN*, two examples are presented here. The first one involves a simulated nonlinear dynamic system and the second one, the river discharges and rainfalls data for the Fuji River.

Example 1 - Nonlinear System

Consider the nonlinear system (Brown and Harris, 1994),

$$\begin{aligned} y(k) = {} & [0.8 - 0.5\exp(-y^2(k-1))]y(k-1) \\ & - [0.3 + 0.9\exp(-y^2(k-1))]y(k-2) \\ & + 0.1\sin(\pi y(k-1)) + e(k) \end{aligned} \qquad (33)$$

where $e(k)$ is a white noise. From (33), m = 2, and $X(k) = [y(k\text{-}1), y(k\text{-}2)]^T$. For $e(k) \equiv 0$, the output of the nonlinear system is a spiral starting from the initial condition $X(1) = [0.1, 0.1]^T$ moving towards a globally attracting limit cycle, as shown in Fig. 1. For $e(k) \sim N(0,0.1^2)$, 300 data are generated with $X(1) = [0, 0]^T$, as shown in Fig, 2.

Only the first 50 data are used to determine the structure of the *SVRBFN*, as the remaining data are on the limit cycle, and hence they do not offer new information for modelling. From the procedure described in section 4, 10 *SV*s are obtained for $\gamma^2 = ½$, $\varepsilon = 0.2$ and $C = 300$, and are marked by circles in Fig. 2. The variance of the modelling errors is 0.0099, and the mean is approximately zero. From the estimated weight $\hat{\theta}_i$, $\hat{y}(k)$ is computed from (28), and the iterative map is shown in Fig. 3.

The *OLS* algorithmmethod given by (17) to (19) is now used to select the centres of the RBFs. In this case, Φ given by (7) contains 300 columns. The same spread, $\gamma^2 = ½$, is also used.

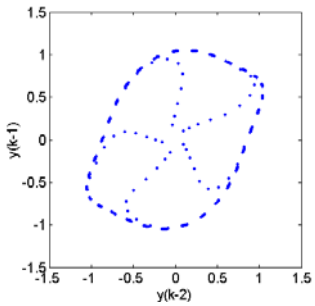


Fig. 1 Iterative map of output for $e(k) \equiv 0$

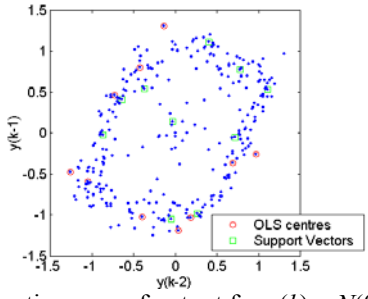


Fig. 2 Iterative map of output for $e(k) \sim N(0,0.1^2)$ showing 10 *OLS* centres and 10 *SV*s

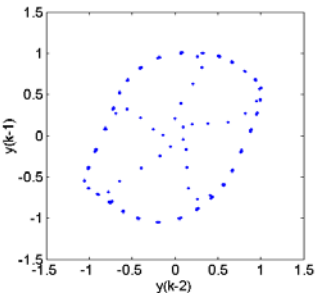


Fig. 3 Iterative map of output of *SVRBFN*

Table 1 RBF networks obtained for different ρ

ρ	0.02	0.021	0.022
OLS Centres	12	10	9
$\hat{\sigma}^2$	0.0097	0.010	0.0106

In Table 1, the data retained are referred to as the *OLS* centres for convenience. The main reason for choosing these values of ρ is that the results obtained are comparable to that from the *SVRBFN* that has 10-*SV*. Only the iterative map of the output of the *RBFN* with 12 *OLS* centres is shown in Fig. 4, as the variance of the modeling errors is the smallest, and all other three iterative maps are similar to it. Although the variance of the modelling errors of the *RBFN* with 12 *OLS* centres are smaller than that of the *SVRBFN* with 10 *SV*s, the iterative map of its output shown in Fig. 4 looks different from that of the nonlinear system shown in Fig. 1. However, the iterative map of the *SVRBFN* with 10-*SV* shown in Fig. 3 is closer to that shown in Fig. 1, indicating a better generalization result is obtained from the *SVRBFN*.

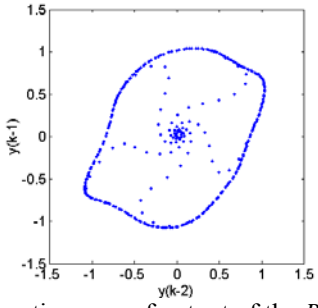


Fig. 4 Iterative map of output of the *RBFN* with 12-*OLS* centres

Example 2 - River discharges of Fuji River

Daily river discharges $y(k) m^3/s$, are collected at Kitamat-suno gauging station, 10.7 *km* from the river mouth with a catchment area of 3540 km^2. Rainfall $u(k)$ *mm* are collected daily at 10 weather stations in and around the basin (Kamikuishiki, Nakatomi, Kawaguchiko, Yamana-ka, Nanbu, Ooizumi, Nirasaki, Kofu, Katsunuma and Ootsuki). A total of 1461 sets of daily rainfall and river discharges data from January 1990 to December1993 are used in this example. The first 1000 normalized data are used to train the *RBFN*, whilst the remaining data are used to validate the networks. From the autocorrelation functions of river discharges, and the cross-correlation functions between rainfall and river discharges, the input $X(k)$ is (Choy and Chan, 2003):

$$X(k) = [\underline{y}(k-1),\ \underline{y}(k-2),\ \underline{u}(k),\ \underline{u}(k-1)]^T .$$

Out of the 1000 normalized data, only 53 data are used to select the *SV*s, as the computing time using all 1000 data may be substantial. These data are obtained as follows. As most data are lying within the range between 0 and 0.1, this range is divided into 10 sub-divisions. Thirty data are obtained by selecting 3 data randomly from each sub-division. The other data are the smallest and all data in the range between 0.1 and 1 (Choy and Chan, 2003). These are the training data for both the *SVRBFN* and the *RBFN* based on the *OLS*. For $\gamma = 1$ and $C = 300$, several *SVRBFN*s are obtained for different ε. The "best" model is obtained for $\varepsilon = 0.09$, as shown in Fig. 5, with 9 *SV*s marked by circles. The mean of the modelling errors is approximately zero and the variance is 88.68^2. Four outliers are observed in the modelling errors, which may arise from the data collection process. Since rainfalls are accumulated over a day, whilst the river discharge data are measured at regular intervals, discrepancies may therefore arise from the data collection process (Choy and Chan, 2004). Intervention variables are introduced to remove these outliers (Box, *et al.*, 1994). The procedure for selecting the *SVRBFN* is repeated with the adjusted data with $\gamma = 0.7$, $C = 300$, $\varepsilon = 0.09$. Ten *SV*s are selected and the variance of the modelling errors is 44.40^2.

The *SVRBFN* is validated by using it to predict the river discharges for the period from 1001 to 1461. The variance of the prediction errors is 78.32^2, and reduces to 55.19^2 after two further outliers in the residuals are removed, giving a result close to that obtained previously.

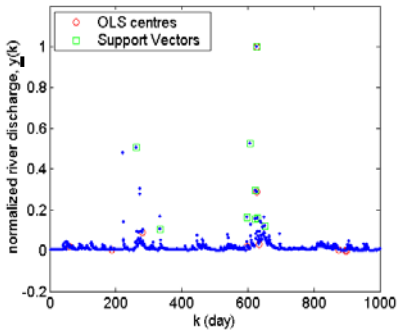


Fig. 5 10 *OLS* centres vs 10 *SV*s

The *OLS* learning algorithm is now used to identify the *RBF* centres using all of the 1000 data. With $\gamma = 1$, $\rho = 0.057$, 10 *OLS* centres are obtained. The variance of the modelling errors is 74.69^2 and 6 outliers are observed. As discussed above, they are removed using 6 intervention variables, and the variance of the modelling errors is 45.32^2. The variance of the prediction errors using validation data is 84.18^2, and 2 outliers are observed. Excluding these 2 outliers, the variance is 60.62^2. Clearly, the *SVRBFN* performs slightly better than the *RBFN* obtained from the *OLS* learning algorithm.

To compare the performance of these networks, the case where it rains for two consecutive days is considered. The prediction of the river discharges for a 5-day period after raining for the first two days are computed from these networks with intervention variables. For convenience, normalized data are used in the following comparison. The normalized rainfall for day 1 is fixed at 0.4, giving $\underline{u}(1) = 0.4$, and $\underline{u}(2)$ is set to a value varying from 0 to 1 with an increment of 0.05. The prediction of river discharge from the *SVRBFN* for these rainfalls is shown in Fig. 6, showing the highly nonlinear relationship.

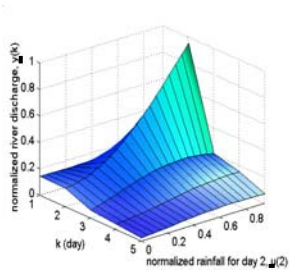
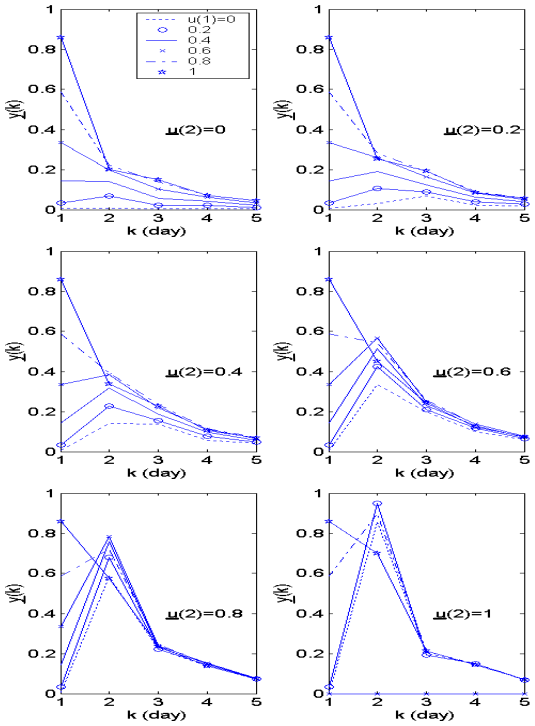


Fig. 6 River discharges after raining for 2 consecutive days with $\underline{u}(1) = 0.4$

The procedure is repeated for both networks with $\underline{u}(2)$ set to 0.4, whilst $\underline{u}(1)$ varies from 0 to 1 by an increment of 0.2. The results are plotted in Fig. 7. It is observed that the predicted river discharges in day 2 from the *RBFN* based on the *OLS* algorithm are almost the same irrespective of the rainfalls on that

day. This result does not seem to be sensible. In contrast, the predicted river discharges in day 2 from the *SVRBFN* varies in-line with changes in the rainfall on that day, showing clearly that the generalization results obtained from the *SVRBFN* is more sensible than that obtained from the *OLS* algorithm.

(a) *RBFN* using *OLS*



(b) *SVRBFN*

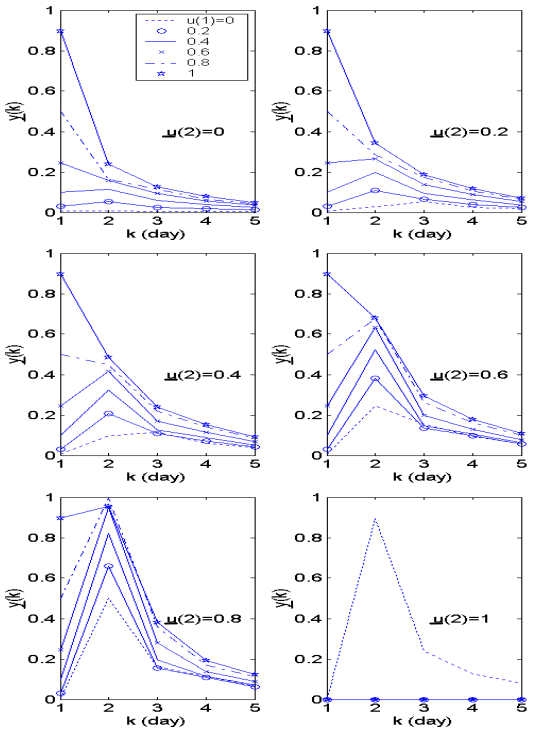


Fig. 7 River discharges after raining for the first 2 days

6. CONCLUSIONS

The selection of the structure of *RBFN* is investigated. From simulated and real data, it is found that the generalization results of the network with its structure selected taking into account the model complexity, i. e., the *SVRBN* is much better than the one that considered only the variance of the modelling errors, i.e., the *OLS* algorithm. This result illustrates the importance of the model complexity in the structure selection of *RBFN*s.

REFERENCES

Box, G. E. P., Jenkins, G. M. and Reinsel, G. C., 1994, *Time Series Analysis — Forecasting and Control* (Englewood Cliffs, N. J.: Prentice Hall).

Brown, M., and Harris, C. J., 1994, *Neurofuzzy Adaptive Modelling and Control* (New York: Prentice Hall).

Chan, W. C., Chan, C. W., Cheung, K. C., and Harris, C. J., 2001, On the modelling of nonlinear dynamic systems using support vector neural networks. *Eng. App. of Artificial Intelligence*, 14, pp. 105-113.

Chen, S., Billings, S. A., and Luo, W., 1989, Orthogonal least squares methods and their application to non-linear system identification. *Int. J. of Contr.*, 50, pp. 1873-1896.

Chen, S., Cowan, C. F. N., and Grant, P. M., 1991, Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks*, 2, pp. 302-309.

Choy, K. Y., and Chan, C. W., 2003, Modelling of River Discharges and Rainfall using Radial Basis Function Networks Based on Support Vector Regression. *Int. J. of Systems Science*, 34, pp. 763-773.

Jang, J.–S. R., Sun, C.-T., and Mizutani, E., 1997, *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence* (Upper Saddle River, NJ: Prentice Hall).

Powell, M. J. D., 1985, Radial basis function for multivariable interpolation: a review. *IMA Conference on Algorithms for the Approximation of Functions and Data*, RMCS, Shrivenham.

Schölkopf, B., and Smola, A. J., 2002, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Cambridge, Mass.: MIT).

Vapnik, V. N., 1998, *Statistical Learning Theory*. (New York: John Wiley & Sons).