Tree Based Orthogonal Least Squares Regression with Repeated Weighted Boosting Search

Lihua, Fu, Hongwei Li*
School of Mathematics and Physics, China University of Geosciences, Wuhan 430074, China
Email:fulihua9270@yahoo.com.cn, hwli@cug.edu.cn

Meng Zhang
Department of Computer Science, Central China Normal University, Wuhan 430079, China
Email: morosezhang@yahoo.com.cn

***Abstract*—Orthogonal Least Squares Regression (OLSR) selects each regressor by repeated weighted boosting search (RWBS). This kind of OLSR is known to be capable of producing a much sparser model than many other kernel methods. With the aid of tree structure search, this paper is to construct an even sparser regression model in the framework of OLSR with RWBS. When RWBS being used to solve the optimization at each regression stage, OLSR is extended by keeping the k (k >1)excellent regressors, which minimize the modeling MSE, rather than only choose the best one at each iteration. In this way, the next regressor will be searched in k subspaces instead of in only one subspace as the conventional method. Furthermore we propose a subtree search to decrease experimental time complexity, by specifying the total number of children in every tree depth. The new schemes are shown to outperform the traditional method in the applications, such as component detection, sparse representation for ECG signal and 2-d time series modeling. Besides, experimental results also indicate that subtree based algorithm is with much lower time complexity than tree based one.**

***Index Terms*—Orthogonal least squares, repeated weighted boosting search, tree structure search**

I. INTRODUCTION

A basic principle in nonlinear data modeling is the parsimonious principle of ensuring the smallest possible model that explains the training data. The state-of-art sparse kernel modeling techniques, such as support vector machines (SVM) and relevance vector machine (RVM) [1,2] have widely been adopted in data modeling applications.

However, most existing kernel methods use the fix parameter for every regressor. For example, standard SVM with Gaussian kernel generally adopts a fix scale parameter for each term by cross-validation. For the non-flat functions which contain both the steep variations and the smooth variations, the fix scale scheme will damage the sparsity of SVM.

Ref. [3] takes this problem as a linear regression in a combined feature space which is implicitly defined by a set of translation invariant kernels with different scales. It was reported that the multiscale SVM could produce much sparser kernel model than the conventional methods. However, multiscale SVM needs to construct a candidate set for scales, which is not hard task for users.

The orthogonal least squares regression(OLSR) algorithm [4] developed in the late 80s for nonlinear system modeling, remains popular for nonlinear data modeling, for the reason that the algorithm is simple and efficient, and is capable of producing very sparse regression model with good generalization performance. Over the time, many improved variants of the OLSR algorithm have been proposed [5-9]. OLSR can tune the scales in every term.

In order to minimize the modeling MSE, at each stage, OLSR selects the best regressor in a subspace which is orthogonal to the linear space spanned by the already selected regressors. Different regressor will lead to different subspace for the next regressor being searched. Thus the selection of every regressor will dramatically affect the choosing of regressor in the next stage. Nevertheless OLSR is a greedy algorithm, which only seeks the best performance in the current stage, and ignores the affect on the next stage. To get a good performance, OLSR needs more regressors than necessary.

This paper extends OLSR by keeping the k excellent regressors which minimize the modeling MSE, rather than ignoring them after getting the best one at each regressor stage. These surviving regressors (regarded as nodes) are used to calculate the k new subspaces and corresponding residuals. Because each node creates k children, this tree-structure search scheme, in the L-th level, involves k^L best nodes rather than only one node in the conventional method (see Fig. 2(a)). In order to decrease experimental time complexity, we specify the total number of children as s in every level (see Fig. 2(b)). For notation clarity, this paper terms the former structure as tree (k) and the latter as subtree(s, k).

In fact, one can implement the tree structure search in many variants of the OLSR algorithm. This paper only considers Ref. [9]. In [9], the optimization at each regression stage is carried out with a simple search algorithm re-enforced by boosting, termed as repeated weighted boosting search (RWBS). In this way, OLSR tunes the centre vector and diagonal covariance matrix of individual regressor by incrementally minimizing the training mean square error (MSE). This kind of OLSR can produces a much sparser regression model than many

*corresponding author

© 2012 ACADEMY PUBLISHER
doi:10.4304/jcp.7.1.187-195

other kernel machines and previous versions of OLSR [9-11], largely because of its flexibility to select its optimal scales.

RWBS is a guided global search algorithm which involves the excellent individual, obtained at the previous iteration, into the new iteration. This scheme makes the algorithm constringency. At each regressor stage, OLSR with RWBS always keeps the optimal solution as the newly selected regressor parameters, and the other excellent individuals mentioned above are ignored.

With the aid of tree(k) and subtree(s, k), an extension for OLSR with RWBS is presented. Because the tree structure search concerns with the performance not only in the current stage, but also in a more global view than conventional scheme, this method will lead to a meaningful improvement to the conventional OLSR.

Experimental results show that both tree based and subtree based OLSR create a much sparser signal representation than conventional OLSR. And subtree based algorithm can avoid the exponentially increasing computational load caused by tree based OLSR.

Tree based algorithm for regression also can be found in matching pursuit field, where a sparse linear combination is searched in a given linear space with finite dimension [12-13].

II. OLSR WITH RWBS

A. Theory

Given N pairs of training data $\{\mathbf{x}(l), y(l)\}_{l=1}^{N}$ and kernel function $\varphi(\cdot,\cdot)$, let

$$\mathbf{\Phi}_i = [\varphi(\mathbf{u}(i), \mathbf{x}(1)), \cdots, \varphi(\mathbf{u}(i), \mathbf{x}(N))]^T, \ i = 1, 2, \cdots, M \quad (1)$$

where M denotes the size of model, $\mathbf{u}(i)$ is the parameter vector, thus the regression matrix is $\mathbf{\Phi} = [\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_M]$, weight vector $\mathbf{w} = [w_1, \cdots, w_M]^T$, output vector $\mathbf{y} = [y(1), \cdots, y(N)]^T$, and error vector $\mathbf{e} = [e(1), \cdots, e(N)]^T$. The regression model can be presented as following matrix form

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \mathbf{e} \quad (2)$$

The goal of modeling data is to find the best linear combination of the column of $\mathbf{\Phi}$ (i.e. the best value for $\mathbf{w}$ and $\mathbf{u}(i)$) to explain $\mathbf{y}$ according to some criteria.

By OLSR algorithm, the solution is searched in a transformed orthogonal space. In more detail, let an orthogonal decomposition of the regression matrix $\mathbf{\Phi}$ be $\mathbf{\Phi} = \mathbf{HA}$, where $\mathbf{A}$ is an upper triangular matrix with the unit diagonal element and $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_M]$ with the orthogonal columns that satisfy $\mathbf{H}_i^T \mathbf{H}_j = 0$ if $i \neq j$. The regression model (3) can alternatively be expressed as

$$\mathbf{y} = \mathbf{H}\boldsymbol{\theta} + \mathbf{e} \quad (3)$$

where the new weight vector $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_M]^T$ satisfies the triangular system $\boldsymbol{\theta} = \mathbf{Aw}$. For the orthogonal regression model, the training MSE can be expressed as

$$J = \mathbf{e}^T\mathbf{e}/N = \mathbf{y}^T\mathbf{y}/N - \sum_{i=1}^{M} \mathbf{H}_i^T\mathbf{H}_i\theta_i^2/N \quad (4)$$

Thus the training MSE for the L -term subset model can be expressed as $J_L = J_{L-1} - \mathbf{H}_L^T\mathbf{H}_L\theta_L^2/N$ with $J_0 = \mathbf{y}^T\mathbf{y}/N$.

At the L-th stage of regression, the L-th regressor is determined by maximizing the error reduction criterion $E_L = \mathbf{H}_L^T\mathbf{H}_L\theta_L^2/N$ with respect to the kernel parameter $\mathbf{u}(L)$. Generally, Gaussian kernel is often the first choice of kernel because of its excellent generalized ability.

B. Algorithm

Some guided random search methods can be used to determine the parameters of the k-th kernel regressor, such as the genetic algorithm and adaptive simulated annealing. RWBS is recently proposed global searching algorithm. It is extremely simple and easy to implement, involving a minimum programming effort. Before implementing RWBS, several parameters need be set, such as the initial size of population Ps, the generation of outer loop NG, and the iteration of inner loop Nb. One can refer to [9,14] for more detail.

When searching the L-th regressor of OLSR with RWBS, the algorithm can be described as Fig. 1

NOTE 1

RWBS can get the global optimization by involving the good individual $\mathbf{u}_t^L, t = 1, \cdots, NG-1$, obtained at the previous iteration, into the new iteration. Always the global optimization $\mathbf{u}_{NG}^L$ is kept as the newly selected regressor parameter, and all other $\mathbf{u}_t^L, t = 1, \cdots, NG-1$ are ignored.

NOTE 2

There are other decision criterion to stop the iteration besides t=NG. But for simplicity, we let the condition t=NG be the sole criterion.

III. TREE BASED ALGORITHM

A. Theory

In this section, the tree based OLSR is described. The algorithm also has a recursive structure. At L-th regressor stage, when implementing RWBS in Fig.1, we select k best vectors from all of the good individuals $\mathbf{u}_t^L, t = 1, \cdots, NG$, to minimize the training MSE. That is

$$\tilde{\mathbf{u}}_i^L = \arg\min_{\mathbf{u}_t^L} J(\mathbf{u}_t^L), \quad \text{and } \tilde{\mathbf{u}}_i^L \neq \{\tilde{\mathbf{u}}_1^L, \tilde{\mathbf{u}}_2^L, \cdots, \tilde{\mathbf{u}}_{i-1}^L\}, \quad i = 1, \cdots, k \quad (5)$$

where $k < NG$. Thus each node can create k child nodes $\tilde{\mathbf{u}}_i^L, i = 1, \cdots, k$. The selected vectors are treated as the candidates for the parameter vector for the L-th regressor. For each $\tilde{\mathbf{u}}_i^L$, corresponding regressor $\mathbf{\Phi}_i^L$ and training error J_i^L are also calculated according to (1) - (4). Once the required tree's depth is reached or the training MSE is small enough, the program is broken and we keep the combination of nodes which produces the best performance as the final solution. We term the algorithm as tree(k) which lets each node creates k new individuals.

© 2012 ACADEMY PUBLISHER

As Fig 2(a) shows, there are k^L branches when the tree depth reaches L, which means that we will search the solution in the k^L subspaces. Even if one let k be a

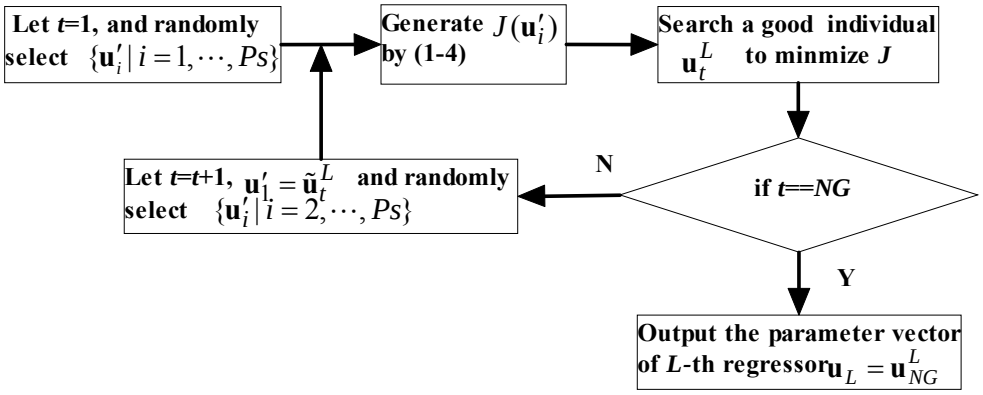

Figure 1. The scheme of OLSR with RWBS

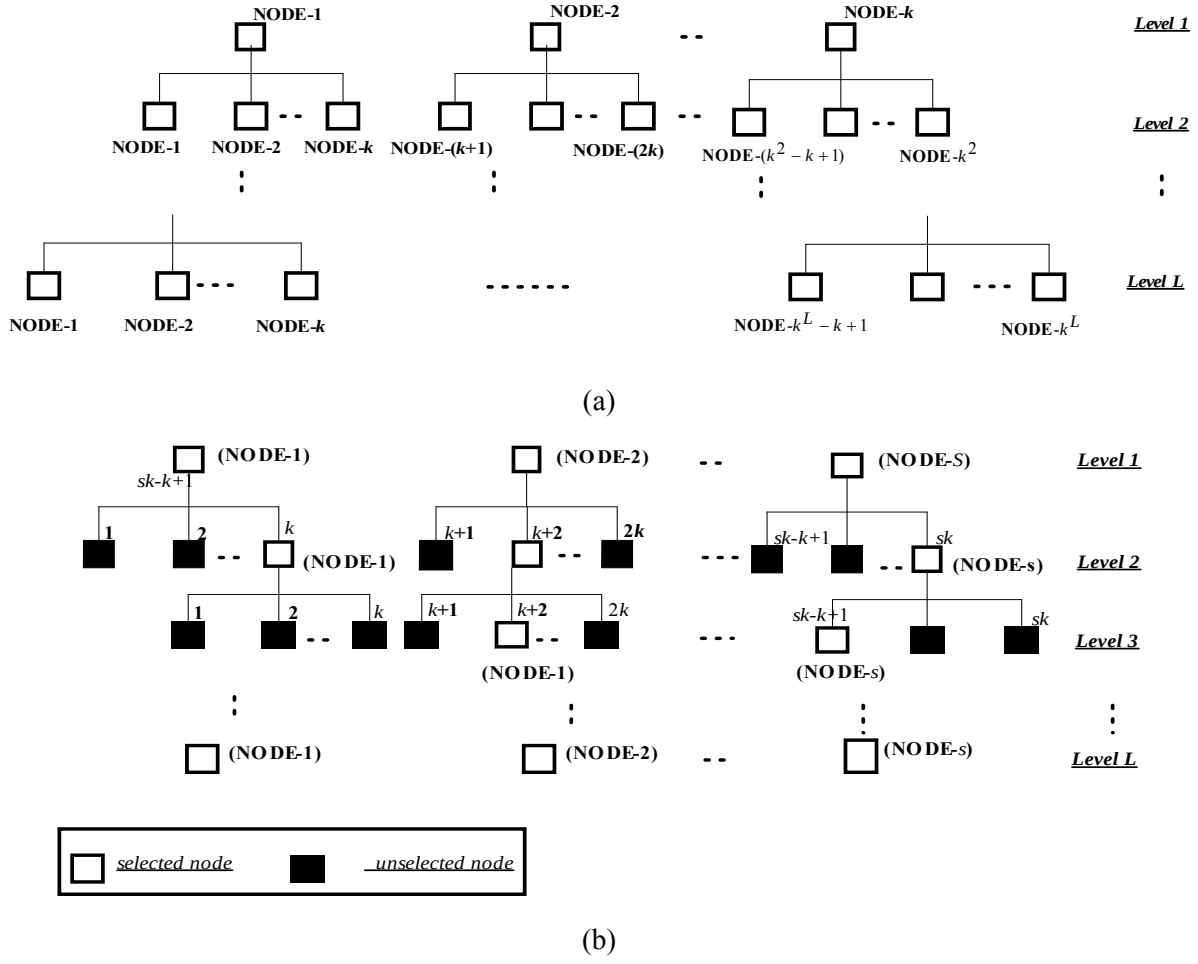

Figure 2. (a): tree(*k*) structure: every node can produce *k* child nodes. (b):sub-tree(*s*, *k*) structure, in each level, *s* nodes are selected to produce new nodes. Each selected node can produce *k* child nodes.

moderate number, the load of computation is prohibitive. This paper also proposes subtree(*s*, *k*) algorithm, which also lets each node have *k* children, but only s best individuals is kept in each level (see Fig. 2(b)), Tree(s, k) avoids the exponential increase of nodes.

Given the thresholds for tree depth *N* and training accuracy ξ, and let every node can generate *k* children. We depict the tree based and subtree based algorithm as following:

For $b = 1 : r(L-1)$ (the positive integer $r(L)$ denotes the total number of new surviving individual generated at

© 2012 ACADEMY PUBLISHER

level L , and initial value $r(0)=k$ for tree(k) or $r(0)=s$ for subtree(s, k)).

With the aid of RWBS, generate the candidate parameter column vectors $\tilde{\mathbf{u}}_i^L, i=1,\cdots,k$ for L-th regressor according to (5). And then create the candidate parameter matrix for the regression model with L terms, $\tilde{\mathbf{u}}\_candidate_{b,i}^L=[\mathbf{u}\_candidate_b^{L-1}, \mathbf{u}_i^L], i=1, 2,\cdots,k$ where $\mathbf{u}\_candidate_b^{L-1}$ denotes the candidate parameter matrix for the model with $L-1$ terms. And the initial element $\mathbf{u}\_candidate_b^0$ is a null matrix. Calculate $J_{b,i}^L=J(\tilde{\mathbf{u}}\_candidate_{b,i}^L)$ according to the equations (1-4).

End for

$\tilde{\mathbf{u}}\_candidate_{b\min,i\min}^L=\arg\min_{\tilde{\mathbf{u}}\_candidate_{b,i}^L} J(\tilde{\mathbf{u}}\_candidate_{b,i}^L)$ If $\min(J_{b,i}^L)<\xi$ (for $b=1:r(L)$, and $i=1:k$)

Break the program, and output $\tilde{\mathbf{u}}\_candidate_{b\min,i\min}^{L-1}$ as the kernels parameters for the regression model with $L-1$ -terms.

End if

We select $r(L)$ best elements from all of $\mathbf{u}\_candidate_{b,i}^L$ (for $b=1:r(L-1)$, and $i=1:k$), which can minimize $J_{b,i}^L$'s . The matrixes $\mathbf{u}\_candidate_c^L$ ($c=1:r(L)$) are used to record the $r(L)$ selected elements.

End for

Note 3.

When one let $r(L-1)=k^L$, this algorithm is tree(k). And if $r(L)\equiv s$ for any positive integer L, the algorithm shrinks to be subtree(s, k). If $r(L)\equiv 1$, one can get OLSR with RWBS.

B. Complexity Analysis

However, in the tree(k) algorithm, the number of nodes at each level grows exponentially. In order to obtain a $L-$term regression model, the total number of nodes needed be computed will be $(k^{L+1}-1)/(k-1)-1$. It will cause a heavy computational load. For subtree(s, k) algorithm, only $(L-1)\cdot(k-1)+1$ nodes need to be calculated. So sub-tree algorithm will reduce the computational load of tree algorithm greatly.

IV. SIMULATIONS

In order to show the approximation and detection performance of the proposed algorithm, we ran the experiments on the 1.86GHz notebook with 512MB of RAM, using the Windows XP operating system and employing the MATLAB software. In the simulations, OLSR with RWBS, tree based and subtree based algorithms were compared, all of them with RBF kernels. We selected the parameters in RWBS by cross-validation method. For the 1-dimensional simulations, the parameters of all tree based and subtree based algorithms are set as $Ps=3, NG=20$ and Nb=6 . And for the 2-dimensional simulation, the parameters are $Ps=4, NG=20$ and Nb=10 .

Simulation 1. Components detection

In this simulation, we used the model consisting of three RBF kernels, which is described as (1)

$$f(x)=\exp[-\frac{(x+5)^2}{2\times 1.9^2}]+2\exp[-\frac{(x-4)^2}{2\times 0.4^2}]+4\exp[-\frac{(x-6)^2}{2\times 0.8^2}] \quad (6)$$

A dataset is generated with size $N=160$ by an additive noise process $y_i=f(x_i)+n_i$, where the inputs were uniformly sampled from the domain [-8, 8] and the noise $n_i\sim N(0, 0.1^2)$. Totally 100 times Monte-Carlo trials were performed. Fig. 3 and 4 show the typical performances of OLSR and tree(3), both regression models with only 3 regressors.

Fig.3 (a-c) show the performances produced by each of the three terms (or regressors), that is regressor 1, 2 and 3 respectively. Fig.3 (d) shows the performances of the OLSR with 3 terms. From Fig.3, one can find that, because the first term is not capable of matching one of the waves in the original function properly (see Fig.3 (a)), the third term can not detect any component (see Fig. 3(c)). It is largely due to OLSR is a greedy algorithm, which only seeks the best performance in the current stage, and ignores the affect on the next stage. And the final result of OLSR is bad (see Fig.3 (d)).

On the contrary, because each term of tree (3) can match one of the waves of original function properly, the final model can detect all of the components in equation (6). Tree(3) outperforms OLSR because it has a more global view than OLSR.

This simulation indicates that tree based algorithm is more suitable than OLSR with RWBS to tackle the component detection problem.

Simulation 2. Sparse Representation for ECG Recording

ECG recordings are extremely non-flat signals in time domain, thus the modeling task for ECG is very difficult. Here, we used the first 1.5 second recording, totally 540 samples in “MIT 100” ECG signal [15].

Fig. 5 shows the convergence rates of algorithms with the threshold of training accuracy 0.0020, which indicates that both tree based and subtree based algorithms have much faster convergence rate than OLSR. The modeling performances of algorithms at the 3-th, 5-th and 7-th step are shown in Fig. 6. For sub-tree based and tree based algorithms, both training error and time (second) are also presented in every sub-figure. One can find that subtree based algorithm has a much lower complexity than tree based algorithm. So the tree based algorithm is suitable for the case in which time consumption is not concerned greatly. And subtree based algorithm can be used as a tradeoff between OLSR with RWBS and tree based method.

The simulation shows, for ECG modeling task, that newly proposed algorithms have much faster convergence rates than OLSR. Thus, with a smaller size of regression model, the newly proposed algorithms have much better approximation performance than OLSR.

© 2012 ACADEMY PUBLISHER

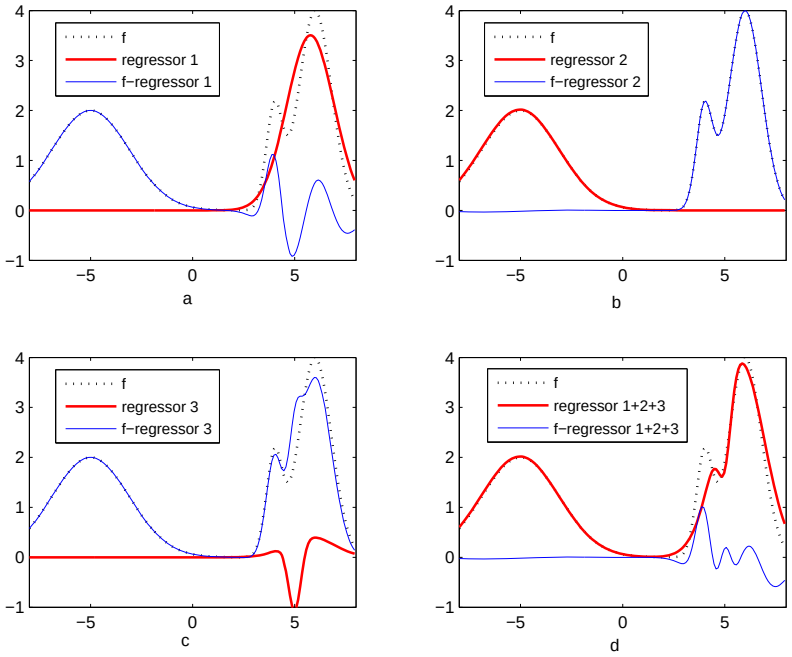

Figure 3. The effect of every regressor of OLSR model with three terms. In Fig.3 (a-c), the dotted line, broad solid line and broad thin line denote the original noise-free function, one of the three regressors of the OLSR model and the difference between them. For Fig.4(d), the thin solid line denotes the model with three terms.

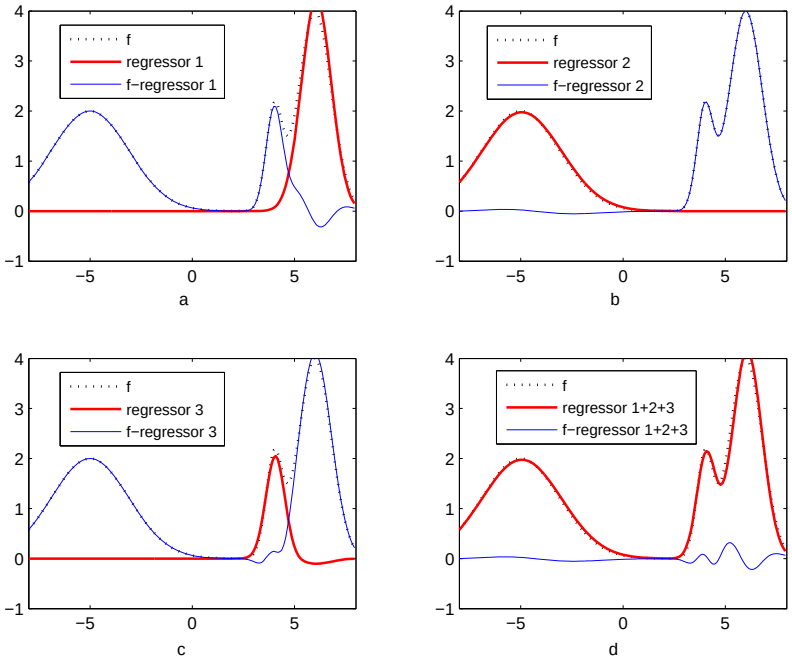

Figure 4. The effect of every regressor of tree(3) model with three terms. For Fig. 4 (a-c), the dotted line, broad solid line and thin solid line denote the original noise-free function, one of the three regressors of the tree(3) model and the difference between them. For Fig.4(d), the thin solid line denotes the model with three terms.

© 2012 ACADEMY PUBLISHER

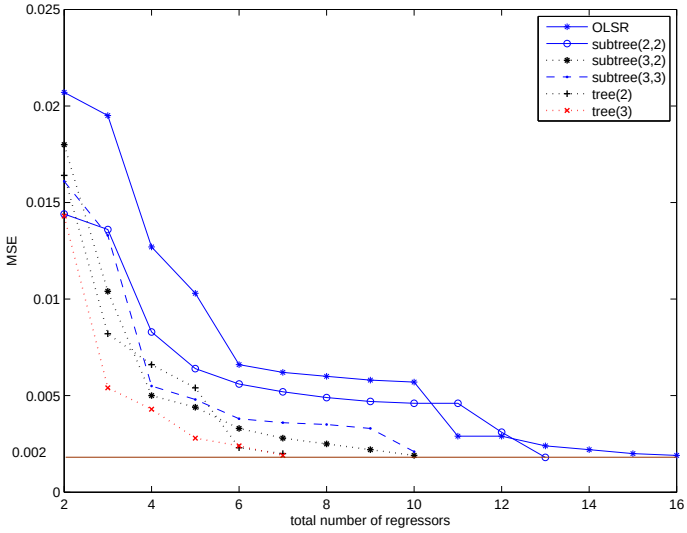

Figure 5. Convergence rates of different algorithms for ECG recording.

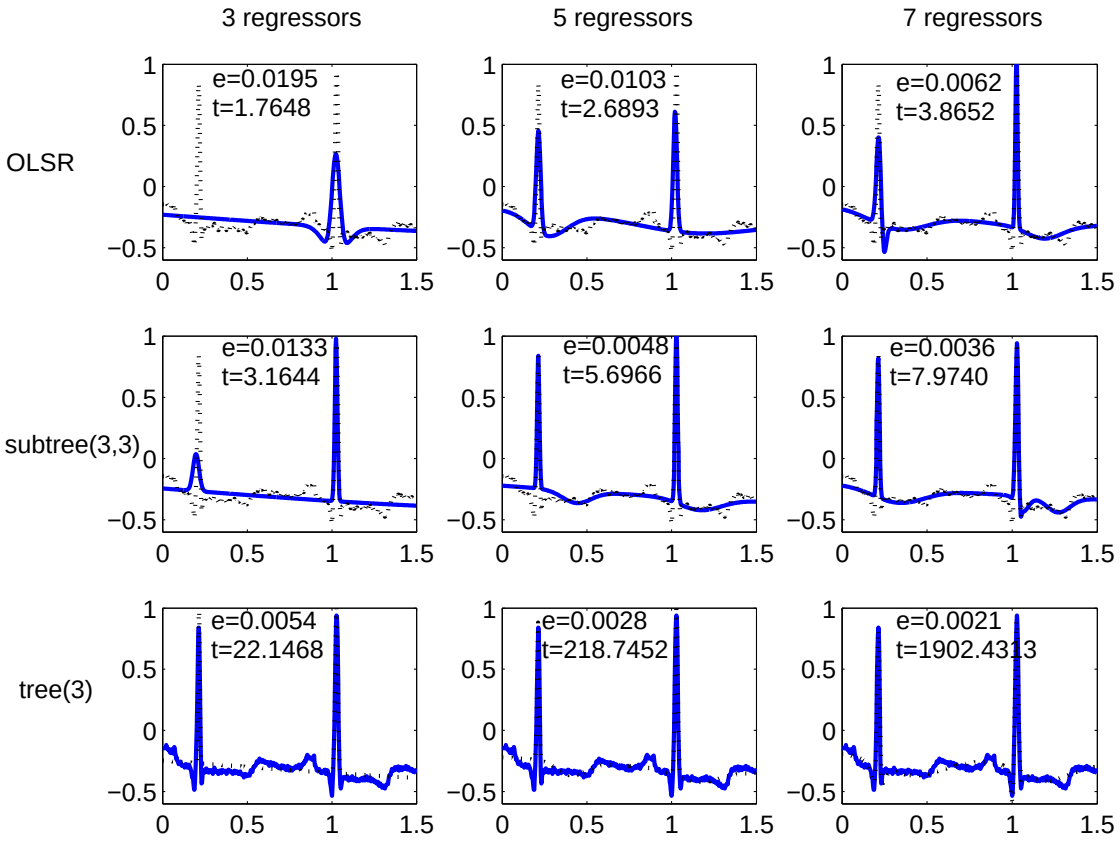

Figure 6. The performances for OLSR, subtree(3,3) and tree(3) algorithms at the different steps. Each algorithm's performance at different step is present in one of the rows.

© 2012 ACADEMY PUBLISHER

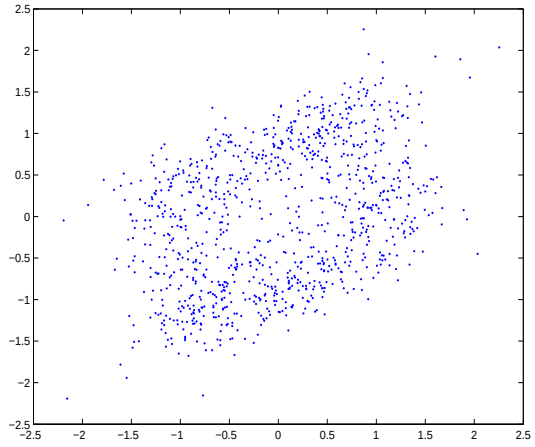

Figure 7. Phase plot of a typical set of noisy training data ($y(0)=y(-1)=0.0$) for the two-dimensional time series modeling problem

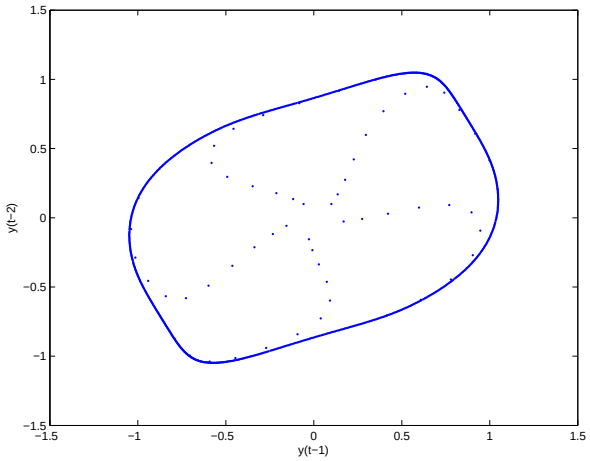

Figure 8. Phase plot of the noise-free two-dimensional time series ($y(0)=y(-1)=0.1$)

Simulation 3. Two- dimensional modeling

This was a two-dimensional simulated nonlinear time series given by

$$y(k)=[0.8-0.5\exp(-y^2(k-1)]y(k-1)$$
$$-[0.3+0.9\exp(-y^2(k-1)]y(k-2)+0.1\sin(\pi y(k-1))+\varepsilon(k)$$

where the noise $\varepsilon(k)$ was Gaussian with zero mean and variance 0.09. One thousand noisy samples were generated given $y(0)=y(-1)=0.0$. This model has been used in [16]. The first 500 data points plotted in Fig.7 were used for training, and the other 500 samples were used for possible cross-validation. The underlying noise-free system

$$y_d(k)=[0.8-0.5\exp(-y^2(k-1)]y(k-1)$$
$$-[0.3+0.9\exp(-y^2(k-1)]y(k-2)+0.1\sin(\pi y(k-1))$$

was shown by 1000 samples given in Fig.8 with $y(0)=y(-1)=0.1$.

Let the input vector $x(k)=[y(k-1), y(k-2)]^T$ and we use OLSR, subtree(2,2), subtree(3,3), subtree(3,2), tree(2) and tree(3) for this 2-dimensional modeling problem.

Let threshold of training error be 0.09, Tab.1 shows the averaged experimental results for 50 times simulations with all tested algorithms and the algorithms mentioned in [16] as well. Fig. 9 shows the modeling performances of different subtree and tree based algorithms at the 5-th step. The simulation shows that both tree and subtree based algorithms can get a better performance in a high-dimensional case than some traditional OLSR algorithms.

V. CONCLUSIONS

In order to a sparse representation, this paper proposes a novel tree based orthogonal least squares regression. Unlike most of the conventional OLSR, the new method keeps the *k* excellent regressors which minimize the modeling MSE, rather than only choose the best one at each iteration. These surviving individuals are used to calculate the *k* new subspaces and corresponding residuals. Because the tree structure search considers the performance not only in the current stage, but also in a global view, this method will lead to a meaningful improvement to the conventional OLSR. Numerical simulations are performed in some signal processing applications, such as component detection, sparse representation for ECG recording, and 2-d time series modeling. In all these simulations, both tree based and subtree based algorithms outperform OLSR with RWBS in convergence rate and accuracy. Because the subtree algorithm avoids the exponentially increasing computational load in tree based algorithm, the former can be regarded as a tradeoff between the latter and OLSR with RWBS.

REFERENCES

[1] A.Smola, "Regression estimation with support vector learning machines," Master's Thesis, Technische University Műnchen, 1996. Available at (http:// www.kernel-machines.org)

[2] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J.Mach. Learning*, pp.211-244,2001.

[3] Z. D. Nian, J. Wang and Y. Zhao, "Non-flat function estimation with a multi-scale support vector regression," *Neurocomputing*, vol. 70, issues 1-3, pp. 420-429, 2006.

[4] S. Chen, S. A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no.5, pp.1873-1896, 1989.

[5] S. Chen, E. S. Chng and K. Alkadhimi, "Regularized orthogonal least squares algorithm for constructing radial basis function networks," *Int. J. Control*, vol.64, no.5, pp.829-837, 1996.

[6] S. Chen, Y. Wu and B. L. Luk, "Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks," *IEEE Trans. Neural Networks*, vol.10, no.5, pp.1239-1243, 1999.

[7] S. Chen, "Locally regularized orthogonal least squares algorithm for the construction of sparse kernel regression models," in Proc. 6th Int. Conf. Signal Processing (Beijing, China), Aug. 26-30, 2002.

© 2012 ACADEMY PUBLISHER

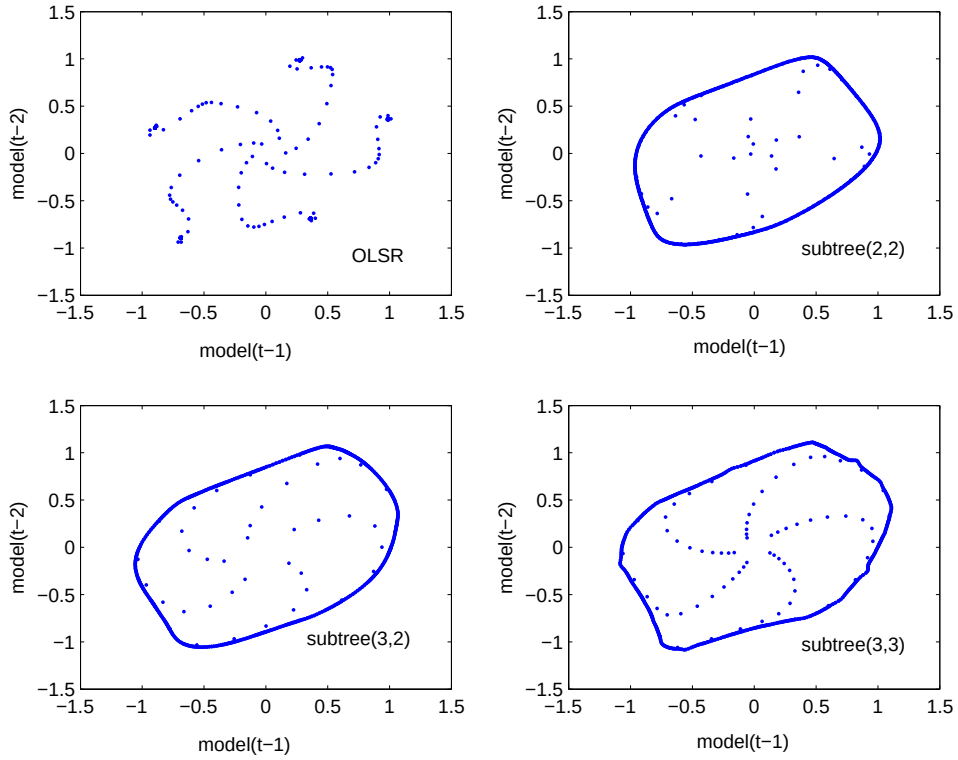

Figure 9. Performances of different algorithms for two-dimensional time series modeling problem at the 5-th step.

Table 1. Performances of different algorithms for two-dimensional modeling. The results of the above 4 rows are cited from [16]

algorithm	MODEL SIZE	MSE
OLS without RWBS	37	0.0881
UROLS	30	0.0911
RVM	19	0.0922
LROLS	18	0.0926
OLSR with RWBS	13.3	0.0903
subtree(2, 2)	7.1	0.0907
subtree(3, 3)	5.2	0.0945
subtree(3, 2)	5.5	0.0967
tree(2)	5.3	0.0970
tree(3)	5.1	0.0964

[8] X. Hong and C. J. Jarris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *IEEE Trans. Neural Networks*, vol.13, no.5 , pp. 1245-1250, 2002.
[9] S. Chen, X. X. Wang, and D.J. Brown, "Orthogonal least squares regression with tunable kernels," *Electronics Letters*, vol. 41 no. 8, 2005.
[10] X. X. Wang, S. Chen, D. Lowe and C.J. Harris, "Sparse support vector regression based on orthogonal forward selection for the generalized kernel model," *Neurocomputing*, vol.70, no.1-3, pp.462-474, 2006.
[11] X. X. Wang, S. Chen, D. Lowe and C.J. Harris, "Parsimonious least squares support vector regression using orthogonal forward selection with the generalized kernel model," *Int. J. Modeling, Identification and Control*, vol.1, no.4, pp.245-256, 2006.
[12] S. F. Cotter, and B.D. Rao, "Application of tree-based searches to matching pursuit," *IEEE. Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3933-3936, 2001.
[13] G.Z. Karabulut, L. Moura, D. Panario, and A. Yongacoglu, "Integrating flexible tree searches to orthogonal matching pursuit algorithm," *IEE ProcVis Image Signal Processing*, vol.153, no.5, pp.538-548, 2006.

© 2012 ACADEMY PUBLISHER

[14] S.Chen. X. X. Wang, and C.J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," *IEEE Trans. Syst. Man Cybern. B, Cybern.* vol.35, no.4, pp. 682-693, 2005.
[15] MIT-BIH Arrhythmia Database [Online]. Available: http://www.physionet.org/physio-bank/database/mitda/
[16] S. Chen, "Local regularization assisted orthogonal least regression", *Neurocomputing*, vol. 69, pp559-585, 2006.

ACKNOWLEDGEMENTS

This work was partially supported by NSFC under Grants 11026145, 61071188 and 90920005, by NSF of Hubei Province under Grants 2010CDB04205, 2009CDB077 and by the Fundamental Research Funds for the Central Universities.

Lihua Fu was born in Zhijiang, China, in 1979. She received the Master degree in mathematics from Hubei University. She is currently working toward Ph.D. degree in applied mathematics in China University of Geosciences (CUG). She is also a Lecturer with the School of Mathematics and Physics in CUG. Her research interests include wavelet and its applications in signal processing.

Hongwei Li is with the School of Mathematics and Physics in China University of Geosciences. His research interests include time series analysis and ICA.

Meng Zhang was born in Wuhan, China, in 1977. He received the Master degree in mathematics and the Ph.D. degree in computer science from Wuhan University in 2002 and 2005, respectively. His research interests include signal processing and neural network.

© 2012 ACADEMY PUBLISHER