A novel adaptive kernel method with kernel centers determined by a support vector regression approach

L.G. Sun, C.C. de Visser, Q.P. Chu, J.A. Mulder

Delft University of Technology, Delft, The Netherlands, 2600GB

Abstract

The optimality of the kernel number and kernel centers plays a significant role in determining the approximation power of nearly all kernel methods. However, the process of choosing optimal kernels is always formulated as a global optimization task, which is hard to accomplish. Recently, an algorithm, namely improved recursive reduced least squares support vector regression (IRR-LSSVR), was proposed for establishing a global nonparametric offline model, which demonstrates significant advantage in choosing representing and fewer support vectors compared with others. Inspired by the IRR-LSSVR, a new adaptive parametric kernel method called WV-LSSVR is proposed in this paper using the same type of kernels and the same centers as those used in the IRR-LSSVR. Furthermore, inspired by the multikernel semiparametric support vector regression, the effect of the kernel extension is investigated in a recursive regression framework, and a recursive kernel method called GPK-LSSVR is proposed using a compound type of kernels which are recommended for Gaussian process regression. Numerical experiments on benchmark data sets confirm the validity and effectiveness of the presented algorithms. The WV-LSSVR algorithm shows higher approximation accuracy than the recursive parametric kernel method using the centers calculated by the k-means clustering approach. The extended recursive kernel method (i.e. GPK-LSSVR) has not shown advantage in terms of global approximation accuracy when validating the test data set without real-time updation, but it can increase modeling accuracy if the real-time identification is involved.

Keywords: support vector machine, recursive identification, adaptive model, kernel basis function

[1]

1. Introduction

Seeking optimality in selecting the number and the centers' position of the kernel basis functions has been a defining feature of applying the kernel basis function based identification method[1][2], particularly those dealing with complex dynamic systems. Given a modeling data set, the task of determining the centers and the number of the kernel basis functions becomes a global optimization problem, which usually requires a high computational load especially when the modeling data set is of large scale[2][3]. Many achievements concerning kernel selection exist in the literature. Among many examples are the *orthogonal least squares* [4], clustering algorithms such as k-means[2], and more recently, many efforts in reducing the dependency between the kernel basis number and the training data number[5][6][3]. Despite the abovementioned achievements in selecting the centers and number of the kernel basis functions, a novel type of learning machine called support vector machine (SVM) has proven to be a good alternative. As suggested by Bernhard et al. [7], the SVM has an advantage in providing high-quality centers for other kernel methods. In other words, the kernel methods using the centers calculated by SVM enjoy higher approximation accuracy than the classical radial basis function (RBF) machine.

Support vector machines (SVMs) proposed basing on structural risk minimization (SRM) principle are state of the art learning algorithms for pattern recognition and function approximation [8][1][2]. To reduce the computational complexity in training process, one of its branches, namely least squares support vector machine (LSSVM), was proposed by Suykens et al.[9] [10]. In LSSVM, only a linear equation set needs to be solved rather than a quadratic programming problem as in classical SVMs. However, LSSVM has also incurred a new problem, that is, its solution suffers from lack of sparseness and robustness[11]. In LSSVM, all training samples become support vectors owing to the equality constraints, and its loss function takes a squared errors form. As a consequence, it needs a high computational load in training and has bad robustness. To overcome these drawbacks, many efforts have been made by Suykens[12], De Kruif and De Vries[13], Hoegaerts[14], Zeng and Chen[15] and Jiao[16] et al. For large data sets, Brabanter et al.[17] and Karsmakers et al.[6] recently developed the fixed-size kernel (SVR) modeling method. More recently, a novel and much sparser

[1] Corresponding author: Liguo Sun, Email: L.SUN@Tudelft.nl

LSSVR method named improved recursive reduced LSSVR (IRR-LSSVR) is proposed by Zhao and Sun et al.[18] after combining a reduced technique [19] with the iterative strategy[16]. In each iteration, all the remaining non-support data points will be evaluated, and the data leading to a maximum reduction in the overall squared fitting errors is introduced. Above all, IRR-LSSVR exploits an improved criterion for selecting optimal support vectors recursively, which takes into account the adaptation of existing weights by the will-selected support vectors when introducing a new support vector. Consequently, IRR-LSSVR leads to a sparser SVM model when compared with other LSSVR methods. That is to say, it needs less support vectors while keeping almost the same approximation accuracy without greatly increasing the computational time in training[18].

Furthermore, the choice of kernel function type plays a paramount role in determining the modeling performance of a kernel method. When the modeling system owns different data trends in different subdomains, the kernel method using single kernel commonly cannot lead to a satisfying result, i.e. the model does not globally fit the data. Multikernel learning algorithms [20] [21][22][23] have been intensively investigated during the last decade. Their superiority of leading to high approximation accuracy and the necessity has been demonstrated by Ong et al.[22].

The objective of this paper is to present an adaptive fixed-size kernel method called WV-LSSVR, which takes advantage of IRR-LSSVR in selecting kernel centers, for global model real-time identification. Meantime, multikernel effect on the kernel basis function based modeling method is also investigated, and an online GPK-LSSVR kernel method is developed by extending the single Gaussian kernel into a compound kernel recommended for Gaussian process regression. Note that, GPK-LSSVR is an improved version of WV-LSSVR.

This paper is structured as follows. In Sec. 2, the preliminaries on reduced LSSVR is briefly introduced. The new adaptive methods will be developed in Sec. 3. Results and analysis will be given in Sec. 4. Finally, the paper is concluded by Sec. 5.

2. Preliminaries on recursive reduced LSSVR

Because the centers of the kernels used in the laterly proposed kernel method will be calculated using recursive reduced LSSVR (RR-LSSVR), the preliminaries on RR-LSSVR is briefly introduced in this section.

2.1. Reduced Least Squares Support Vector Regression

Given a training data set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^m$ is the input with m-dimension and $d_i \in \mathbb{R}$ is its corresponding model output. The least squares support vector regression is solved by solving the following optimization problem:

$$\min_{w,e,b} \left\{ \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{N} e_i^2 \right\} \tag{1}$$

s.t. $d_i = w^T \varphi(x_i) + b + e_i$, $i = 1, ..., N$. where w is the normal vector of the hyperplane, b is the offset, $e = [e_1, ..., e_N]^T$ denotes the prediction residual vector, $C \in \mathbb{R}^+$ is the regularization parameter, $\varphi(\cdot)$ is the mapping from the input space to the feature space. In literature, this constrained optimization problem is solved by introducing Lagrangian factors:

$$L(w, b, e, \alpha) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{N} \alpha_i \left(d_i - w^T \varphi(x_i) - b - e_i \right) \tag{2}$$

where α is the Lagrangian multiplier vector. The conditions for optimality are

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i), \tag{3a}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0, \tag{3b}$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = C e_i, \tag{3c}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i - d_i = 0, \tag{3d}$$

Eliminating the vectors w and e, the following linear equations set is obtained:

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & \bar{K} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ d \end{bmatrix} \tag{4}$$

where $\vec{1} = [1_1, ..., 1_N]^T$,$d = [d_1, ..., d_N]^T$,$\bar{K}_{ij} = k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) + \delta_{ij}/C$ with

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}, i, j = 1, ..., N$$

$k(x_i, x_j)$ is the kernel function on the paired input vectors $\{(x_i, x_j), i = 1, ..., N; j = 1, ..., N\}$. The commonly used kernel function is the Gaussian defined by $k(x_i, x_j) = exp\left(-\|x_i - x_j\|^2 / 2\gamma^2\right)$. After obtaining the solution α from Eq. 4, for any new testing sample $x \in \mathbb{R}^m$, the predicting value is derived as follows:

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x) + b \tag{5}$$

Assuming that $\{(\mathbf{x}_i, d_i)\}_{i=1}^{M}$ is a reduced subset of original training samples, and are used to construct an approximation to the aforementioned normal LSSVR. From Eq. 3, the sparse expression of w can be rewritten as $w = \sum_{i \in S} \alpha_i \varphi(x_i)$ with S the index subset of $\{1, ..., N\}$.

After substituting w into Eq. 1, we get the equivalent formulation of the optimization problem[24]:

$$\min_{b,\alpha_S}\left\{L\left(b\,\alpha_S\right)=\frac{1}{2}\alpha_S^TK\alpha_S+\frac{C}{2}\sum_{i=1}^{N}\left(d_i-\sum_{j\in S}\alpha_j\cdot\right.\right.$$
$$\left.\left.\varphi\left(x_j\right)^T\varphi\left(x_i\right)-b\right)^2\right\} \tag{6}$$

where $K_{ij}=k\left(x_i,x_j\right),\, i,j\in S$, α_S denotes the subvector of α indexed by S. Eq. 6 can be reformulated into the following economy form[24]:

$$\min_{b,\alpha_S}\left\{L=\left[b\,\alpha_S^T\right]\left(\begin{bmatrix}0 & \vec{0}^T\\ \vec{0} & K/C\end{bmatrix}+\left[\begin{array}{c}\vec{1}^T\hat{K}\end{array}\right]\left[\begin{array}{c}\vec{1}\\ \hat{K}^T\end{array}\right]\right)\left[\begin{array}{c}b\\ \alpha_S\end{array}\right]\right.$$
$$\left.-2\left(\left[\vec{1}^T\hat{K}\right]\mathbf{d}\right)^T\left[\begin{array}{c}b\\ \alpha_S\end{array}\right]\right\} \tag{7}$$

where $\hat{K}_{ij}=k\left(x_i,x_j\right),\, i,j\in S$, $\vec{1}$ is a vector of all ones in an appropriate dimension, $\vec{0}$ is a vector of all zeros in an appropriate dimension. Let $\partial L\ \partial b=0$ and $\partial L\ \partial\alpha_S=0$, Eq. 7 can reach its optimal value at

$$\left(R+ZZ^T\right)\left[\begin{array}{c}b\\ \alpha_S\end{array}\right]=Z\mathbf{d} \tag{8}$$

where

$$R=\begin{bmatrix}0 & \vec{0}^T\\ \vec{0} & K/C\end{bmatrix},\; Z=\left[\begin{array}{c}\vec{1}^T\\ \hat{K}\end{array}\right]$$

By solving Eq. 8, a reduced LSSVR estimator is derived for a testing sample x:

$$f\left(x\right)=\sum_{i\in S}^{N}\alpha_ik\left(x_i,x\right)+b \tag{9}$$

2.2. Improved algorithm for selecting support vectors

In IRR-LSSVR [18], an improved criterion was developed for RR-LSSVR. In each iteration, all the remaining non-support data points will be evaluated, and the data leading to a maximum reduction in the overall squared fitting errors is introduced. Above all, the adaptation of the weights by the will-selected support vectors is taken into account when introducing a new support vector.

3. Parametric kernel methods with their centers determined by LSSVR

The property of a kernel, a nonlinear mapping function from the original space to the feature space, directly determines the approximation accuracy of SVM. Meantime, sparseness plays a crucial role in making the kernel models enjoy better prediction performance in terms of generalization performance and testing time. However, the kernel center selection process usually has a high computational time requirement. Since IRR-LSSVR has an superiority in fastly selecting optimal locations and number of all kernels, it provides a tool for all the general kernel methods to determine the number and locations of the kernels when constructing global model. In both kernel methods proposed in this paper, IRR-LSSVR is utilized to deal with the pre-collected static training data pool to get the optimal support vectors $S=\{x_1,x_2,...,x_M\}$, which are laterly used as kernel centers.

In many circumstances, like for example in real-time model identification, an online global kernel method provides a good alternative to other possible online identification algorithms. In the remainder of this paper, we will focus on developing an online kernel method basing on IRR-LSSVR. As an attempt, a fixed-size online parametric kernel method has been developed by introducing classical weights updation technique after the kernel centers have been determined using IRR-LSSVR.

As a further step, kernel extention has been studied inspired by the multikernel support vector regression, which has shown its superiority in terms of improving approximation power. As indicated by Zhao et al. [23], apart from introducing new support vectors, an alternative way to enhance modeling accuracy is to expand the kernel basis functions, like for example adding a constant term and a linear term to the Gaussian kernel. Specially, kernel extention becomes essential when the local data trend among different subdomains of the system varies greatly.

3.1. Recursive kernel method using the support vectors

In IRR-LSSVR, a Gaussian kernel was adopted by Zhao et al.[18] with the expression:

$$k\left(\mathbf{x}_n,\mathbf{x}_m\right)=exp\left\{-\frac{\left\|\mathbf{x}_n-\mathbf{x}_m\right\|^2}{2\gamma^2}\right\} \tag{10}$$

Note that the Gaussian kernels are centered on the selected support vectors as shown in Eq. 9, and the outputs of the kernel basis functions $ker\left(\mathbf{x}_i,\cdot\right)$ constitute inputs in the feature space rather than outputs. As shown in Eq. 9, the weights α are associated with each of the support vectors. If we want to develop a new online fixed-size kernel method for identifying global model in real-time, we can collect abundant modeling data with enough excitation information at first, and then use IRR-LSSVR to calculate the centers for the parametric kernel method. Having determined the number and the centers of the kernels, we can then use classical recursive least squares approach to update the weights for each kernel basis functions at each time step. In each iteration, the mapping relationship between the two input spaces remains the same, but the surface shape of the implicit function defined in the high dimensional feature space is always adapted.

In this section, a novel online fixed-size kernel method is developed. This method employs Gaussian kernels as well as the kernel number and positions calculated using IRR-LSSVR. Having chosen the kernel determination technique, we can then realize an online regression framework by applying a classical recursive least squares method. In specific, the weights α are set as initial weights, which determine the initial shape of the function output surface in the feature space. Unlike the weights α, the scalar variable b shown in Eq. 9 is not updated at each time step. Since the weights of the proposed method vary with time, the method is refered to as weights varying LSSVR (WV-LSSVR) for short.

Before going further, we need to have a discussion on the WV-LSSVR. IRR-LSSVR is an offline learning machine, therefore, we need to collect, in advance, enough representing modeling data with full coverage of the system under identification. Besides, the kernel selection method (i.e. IRR-LSSVR) is a nonparametric method, and the scale of the model increases continuously with the increment of the support vector number. Apart from the kernel center selection process, the WV-LSSVR becomes a parametric online kernel method suitable for identifying dynamic global model in real-time. In sum, the WV-LSSVR is a hybrid kernel method, which comprises two processes: kernel center selection process using pre-collected modeling samples and recursive identification of the kernel weights.

3.2. Extension of the kernel basis function

A widely used kernel function for Gaussian process regression is given by the exponential of a quadratic form, with the addition of constant and linear terms [2] [pp.307]:

$$k\left(\mathbf{x}_n, \mathbf{x}_m\right) = a_{n0} \cdot exp\left\{-\frac{a_{n1}\left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2}{2}\right\} + a_{n2} + a_{n3}\mathbf{x}_n^T\mathbf{x}_m \tag{11}$$

where $\mathbf{x}_n$ is the input vector of the current evaluation data and $\mathbf{x}_m$ is one of the support vectors selected offline using IRR-LSSVR. Note that the term involving θ_3 corresponds to a parametric model that is a linear function of the input variables.

By introducing $a_{n1} = \frac{1}{\gamma^2}$, we can get

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \theta_{i1} \cdot exp\left\{-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\gamma^2}\right\} + \theta_{i2} + \theta_{i3}\mathbf{x}_i^T\mathbf{x}_j \tag{12}$$

where $\mathbf{x}_i \in S$, and $S = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ is the selected sub set of support vectors calculated using IRR-LSSVR. Once again, the kernel functions are centered on the selected support vectors, and the their bandwidth are chosen the same as those used in IRR-LSSVR. This novel adaptive kernel method will be referred to as Gaussian process kernel based LSSVR (GPK-LSSVR) in the remainder of this paper. The parameters of the kernel model that needs to be recursively updated has the following expression: $\mathbf{c} = [\theta_{11}, \theta_{12}, \theta_{13}, \cdots, \theta_{i1}, \theta_{i2}, \theta_{i3}, \cdots, \theta_{M1}, \theta_{M2}, \theta_{M3}]^\top$, where $i \in [1, \cdots, M]$ and M is the number of support vectors. In what follows, the implementing flowchart of GPK-LSSVR is depicted in Algorithm 1.

Algorithm 1. GPK-LSSVR

```
step.1 Basing on the a priori experience of the model, gather
       and select offline input-output training data set $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$
       with as large coverage of the input space as possible.
step.2 Choose the optimal support vector set $\{(\mathbf{x}_i, d_i)\}_{i=1}^M$
       from $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$ and calculate the unknown parameters
       $\alpha_S$, $b$ as shown in Eq. 8 using IRR-LSSVR from [18].
step.3 Extend the nominal Gaussian kernel shown in Eq. 10
       into the kernel shown in Eq. 12. The kernel centers
       are determined by the support vector set $\{(\mathbf{x}_i, d_i)\}_{i=1}^M$.
       The initial value of $\theta_1 = [\theta_{11}, \theta_{21}, \cdots, \theta_{M1}]$ is determined
       by $\alpha_S$, and the initial values of $\theta_2$ and $\theta_3$ are
       set to be zero.
step.4 Update the parameters $\theta_1$, $\theta_2$ and $\theta_3$ using recursive
       least squares method when evaluating new testing
       data. Execute step.4 recursively for each new data.
```

3.3. Computational complexity

According to [18], both WV-LSSVR and GPK-LSSVR have the same computational complexity in time $\mathcal{O}\left(M \cdot N^2\right)$ for each iteration in the kernel center determination process shown in step.2, where M is the number of selected support vectors and N is the total number of modeling data. The computational complexity in recursively updating the model using a newly available data is tabulated in Table 2 with m the dimension of the input variables. As shown in Table 1, the computational complexity of both methods comprises two parts: kernel basis regression vector computation $\mathcal{O}(m \cdot M)$, and the parameter updation using the recursive least squares method $\mathcal{O}\left(M^2\right)$ or $\mathcal{O}\left((3M)^2\right)$. According to Table 1, the computational complexity in evaluating a new data will increase quadratically if the number of support vectors increases.

Table 1: Computational complexity in each time step

algorithms	time	memory
WV-LSSVR	$\mathcal{O}\left(m \cdot M + M^2\right)$	$\mathcal{O}\left(m \cdot M + M^2\right)$
GPK-LSSVR	$\mathcal{O}\left(m \cdot M + (3M)^2\right)$	$\mathcal{O}\left(m \cdot M + (3M)^2\right)$

4. Experiments and Results

To demonstrate the feasibility and efficiency of WV-LSSVR and GPK-LSSVR methods, we will carry out a set of 14 experiments using benchmark data sets and a data set from a real-world object. Among the benchmark data sets, motor-UPDRS, total-UPDRS, winequality-red, winequality-white, concrete, autoMPG, abalone, Boston housing are from the well-known UCI repository [2], and stock, delta elevators, delta ailerons, kinematics, cpu-act, puma8NH are found from the itorgo collection[3]. In ad-

[2] Available from URL:http://archive.ics.uci.edu/ml/.

[3] URL:http://www.liaad.up.pt/ ltorgo/Regression/DataSets.html.

dition, one data set from a real-world object is also used to validate the methods, that is, the data from the diesel engine[25]. All experiments are finished on an Intel i5 CPU E31270 (3.40 GHz) processor with 16.00 GB RAM in a Matlab2010b environment.

The rooted mean squared errors (RMSE) and normalized mean squared errors (NMSE) criteria are used to evaluate the approximation power of the proposed methods, where the RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{d}_i - d_i\right)^2}{N}} \tag{13}$$

The NMSE is defined as:

$$NMSE = \frac{1}{\triangle^2\, N}\sum_{i=1}^{N}\left(d_i - \hat{d}_i\right)^2 \tag{14}$$

where

$$\triangle^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(d_i - \bar{d}\right)^2$$

with $\bar{d}$ the mean of the measured values, $\hat{d}_i$ the predicted value, and d_i the measured value.

4.1. Comparison with the k-means clustering approach

For comparison purpose, a classical k-means clustering (KMC) based RBF kernel method[2][7] was also implemented. Three kernel methods, named KMC-RBF, WV-LSSVR and GPK-LSSVR separately, were compared with each other with special concentration on the optimality of the selected kernel centers. The aforementioned benchmark data sets were applied to validate the performance of the kernels chosen using IRR-LSSVR and the KMC method separately. As a first step in the numerical experiments, the training data set was utilized to calculate the number and centers for all the kernel basis functions. As a second step, the same training data set was used again to train the model in a recursive manner. As a third step, the identified model from the previous process was evaluated using the testing benchmark data sets. Note that all the kernel parameters are selected the same as those used in IRR-LSSVR[18].

Fig. 1 and Fig. 2 depict the effects of the k-means clustering threshold σ and the kernel number on KMC-RBF basing on the 'total UPDRS' data set. As can be seen from Fig. 1, when the threshold σ deceases, RMSE also decreases. In addition, once σ is smaller than some value (i.e. 0.01), the influence of its reduction on the approximation accuracy becomes negligible. In Fig. 1, the RMSE first decreases, then increases, and gets its minimum around 750. Fig. 2 gives the same trend information on NMSE.

Fig. 3 and Fig. 4 give the comparison results of KMC-RBF, GPK-LSSVR and WV-LSSVR basing on the 'total UPDRS' data set. Although IRR-LSSVR is an offline

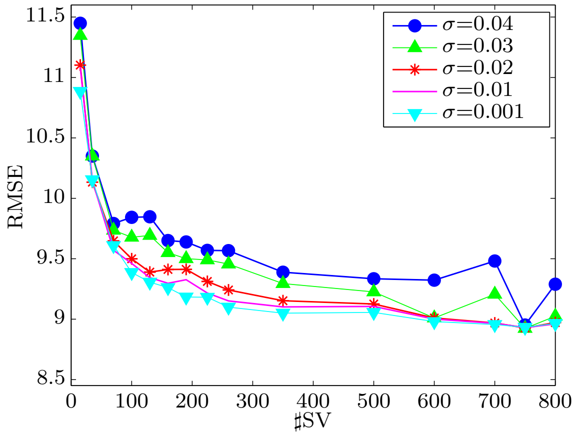

Fig. 1: RMSE of KMC-RBF, under different σ, total UPDRS.

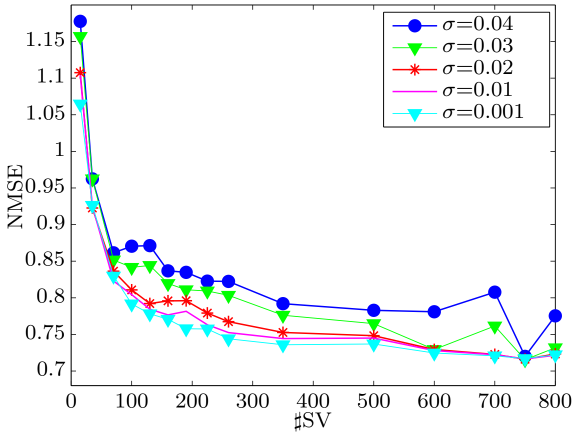

Fig. 2: NMSE of KMC-RBF, under different σ, total UPDRS.

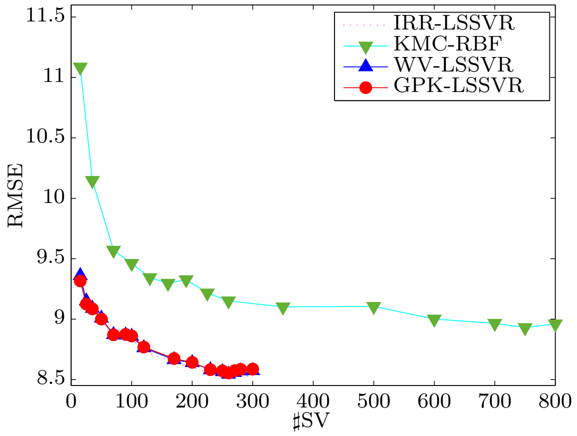

Fig. 3: RMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$, $\sigma = 0.01$.

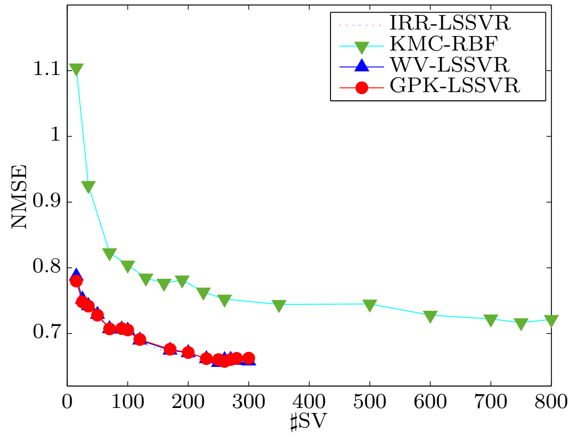

Fig. 4: NMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$, $\sigma = 0.01$.

method, it is utilized to provide a baseline for the comparison results of the abovementioned methods. The effects of different kernel number and different kernel centers on the approximation power are demonstrated. As can be seen from Fig. 3, GPK-LSSVR enjoys the same level of approximation accuracy as both WV-LSSVR and IRR-LSSVR, while it has a far more higher approximation accuracy than KMC-RBF. Due to similarity and limitation of space, except for 'total UPDRS', the comparison results for all of the other benchmark data sets are tabulated in Table 2. In all of the following numerical experiments, σ is selected as 0.01, and other parameters for IRR-LSSVR are chosen the same as those in [18].

In Table 2, seTime denotes the consumed time for kernel selection process, and trTime is the time for identifying the global model recursively. Besides, teTime represents the evaluation time using the testing samples without model adaption, while trNum and teNum stand for the number of training data points and testing data points separately. Last but not the least, ♯SV gives the number of support vectors. The numbers of the support vectors are chosen the same as those used in [18].

According to Table 2, both WV-LSSVR and GPK-LSSVR enjoy a lower RMSE when compared with the KMC-RBF method. This indicate that the kernels used in both WV-LSSVR and GPK-LSSVR methods, which are determined using IRR-LSSVR, own higher approximation power than the kernels used in KMC-RBF method, which are calculated using the KMC algorithm. When compared with WV-LSSVR, GPK-LSSVR leads to more or less the same global approximation accuracy.

Furthermore, the seTime of IRR-LSSVR is comparable to, even if it is not always smaller than, that of the KMC.

4.2. Recursive identification results on extended kernel basis functions

Indicated by [23], multikernel SVMs have demonstrated superiority in enhancing approximation power, especially when the local data trend in one sub domain differs greatly from that in another sub domain. In view of this, the effects of extending the kernel basis functions are explored in a recursive model identification framework. To illustrate the benefits of the extended kernel basis function, the GPK-LSSVR method is compared with WV-LSSVR, and the results are tabulated in Table 3. This experiment consists of two steps. At the first step, IRR-LSSVR was utilized to calculate the kernel centers for both WV-LSSVR and GPK-LSSVR using the benchmark training data. Subsequently, both of the previous methods were employed to learn and evaluate the testing benchmark data sets at the second step. We also should note that GPK-LSSVR and WV-LSSVR use the same kernel centers as those of IRR-LSSVR, which warrants a fair comparison.

Fig. 5 and Fig. 6 give the comparison results of IRR-RBF, GPK-LSSVR and WV-LSSVR using 'total UPDRS' data set. WV-LSSVR and GPK-LSSVR methods lead to a higher approximation accuracy than IRR-LSSVR, which is mainly because that the testing benchmark data are also involved in updating the model in the former methods. In addition, as can be seen from Fig. 5, GPK-LSSVR enjoys a higher level of approximation accuracy than WV-LSSVR. This improvment should be ascribed to the extention of the kernel basis function, since the extended kernel allows a fast adaptation to higher degree of nonlinearity in local data trends and provides higher degree of freedom on modeling parameters for optimization. Again, due to similarity and limitation of space, except for 'total UPDRS', the comparison results for all of the other benchmark data sets are tabulated in Table 3. In all of the following numerical experiments, kernel parameters are chosen the same as those used in IRR-LSSVR from [18].

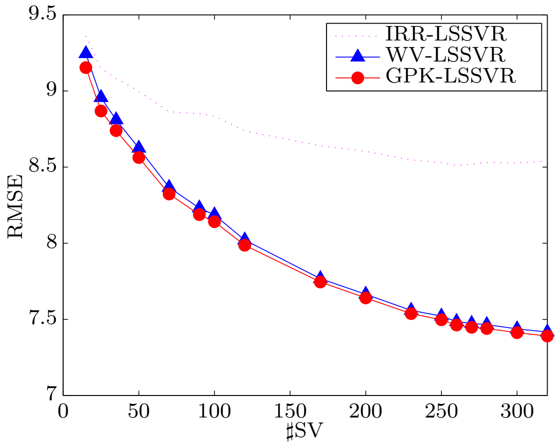

Fig. 5: RMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$.

Table 2: Comparison results with the K-means clustering method

data sets	algorithms	RMSE	NMSE	seTime[s]	trTime[s]	teTime[s]	trNum	teNum	♯SV
motor UPDRS $C^\star = 2^5$	KMC-RBF	6.9765	7.5682E-01	216.3110	47.1279	1.7748	3000	2875	350
$\gamma^\star = 2^{-2}$, 40 inputs	WV-LSSVR	6.6354	6.8461E-01	46.3323	48.3603	2.1684	3000	2875	350
	GPK-LSSVR	6.6393	6.8542E-01	46.8159	818.0848	5.3196	3000	2875	350
total UPDRS $C^\star = 2^5$	KMC-RBF	9.1513	7.5247E-01	136.8129	23.0257	1.5288	3000	2875	260
$\gamma^\star = 2^{-2}$, 16 inputs	WV-LSSVR	8.5437	6.5587E-01	30.1862	25.0226	1.5600	3000	2875	260
	GPK-LSSVR	8.5546	6.5753E-01	30.2642	522.6190	3.9312	3000	2875	260
wineequality-red $C^\star = 2^5$	KMC-RBF	9.4202E-01	13.6340E-01	0.2808	0.2652	0.0312	1000	599	19
$\gamma^\star = 2^{-1}$, 11 inputs	WV-LSSVR	6.2926E-01	6.0837E-01	1.2324	0.1560	0.0312	1000	599	19
	GPK-LSSVR	5.2162E-01	5.9368E-01	1.2324	0.7956	0.3588	1000	599	19
wineequality-white $C^\star = 2^6$	KMC-RBF	7.3611E-01	6.9254E-01	11.4349	3.3072	0.5460	3500	1398	70
$\gamma^\star = 2^0$, 11 inputs	WV-LSSVR	7.1246E-01	6.4875E-01	17.7373	3.4944	0.2808	3500	1398	70
	GPK-LSSVR	7.1283E-01	6.4943E-01	17.8465	15.5845	0.6240	3500	1398	70
concrete $C^\star = 2^6$	KMC-RBF	7.2130	19.192E-02	7.8000	1.4352	0.5616	700	330	125
$\gamma^\star = 2^{-1}$, 8 inputs	WV-LSSVR	5.8509E-01	12.628E-02	1.1700	1.6848	0.2652	700	330	125
	GPK-LSSVR	5.9394	13.013E-02	1.1232	13.7437	0.4212	700	330	125
Boston housing $C^\star = 2^5$	KMC-RBF	4.4051	2.0304E-01	1.7628	0.4992	0.0624	400	106	90
$\gamma^\star = 2^{-1}$, 13 inputs	WV-LSSVR	2.6189	7.1766E-02	0.3432	0.6240	0.0468	400	106	90
	GPK-LSSVR	2.6397	7.2912E-02	0.3432	3.8064	0.0936	400	106	90
delta elevators $C^\star = 2^5$	KMC-RBF	1.4578E-03	3.7374E-01	3.0576	3.0108	0.7644	6000	3517	36
$\gamma^\star = 2^0$, 6 inputs	WV-LSSVR	1.4505E-03	3.7001E-01	56.2384	2.6208	0.7800	6000	3517	36
	GPK-LSSVR	1.4508E-03	3.7017E-01	56.0980	9.0481	1.0452	6000	3517	36
puma8NH $C^\star = 2^5$	KMC-RBF	4.0173	5.1544E-01	41.0751	5.9592	0.9984	4500	3693	90
$\gamma^\star = 2^{-1}$, 8 inputs	WV-LSSVR	3.3622	3.6104E-01	31.5122	5.6940	0.9672	4500	3693	90
	GPK-LSSVR	3.3633	3.6128E-01	34.4606	49.4991	2.1216	4500	3693	90
cpu act $C^\star = 2^8$	KMC-RBF	4.3084	5.8782E-02	790.4415	22.1989	1.7628	5000	3192	200
$\gamma^\star = 2^0$, 21 inputs	WV-LSSVR	2.5893	2.1231E-02	55.2244	22.9477	1.7940	5000	3192	200
	GPK-LSSVR	2.5898	2.1238E-02	66.8464	332.0325	3.7752	5000	3192	200
delta ailerons $C^\star = 2^5$	KMC-RBF	1.6508E-04	3.0721E-01	5.1324	4.3056	0.7332	5000	2129	60
$\gamma^\star = 2^{-1}$, 40 inputs	WV-LSSVR	1.6288E-04	2.9906E-01	37.0658	5.1168	0.8424	5000	2129	60
	GPK-LSSVR	1.6284E-04	2.9891E-01	37.5650	16.4269	0.6552	5000	2129	60
Ablone $C^\star = 2^5$	KMC-RBF	2.5246	5.6284E-01	0.0780	0.1404	0.0312	3000	1177	15
$\gamma^\star = 2^1$, 8 inputs	WV-LSSVR	2.1808	4.2000E-01	11.6377	0.5304	0.0312	3000	1177	15
	GPK-LSSVR	2.1829	4.2080E-01	11.9029	1.8876	0.2808	3000	1177	15
stock $C^\star = 2^{10}$	KMC-RBF	12.3160E-01	3.6105E-02	2.3868	1.0608	0.3432	600	350	100
$\gamma^\star = 2^{-1}$, 9 inputs	WV-LSSVR	8.1528E-01	1.5822E-02	0.6864	0.7956	0.2496	600	350	100
	GPK-LSSVR	8.1737E-01	1.5904E-02	0.7020	7.1448	0.6864	600	350	100
AutoMPG $C^\star = 2^{10}$	KMC-RBF	3.1642	1.4336E-01	0.0156	0.0156	0.0060	350	42	11
$\gamma^\star = 2^1$, 7 inputs	WV-LSSVR	2.6098	0.9753E-01	0.1716	0.0624	0.0240	350	42	11
	GPK-LSSVR	2.5320	0.9179E-01	0.1716	0.1248	0.0300	350	42	11
kinematics $C^\star = 2^4$	KMC-RBF	9.8647E-02	14.4530E-02	753.8748	179.9004	3.6504	4500	3693	500
$\gamma^\star = 2^{-1}$, 8 inputs	WV-LSSVR	7.9163E-02	9.3077E-02	166.1255	239.1495	3.9312	4500	3693	500
	GPK-LSSVR	7.9123E-02	9.2981E-02	162.7090	3851.6335	9.4693	4500	3693	500
diesel engine $C^\star = 2^{10}$	KMC-RBF	6.7024E-02	1.1151E-03	0.0156	0.0156	0.0156	210	198	15
$\gamma^\star = 1.69$, 3 inputs	WV-LSSVR	2.2658E-02	1.2743E-03	0.0936	0.0624	0.0158	210	198	15
	GPK-LSSVR	2.2389E-02	1.2442E-03	0.1092	0.1248	0.0514	210	198	15

In Table 3, trTime gives the computational time of selecting kernel centers using IRR-LSSVR, while teTime shows the total recursive identification time on the benchmark testing data sets. Besides, the column marked with baseline RMSE shows the RMSE values of the baseline method (i.e.IRR-LSSVR) when evaluating the testing data without updating the model. As can be seen from the RMSE column, the approximation accuracy of both WV-LSSVR and GPK-LSSVR methods is higher than that of the baseline method, this is mainly because the testing data are also used to update the global model in real-time. Furthermore, If we compare GPK-LSSVR with WV-LSSVR, it is found that the former enjoys a higher approximation accuracy than the latter in recursive identification using the aforementioned parametric kernel methods. In addition, GPK-LSSVR always owns a higher approximation accuracy than WV-LSSVR though the magnitude of the improvement varies on different data sets. This is due to the fact that the Gaussian process kernel provide more freedom than the single Gaussian kernel for the recursive parametric kernel method, which allows the kernel model to track the different data trends among different sub domains. At last, it should also be noted that the computational time of GPK-LSSVR keeps still comparable to that of the WV-LSSVR, which is in consistent with the analysis result shown in Table 1.

5. Conclusions

In the field of SVMs, convincing results have been achieved on techniques of selecting optimal support vec-

Table 3: Comparison results concerning the extended kernel functions

data sets	baseline RMSE	algorithms	RMSE	NMSE	trTime[s]	teTime[s]	trNum	teNum	♯SV
motor UPDRS $C^\star = 2^5$	6.5859	WV-LSSVR	5.7273	5.1004E-01	37.4035	11.3455	3000	2875	350
$\gamma^\star = 2^{-2}$, 40 inputs		GPK-LSSVR	5.6997	5.0515E-01	44.6163	722.7526	3000	2875	350
total UPDRS $C^\star = 2^5$	8.5098	WV-LSSVR	7.4859	5.0351E-01	30.6386	27.0350	3000	2875	260
$\gamma^\star = 2^{-2}$, 16 inputs		GPK-LSSVR	7.4632	5.0046E-01	28.5638	384.4957	3000	2875	260
winequality-red $C^\star = 2^5$	6.2870E-01	WV-LSSVR	5.7554E-01	5.0892E-01	1.2324	0.3120	1000	599	19
$\gamma^\star = 2^{-1}$, 11 inputs		GPK-LSSVR	5.4157E-01	4.5063E-01	1.0920	0.4992	1000	599	19
winequality-white $C^\star = 2^6$	7.1215E-01	WV-LSSVR	6.7444E-01	5.8136E-01	17.5501	1.2480	3500	1398	70
$\gamma^\star = 2^0$, 11 inputs		GPK-LSSVR	6.6549E-01	5.6603E-01	18.1429	7.5036	3500	1398	70
concrete $C^\star = 2^6$	5.8783	WV-LSSVR	3.2515E-01	3.9001E-02	1.1076	0.7176	700	330	125
$\gamma^\star = 2^{-1}$, 8 inputs		GPK-LSSVR	3.0922	3.5272E-02	1.0608	8.0653	700	330	125
Boston housing $C^\star = 2^5$	2.6354	WV-LSSVR	0.7371	0.5685E-02	0.3900	0.2184	400	106	90
$\gamma^\star = 2^{-1}$, 13 inputs		GPK-LSSVR	0.5896	0.3637E-02	0.4680	1.1700	400	106	90
delta elevators $C^\star = 2^5$	1.4503E-03	WV-LSSVR	1.4293E-03	3.5926E-01	38.4889	0.3900	6000	3517	36
$\gamma^\star = 2^0$, 6 inputs		GPK-LSSVR	1.4223E-03	3.5575E-01	53.7423	4.6488	6000	3517	36
puma8NH $C^\star = 2^5$	3.3616	WV-LSSVR	3.1171	3.1032E-01	33.0566	5.7876	4500	3693	90
$\gamma^\star = 2^{-1}$, 8 inputs		GPK-LSSVR	3.0954	3.0602E-01	31.1690	35.02094	4500	3693	90
cpu act $C^\star = 2^8$	2.5816	WV-LSSVR	2.2841	1.6521E-02	64.4440	17.0821	5000	3192	200
$\gamma^\star = 2^0$, 21 inputs		GPK-LSSVR	2.2462	1.5977E-02	56.1604	179.8536	5000	3192	200
delta ailerons $C^\star = 2^5$	1.6278E-04	WV-LSSVR	1.5366E-04	2.6617E-01	27.2879	0.5148	5000	2129	60
$\gamma^\star = 2^{-1}$, 40 inputs		GPK-LSSVR	1.5238E-04	2.6175E-01	38.5946	6.6768	5000	2129	60
Ablone $C^\star = 2^5$	2.1804	WV-LSSVR	2.1292	4.0037E-01	11.2945	0.3120	3000	1177	15
$\gamma^\star = 2^1$, 8 inputs		GPK-LSSVR	2.0843	3.8365E-01	11.6689	0.6396	3000	1177	15
stock $C^\star = 2^{10}$	8.1466E-01	WV-LSSVR	5.155E-01	6.3247E-03	0.4277	0.1093	600	350	100
$\gamma^\star = 2^{-1}$, 9 inputs		GPK-LSSVR	4.9072E-01	5.7322E-03	0.7332	4.0560	600	350	100
AutoMPG $C^\star = 2^{10}$	2.6099	WV-LSSVR	2.0339	5.9231E-02	0.1519	0.0624	350	42	11
$\gamma^\star = 2^1$, 7 inputs		GPK-LSSVR	1.8101	4.6914E-02	0.1872	0.0624	350	42	11
kinematics $C^\star = 2^4$	7.8735E-02	WV-LSSVR	5.7416E-02	4.8961E-02	150.0730	153.2242	4500	3693	500
$\gamma^\star = 2^{-1}$, 8 inputs		GPK-LSSVR	5.6991E-02	4.8240E-02	152.8338	3414.4875	4500	3693	500
diesel engine $C^\star = 2^{10}$	2.2034E-02	WV-LSSVR	1.9435E-02	0.93758E-03	0.1092	0.0624	210	198	15
$\gamma^\star = 1.69$, 3 inputs		GPK-LSSVR	1.8844E-02	0.8815E-03	0.1092	0.0624	210	198	15

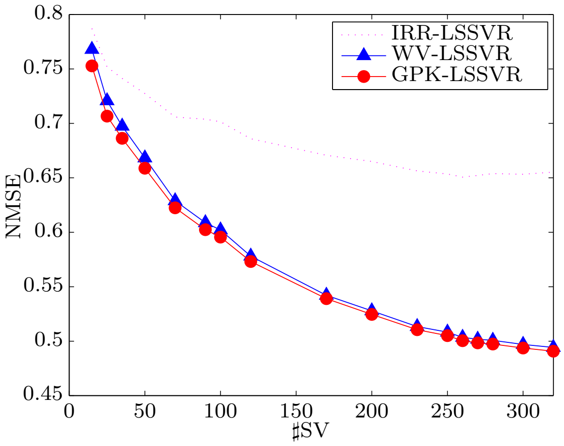

Fig. 6: NMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$.

tors for LSSVR. In specific, the support vector selection technique employed in IRR-LSSVR has shown its superiority to other methods. To achieve the same level of modeling accuracy, IRR-LSSVR needs fewer support vectors than other LSSVR methods.

In this paper, we propose a general recursive parametric kernel method called WV-LSSVR, which is suitable for identifying global model in real-time. In order to calculate the centers for all kernels, the support vector selection technique from the field of SVMs is made use of. Instead of directly solving a global optimization problem, WV-LSSVR gets all the kernel centers by using IRR-LSSVR, where support vectors are chosen in a recursive way. In specific, IRR-LSSVR is employed to analyze a pre-collected modeling data set so as to acquire a suitable number of support vectors, which are laterly set to be the centers of the kernels in WV-LSSVR.

For the purpose of improving the approximation accuracy of the global model, especially when considering different data trends among different subdomains, an improved version of WV-LSSVR called GPK-LSSVR is developed by extending the kernel basis functions. A kernel basis function recommended for Gaussian process regression is adopted in GPK-LSSVR. The advantage of the Gaussian process kernel is that it helps to simplify the implementation of the proposed recursive kernel method.

To demonstrate the efficacy of WV-LSSVR and GPK-LSSVR, a set of numerical experiments are carried out using benchmark data sets. At first, classical KMC-RBF is implemented so as to provide a comparison baseline. Compared with KMC-RBF, the kernel centers calculated using IRR-LSSVR lead to a higher approximation accuracy in identifying the global model. Subsequently, GPK-LSSVR method, in which the Gaussian kernels are extended with a linear term and a constant term, is compared with WV-LSSVR in a recursive identification framework. In this numerical experiment, the testing benchmark data are evaluated using WV-LSSVR and GPK-LSSVR while the recursive identification is still going on. The results validate the hypothesis that a suitable extension of the kernel basis function helps to enhance the approximation power of the recursive parametric kernel method. In the future, techniques about choosing representing modeling data in advance need to be investigated, because all of the kernel centers are selected only from the pre-collected modeling data.

References

[1] B. Scholkopf, A. J. Smola, Learning with Kernels, Cambridge:MIT Press, 2002.

[2] M. B. Christopher, Pattern Recognition and Machine Learning, Springer, 2006.

[3] L. Bottou, O. Chapelle, D. DeCoste, J. Weston, Large-Scale Kernel Machines, MIT Press, 2006.

[4] S. Chen, C. F. N. Cowan, P. M. Grant, Orthogonal least squares learning algorithm for radial basis function networks., IEEE Transactions on neural networks 2 (2) (1991) 302–309.

[5] G. Pillonetto, M. H. Quang, A. Chiuso, A new kernel-based approach for nonlinear system identification, IEEE Trans. on Automatic Control 56 (12) (2011) 2825–2840.

[6] P. Karsmakers, K. Pelckmans, K. De Brabanter, H. Van hamme, J. A. K. Suykens, Sparse conjugate directions pursuit with application to fixed-size kernel models, Mach. Learn. 85 (1) (2011) 109–148.

[7] S. Bernhard, K.-K. Sung, J. C. Charis, F. G. Burges, Comparing support vector machines with gaussian kernels to radial basis function classifiers., IEEE Transactions on Signal Processing 45 (11) (1997) 2758–2765.

[8] V. N. Vapnik, The Nature of Statistical Theory, New York:Springer-Verlag, 1995.

[9] J. A. K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural process, Lett. 9 (3) (1999) 293–300.

[10] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least squares support vector machines, World Scientific, 2002.

[11] J. A. K. Suykens, J. De Brabanter, L. Lucas, J. Vandewalle, Weighted least squares support vector machines:robustness and sparse approximation, Neuralcomputing 48 (1) (2002) 85–105.

[12] J. A. K. Suykens, L. Lucas, J. Vandewalle, Sparse approximation using least squares vector machines, in: IEEE International Symposium on Circuits and Systems, IEEE, Geneva, 2000, pp. 757–760.

[13] B. J. De Kruif, T. J. A. De Vries, Pruning error minimization in least squares support vector machines, IEEE Trans. Neural Networks 14 (3) (2004) 696–702.

[14] L. Hoegaerts, J. A. K. Suykens, J. Vandewalle, B. De Moor, A comparison of pruning algorithms for sparse least squares support vector machines, in: International Conference on Neural Information Processing, Springer, Calcutta, India, 2004, pp. 1247–1253.

[15] X. Y. Zeng, X. W. Chen, Smo-based pruning methods forsparse least squares support vector machines, IEEE Trans. Neural Networks 16 (6) (2005) 1541–1546.

[16] L. Jiao, L. Bo, L. Wang, Fast sparse approximation for least squares support vector machine, IEEE Trans. Neural Networks 18 (3) (2007) 685–697.

[17] K. De Brabanter, J. De Brabanter, J. A. K. Suykens, B. De Moor, Optimized fixed-size kernel models for large data sets, Computa. Stat. Data Anal. 54 (6) (2010) 1484–1504.

[18] Y. P. Zhao, J. G. Sun, Z. H. Du, Z. A. Zhang, An improved recursive reduced least squares support vector regression, Neurocomputing 87 (1) (2012) 1–9.

[19] G. Cawley, N. L. C. Talbot, Reduce rank kernel ridge regression, letter, Neural Process 16 (3) (2002) 293–302.

[20] J. Bi, T. Zhang, K. Bennett, Column-generation boosting methods for mixture of kernels, in: Proceeding of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD, New York, 2004, pp. 521–526–1253.

[21] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, M. I. Joran, Learning the kernel matrix with semidefinite programming, Journal of machine learning research 5 (2004) 27–72.

[22] C. S. Ong, A. J. Smola, R. C. Williamson, Learning the kernel with hyperkernels, Journal of machine learning research 6 (2005) 1043–1071.

[23] Y. P. Zhao, J. G. Sun, Multikernel semiparametric linear programming support vector regression, Expert Systems with Applications 38 (2011) 1611–1618.

[24] Y. P. Zhao, J. G. Sun, Recursive reduced least squares support vector regression, Pattern Recognition 42 (5) (2009) 837–842.

[25] S. A. Billings, S. Chen, The identification of linear and nonlinear models of a turbocharged automotive diesel engine., Mech. Syst. Signal Process 3 (2) (1989) 123–142.