

Significant Vector Learning to Construct Sparse Kernel Regression Models

Junbin Gao*

*School of Computer Science
Charles Sturt University, Bathurst, NSW 2795, Australia
Email: jbgao@csu.edu.au*

Daming Shi

*School of Computer Engineering
Nanyang Technological University, Singapore 639798
Email: asdmshi@ntu.edu.sg*

Xiaomao Liu

*Department of Mathematics
Huazhong University of Science and Technology, Wuhan 430074, China*

Abstract

A novel significant vector (SV) regression algorithm is proposed in the paper based on the analysis on Chen's orthogonal least squares (OLS) regression algorithm. The proposed regularized SV algorithm finds the significant vectors in a successive greedy process in which, compared to the classical OLS algorithm, the orthogonalization has been removed from the algorithm. The performance of the proposed algorithm is comparable to the OLS algorithm while it saves a lot of time complexities in implementing orthogonalization needed in the OLS algorithm.

1 Introduction

In practical nonlinear data modelling more and more interests are paid to the basic principle of parsimonious models that ensure the smallest possible model that explains the data well. Apart from obvious computational advantage, small models often generalize better for the unseen data. In recent years the support vector machine (SVM) (Vapnik, 1998) and kernel machine models (KMM) (Schölkopf and Smola, 2002; Chen, 2006) considerably attract one's interests. These techniques have been gaining more and more popularity and have been regarded as the state-of-art technique for regression and classification problems with tremendously successful applications in many areas. The theoretical fundamental of SVM is the structural risk minimization principle which results in excellent generalization properties with a sparse model representation (Poggio and Girosi, 1998). However it has been shown that the standard SVM technique is not always able to construct parsimonious models in system identification (Drezet and Harrison, 1998). This inadequateness motivates exploring new methods for the parsimonious models under the framework of both SVM and KMM. Tipping (2001) first introduced

*The author to whom all the correspondence should be addressed.

the relevance vector machine (RVM) method which can be viewed from a Bayesian learning framework of kernel machine and produces an identical functional form to the SVM/KMM. The results given in (Tipping, 2001) have demonstrated that the RVM has a comparable generalization performance to the SVM but requires dramatically fewer kernel functions or model terms than the SVM. A drawback of the RVM algorithm is a significant increase in computational complexity, compared with the SVM method. Recently Chen et al (Chen, 2002, 2006; Chen et al., 2003) derived a novel method for constructing sparse kernel models based on his orthogonal least squares (OLS) algorithm (Chen et al., 1989, 1991) and kernel techniques (Schölkopf and Smola, 2002). The OLS algorithm has been demonstrated as efficient learning procedure for constructing sparse regression models and gives good performances in nonlinear system identification. There are a lot of literatures concerning the problem of regressor selection, see for example (Kruif and Vries, 2002; Gestel et al., 2003; Valyon and Horváth, 2003; Suykens et al., 2002).

The OLS algorithm involves sequential selection of the regressors, which ensures that each new regressor vector defined on the training data is orthogonal to the previous selections. It employs the well-known Gram-Schmidt orthogonalization method in applied mathematics. In choosing the best regressor, the contribution of each regressor to the modelling error decrease is measured. Each chosen regressor maximally decreases the squared error of the model output, and the method stops when this error reaches an acceptable level or when the desired number of regressors have been chosen. Following the same idea, we found that orthogonalization procedure employed by the OLS algorithm can be removed and a comparable training result can be achieved, so that we can save a lot of computational complexity in the training procedure.

In this paper, a novel approach is proposed to determine the regressors of the kernel regression modelling based on the so-called significant vectors (SV). The rest of this paper is organized as follows: In section 2, the basic idea of the OLS algorithm is reviewed and the concepts of significant vector analysis are given. In section 3, the algorithm for finding significant vectors is presented. The experiments are carried out in section 4, followed by our conclusions in Section 5.

2 The OLS Algorithm and Significant Vectors

To introduce our method, we first review the basic algorithm of OLS and follow the notations used in (Chen et al., 2003).

Consider the general discrete-time nonlinear system represented by the nonlinear model (Chen and Billings, 1989):

$$y(k) = f(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) + e(k) = f(\mathbf{x}(k)) + e(k), \quad (2.1)$$

where $u(k)$ and $y(k)$ are the system input and output variables, respectively, n_y and n_u are positive integers representing the lags in $y(k)$ and $u(k)$, respectively, $e(k)$ is the system white noise, $\mathbf{x}(k) = [y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)]^T$ denotes the system ‘‘input’’ vector, and f is the unknown system mapping. The system identification involves in construct a function (model) to approximate the unknown mapping f based on an N -sample observation data set $\mathbf{D} = \{\mathbf{x}(k), y(k)\}_{k=1}^N$, i.e., the system input-output observation data $\{u(k), y(k)\}$. The most popular class of such approximating functions is the kernel regression model of the form:

$$y(k) = \hat{y}(k) + e(k) = \sum_{i=1}^N w_i \phi_i(k) + e(k), \quad 1 \leq k \leq N \quad (2.2)$$

where $\hat{y}(k)$ denotes the ‘‘approximated’’ model output, w_i ’s are the model weights, and $\phi_i(k) = k(\mathbf{x}(i), \mathbf{x}(k))$ are the regressors generated from a given kernel function $k(\mathbf{x}, \mathbf{y})$, see (Schölkopf and Smola, 2002). If we choose $k(\mathbf{x}, \mathbf{y})$ as the Gaussian kernel, then (2.2) describes a RBF network with each data as a RBF center and a fixed RBF width. The model (2.2) can be made more general if we choose each ϕ_i as different function regressors, such that it can include, for example, all the kernel based models, the polynomial-based models and all the generalized linear nonlinear model (i.e., linear-in-the-weight models). But in this paper we will focus on the case in which all the regressors ϕ_i are generated from a single kernel function just as defined in (2.2). Our analysis in this paper can be easily applied to all the other cases.

Let

$$\begin{aligned}\Phi_i &= [\phi_i(1), \dots, \phi_i(N)]^T = [k(\mathbf{x}(i), \mathbf{x}(1)), \dots, k(\mathbf{x}(i), \mathbf{x}(N))]^T; \\ \Phi &= [\Phi_1, \dots, \Phi_N]; \\ \mathbf{w} &= [w_1, w_2, \dots, w_N]^T; \\ \mathbf{y} &= [y(1), y(2), \dots, y(N)]^T; \\ \mathbf{e} &= [e(1), e(2), \dots, e(N)]^T\end{aligned}\tag{2.3}$$

then the regression model (2.2) can be written in the following matrix form

$$\mathbf{y} = \Phi \mathbf{w} + \mathbf{e}.\tag{2.4}$$

The goal is to find the best linear combination of the columns of Φ (i.e. the best value for \mathbf{w}) to explain \mathbf{y} according to some criterion. The normal criterion is to minimize the sum of squared errors,

$$E = \mathbf{e}^T \mathbf{e}\tag{2.5}$$

where the solution \mathbf{w} is called the least squares solution to the above model, which is equivalent to finding the orthogonal projection of the target vector \mathbf{y} in the subspace spanned by all the regressor vectors Φ_i given by (2.3). If the dimension of the subspace is N , then the least square solution is strict interpolation solution in which the observation dataset is exactly reproduced by the resulted model. In fact too many regressors in the model always cause the model to be over sensitive to the details of the data and result in poor generalization performance (over-fitting).

There are two main techniques to tackle with over-fitting problems. The first, regularization (Tikhonov and Arsenin, 1977; Bishop, 1991), introduces a regularized measure on the weight parameters w_i , for example, the well-known ridge regression method etc. The second way to avoid over-fitting is to reduce the number of regressors used in model (2.2), i.e., explicitly limit the complexity of the model. This method has the added advantage of producing parsimonious models. Thus, the object is to select the smallest number of n_M of regressor functions (i.e., to select n_M columns from the matrix Φ) that can model the training data to the desired degree of accuracy. The resulted model has the parsimonious property. There is a serious problem in selecting the optimal regressors in the sense that there are a lot of different possible ways to choose n_M columns from a $N \times N$ matrix Φ . The number of possible selection is $\frac{N!}{n_M!(N-n_M)!}$.

An interesting and powerful method for choosing the subset n_M of N regressor functions is the orthogonalization procedure (Chen et al., 1991) and its improved version (Orr, 1995). The sort of methods is referred to as the forward selection scheme as the regressors are picked one at a time from the pool of columns of Φ and added to an initially empty subset model until some criterion is met.

To find the contributions and the output from different regressor vectors, these vectors are first orthogonalized with respect to each other

$$\Phi = \mathbf{S}\mathbf{A}$$

where \mathbf{A} is an upper triangular matrix with 1 as the diagonal elements and all the columns of $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N]$ are orthogonal. Thus model (2.4) is converted into a new one:

$$\mathbf{y} = \mathbf{S}\mathbf{g} + \mathbf{e}. \quad (2.6)$$

with new weight parameters $\mathbf{g} = [g_1, g_2, \dots, g_N]^T$. The least squares criterion gives the estimate

$$\hat{g}_i = \frac{\mathbf{s}_i^T \mathbf{y}}{\mathbf{s}_i^T \mathbf{s}_i} \quad (2.7)$$

And the relationship between the error \mathbf{e} and the output \mathbf{y} is given by

$$\mathbf{y}^T \mathbf{y} = \sum_{i=1}^{n_M} \hat{g}_i^2 \mathbf{s}_i^T \mathbf{s}_i + \mathbf{e}^T \mathbf{e} \quad (2.8)$$

where n_M is the number of significant regressors in the model. The error reduction ratio due to \mathbf{s}_i is defined as

$$[err]_i = \hat{g}_i^2 \frac{\mathbf{s}_i^T \mathbf{s}_i}{\mathbf{y}^T \mathbf{y}} \quad (2.9)$$

The new regressor candidate is chosen from the remaining orthogonalized regressors so that the newly chosen regressor makes the maximal error reduction ratio. If we take a close look at the error reduction ratio and substitute (2.7) into (2.9), we can obtain

$$[err]_i = \frac{(\mathbf{s}_i^T \mathbf{y})^2}{(\mathbf{s}_i^T \mathbf{s}_i)(\mathbf{y}^T \mathbf{y})} \quad (2.10)$$

which is the squared cosine value of the angle between the vector \mathbf{s}_i and \mathbf{y} (or the normalized inner product between the two vectors). As we know that the inner product between two vectors gives the similarity of the vectors, we can say the maximal error reduction criterion chooses the orthogonalized regressor \mathbf{s}_i which gives the maximal similarity to the target vector. In other words, the regressor vector gives the maximal similarity to the target vectors that mostly explain the data. The above analysis suggests that we could define a set of significant vectors which mostly explain the data in a forward greedy procedure.

At the beginning the set of significant vectors is empty. Denote $\mathbf{y}^{(0)} = \mathbf{y}$. Then for each regressor vector Φ_i from the columns of Φ we solve a one-parameter least square problem defined as

$$\min_{\omega_i^{(1)}} \|\mathbf{y} - \omega_i^{(1)} \Phi_i\|^2 = \sum_{k=1}^N (y(k) - \omega_i^{(1)} \Phi_i(k))^2$$

Suppose the solution to the above least squares problem is $\omega_i^{(1)*}$. Then find the first significant regressor vector Φ_{i_0} from

$$\Phi_{i_1} = \min_{\Phi_i \in \Phi} \|\mathbf{y} - \omega_i^{(1)*} \Phi_i\|^2$$

Now drop Φ_{i_1} from the columns of Φ and denote the remnant by $\Phi^{(1)}$. Then the second significant regressor vector will be chosen from the columns of the new $\Phi^{(1)}$. As there is no orthogonality between the columns of Φ , the selection criterion should be rectified. It is better for us to choose the second significant regressor vector such that it will mostly explain the residual between the target vector and the first significant regressor vector, i.e., $\mathbf{y}^{(1)} = \mathbf{y}^{(0)} - \omega_{i_1}^{(1)*} \Phi_{i_1}$. Thus we define the second significant regressor vector as

$$\Phi_{i_2} = \min_{\Phi_i \in \Phi^{(1)}} \min_{\omega_i^{(2)}} \|\mathbf{y}^{(1)} - \omega_i^{(2)} \Phi_i\|^2$$

Generally, suppose that, at the time m , we have a set of significant regressor vectors $\{\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_{m-1}}\}$. Denote by $\mathbf{y}^{(m-1)}$ the residual vector incurred by the least squares approximation by the subspace spanned by $\{\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_{m-1}}\}$ and $\Phi^{(m)}$ the remnant regressor vectors from Φ by removing $\{\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_{m-1}}\}$. For each $\Phi_i \in \Phi^{(m)}$, solve a one-parameter least square problem,

$$\omega_i^{(m)*} = \arg \min_{\omega_i^{(m)}} \|\mathbf{y}^{(m-1)} - \omega_i^{(m)} \Phi_i\|^2, \quad (2.11)$$

Totally there are $N - m + 1$ such problems and it are easy to solve them. Then the m th significant regressor vector is defined by

$$\Phi_{i_m} = \min_{\Phi_i \in \Phi^{(m)}} \min_{\omega_i^{(m)}} \|\mathbf{y}^{(m-1)} - \omega_i^{(m)} \Phi_i\|^2 = \min_{\Phi_i \in \Phi^{(m)}} \|\mathbf{y}^{(m-1)} - \omega_i^{(m)*} \Phi_i\|^2. \quad (2.12)$$

The estimated parameter $\omega_i^{(m)*}$ from (2.11) will be the m th coefficient in the original model (2.2).

It is easy to prove that

$$\begin{aligned} \|e^{(m)}\|^2 &= \|\mathbf{y}^{(m-1)} - \omega_i^{(m)*} \Phi_i\|^2 \\ &= \mathbf{y}^{(m-1)T} \mathbf{y}^{(m-1)} - \frac{(\Phi_i^T \mathbf{y}^{(m-1)})^2}{\Phi_i^T \Phi_i} \end{aligned}$$

Hence

$$\frac{\|e^{(m)}\|^2}{\mathbf{y}^{(m-1)T} \mathbf{y}^{(m-1)}} = 1 - \frac{(\Phi_i^T \mathbf{y}^{(m-1)})^2}{(\mathbf{y}^{(m-1)T} \mathbf{y}^{(m-1)})(\Phi_i^T \Phi_i)} \quad (2.13)$$

Thus the criterion for choosing a new regressor vector used in (2.12) is similar to the one used in OLS, see (2.9).

In this paper, the above procedure for selecting significant regressor vectors is called the *Significant Vector* (SV) algorithm. The input vectors $\{\mathbf{x}(i_1), \dots, \mathbf{x}(i_m)\}$ corresponding to $\{\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_m}\}$ are called the *Significant Vectors*.

One of interesting questions is that when the above procedure should be terminated, i.e., how many significant vectors should be selected such that the resulting model can be generalized better. One simple criterium is that the procedure would be terminated if the residual vector satisfies a given threshold or a given number of the significant vectors has been achieved. In fact we know the relative error $\frac{\|e^{(m)}\|^2}{\mathbf{y}^{(m-1)T} \mathbf{y}^{(m-1)}}$ at the m th step is determined by the squared cosine value of the angle between the residual vector $\mathbf{y}^{(m-1)}$ and the regressor vector Φ_i , see

(2.13). Similar to the stopping criteria of the OLS algorithm, we terminate the procedure if for one of the remnant regressor vectors Φ_i ,

$$\frac{(\Phi_i^T \mathbf{y}^{(m-1)})^2}{(\mathbf{y}^{(m-1)T} \mathbf{y}^{(m-1)})(\Phi_i^T \Phi_i)} > \xi$$

where $0 < \xi < 1$ is a pre-specified credit value.

Obviously the resulting model may cause an overfitting problem as we will see in numerical example 1 in section 4. To avoid overfitting one may use more sophisticated stopping criteria, for example the one introduced in (Kruif and Vries, 2002) which can be used in our procedure, or one may use a regularized technique, see section 4.

Obviously the algorithm proposed here is sub-optimal indeed in contrast to the OLS algorithm. However the orthogonalization process, which costs a lot of computational complexity in the OLS algorithm, is omitted in the new algorithm. In section 4, we will demonstrate that the performance of the new algorithm is comparable to the OLS algorithm but the computational complexity has decreased. To compare the computational complexity of the SV algorithm with that of OLS algorithm, we simply calculate the number of multiplication used in both algorithms. From the calculation formulae used in (Chen et al., 1991), one can see that, at step m , the number of multiplication is $(3N + 1) * (N - m + 1)$ in orthogonalization process, and is $(3N + 1)$ for getting the elements in the new weight vector \mathbf{g} . One may note that finding the solution to a one-parameter least square problem only needs $2N + 1$ multiplications where N is the length of vector, thus the total number of multiplications for finding the m th significant vector is $(2N + 1) * (N - m + 1)$. Overall to select the first m_0 regressor vectors one needs roughly $3N^2 m_0$ multiplications by OLS algorithm, and $2N^2 m_0$ multiplications by the SV algorithm, respectively.

Another advantage of our SV algorithm is that it is easily generalized to the case of non-square error measures. For example, if we use the absolute error function instead of the square error defined by (2.5), i.e.,

$$E = |\mathbf{e}| = \sum_{k=1}^N |e(k)|$$

then OLS algorithm won't work well, because the optimal vector in the linear span space of regressor vectors is not the projection vector of \mathbf{y} onto the space in the common sense. However the significant vector algorithm still works well in which one only needs to solve one-parameter least absolute estimate problem, that is, for a given regressor Φ_i solving

$$\min_{\omega} \|\mathbf{y} - \omega \Phi_i\|_1 = \sum_{k=1}^N |y(k) - \omega \Phi_i(k)| \quad (2.14)$$

It is easy to find out ω from the problem (2.14) without using any optimal procedures.

Further we can apply the significant vector algorithm to the support vector machine regression involving with the so-called ϵ -insensitive error function, see (Suykens et al., 2002),

$$E = |\mathbf{e}|_{\epsilon} = \sum_{k=1}^N |e(k)|_{\epsilon}$$

where

$$|e(k)|_\epsilon = \begin{cases} 0 & \text{if } |e(k)| \leq \epsilon \\ |e(k)| - \epsilon & \text{if } |e(k)| > \epsilon \end{cases}$$

The proposed SV algorithm is similar to the one derived by Friedman (2001) in the sense of greedy feature, i.e., we prefer to the vectors who make the “best” explanation to the target in the sense of least square errors. One advantage of the new algorithm is that only one-parameter least squares problems are solved in each stage to find a new significant vector, thus speed of the algorithm is very fast as we have demonstrated.

It is also worthwhile to mention the link between the proposed method and the algorithm derived by Orr (1995). In (Orr, 1995) the criterion for selecting new regressor vectors in each stage is based on the amount of the error incurred by the subspace spanned by the new vector and all the other vectors selected in the previous stages. The criterion is global in the sense of approximation capability to the target vector by the subspace spanned by the selected regressor vectors. As pointed out by Orr (1995) the computational complexity of Orr’s algorithm is significantly increased when more and more vectors are selected in the procedure. Thus in order for the computational cost to be reduced the orthogonalization idea was borrowed from Chen’s OLS algorithm in (Orr, 1995).

3 Regularized Significant Vector Algorithm

Like the original OLS algorithm (Chen et al., 1989) the algorithm proposed in the last section suffers from the over-fitting to the noise in the data which results in poorer performance in generalization for testing samples. A numerical example has been shown in Figure 1(a) where a Gaussian radial basis function (RBF) network was used for modelling the scalar function

$$f(x) = \sin(2\pi x), \quad 0 \leq x \leq 1.$$

The Gaussian kernel function used had a variance of 0.04. One hundred training data were generated from $y = f(x) + \epsilon$, where x was taken from the uniform distribution in $(0, 1)$ and the noise ϵ had a Gaussian distribution with zero mean and variance 0.16. At the beginning, there were $N = 100$ regressors in model 2.2. After training using the proposed algorithm 14 significant vectors were chosen. As the training data were very noisy, the generalization performance is very poor as seen in Figure 1(a).

As done in the relevance vector machine (RVM) (Tipping, 2001), an equivalent regularization formula can be adopted in the significant vector algorithm for the regularized objective. The regularized significant vector algorithm is based on the following regularized error criterion

$$E(\mathbf{w}, \boldsymbol{\alpha}, \beta) = \beta \mathbf{e}^T \mathbf{e} + \sum_{i=1}^{n_M} \alpha_i \omega_i^2 = \beta \mathbf{e}^T \mathbf{e} + \boldsymbol{\omega}^T \mathbf{H} \boldsymbol{\omega} \quad (3.1)$$

where n_M is the number of involved significant vectors, β is the noise parameter and $\mathbf{H} = \text{diag}\{\alpha_1, \dots, \alpha_{n_M}\}$ consisting of the hyperparameters used for regularizing weights. The key issue in regularized regression formulation is to automatically optimize the regularization parameter. The Bayesian evidence technique (MacKay, 1992) can readily be used for this objective. Estimating hyperparameters is implemented in a loop procedure based on the calculation of the log evidence for β and $\boldsymbol{\alpha}$ (Nabney, 2001).

Define

$$\mathbf{A} = \beta \Phi^T \Phi + \mathbf{H}$$

and

$$\gamma_i = 1 - \alpha_i (\mathbf{A}^{-1})_{ii}, \quad \gamma = \sum_{i=1}^{n_M} \gamma_i \quad (3.2)$$

Then the update formulas for hyperparameters β and α_i can be given by

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{2\omega_i^2}, \quad \beta^{\text{new}} = \frac{N - \gamma}{2\mathbf{e}^T \mathbf{e}} \quad (3.3)$$

The iterative hyperparameter and model selection procedure can be summarized:

Initialization Set initial value for β and α_i for $i = 1, 2, \dots, N$, for example, using estimated noise variance for the inverse of β and a small value 0.0001 for all α_i .

Step 1 Given the current β and α_i , use the procedure described in section 2 to select a subset model with n_M significant vectors.

Step 2 Update α_i and β using (3.3). If α_i and β remains sufficiently unchanged in two successive iterations or a pre-set maximum iteration number is reached, then stop the algorithm; Otherwise go to step 1.

4 Modelling Examples

We are now in a position to compare our algorithm with Chen’s LROLS algorithm (Chen, 2006), the relevance vector machine (RVM) algorithm (Tipping, 2001) and the standard modified Gram-Schmidt (MGS) algorithm (Golub and van Loan, 1996). The RVM algorithm begins with all the regressors and unimportant regressors will be removed in the iterative procedure. It can be considered as a top-down procedure of building relevance regressors while our algorithm is a bottom-up procedure. Generally the proposed new algorithm is faster because it only involves one-parameter least square problems while in the initial steps RVM is very slow in computing large scale matrix inverses at the size of data number. As an orthogonalizing algorithm, MGS does not offer a mechanism for selecting the best column (regressor) in each step instead picking up columns in order. For a fair comparison we conducted MGS the standard evidence procedure (Nabney, 2001) for several random orders and used the mean result.

Our modelling simulation is conducted on the three examples used in (Chen, 2006) for the purpose of comparison.

Example 1: In this example we use a Gaussian radial basis function (RBF) network to model the scalar function

$$f(x) = \sin(2\pi x), \quad 0 \leq x \leq 1.$$

The width of RBF kernel function is 0.2, i.e., $\sigma^2 = 0.04$. A set of training data $\mathcal{D} = \{(x_k, t_k)\}_{k=1}^{100}$ is generated for the input x_k by drawing from the uniformly distribution over $[0, 1]$ and the target noise within t_k was given by Gaussian with zero mean and variance 0.16, i.e., the deviation 0.4. The target is quite noisy compared to the maximal target values ± 1 . The full RBF model is defined by all the RBF regressors with centers at each input training data, thus $N = 100$. As we have pointed out in section 3 without regularization the constructed models suffered from a serious over-fitting problem, see Figure 1. The regularized significant vector algorithm derived

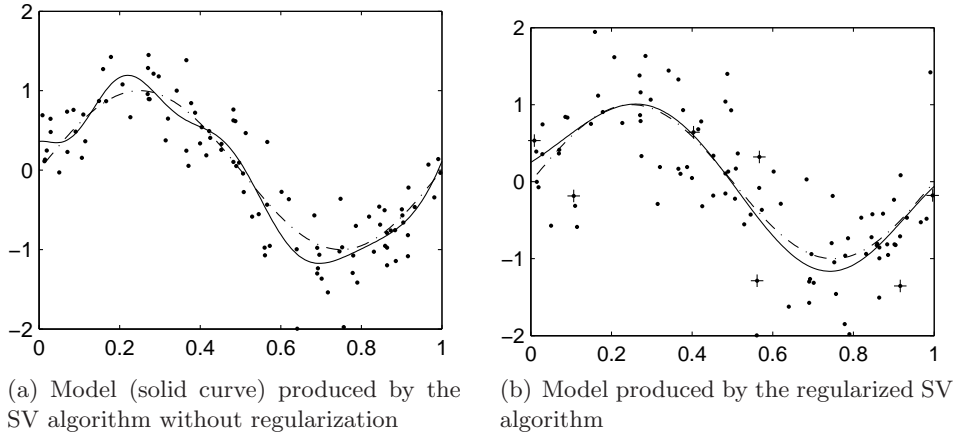


Figure 1: The results for the simple scalar function modelling problem: dots are the noise training data, the dash-curve is the underlying function $f(x)$, the solid curves are models generated from the proposed algorithms and the marker + indicates the significant vectors selected by RSV algorithm

Table 1: Mean Square Errors for Example 1

Methods	Training MSE	Test MSE
LROLS (7 regressors)	0.12482	0.00709
RVM (7 regressors)	0.12561	0.00872
SV (14 regressors)	0.11559	0.01951
RSV (6 regressors)	0.12408	0.00848
Random MGS(6 regressors)	0.13463	0.01716

in the last section was able to overcome this problem and produced a sparser six-term model, with the means square error (MSE) values over the noisy training set and the noise-free testing set being 0.12408 and 0.000848, respectively.

Table 1 compares the MSE values over the training and testing sets for the models constructed by the LROLS (Chen, 2006), the RVM algorithm, the standard SV algorithm (without regularization), the regularized significant vector algorithm (RSV) and the regressor random selection algorithm. To do a fair comparison, we randomly select 6 regressors at a time and run the standard evidence procedure (Nabney, 2001) 50 folds. The mean training error and test error are listed in column 5 of Table 1. Obviously the result of RSV algorithm is better than that of regressor random selection. The result given by RSV is comparable to the result generated by LROLS algorithm while the computational cost for orthogonalization had been saved in RSV algorithm. The model map of the 6-term model produced by the RSV algorithm is shown in Figure 1 (b) where the significant vectors (or selected regressors) are marked as +. We can see some of the significant vectors are close to the boundary of the data set and at the places where the data change shape (intrinsic features among the data).

Example 2: This is a two-dimensional simulated nonlinear time series given by

$$\begin{aligned}
y(k) = & (0.8 - 0.5 \exp\{-y^2(k-1)\})y(k-1) - (0.3 + 0.9 \exp\{-y^2(k-1)\})y(k-2) \\
& + 0.1 \sin(\pi y(k-1)) + e(k)
\end{aligned} \tag{4.1}$$

where the noise $e(k)$ is Gaussian with zero mean and variance 0.09. We generated one thousand

Table 2: Mean Square Errors (rg=regressors)

Methods	LROLS (17 rg)	RVM (17 rg)	RSV (18 rg)	Random (18 rg)
Training MSE	0.11862	0.10219	0.10108	0.19231
Test MSE	0.08917	0.09017	0.09299	0.10013

Table 3: Mean Square Errors (regrs=regressors)

Methods	LROLS (34 rg)	RVM (36 rg)	RSV (36 rg)	Random (36 rg)
Training MSE	0.000439	0.000436	0.000524	0.001673
Test MSE	0.000485	0.000487	0.000438	0.001538

noisy samples with the initial conditions $y(0) = y(1) = 0.0$. The first 500 data points were used for training, and the other 500 samples were used for possible cross-validation. The underlying noise-free system was specified by a limit circle, as shown by the one thousand samples given in Figure 2 (b) with initial value $y(0) = y(-1) = 0.1$. We use a Gaussian RBF model in the form

$$\hat{y}(k) = \hat{f}_{\text{RBF}}(\mathbf{x}(k)) \quad \text{with} \quad \mathbf{x}(k) = [y(k-1), y(k-2)]^T$$

The modelled results with 18 significant vectors are shown in Figure 2. Figure 2(c) plots the result generated by one-step prediction from the learnt model and Figure 2(d) shows the model output generated by iterative model prediction. As the SV algorithm is obviously inferior to all the other algorithm, we did not test the SV algorithm for this example. Both the training and test MSEs for all the four algorithms are reported in Table 2. The result of RSV is comparable to that given by the LROLS and RVM with lower computational complexity as mentioned before and better than that given by the random MGS.

Example 3: The third example is a practical modelling problem. In this example, we are about to construct a model representing the relationship between the fuel rack position (input) and the engine speed (output) for a Leyland TL11 turbocharged, direct inject diesel engine operated at low engine speed. Detailed system description and experimental setup can be found in (Billings et al., 1989). The data set consists of 410 samples. We use the first 210 data points as training data in modelling and the last 200 points in model validation. An RBF model of the form

$$\hat{y}(k) = \hat{f}_{\text{RBF}}(\mathbf{x}(k)) \tag{4.2}$$

but this time the input vector $\mathbf{x}(k)$ is defined as

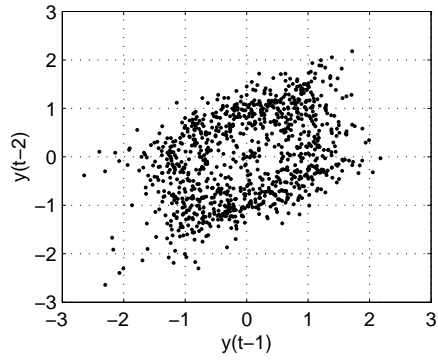
$$\mathbf{x}(k) = [y(k-1), u(k-1), u(k-2)]^T \tag{4.3}$$

where u means the fuel input. The variance of the RBF kernel function was chosen to be 1.69. The total number of regressors is $N = 210$ in the initial stage. By running RSV algorithm a model with 36-term significant regressors was constructed with MSE values over the training and testing data were 0.0005241 and 0.0004379 respectively, see Table 3. The result of RSV is still comparable to the ones given by other algorithms.

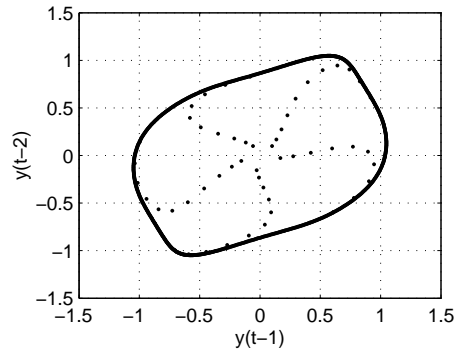
The constructed RBF model by the RSV algorithm was used to generate the one-step prediction $\hat{y}(k)$ of the system output according to (4.3). The iterative model output $\hat{y}_d(k)$ was also produced by (4.2) with

$$\mathbf{x}_d(k) = [\hat{y}_d(k-1), u(k-1), u(k-2)]^T \tag{4.4}$$

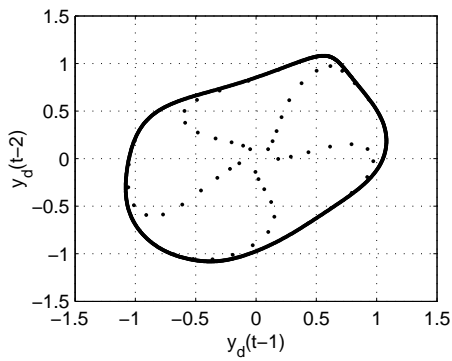
The one-step model prediction and iterative model output for this 36-term model selected by RSV algorithm are shown in Figure 3 in comparison with the system output.



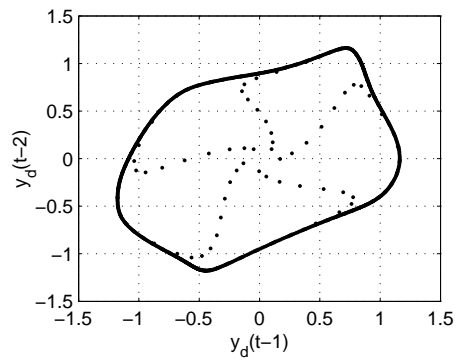
(a) Phase plot of noisy training data set ($y(0) = y(-1) = 0$)



(b) Phase plot of noise-free data generated by (4.1) without $e(k)$

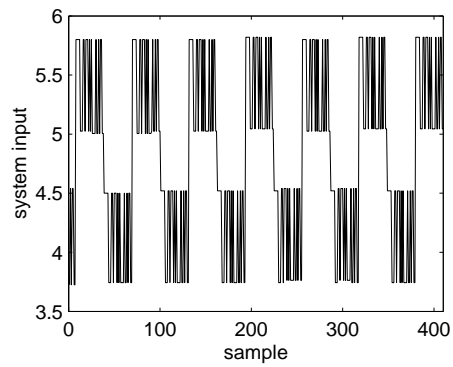


(c) Phase plot of one-step model prediction

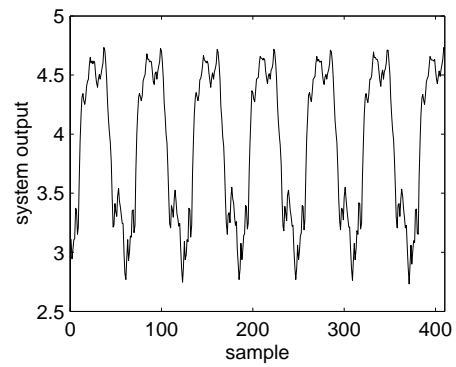


(d) Phase plot of iterative model prediction

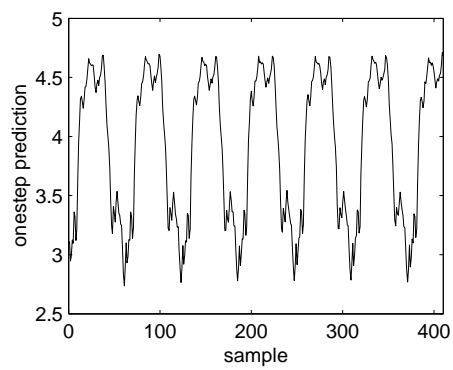
Figure 2: The results for modelling the nonlinear system defined by (4.1)



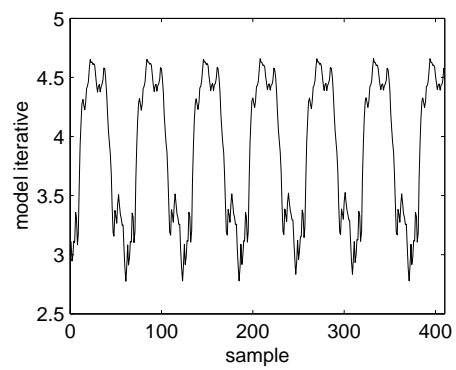
(a) Engine input data $u(k)$



(b) Engine output data $y(k)$



(c) Output plot of one-step model prediction



(d) Output plot of iterative model prediction

Figure 3: The results for modelling the relationship between the engine speed and the fuel rack position

5 Conclusions

The regularized significant vector algorithm has been proposed for nonlinear system identification using the kernel regression model. Compared to the LROLS algorithm the new algorithm has less computational complexity by removing the orthogonalization procedure employed in LROLS while the overall performance offered by the RSV algorithm is considerably comparable to the results given by the LROLS algorithm, which has been demonstrated by three modelling problems. As reported in (Chen, 2006) the LROLS is comparable to the RVM algorithm but as a top-down pruning procedure the RVM has higher computational complexity than both the LROLS and the proposed new algorithm. The computational requirements of this iterative model algorithm are very simple and its implementation is straightforward. The core idea can be easily extended to other cases such as robust loss measures and error/loss functions for classification problems.

Acknowledgements

The authors are grateful to anonymous reviewers for their constructive suggestion. Also thanks to Prof. Sheng Chen of University of Southampton, UK, who made a lot of useful comments and provided us the raw data used in Example 3. This work is supported by the grant number 60373090 from the National Natural Science Foundation of China (NSFC) and by the internal grant from Charles Sturt University.

References

- Billings, S., S. Chen, and R. Backhouse (1989). The identification of linear and nonlinear models of a turbocharged automotive diesel engine. *Mech. Syst. Signal Processing* 3(2), 123–142.
- Bishop, C. (1991). Improving the generalization properties of radial basis function neural networks. *Neural Computation* 3(4), 579–581.
- Chen, S. (2002). Locally regularized orthogonal least squares for the construction of sparse kernel regression models. In *Proceeding of 6th Int. Conf. Signal Processing*, Volume 2, Beijing, China, pp. 1229–1232.
- Chen, S. (2006). Local regularization assisted orthogonal least squares regression. *NeuroComputing* 69, 559–585.
- Chen, S. and S. Billings (1989). Representations of nonlinear systems: the NARMAX model. *International Journal of Control* 49(3), 1013–1032.
- Chen, S., S. Billings, and W. Luo (1989). Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control* 50(5), 1873–1896.
- Chen, S., C. Cowan, and P. Grant (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks* 2, 302–309.
- Chen, S., X. Hong, and C. Harris (2003). Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design. *IEEE Trans. Automatic Control* 48(6), 1029–1036.

- Drezet, P. and R. Harrison (1998). Support vector machines for system identification. In *Proceeding of UKACC Int. Conf. Control'98*, Swansea, U.K., pp. 688–692.
- Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29(5), 1189–1232.
- Gestel, T., M. Espinoza, J. Suykens, C. Brasseur, and B. deMoor (2003). Bayesian input selection for nonlinear regression with LS-SVMS. In *Proceedings of 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, pp. 27–29.
- Golub, G. and C. van Loan (1996). *Matrix Computations* (3 ed.). Maryland: The Johns Hopkins University Press.
- Kruif, B. and T. Vries (2002). Support-Vector-based least squares for learning non-linear dynamics. In *Proceedings of 41st IEEE Conference on Decision and Control*, Las Vegas, USA, pp. 10–13.
- MacKay, D. (1992). Bayesian interpolation. *Neural Computation* 4(3), 415–447.
- Nabney, I. (2001). *NetLab: Algorithms for Pattern Recognition*. London, Berlin: Springer.
- Orr, M. (1995). Regularization in the selection of radial basis function centres. *Neural Computation* 7(3), 606–623.
- Poggio, T. and F. Girosi (1998). A sparse representation for function approximation. *Neural Computation* 10, 1445–1454.
- Schölkopf, B. and A. Smola (2002). *Learning with Kernels*. Cambridge, Massachusetts: The MIT Press.
- Suykens, J., T. van Gestel, J. DeBrabanter, and B. DeMoor (2002). *Least Square Support Vector Machines*. Singapore: World Scientific.
- Tikhonov, A. and V. Arsenin (1977). *Solution of Ill-posed Problems*. Washington, D.C.: W.H. Winston.
- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. Machine Learnign Research* 1, 211–244.
- Valyon, J. and G. Horváth (2003). A generalized LS-SVM. In J. Principe, L. Gile, N. Morgan, and E. Wilson (Eds.), *Proceedings of 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands.
- Vapnik, V. (1998). *Statistical Learning Theory*. New York: Wiley.