http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (2): 320-325, 2008
ISSN 1812-5638

Multi-Resolution Signal Decomposition and Approximation Based on SVMS

[1,2]Wang Liejun, [2]Jia Zhenhong and [1]Lu Zhaogan
[1]Department of Information and Communication Engineering, Xi'an Jiaotong University, People's Republic of China
[2]School of Information Science and Engineering, Xinjiang University, People's Republic of China

Abstract: Support Vector Machines (SVMs) and Multi-Resolution Analysis (MRA) both have been developed for solving signal approximation problem. Replacing the approximation criterion of MRA by which be used in SVMs, multi-resolution signal decomposition and approximation algorithm based on SVMs can be derived. The advantage of this algorithm not only reduces the approximation error by introducing structure risk, but also has better smoothness of approximation function. Experiment illustrates that this algorithm has better approximation performance than conventional MRA when applying it to the approximation of stationary signal.

Key words: Support vector machines, multi-resolution analysis, signal approximation, approximation criterion

INTRODUCTION

A random sequence can be formed as follows:

$$y(n) = f(x_n)+\varepsilon_n \quad \text{for all } n = 1,2, \ldots N \tag{1}$$

Where, f(x) is an unknown signals in square integrable space L_2 (R) and ε_n is random noise with unknown probability distribution. y(n) can be seen as the observed samples of f(x) at the point x_n. Now define the following approximation problem: recovering the function $\tilde{f}(x)$ from the sequence $\{y(n)^N{}_{n=1}\}$ and $\tilde{f}(x)$ being the most similar to f(x) at the points of $\{x_n)^N{}_{n=1}\}$ in the sense of criterion L. Aforementioned approximation problem can be seen as the regression problem or the de-noising problem from different point of view.

MRA (Mallat, 1989) is one of methods that solve the approximation problem what (1) presents. The key idea of MRA is to stepwise approximate f(x) at different resolution. $P_j f(x)$, the approximation of f(x) at resolution j, is the projection of f(x) on the scale subspaces V_j. Among all the approximated function in V_j, $P_j f(x)$ is the function the most similar to f(x) in the sense of minimal Mean Square Error (MSE). Several researchers have theoretically investigated the approximation performance of multi-resolution theory and wavelet basis (Strang, 1989; Sweldens and Piessens, 1994; Unser, 1996).

SVMs has been introduced as novel tools for solving aforementioned approximation problems. In recent years, there has been a lot of interest in studying the SVMs (Vapnik, 1995; Vapnik *et al*., 1997; Hearst *et al*., 1998; Williamson *et al*., 1999). SVMs is based on the idea of structural risk minimization, which shows that the generalization error is bounded by the sum of the training set error and a term depending on the VC dimension of the learning machine. By minimizing this upper bound, high generalization performance can be achieved. Moreover, the quality and complexity of the SVMs solution does not depend directly on the dimensionality of the input space. To date, SVMs has been applied successfully to a wide range of problems, such as classification, regression and density estimation.

Combining the ideas of MRA and SVMs, this paper proposes a new algorithm named multi-resolution signal decomposition and approximation based on SVMs. The algorithm not only approximates f(x) gradually at different resolution, but also ensures better approximation performance than conventional MRA due to using the approximation criterion of SVMs.

MULTI-RESOLUTION SIGNAL DECOMPOSITION AND APPROXIMATION

Suppose MRA is a multi-resolution analysis in L_2 (R) such that the scale subspaces V_j and the wavelet subspaces W_j satisfy:

$$V_{j-1} = V_j \oplus W_j \tag{2}$$

$$\bigcup_{j\in z} V_j = L^2(R) \tag{3}$$

$$\bigcap_{j\in z} V_j = \langle 0 \rangle \tag{4}$$

Corresponding Author: Wang Liejun, Department of Information and Communication Engineering, Xi'an Jiaotong University, People's Republic of China

$\oplus$ and z denote direct sum and the set of integers, respectively. Let $\phi_{jk}(x) = 2^{-j/2}\,\phi(2^{-j}x\text{-}k)$ and $\psi_{jk}(x) = 2^{-j/2}\,\psi(2^{-j}x\text{-}k)$ be orthogonal basis of V_j, W_j subspaces, respectively and satisfy dilation equation:

$$\phi\left(2^{-j}x\right)=\sqrt{2}\sum_k h_{0k}\phi\left(2^{-j+1}x-k\right) \tag{5}$$

$$\psi\left(2^{-j}x\right)=\sqrt{2}\sum_k h_{1k}\psi\left(2^{-j+1}x-k\right) \tag{6}$$

Then the projection of f (x) on V_{j-1} can be computed as follows:

$$P_{j-1}f(x)=P_jf(x)+D_jf(x) \tag{7}$$

$P_jf(x)$ is smooth approximation of f(x) at resolution j and $D_jf(x)$ is smooth details. They can be expressed:

$$D_jf(x)=\sum_k d_k^{(j)}\psi_{jk}(x) \tag{8}$$

$$P_jf(x)=\sum_k c_k^{(j)}\phi_{jk}(x) \tag{9}$$

Where, $c_k^{(j)}$ and $d_k^{(j)}$ are discrete approximation and discrete details of f (x) at resolution j. We use following formula to compute $c_k^{(j)}$ and $d_k^{(j)}$:

$$d_k^{(j)}=\left\langle f(x),\psi_{jk}(x)\right\rangle \tag{10}$$

$$c_k^{(j)}=\left\langle f(x),\phi_{jk}(x)\right\rangle \tag{11}$$

If MRA is used to solve aforementioned approximation problems, Eq. 9 and 11 must be rewritten as follows because f(x) is unknown:

$$P_jy(n)=\sum_k c_k^{(j)}\phi_{jk}(n) \tag{12}$$

$$c_k^{(j)}=\left\langle y(n),\phi_{jk}(n)\right\rangle \tag{13}$$

At last, the approximation of $f(x_n)$ gets the following form:

$$\tilde{f}(x_n)=P_jy(n)=\sum_k c_k^{(j)}\phi_{jk}(n) \tag{14}$$

In this study, we call above algorithm conventional multi-resolution decomposition and approximation (*abbr.* conventional algorithm) which will be compared with our algorithm. The approximation criterion of conventional algorithm is:

$$\tilde{f}(x_n)=\arg\min_{g\in V_j}\left\|g(x_n)-f(x_n)\right\|_{L_2}^2 \tag{15}$$

Where, $\|\bullet\|_{L_2}$ denotes the norm defined on $L_2(R)$.

SUPPORT VECTOR MACHINE FOR APPROXIMATION (SVMA)

Here, we sketch the ideas behind the support vector machine for approximation (SVMA). In SVMA the basic idea is to map the data set $\{x_n, y(n)\}_{n=1}^N$ into a high dimensional feature space F via a nonlinear mapping φ and to do linear approximation in this space. Thus linear approximation in a high dimensional space corresponds to nonlinear approximation in low dimensional input space $L_2(R)$. A convex optimization problem as follow can be derived when introducing slack variables ξ_n, ξ_n^* and a ε insensitive loss function.

$$\begin{aligned}&\min\quad \frac{1}{2}\|w\|^2+C\sum_{i=1}^N\left(\xi_i+\xi_i^*\right)\\&\text{s.t. } y(n)-\left\langle w,\varphi(x_n)\right\rangle-b\le\varepsilon+\xi_n\;\left\langle w,\varphi(x_n)\right\rangle+b-y(n)\le\varepsilon+\xi_n^*\\&\quad \xi_n,\xi_n^*\ge 0\quad \text{for all}\quad n=1,2,\cdots N.\end{aligned} \tag{16}$$

C is a regularization constant and b is a threshold. C determines the trade off between the flatness of $\tilde{f}$ and the amount up to which deviations larger than ε are tolerated. By constructing a Lagrange function from both the objective function and the corresponding constraints we yields the wolf dual optimization problem:

$$\begin{aligned}&\min\; W(a,a^*)=\frac{1}{2}\sum_{i,n=1}^N\left(\alpha_i-\alpha_i^*\right)\left(\alpha_n-\alpha_n^*\right)K(x_n,x_i)-\sum_{n=1}^N\left(\alpha_n-\alpha_n^*\right)y(n)+\varepsilon\sum_{i=1}^N\left(\alpha_i+\alpha_i^*\right)\\&\text{s.t.}\quad \sum_{i=1}^N\left(\alpha_i-\alpha_i^*\right)=0\qquad \alpha_n,\alpha_n^*\in[0,C]\quad \text{for all } n=1,2,\cdots N\end{aligned} \tag{17}$$

α_i,α_i^* in Eq. 17 are Lagrange multipliers. K describes the dot product of $\varphi(x)$ in the feature space F

$$K(x_n,x_i)=\varphi(x_n)\bullet\varphi(x_i)$$

K is so called kernel function and must satisfy Mercer conditions. At last the approximation of f is given by:

$$\tilde{f}(x_n)=\sum_{i=1}^N\left(\alpha_i-\alpha_i^*\right)K(x_n,x_i)+b \tag{18}$$

b can be computed by Karush-Kuhn-Tucker (KKT) conditions. Substituting $\beta_i=\alpha_i-\alpha_i^*$ into (3.3), we have the form:

$$\tilde{f}(x_n)=\sum_{i=1}^N\beta_iK(x_n,x_i)+b \tag{19}$$

Following (Girosi,1998), the approximation criterion of SVMA is:

$$\tilde{f}(x_n) = \underset{g \in F}{\arg\min}\left\{C\sum_{n=1}^{N}\left|y(n) - g(x_n)\right|_\varepsilon + \frac{1}{2}\|g\|_F^2\right\} \tag{20}$$

Where, $\|\bullet\|_F$ denotes the norm defined on the feature space F.

DERIVATION OF OUR ALGORITHM

We compute the following approximation of $f(x_n)$ according to the approximation criterion of SVMs in the scale subspaces V_j

$$\tilde{f}(x_n) = \sum_k c_k^{(j)}\phi_{jk}(n) + b \tag{21}$$

In order to make our notation consistent with SVMA, we have to modify slightly the expression from Eq. 14-21. The approximation criterion Eq. 20 of SVMs can be rewritten out more explicitly as:

$$\min_{\tilde{f}} H(\tilde{f}) = C\sum_{n=1}^{N}\left|y(n) - \tilde{f}(x_n)\right|_\varepsilon + \frac{1}{2}\|\tilde{f}\|_{V_j}^2 \tag{22}$$

Where:

$$\|\tilde{f}\|_{V_j}^2 = \sum_k \frac{c_k^{(j)2}}{\lambda_k^{(j)}} \tag{23}$$

is known form literature (Girosi, 1998). We substitute Eq. 23 into Eq. 22 and then:

$$\min_{\tilde{f}} H(\tilde{f}) = C\sum_{n=1}^{N}\left|y(n) - \tilde{f}(x_n)\right|_\varepsilon + \frac{1}{2}\sum_k \frac{c_k^{(j)2}}{\lambda_k^{(j)}} \tag{24}$$

Since it is difficult to deal with the ε insensitive function, the problem above is replaced by the following equivalent problem, in which an additional set of variables ξ_n, ξ_n^* are introduced:

$$\min_{\tilde{f}} H(\tilde{f}) = C\sum_{n=1}^{N}\left(\xi_n + \xi_n^*\right) + \frac{1}{2}\sum_k \frac{c_k^{(j)2}}{\lambda_k^{(j)}} \tag{25}$$

For the purpose of solving the above constrained minimization problem, we use the technique of lagrange multipliers. The lagrangian corresponding to the problem above is:

$$W = C\sum_{n=1}^{N}\left(\xi_n + \xi_n^*\right) + \frac{1}{2}\sum_k \frac{c_k^{(j)2}}{\lambda_k^{(j)}} + \sum_{n=1}^{N}\alpha_n^{*(j)}\left(y(n) - \tilde{f}(x_n) - \varepsilon - \xi_n^*\right) + \sum_{n=1}^{N}\alpha_n^{(j)}\left(\tilde{f}(x_n) - y(n) - \varepsilon - \xi_i\right) + \sum_{n=1}^{N}\left(r_n^{(j)}\xi_n + r_n^{*(j)}\xi_n^*\right) \tag{26}$$

Where, $\alpha_n^{(j)}, \alpha_n^{*(j)}, r_n^{(j)}, r_n^{*(j)}$ are Lagrange multiplier. We set to zero the derivatives with respect to $c_k^{(j)}$ and obtaining Eq. 28

$$\frac{\partial W}{\partial c_k^{(j)}} = 0 \tag{27}$$

$$c_k^{(j)} = \lambda_k^{(j)}\sum_{n=1}^{N}\left(\alpha_n^{*(j)} - \alpha_n^{(j)}\right)\phi_{jk}(n) \tag{28}$$

And $d_k^{(j)}$ can be obtained by similar steps in W_j space:

$$d_k^{(j)} = \lambda_k^{(j)}\sum_{n=1}^{N}\left(\alpha_n^{*(j)} - \alpha_n^{(j)}\right)\psi_{jk}(n) \tag{29}$$

Substituting $\beta_n^{(j)} = \alpha_n^{(j)} - \alpha_n^{*(j)}$ into Eq. 28, we get the more compact expression of $c_k^{(j)}$:

$$c_k^{(j)} = 2^{-j/2}\sum_{n=1}^{N}\beta_n^{(j)}\phi\left(2^{-j}n - k\right) \tag{30}$$

In Eq. 30, $\beta_n^{(j)}$ is given by solving following quadratic optimization problem:

$$\min\ W(a, a^*) = \frac{1}{2}\sum_{i,n=1}^{N}\left(\alpha_i^{(j)} - \alpha_i^{*(j)}\right)\left(\alpha_n^{(j)} - \alpha_n^{*(j)}\right)K_j(x_n, x_i) - \sum_{n=1}^{N}\left(\alpha_n^{(j)} - \alpha_n^{*(j)}\right)y(n) + \varepsilon\sum_{n=1}^{N}\left(\alpha_n^{(j)} + \alpha_n^{*(j)}\right) \tag{31}$$

$$\text{s.t.}\quad \sum_{i=1}^{N}\left(\alpha_n^{(j)} - \alpha_n^{*(j)}\right) = 0 \qquad \alpha_n^{(j)}, \alpha_n^{*(j)} \in [0, C] \qquad \text{for all } n = 1, 2, \cdots N$$

Where, the kernel function K_j takes the form of :

$$K_j(x, y) = \sum_k 2^{-j}\phi\left(2^{-j}x - k\right)\phi\left(2^{-j}y - k\right) \tag{32}$$

In summary, our algorithm is a two-step process. As the first step, multi-resolution signal decomposition based on SVMs is carried out through computing $c_k^{(j)}$ by Eq. 30. And as the second step, multi-resolution signal approximation based on SVMs is carried out through substituting obtained $c_k^{(j)}$ into (21) to compute $\tilde{f}(x_n)$. We do not compute $d_k^{(j)}$ in this study.

RESULTS AND DISCUSSION

Our algorithm involves the numerical computation of reproducing kernel that takes the form of (32). We call this kind of reproducing kernel scale kernel. Usually the function ϕ in Eq. 32 have no explicit expressions, hence the computation of scale kernel differs from common kernels such as Gauss kernel. In this study, the function ϕ we chosen is the Daubechies (DB) scale function.

Experiment 1: the comparision of two algorithms: A SINC original signal f(x) = sin c(x/48) and x∈[-192,192] are given in this experiment. We generate the sequence

$$\{y(n)\}_{n=1}^{N}$$

from f(x) with noise $\varepsilon_n = 0$ and the sampling rate dx = 8. It can be inferred that N = 49 because of uniformly sampling. Now two algorithms are performed on

$$\{y(n)\}_{n=1}^{N},$$

respectively and compute the decomposition (i.e., $c_k^{(j)}$) and approximation (i.e., $\tilde{f}(x_n)$) of the signal at the scale j(j = -1,0,1,2). The parameters our algorithm chosen are C = ∞, $\varepsilon = 10^{-7}$ and DB10 scale kernel. The results of decomposition and approximation are illustrated in Fig. 1 and 2. As shown in Fig. 1, the $c_k^{(j)}$ curves obtained by two algorithms are similar. The difference of them is shown in the third column. And shown in Fig. 2, two algorithms both achieve excellent approximation result when the scale j is small. But the way to approximation is obviously different when the scale j becomes larger gradually.

Experiment 2: multi-resolution stationary signal approximation: The experiment is designed for quantitatively measuring our algorithm's approximation performance for stationary signal. Because the approximation model can be seen as the de-noising model, we use input signal-noise-ratio (*abbr.* Input SNR), Output SNR, Improved SNR and mean square error (MSE) to measure the approximation performance. Four indexes are defined as follows:

$$\text{InputSNR} = 20\log_{10}\left[\frac{\text{std}(f(x_n))}{\text{std}(\varepsilon_n)}\right] \tag{33}$$

$$\text{OutputSNR} = 20\log_{10}\left[\frac{\text{std}(f(x_n))}{\text{std}(f(x_n)-\tilde{f}(x_n))}\right] \tag{34}$$

$$\text{ImprovedSNR} = \text{OutputSNR} - \text{IntputSNR} \tag{35}$$

$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}\left[f(x_n)-\tilde{f}(x_n)\right]^2 \tag{36}$$

Where, std(•) denotes the standard deviation function.

In the experiment let the stationary signal be f(x) = sin c(x) with x∈[-3.0,3.0]. Analogously to the experiment 1, we sample the signal f(x) with dx = 0.1 and N = 61. The

$$\{y(n)\}_{n=1}^{N}$$

are generated from f(x) by adding Gaussian noise with different Input SNR. Based on

$$\{y(n)\}_{n=1}^{N}$$

the experiment is conducted and then each of two algorithm's approximation result (i.e., $\tilde{f}(x_n)$) is obtained at

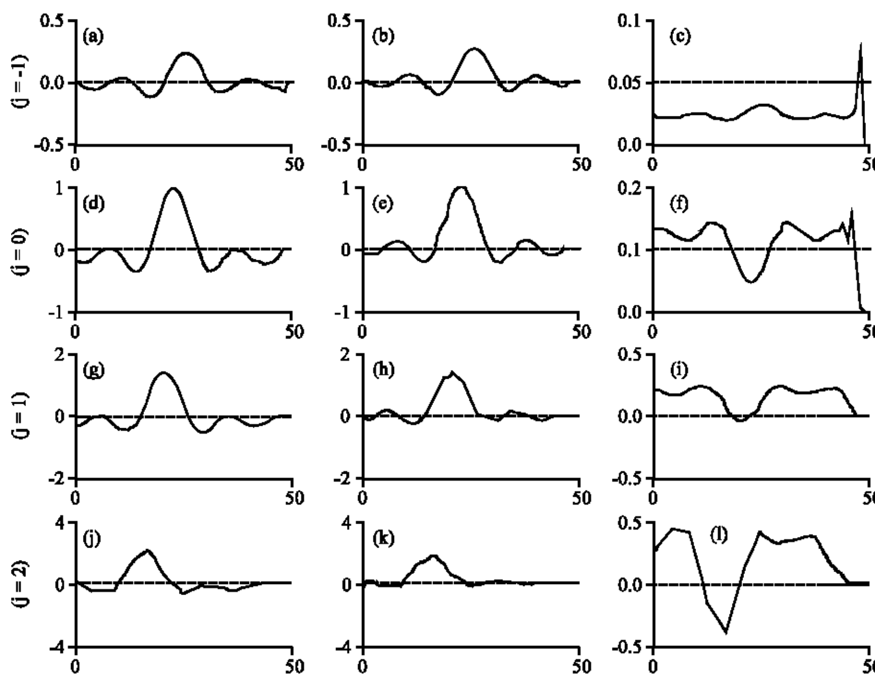

Fig. 1: The decomposition $c_k^{(j)}$ curves of two algorithms at different scale j. The subplots in the column 1 [i.e., (a), (d), (g) and (j)] are obtained by our algorithm, the column 2 by conventional algorithm and the column 3 are the differences of the column 1 and the column 2. The $c_k^{(j)}$ curves have been interpolated when j>0

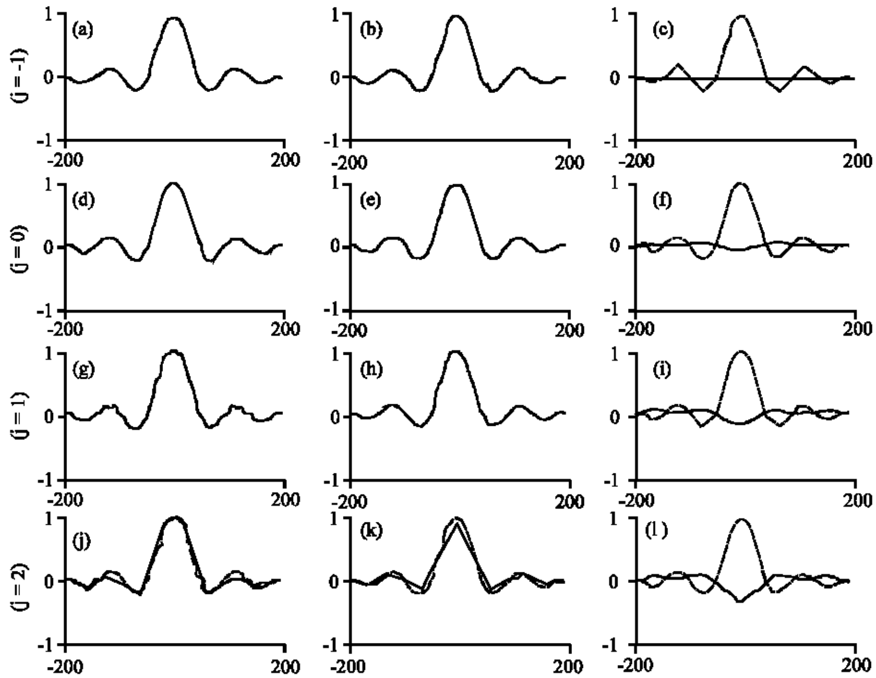

Fig. 2: The approximation $\tilde{f}(x_n)$ curves of two algorithms at different scale j. The subplots in the column 1 [i.e., (a), (d), (g) and (j)] are obtained by our algorithm, the column 2 by conventional algorithm and the column 3 are the differences of the column 1 and the column 2. The dot lines in the figure denote the sequence $\{y(n)\}_{n=1}^{N}$

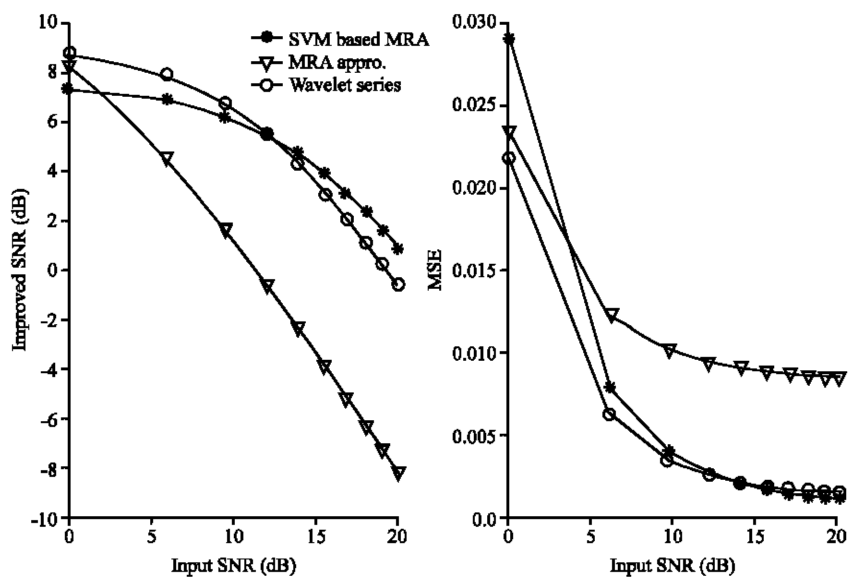

Fig. 3: The Improved SNR and MSE versus Input SNR for three algorithms in the experiment 1. SVM based MRA represents our algorithm, MRA appro. represent conventional algorithm and wavelet series represent wavelet series-based de-noising algorithm in the legend

the scale j = 0. Finally substituting the results obtained into Eq. 34 and 35, two algorithm's Improved SNR can be computed.

Based on the equivalence of the approximation model and the de-noising model, the paper compares above two algorithms with wavelet series-based de-noising algorithm besides the intercomparsion of two algorithms. There are typically three steps in the process of the wavelet series-based de-noising. Firstly, apply wavelet transforms to the samples y(n), obtaining empirical wavelet coefficients. And then apply one of following methods coordinatewise to the wavelet coefficients with specially chosen threshold. Finally, $\tilde{f}(x_n)$ can be recovered by inverse wavelet transforms. Usually threshold-estimating methods include SureShrink, VisulShrink, Minimaxi and HeurSure *et al*., In experiment 2 we employ SureShrink which is the abbreviation of Stein unbiased risk estimation. Simultaneously we use hard thresholding to deal with wavelet coefficients.

Figure 3 compares the Improved SNR and MSE varying with respect to Input SNR of above mentioned three algorithms. It is possible to conclude that our algorithm has higher Improved SNR and smaller MSE than conventional algorithm at the same Input SNR level. The larger the Input SNR is, the more obvious above conclusion is. The sole exception happens when the Input SNR is zero. When Input SNR>12dB, compared with wavelet series-based de-noising, our algorithm also has higher Improved SNR and smaller MSE. But the whole scenario is reversed when Input SNR<12dB).

CONCLUSION

The main contribution of the study is that the algorithm of multi-resolution signal decomposition and approximation based on SVMs has been proposed. The merit of the algorithm proposed are not only reduces the approximation error but also have better smoothness of approximation function. It is evident from the experimental data that the algorithm proposed not only can be applied to the approximation of stationary signals, but also has higher Improved SNR and smaller MSE than conventional algorithm at the same Input SNR level. Compared with wavelet series-based de-noising, the algorithm proposed seems consistent with the above conclusions when the Input SNR is larger than a threshold.

REFERENCES

Girosi, F., 1998. An equivalence between sparse approximation and support vector machines. Neural Comput., 10 (6): 1455-1480.

Hearst, M.A., B. Scholkopf and S. Dumais, 1998. Trends and controversies-support vector machines. IEEE Intel. Syst., 13 (4): 18-28.

Mallat, S., 1989. Multiresolution approximation and wavelet orthonormal bases of L2. Trans. Am. Math. Soc., 315: 69-87.

Strang, G., 1989. Wavelets and dilation equations. SIAM Rev., 23 (5): 614-627.

Sweldens, W. and R. Piessens, 1994. Quadrature formulate and asymptotic error expansions for wavelet approximation of smooth functions. SIAM. J. Numer. Anal., 31 (2): 427-441.

Unser, M., 1996. Approximation power of biorthogonal wavelet expansions. IEEE Trans. Signal Processing, 44 (3): 519-527.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, N.Y.

Vapnik, V., S. Golowich and A.J. Smola, 1997. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. Neural Information Processing Syst. MIT Press.

Williamson, R., A. Smola and B. Scholkopf, 1999. Entropy Numbers, Operators and Support Machine Kernel. In: Advances in Kernel Methods-Support Learning. Cambridge, M.A., MIT Press, pp: 127-144.