WAVELET ESTIMATION FOR NONPARAMETRIC REGRESSION: BEYOND GAUSSIAN NOISE I

by

Yuhai Wu
Purdue University

Technical Report #99-02

Department of Statistics
Purdue University
West Lafayette, IN USA

April 1999

*

WAVELET ESTIMATION FOR NONPARAMETRIC REGRESSION: BEYOND GAUSSIAN NOISE I

by

Yuhai Wu
Purdue University

Abstract

As orthogonal wavelet transform concentrates most of the information of a signal into a few large wavelet coefficients and leaves the rest of the wavelet coefficients close to zero. At the same time, it evenly distributes i.i.d. Gaussian noise into i.i.d. Gaussian noise. The true regression curve therefore could be recovered efficiently from the several significant large wavelet coefficients only. This is the reason for the success of wavelet shrinkage methods in nonparametric regression when the noise is Gaussian. Does this method still work when the noise is non-Gaussian? As the true curve stores most of its information in a few coefficients no matter what kind of noise is involved, this information can be successfully recovered as long as the estimators keep the significant coefficients and discard most of the noise. This indicates that the wavelet shrinkage methods should be successful at least for some non-Gaussian cases. However, because an orthogonal wavelet transform cannot keep i.i.d non-Gaussian noise in the wavelet domain from i.i.d. non-Gaussian regression noise, technical difficulties appear in the statistical analysis. The success of the wavelet shrinkage methods for non-Gaussian regression noise is demonstrated in the example of the shifted exponential noise. In this case, the wavelet estimators achieve "noise-free" reconstruction, near "idea" risk and adaptivity to wide function classes.

Wavelet Estimation for Nonparametric Regression: Beyond Gaussian Noise I

Yuhai WU, Department of Statistics, Purdue University, West Lafayette, IN 47907

An orthogonal wavelet transform concentrates most of the information of a signal into a few large wavelet coefficients and leaves the rest of the wavelet coefficients close to zero. At the same time, it evenly distributes i.i.d. Gaussian noise into i.i.d. Gaussian noise. The true regression curve therefore could be recovered efficiently from the several significant large wavelet coefficients only. This is the reason for the success of wavelet shrinkage methods in nonparametric regression when the noise is Gaussian. Does this method still work when the noise is non-Gaussian? As the true curve stores most of its information in a few coefficients no matter what kind of noise is involved, this information can be successfully recovered as long as the estimators keep the significant coefficients and discard most of the noise. This indicates that the wavelet shrinkage methods should be successful at least for some non-Gaussian cases. However, because an orthogonal wavelet transform cannot keep i.i.d. non-Gaussian noise in the wavelet domain from i.i.d. non-Gaussian regression noise, technical difficulties appear in the statistical analysis. The success of the wavelet shrinkage methods for non-Gaussian regression noise is demonstrated in the example of the shifted exponential noise. In this case, the wavelet estimators achieve "noise-free" reconstruction, near "ideal" risk and adaptivity to wide function classes.

1. INTRODUCTION

Consider the standard nonparametric regression problem

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n \ (n = 2^J), \tag{1}$$

where $x_i = \frac{i}{n+1}$, and $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. $N(0, \sigma^2)$. Donoho and Johnstone(1994) proposed a wavelet shrinkage procedure which consists of the following three steps:

(1). Transform the observed data into the empirical wavelet coefficients using the interval-adapted pyramidal filtering algorithm [Cohen, Daubechies, Jawerth and Vial(1992)].

(2). Keep or throw away(set equal to zero) each empirical wavelet coefficient by using a hard or soft thresholding rule with a specially-chosen threshold value.

(3). Construct the estimator of the regression curve by transforming the estimated wavelet coefficients back.

This procedure looks similar to the Fourier series method (orthogonal series method). However, they have significant differences in two aspects. First they have different bases. Wavelet shrinkage procedures use a wavelet basis while Fourier series methods use a Fourier basis. Second wavelet shrinkage procedures use "keep" or "kill" thresholding rules which are nonlinear. On the other hand, Fourier series methods use rescale smoothing rules which are linear.

The advantages of wavelet bases over Fourier bases come from the following facts:

(1). Wavelet bases are unconditional bases of various functional classes such as Holder, Sobolov, Besov and Triebel etc., while Fourier bases are unconditional bases only for Sobolov spaces. [Donoho(1993)]

(2). The special multiresolution structure of the wavelet basis has the localization property in both time and frequency domains and results in the sparse reconstruction that makes the wavelet transformation an efficient tool for data compression and de-noising. On the other hand, Fourier bases lack the natural localization property.

(3). The wavelet transformation uses only $O(n)$ operations which is faster than the $O(nlog(n))$ operations of the FFT.

The advantage of the nonlinear wavelet threshold methods over the linear Fourier series smoothing methods is that the wavelet methods achieve or almost achieve the optimal rate of convergence for wide function classes which the Fourier methods cannot do. One of the consequences is that the wavelet thresholding method is adaptable to a wide range of spatial and frequency inhomogeneous function classes to which the Fourier method cannot adapt.

For the wavelet shrinkage procedure, a proper threshold value is crucial from both a theoretical and practical point of view. The fundamental rule for choosing a threshold value is the Universal Threshold which is incorporated into the VisuShrink procedure of Donoho and Johnstone(1994). The Universal Threshold value is set as $T_{UV} = \sqrt{2log(n)}\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of the noise σ which can be derived from the median absolute deviation of

the wavelet coefficients at the finest resolution level J-1. The VisuShrink procedure has the following properties simultaneously for various smoothness classes.

(1). Guarantees the noise-free reconstruction.

(2). The risk $R_{n,\sigma}(\hat{f}, f)$ of $\hat{f}$ is within a logarithmic factor of the ideal risk $R_{n,\sigma}(DP, f)$:

$$R_{n,\sigma}(\hat{f}, f) \leq (2logn + 1)\left\{R_{n,\sigma}(DP, f) + \sigma^2\right\}.$$

There are also several other procedures that offer different threshold selections based on different criterions. Some examples are RiskShrink in Donoho and Johnstone(1994), SureShrink in Donoho and Johnstone(1995) and Cross-validation in Nason(1996). Also see Abramocich and Benjamini(1996), Ogden and Parzen(1996), and Cai(1997) among others. For the discussion of Gaussian noise with correlations, see Wang(1995) or Johnstone and Silverman(1996). These methods are mainly based on Gaussian noise although some of them may work for non-Gaussian noise. What happens when the noise are not Gaussian? Neumann and Sachs(1996) showed that for the wavelet coefficients at coarser levels, the non-Gaussian noise problems can be treated the same as the Gaussian noise case because the central limit theorem plays a role. However, it is not clear how to deal with the wavelet coefficients at the finest levels. Unfortunately for a real problem the wavelet coefficients at the finest levels are the most important concern because 75% of all the wavelet coefficients are on the two finest levels. The question then is whether the DJ VisuShrink procedure still works for the nonparametric regression with non-Gaussian noise?

There are other difficulties appearing when a regression problem has an i.i.d. non-Gaussian noise. The wavelet coefficients of our observations through the orthogonal wavelet transform are no longer independent, and they are also not identically distributed at different levels. Even worse is that we cannot ignore the wavelet basis itself when we select the optimal threshold value. The wavelet functions influence the selection of the threshold value.

In this paper, we will try to shed some light on wavelet thresholding estimation for non-parametric regression with non-Gaussian noise. We will use wavelet shrinkage methods to estimate the regression curves when the noise is i.i.d. and shifted exponentially distributed. The rest of the paper is arranged as follow: Section 2 briefly introduces wavelet and wavelet

estimation in regression. The basic results of our study are in Section 3 while some extensions and discussion are in Section 4. Numerical simulations are displayed in Section 5. The final conclusion is left to Section 6. Relevant proofs are put in the appendix.

2. WAVELET ESTIMATION FOR NONPARAMETRIC REGRESSION

2.1 Wavelets

For any function ψ satisfying the condition $\int \psi(x)\,dx = 0$, we call ψ the mother wavelet. The most useful wavelets in functional analysis and statistics are given by $\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi(2^j x - k)$, $j, k \in Z$ where $(\psi_{j,k})_{j,k\in Z}$ consists of an orthonormal basis for the space $L^2(R)$. The mother wavelet ψ is described by two parameters M = the number of vanishing moments, and S = the support length. A function $f \in L^2(R)$ can then be written in the form of a wavelet decomposition

$$f(x) = \sum_{j,k\in Z} \theta_{jk}\psi_{jk}(x)$$

with $\theta_{jk} = \int f(x)\psi_{jk}(x)\,dx$, the wavelet coefficients of f. The oldest known wavelets are generated by the Haar function

$$\psi(x) = \begin{cases} -1, & 0 \le x \le \frac{1}{2} \\ 1, & \frac{1}{2} \le x \le 1 \\ 0, & otherwise. \end{cases} \tag{2}$$

The most popular wavelets are Daubechies wavelets whose mother wavelets have different compact supports and finite vanishing moments. Unfortunately, all well known compactly supported mother wavelet functions (except for the Haar function) do not have closed-form analytic formula. However, they can be computed with arbitrarily high precision by a numerical method which is called cascade algorithm. Therefore, a compactly supported mother wavelet function can be expressed by a numerical vector $\overrightarrow{a} = (a_1, a_2, \ldots, a_n)$. See Daubechies(1991). This vector play an important role in computing the thresholding level later in this paper. By modifying some of the $\psi_{j,k}$, we can form an orthonormal basis for the function space $L^2[0,1]$.

2.2 Wavelet Estimation

Given the observations $y = (Y_1, Y_2, \ldots, Y_n)$ in the regression model (1), we can derive its wavelet coefficients $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ by the discrete wavelet transformation which can be written as an orthogonal matrix $\mathcal{W}$ such that $\omega = \mathcal{W}y$. If $\theta = \mathcal{W}f$ and $z = \mathcal{W}\epsilon$, where $f = (f(\frac{1}{n+1}), f(\frac{2}{n+1}), \ldots, f(\frac{n}{n+1}))$, $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$, then we have

$$\omega_i = \theta_i + z_i, \qquad i = 1, 2, \ldots, n.$$

In order to reflect the special wavelet structure, the coefficients are written as

$$\omega_{jk} = \theta_{jk} + z_{jk}, \qquad j = 0, 1, \ldots, J-1, \qquad k = 0, 1, \ldots, 2^j - 1,$$

the left one is denoted as $\omega_{-1,0}$. Define the soft threshold function $T_s(a, b) = sgn(a)(|a| - b)_+$. For each wavelet coefficient ω_i , we can apply this function with threshold value $b = \sqrt{2log(n)}\hat{\sigma}$ to get an estimator $\hat{\theta}_i = T_s(\omega_i, b)$ of θ_i. The Donoho and Johnstone VisuShrink estimator is then constructed by transforming $\hat{\theta}_i$ back to give

$$\hat{f} = \mathcal{W}^T\hat{\theta}.$$

The risk of $\hat{f}$ is evaluated by $E\|\hat{f} - f\|^2$ which can be approximated by $\frac{1}{n}E\|\hat{f} - f\|_{2n}^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{f}(x_i) - f(x_i))^2$. In practice, a fast $O(n)$ algorithm is used to carry out the matrix transformations $\mathcal{W}$ and $\mathcal{W}^T$.

2.3 Threshold rule

The essential part of the wavelet thresholding method is how to select the threshold value. Two different criterions have been used to construct wavelet threshold estimators. One is de-noising, to get a "noise-free" and "good visual" estimator; the other is to minimize the mean-squared error. The optimal estimators under the first rule are, with high probability, as smooth as the true underlying function and can achieve near ideal risk. The VisuShrink procedure is a typical example under this rule. Two examples under the second rule are the SureShrink method and the Cross-Validation method which try to find the best bias-variance trade off to obtain the minimum mean-squared error based on the observed data.

3. NONPARAMETRIC REGRESSION WITH NON-GAUSSIAN ERRORS

Since the discrete wavelet transformation is orthonormal, Gaussian noise contaminates all wavelet coefficients equally so that after the discrete wavelet transformation the noise z in the wavelet domain remains i.i.d. normal. However, when the noise in the model (1) do not follow a normal distribution, its corresponding noise in wavelet coefficients z, will not keep the i.i.d. property. They are in general no longer independent and also do not have the same distribution any more at different wavelet resolution levels. The Haar wavelets are an exception. Under the Haar wavelet transform, i.i.d. observations with any noise distribution will keep the i.i.d. property among their wavelet coefficients in each resolution level although the wavelet coefficients have different distributions at different resolution levels. Even though things become more complicated and harder to analyze for the non-Gaussian noise problems, we still hope that we can use the wavelet thresholding procedure to solve the problem. Then the main question is whether we can keep using the threshold values derived for Gaussian noise and if we have to use different threshold levels, what kind of property does the new thresholding estimator have. We will answer these questions through the following scenario.

For the nonparametric regression model (1), instead of having a Gaussian noise, we consider the following exponential distributed noise: Let $\epsilon_1, \ldots, \epsilon_n$ be i.i.d. exponential distributed with mean 0 and variance σ^2, that is, let the density function $f(x) = \frac{1}{\sigma}\exp\{-(\frac{x}{\sigma} + 1)\}1_{x+\sigma>0}$. From the wavelet transformation, we get a coefficient sequence in the wavelet domain

$$\omega_{j,k} = \theta_{j,k} + Z_{j,k}, \qquad k = 0, \ldots, 2^j - 1, \qquad j = 0, \ldots, J-1,$$

and $\omega_{-1,0}$. We estimate the wavelet coefficients $(\theta_{j,k})$ with the soft threshold rule

$$\hat{\theta}_{j,k} = sgn(\omega_{j,k})(|\omega_{j,k}| - \lambda)_+$$

and consider the risk measure

$$R(\hat{\theta}, \theta) = E||\hat{\theta} - \theta||_{2n}^2 = \sum_{i=1}^{n} E(\hat{\theta}_i - \theta_i)^2.$$

In the rest of this section we will focus our attention on the Haar wavelets and discuss the general situation in the next section. The main feature for the arbitrary wavelets already appear in this special case and the results and proofs are more transparent.

For the above shifed exponential noise and Haar system, we have the following theorem.

Theorem 1. Within each wavelet coefficient resolution level, the noise terms $\{Z_{j,k}\}$ are independent and identically distributed. At the finest level, the noise variables $Z_{J-1,k}, k = 0\ldots, 2^{J-1}-1$ are i.i.d. and also have the same density function

$$f_{Z_{J-1,k}}(x) = \frac{\exp\{-\frac{\sqrt{2}x}{\sigma}\}}{\sqrt{2}\sigma} 1_{x>0} + \frac{\exp\{\frac{\sqrt{2}x}{\sigma}\}}{\sqrt{2}\sigma} 1_{x<0}$$

In order to exclude the noise in the reconstruction, an appropriate threshold value must be determined. This is the crucial step in the wavelet thresholding procedure. Unfortunately, the choice of the threshold values depends on the tail behavior of the empirical coefficients. Therefore, the optimal selection of the threshold value has to be determined case by case and level by level. The following theorem gives the optimal choice of the threshold value for the case we are discussing here.

Theorem 2. Under the principal of de-noising, the optimal threshold value at the finest resolution level $j = J-1$ is equal to $2^{-\frac{1}{2}} log(n)\hat{\sigma}(1+o(1))$, where $\hat{\sigma}$ is an estimate of σ.

From Lemma 1 in the appendix, we can see that with these threshold values, we can almost surely throw away all the noise at each finest resolution level asymptotically. Therefore every sample in the wavelet transform in which the underlying signal is exactly zero will, in high probability, be estimated as zero. On the other hand, the noise is almost surely to be present if we choose the threshold value $\sqrt{2log(n)}\hat{\sigma}$, which is the optimal selection of the Universal Threshold value based on Gaussian noise. To increase the de-noising quality in finite sample problems, we can use $2log(n)\hat{\sigma}$ instead of $log(n)\hat{\sigma}(1+o(1))$, where the small order in $1+o(1)$ is usually taken to be $klog(log(n))$ with a positive constant of k.

Figure 1 displays an example of wavelet estimation using two thresholding values on a simulated data set ($n = 1024$) and Haar wavelets. The underlying function is the Blocks function which was used in Donoho and Johnstone(1994). The noisy version of the Blocks

function is generated by adding exponential noise E(0,1) and then normalized to have signal-to-noise ratio $||\text{signal}||_{2n}/||\text{noise}||_{2n} = 7$. The estimators of the Universal Threshold value $\sqrt{2log(n)}\hat{\sigma}$ derived from Gaussian noise and our new threshold value $2log(n)\hat{\sigma}$ are compared here. The DJ's Universal method can not give us a noise-free estimate because the noise here has a heavier tail. More numerical simulation results are given in next section. The software used in this paper is S+Wavelets StatSci(1993).

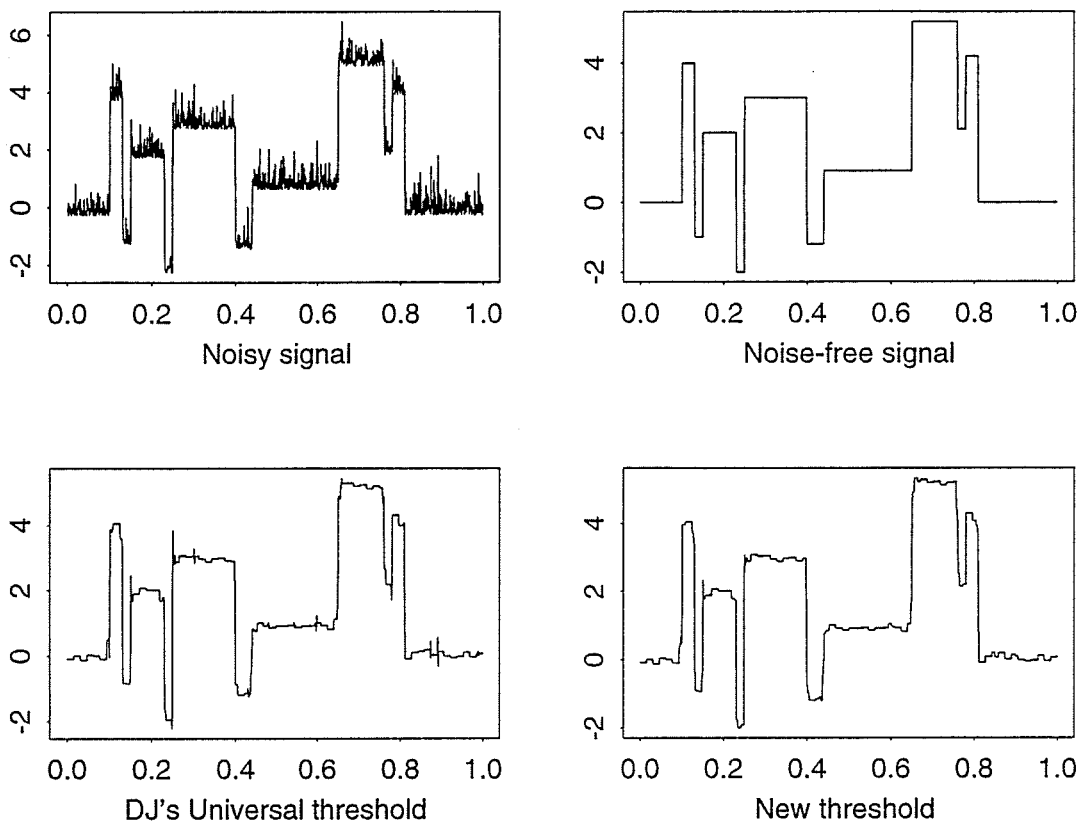

Figure 1. n=1024,wavelet=Haar, snr=7

The following theorem says that the risk will always be close to the ideal oracle risk if we use the thresholding procedure under the new threshold selection

Theorem 3. The mean squared error of our new procedure at the finest resolution level satisfies

$$E||\hat{\theta} - \theta||_{2n_1}^2 \leq \frac{(log(n_1))^2}{2}\left\{\frac{\sigma^2}{2} + \sum_{i=0}^{n_1-1}\min(\theta_i^2, \sigma^2)\right\} \tag{3}$$

where $n_1 = \frac{n}{2}$ is the sample size at the finest resolution level and θ is the true wavelet coefficients in the finest resolution level .

We can similarly prove that the above results will also apply to the other finest resolution levels:

Corollary 1. Under the de-noising rule, the threshold value at each resolution level $k = J-2, J-3, \ldots$ is $2^{\frac{k-J}{2}} log(n_k)\hat{\sigma}(1+o(1))$, and its corresponding risk is within a square of log term of the ideal risk.

In words, the soft threshold estimator based on our new selection of the threshold value guarantees that its risk is at most within a factor of essentially $\frac{(logn)^2}{2}$ of the ideal risk with help of the oracle (see Donoho and Johnstone 1994). Hence, for the non-Gaussian cases, the thresholding method with the new threshold value can also enjoy all the optimal properties such as the noise-free and near ideal risk.

4. DISCUSSION

4.1 Oracle and Ideal risk

Suppose we have the observations $\omega = (\omega_i)_{i=1}^n$ according to

$$\omega_i = \theta_i + \epsilon z_i \quad (i = 1, \ldots, n),$$

where z_i have mean zero and variance 1, and $\epsilon > 0$ is the known noise level. We want to estimate the object $\theta = (\theta_i)$ with l_2-loss. We consider a family of diagonal linear projections

$$T_{DP}(\omega, \theta) = (\delta_i \omega_i)_{i=1}^n, \quad \delta_i \in \{0, 1\}.$$

Such estimators 'keep' or 'kill' each coordinate. If we had complete information about $(\theta_i)_{i=1}^n$, which is called the "oracle" in Donoho and Johnstone(1994), then the ideal projection rule is to set $\delta_i = 1_{(|\theta_i| > \epsilon)}$. So the ideal diagonal projection retains or throws away θ_i by comparing the signals with the noise level. This yields the ideal risk

$$R_\epsilon(DP, \theta) = \sum_{i=1}^n min(\theta_i^2, \epsilon^2).$$

As $(\theta_i)_{i=1}^n$ is what we want to estimate, we do not have such an "oracle". Hence, in general the ideal risk $R_\epsilon(DP, \theta)$ cannot be attained for all θ by any estimator. However, our threshold estimator gets close to it within a squared log term.

4.2 Haar function

Consider the model (1) again: the wavelet coefficients are always i.i.d. normal only if the regression noise are i.i.d. normal distributed and it does not matter which wavelet basis is used. Unfortunately, this property cannot be shared by any other noise distribution. For any i.i.d. non-Gaussian regression noise, the wavelet coefficients of the regression observations are not i.i.d. in general. At most they are uncorrelated within each resolution level. However, there is one exception: if we use the Haar wavelet basis, then in the model (1) with any distribution of i.i.d. regression noise their wavelet coefficients are still i.i.d. within each resolution level. In other words, among all noise distributions, the Gaussian distribution is the only one that preserves the i.i.d. property from regression noise to wavelet coefficients, for all wavelet basis. On the other hand, amongst all commonly used wavelets, the Haar function is the only wavelet that preserves the i.i.d. property from regression noise to wavelet coefficients in each resolution level, no matter what kind of a distribution the noise has. This can be easily checked by looking at the length of the support of a mother wavelet (see Daubechies 91).

4.3 Threshold selection with general wavelet basis

For Gaussian noise, we know that the wavelet basis itself has no influence on the selection of threshold value. We do not have to modify our procedure if we want to use a different wavelet. However, when the regression has non-Gaussian noise, things will be totally different. The wavelet basis will be involved in the selection of threshold value. Theoretically we need to modify the threshold values if we want to change the wavelet basis. Furthermore since the wavelet coefficients in general have different distributions at different resolution levels, we also need to change the threshold values at different resolution levels. Therefore, when we go from Gaussian noise to non-Gaussian noise, not only are the wavelet coefficients no longer i.i.d., but also the wavelet basis itself plays a role in the threshold value selection.

In Section 3, we investigated model (1) for the exponentially distributed noise with a Haar wavelet basis. The Haar wavelet transformation keeps the i.i.d. property within each wavelet resolution level for any i.i.d. regression noise distribution. Now we continue our discussion of Section 3: we keep the i.i.d. exponential distributed noise, but we change from Haar wavelet function to a general wavelet function like $D4$ or $S8$ of Daubechies wavelets. The follow theorem tells us how the wavelet basis affects the selection of threshold value:

Theorem 4. To achieve the optimal de-noising, the threshold value at each resolution level $k = J-1, J-2, J-3, \ldots$ is

$$\lambda_k = 2^{\frac{k-J}{2}} \max_{1 \le i \le S} \{a_i\} log(n_k) \hat{\sigma}(1 + o(1)),$$

where a_i, $1 \le i \le S$, are the absolute value of the components of the underlying mother wavelet function. The corresponding risk of the new thresholding estimator is still within a squared log term of the ideal risk.

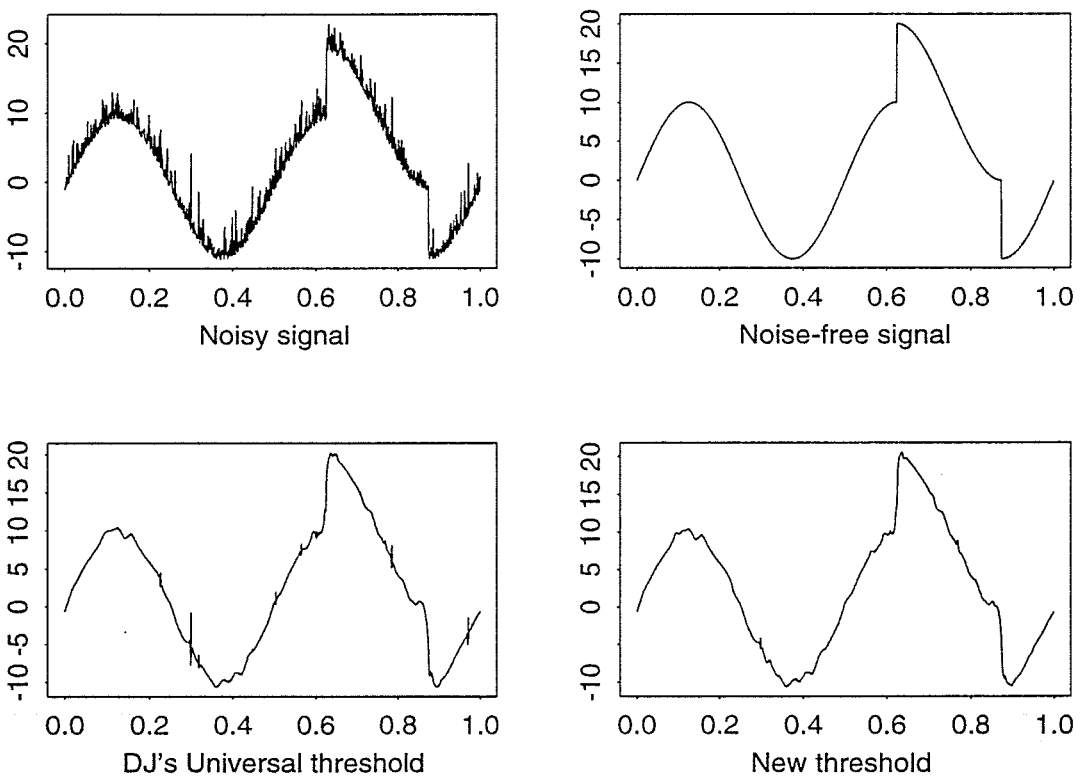

Figure 2. n=1024, wavelet=S8, snr=7

Figure 2 shows the noisy Jumpsine function with exponential noise $E(0,1)$ normalized to have signal-to-noise 7. Daubechies's S8 wavelet is used for both the DJ-Universal estimator and our new threshold estimator. The data size is 1024.

For non-Gaussian noise, the wavelet basis has some effect on both the selection of the threshold value and its induced risk. However, this effect is limited to within a constant factor. It is not large enough to give a different rate. A brief proof of this result is given in the appendix.

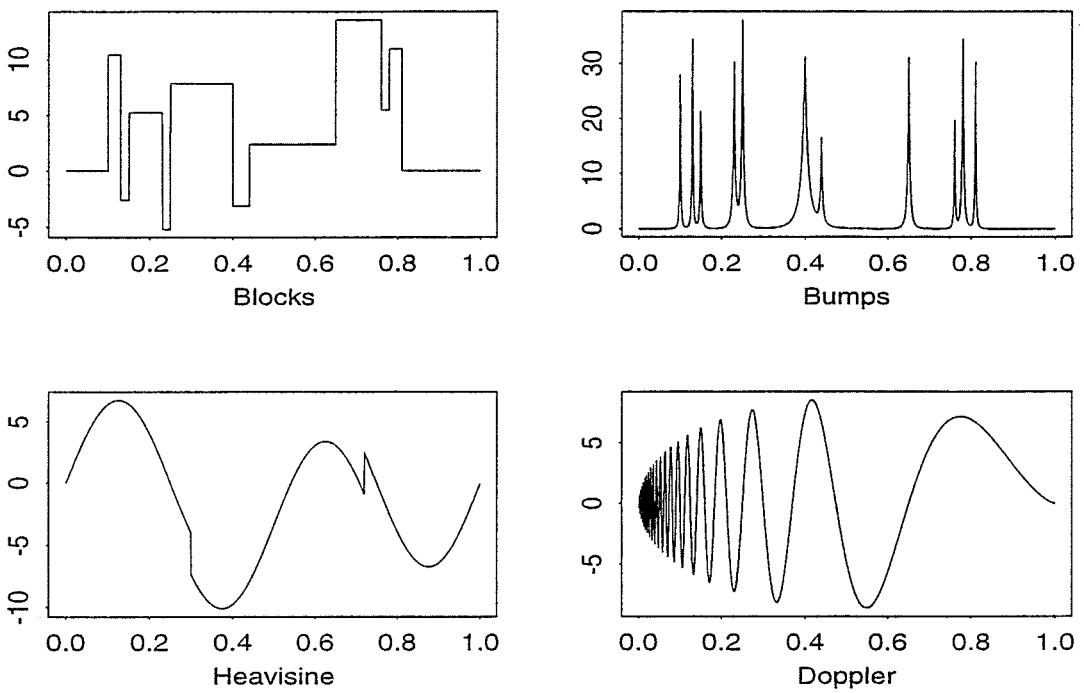

Figure 3. Four functions with n=2048, snr=5

5. SIMULATION AND CONCLUSION

5.1 Simulated results

Some numerical simulations are conducted to compare the performance of our new threshold rule with the DJ-Universal, the DJ-Minimax, as well as the DJ-Adaptive(hybird of SURE and Universal) methods. Figure 3 displays four functions, Blocks, Bumps, HeaviSine and Doppler, which were used in Donoho and Johnstone(1994,1995). Figures 4 and 5 are the noisy versions with the four functions added with Gaussian noise $N(0,1)$ and Exponential

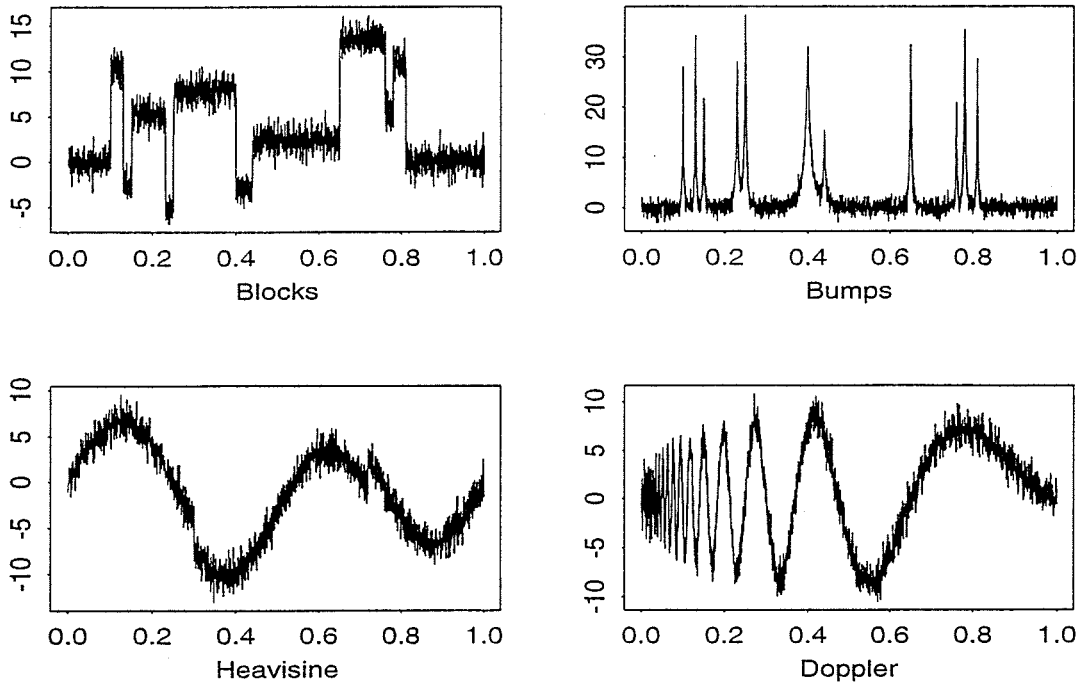

Figure 4. Four functions added with Gaussian noise N(0,1), n=2048 and snr=5

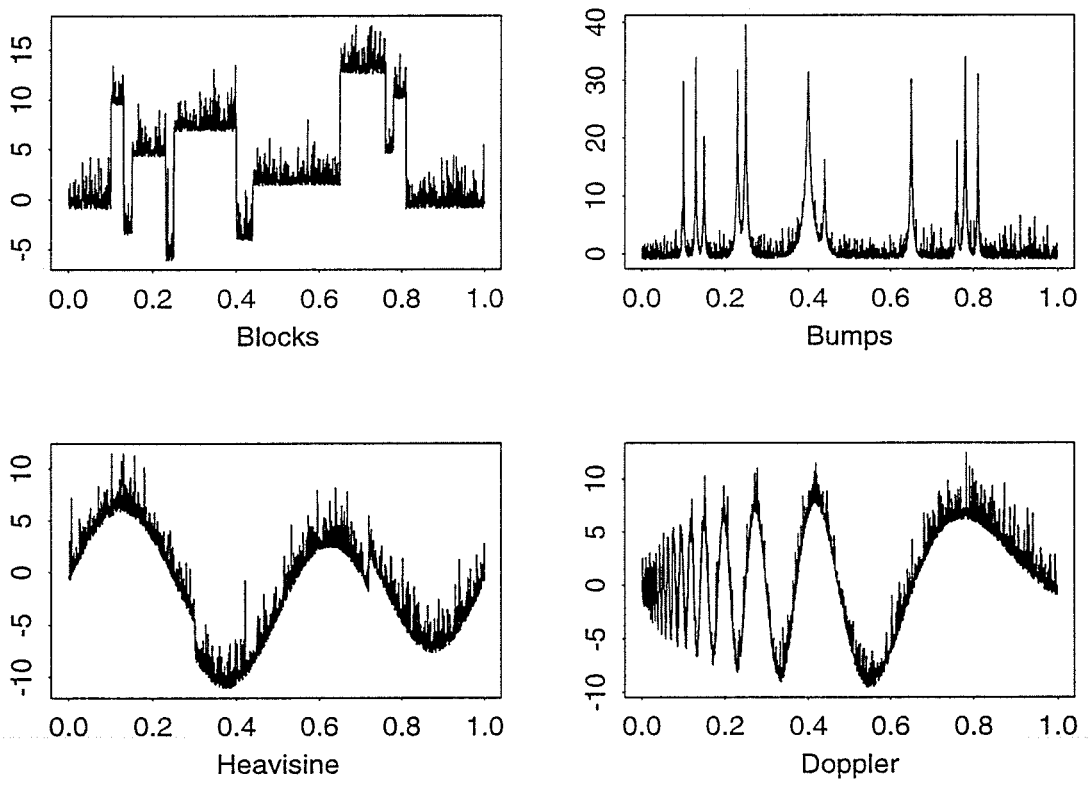

Figure 5. Four functions added with Exponential noise E(0,1), n=2048 and snr=5

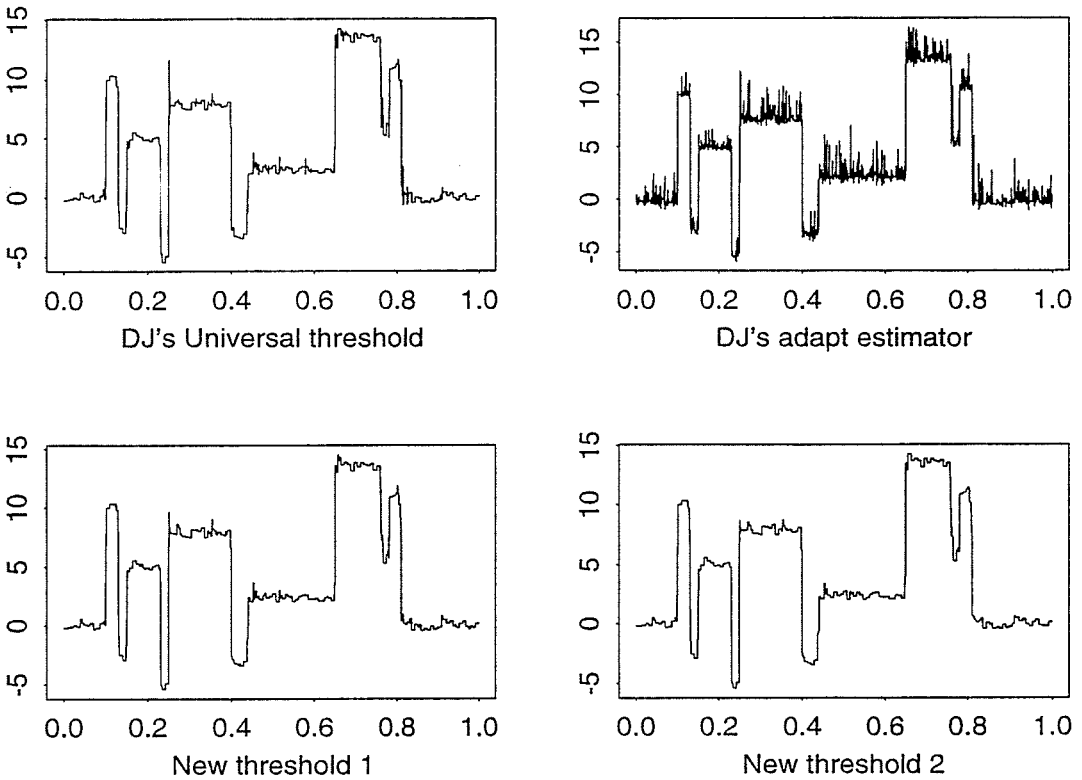

Figure 6. Threshold estimators for the Exponential noisy data

distributed noise $E(0,1)$ respectively. The signal-to-noise ratio is set to be 5. More noisy spikes are seen from the Exponentially distributed noise than those from Gaussian noise, even though they have the same standard deviation. Figures 6-9 show the results of the four wavelet threshold estimates for the four functions added with Exponential noise $E(0,1)$ which are displayed in Fig.4. In order to save space, we omit the displays of the estimates for the Gaussian noise $N(0,1)$ cases. These can be found in Donoho and Johnstone(1994).

For the Blocks function, we use the Haar basis functions. The two new estimators are constructed based on the threshold values which are determined by $log(n)$ and $2*log(n)$ with the level adjustments of Theorem 2 and Corollary 1. For the functions Bumps, HeaviSine and Doppler, we use the wavelet function $S8$. The new estimators 1 and 2 are constructed respectively with $1.517166*log(n)$ and $2*log(n)$ based on the Theorem 4. From the pictures we see that the reconstructed estimator based on the Adaptive method displays a lot of noise. The Universal estimator shows much less noise while there are still some blips left. The New

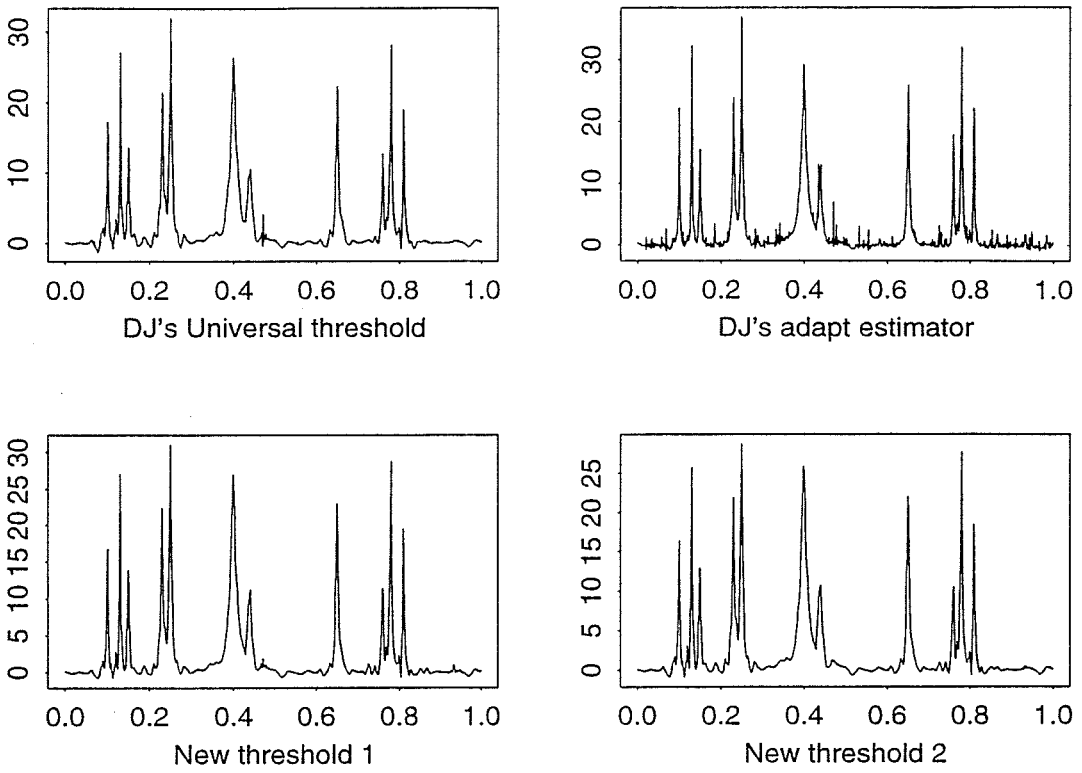

Figure 7. Threshold estimators for the Exponential noisy data

threshold 1 gives us an even better display which is based on $log(n)$ for the Blocks function and $1.517166 * log(n)$ for the other three functions. The New threshold 2 gives us an excellent 'noise-free' estimator which is constructed from $2 * log(n)$. From visual assessments of quality of fit, the new estimators dominate the DJ-methods when the noise follows an Exponential distribution.

However, this is only the first half of the story. It is known that an estimator which is visually preferable shows the worse numerical result when the noise are Gaussian. This is described as the divergence between the usual numerical and visual assessments of quality of fit, see Donoho and Johnstone(1994). Now we look at the second half of the story. Table 1 and 2 give the results of the numerical study. The only difference between Table 1 and Table 2 is their signal-to-noise ratio. Table 1 contains the simulation results when the signal-to-noise ratio is equal to 7, and Table 2 is the results when the signal-to-noise ratio is 5. The columns for Universal, Minimax, Adaptive in the tables are the results of Donoho and Johnstone's three estimators. The columns listed as New1, New2 and New3 correspond

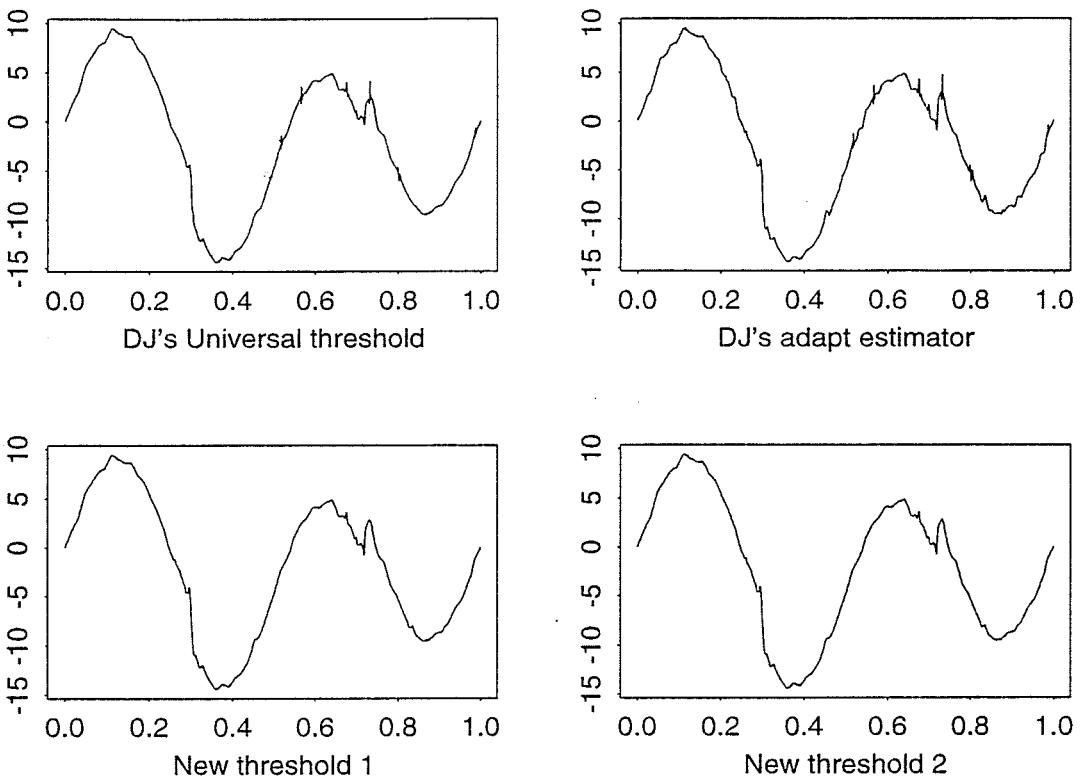

Figure 8. Threshold estimators for the Exponential noisy data

to our new estimators based on $log(n)$, $1.517166 * log(n)$, and $2 * log(n)$. For each of the several sample sizes, each of the four functions, and each of the two noises, 100 replications were performed. The average root mean squared errors and their standard deviations are tabulated. The dark numbers are the smallest average root mean squared errors for each row.

From the tables, we can see that Table 1 and 2 give us similar features:

First let us look at the comparison between the two categories of the DJ-methods and our new methods:

(1). The new estimators are better for all cases studied and every large samples ($n \geq 2048$) when using $E(0,1)$ noise.

(2). The new estimators completely dominate the HeaviSine case and the Doppler case for large samples ($n \geq 1024$) .

(3). The DJ-estimators win all Blocks and Bumps studied when the noise is $N(0,1)$. They also control the small sample studies of Doppler.

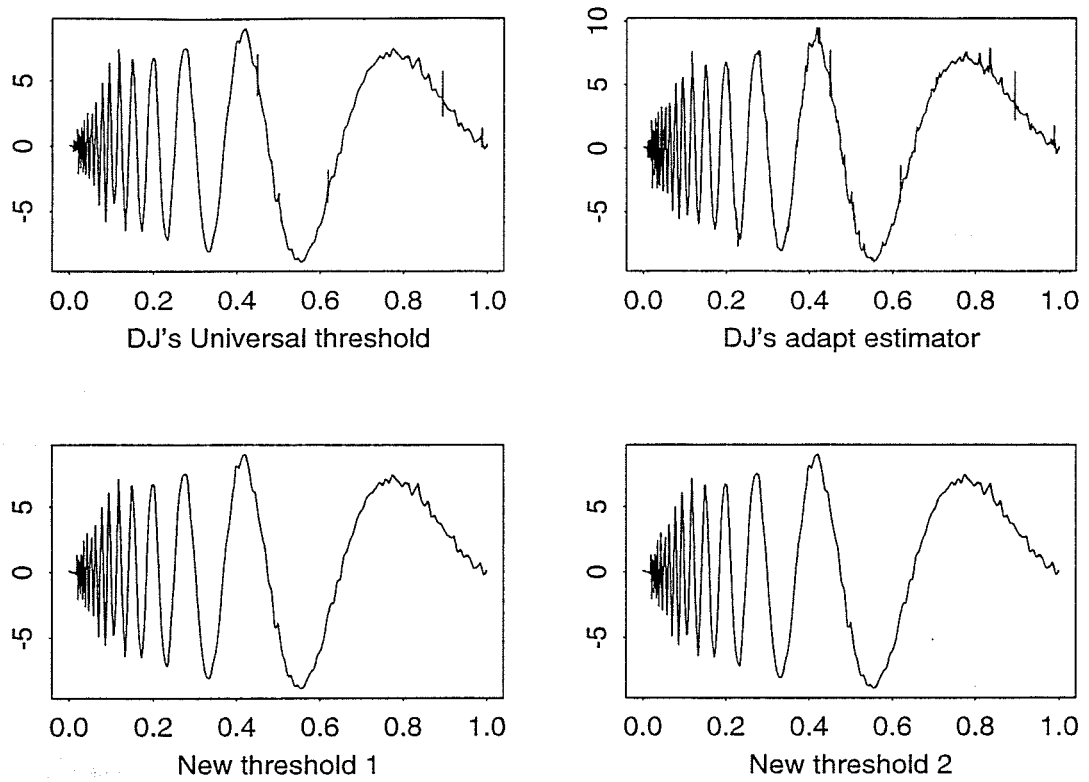

Figure 9. Threshold estimators for the Exponential noisy data

Second, we can see the divergence phenomenon between the usual numerical and visual assessments among the three new estimators with the exceptions of HeaviSine where New 2 and New 3 dominates the $N(0,1)$ noise and $E(0,1)$ noise categories.

Third, amongst the DJ-methods, the Adaptive Hybird method totally controls the Bumps with $N(0,1)$ noise case and small samples of $E(0,1)$ noise. It is also competitive or close to the best for the Blocks case with $N(0,1)$ noise. However, it did poorly for the Blocks case with $E(0,1)$ noise, especially, it is the worst for large samples. The Minimax method shows the power in Doppler and HeaviSine with $N(0,1)$ noise. The Universal method wins HeaviSine with $E(0,1)$ noise.

Finally, the differences can partially be explained by the two different categories of the underlying functions, Blocks and Bumps versus HeaviSine and Doppler:
Blocks and Bumps have many features in the high resolution levels which favors the small threshold values. On the other hand, HeaviSine and Doppler have less or no features in the

high resolution levels which results in the advantage of the large threshold values.

In summary, the DJ-methods lose power for the Exponential noise study, because their methods are all based on the Gaussian noise. Without the normality assumption, Stein's unbiased estimator, minimax estimation or minimax Bayes estimation at least need to be justified or modified if they are still to work for non-Gaussian noise.

	n	*Universal*	*Minimax*	*Adaptive*	*New 1*	*New 2*	*New 3*
E(0,1)	*256*	*0.986(0.112)*	***0.719(0.076)***	*0.725(0.085)*	*0.746(0.081)*	*0.962(0.111)*	*1.191(0.137)*
Blocks	*512*	*0.776(0.073)*	***0.602(0.056)***	*0.682(0.063)*	*0.606(0.058)*	*0.767(0.072)*	*0.950(0.087)*
Haar	*1024*	*0.607(0.036)*	*0.514(0.035)*	*0.658(0.059)*	***0.500(0.032)***	*0.605(0.034)*	*0.718(0.035)*
	2048	*0.465(0.027)*	*0.413(0.029)*	*0.628(0.052)*	***0.412(0.026)***	*0.478(0.027)*	*0.567(0.028)*
	4096	*0.412(0.016)*	*0.397(0.018)*	*0.621(0.035)*	***0.392(0.015)***	*0.458(0.015)*	*0.531(0.015)*
N(0,1)	*256*	*1.268(0.124)*	*0.806(0.078)*	***0.643(0.056)***	*0.888(0.089)*	*1.236(0.119)*	*1.536(0.128)*
Blocks	*512*	*0.979(0.072)*	*0.653(0.049)*	***0.552(0.037)***	*0.706(0.053)*	*0.966(0.069)*	*1.208(0.078)*
Haar	*1024*	*0.750(0.041)*	*0.527(0.030)*	***0.494(0.027)***	*0.565(0.032)*	*0.733(0.034)*	*0.865(0.034)*
	2048	*0.559(0.022)*	***0.417(0.017)***	*0.434(0.017)*	*0.453(0.018)*	*0.584(0.022)*	*0.698(0.022)*
	4096	*0.484(0.022)*	***0.379(0.014)***	*0.406(0.015)*	*0.436(0.014)*	*0.544(0.013)*	*0.629(0.012)*
E(0,1)	*256*	*1.850(0.167)*	*1.234(0.109)*	***0.856(0.072)***	*1.422(0.118)*	*1.830(0.135)*	*2.131(0.149)*
Bumps	*512*	*1.384(0.100)*	*0.964(0.065)*	***0.764(0.059)***	*1.031(0.069)*	*1.345(0.087)*	*1.595(0.092)*
S8	*1024*	*1.037(0.057)*	*0.763(0.041)*	***0.684(0.046)***	*0.789(0.040)*	*0.982(0.047)*	*1.152(0.053)*
	2048	*0.736(0.028)*	*0.583(0.023)*	*0.557(0.044)*	***0.549(0.020)***	*0.661(0.022)*	*0.767(0.024)*
	4096	*0.550(0.015)*	*0.477(0.016)*	*0.490(0.046)*	***0.433(0.014)***	*0.485(0.013)*	*0.541(0.014)*
N(0,1)	*256*	*2.089(0.145)*	*1.382(0.107)*	***0.889(0.074)***	*1.583(0.107)*	*2.022(0.112)*	*2.352(0.127)*
Bumps	*512*	*1.600(0.106)*	*1.076(0.070)*	***0.735(0.035)***	*1.164(0.071)*	*1.520(0.081)*	*1.788(0.087)*
S8	*1024*	*1.266(0.050)*	*0.862(0.038)*	***0.721(0.028)***	*0.895(0.032)*	*1.129(0.038)*	*1.321(0.042)*
	2048	*0.857(0.027)*	*0.625(0.022)*	***0.509(0.016)***	*0.601(0.019)*	*0.748(0.021)*	*0.869(0.021)*
	4096	*0.618(0.014)*	*0.486(0.013)*	***0.440(0.010)***	*0.453(0.011)*	*0.531(0.012)*	*0.604(0.013)*
E(0,1)	*256*	*0.489(0.041)*	*0.494(0.064)*	*0.495(0.087)*	*0.460(0.057)*	***0.441(0.043)***	*0.448(0.038)*
Heavisine	*512*	*0.420(0.034)*	*0.441(0.050)*	*0.425(0.056)*	*0.399(0.045)*	***0.375(0.035)***	*0.381(0.033)*
S8	*1024*	*0.357(0.032)*	*0.400(0.043)*	*0.366(0.052)*	*0.361(0.035)*	*0.329(0.028)*	***0.326(0.026)***
	2048	*0.311(0.019)*	*0.355(0.026)*	*0.315(0.035)*	*0.318(0.022)*	*0.295(0.018)*	***0.293(0.017)***
	4096	*0.284(0.015)*	*0.329(0.019)*	*0.291(0.041)*	*0.292(0.014)*	*0.270(0.012)*	***0.268(0.012)***
N(0,1)	*256*	*0.503(0.037)*	*0.437(0.045)*	*0.471(0.038)*	***0.424(0.044)***	*0.439(0.040)*	*0.457(0.037)*
Heavisine	*512*	*0.435(0.025)*	*0.382(0.028)*	*0.408(0.028)*	***0.365(0.027)***	*0.374(0.028)*	*0.394(0.028)*
S8	*1024*	*0.359(0.020)*	*0.325(0.020)*	*0.341(0.021)*	*0.321(0.020)*	***0.320(0.020)***	*0.330(0.019)*
	2048	*0.308(0.016)*	*0.295(0.016)*	*0.301(0.016)*	*0.295(0.016)*	***0.293(0.016)***	*0.298(0.016)*
	4096	*0.279(0.011)*	*0.272(0.012)*	*0.275(0.011)*	*0.274(0.011)*	***0.270(0.011)***	*0.273(0.011)*
E(0,1)	*256*	*1.127(0.099)*	*0.812(0.070)*	***0.743(0.087)***	*0.879(0.077)*	*1.116(0.093)*	*1.303(0.089)*
Doppler	*512*	*0.833(0.051)*	***0.648(0.044)***	*0.674(0.052)*	*0.681(0.039)*	*0.799(0.036)*	*0.878(0.034)*
S8	*1024*	*0.644(0.031)*	*0.543(0.032)*	*0.532(0.039)*	***0.510(0.028)***	*0.580(0.027)*	*0.655(0.026)*
	2048	*0.486(0.019)*	*0.449(0.020)*	*0.435(0.032)*	***0.399(0.017)***	*0.421(0.016)*	*0.454(0.017)*
	4096	*0.371(0.014)*	*0.375(0.018)*	*0.367(0.040)*	***0.324(0.014)***	*0.325(0.012)*	*0.345(0.012)*
N(0,1)	*256*	*1.292(0.106)*	*0.875(0.072)*	***0.822(0.123)***	*0.975(0.076)*	*1.254(0.085)*	*1.429(0.071)*
Doppler	*512*	*0.951(0.060)*	***0.690(0.047)***	*0.712(0.059)*	*0.736(0.040)*	*0.860(0.037)*	*0.942(0.040)*
S8	*1024*	*0.718(0.029)*	*0.546(0.024)*	*0.551(0.034)*	***0.532(0.023)***	*0.636(0.023)*	*0.712(0.019)*
	2048	*0.523(0.016)*	*0.425(0.016)*	*0.403(0.027)*	***0.396(0.015)***	*0.443(0.014)*	*0.485(0.015)*
	4096	*0.387(0.012)*	*0.338(0.012)*	*0.336(0.011)*	***0.313(0.011)***	*0.339(0.012)*	*0.366(0.011)*

Table 1. Average root mean square errors of estimation with SNR=7

5.2 Conclusion

For nonparametric regression with non-normal noise, the problem becomes very complicated, because within each resolution level we lose the independent property and among the different levels, we lose the identical distribution property. The wavelet basis itself is involved

in the selection of the threshold value. We have to adjust the threshold values according to the tail behavior of the empirical wavelet coefficients at each resolution level. We showed the selection rule of the optimal threshold value through the shifted Exponetial distribution and proved that for non-Gaussian noise regression, the wavelet shrinkage procedure enjoys the adaptive property for various smoothness classes and the risk is within a log factor to the ideal risk. In words, the wavelet shrinkage procedure can be applied to non-Gaussian noise with certain adjustment of the threshold value. For a general discussion of estimating nonparametric regression with i.i.d. non-Gaussian noise through wavelet thresholding method, see Wu(1997b).

	n	*Universal*	*Minimax*	*Adaptive*	*New 1*	*New 2*	*New 3*
E(0,1)	*256*	*0.995(0.121)*	***0.726(0.083)***	*0.735(0.096)*	*0.755(0.091)*	*0.968(0.114)*	*1.164(0.121)*
Blocks	*512*	*0.761(0.054)*	***0.607(0.053)***	*0.695(0.084)*	*0.609(0.050)*	*0.755(0.054)*	*0.911(0.060)*
Haar	*1024*	*0.590(0.033)*	*0.509(0.033)*	*0.628(0.086)*	***0.491(0.030)***	*0.571(0.029)*	*0.653(0.030)*
	2048	*0.461(0.024)*	*0.431(0.026)*	*0.6221(0.053)*	***0.412(0.023)***	*0.470(0.022)*	*0.539(0.021)*
	4096	*0.402(0.014)*	*0.395(0.018)*	*0.625(0.037)*	***0.386(0.013)***	*0.433(0.013)*	*0.483(0.013)*
N(0,1)	*256*	*1.244(0.103)*	*0.813(0.073)*	***0.730(0.114)***	*0.893(0.081)*	*1.182(0.088)*	*1.387(0.076)*
Blocks	*512*	*0.944(0.068)*	*0.640(0.047)*	***0.615(0.054)***	*0.696(0.050)*	*0.926(0.063)*	*1.113(0.060)*
Haar	*1024*	*0.722(0.036)*	*0.515(0.030)*	***0.482(0.028)***	*0.539(0.027)*	*0.665(0.029)*	*0.767(0.028)*
	2048	*0.548(0.024)*	***0.415(0.021)***	*0.430(0.020)*	*0.447(0.021)*	*0.551(0.020)*	*0.615(0.016)*
	4096	*0.462(0.012)*	***0.371(0.012)***	*0.413(0.011)*	*0.416(0.011)*	*0.491(0.010)*	*0.533(0.009)*
E(0,1)	*256*	*1.626(0.147)*	*1.119(0.105)*	***0.826(0.079)***	*1.249(0.102)*	*1.557(0.119)*	*1.795(0.133)*
Bumps	*512*	*1.268(0.093)*	*0.908(0.065)*	***0.761(0.061)***	*0.950(0.062)*	*1.194(0.072)*	*1.386(0.080)*
S8	*1024*	*0.981(0.048)*	*0.737(0.036)*	***0.678(0.042)***	*0.733(0.032)*	*0.894(0.037)*	*1.032(0.041)*
	2048	*0.692(0.028)*	*0.558(0.024)*	*0.528(0.036)*	***0.514(0.021)***	*0.603(0.023)*	*0.688(0.025)*
	4096	*0.507(0.015)*	*0.452(0.015)*	*0.453(0.040)*	***0.403(0.013)***	*0.441(0.013)*	*0.490(0.013)*
N(0,1)	*256*	*1.829(0.120)*	*1.253(0.093)*	***0.868(0.066)***	*1.380(0.087)*	*1.715(0.100)*	*1.970(0.106)*
Bumps	*512*	*1.469(0.071)*	*1.008(0.051)*	***0.783(0.093)***	*1.061(0.047)*	*1.341(0.053)*	*1.558(0.061)*
S8	*1024*	*1.117(0.048)*	*0.794(0.034)*	***0.654(0.022)***	*0.795(0.029)*	*0.988(0.037)*	*1.141(0.038)*
	2048	*0.793(0.028)*	*0.589(0.023)*	***0.519(0.029)***	*0.553(0.019)*	*0.671(0.021)*	*0.775(0.025)*
	4096	*0.563(0.014)*	*0.451(0.013)*	***0.404(0.010)***	*0.414(0.011)*	*0.480(0.012)*	*0.539(0.011)*
E(0,1)	*256*	*0.417(0.058)*	*0.449(0.084)*	*0.426(0.099)*	*0.420(0.079)*	*0.390(0.063)*	***0.385(0.053)***
Heavisine	*512*	*0.373(0.035)*	*0.423(0.054)*	*0.378(0.046)*	*0.378(0.048)*	*0.345(0.040)*	***0.343(0.036)***
S8	*1024*	*0.329(0.027)*	*0.377(0.039)*	*0.331(0.032)*	*0.339(0.031)*	*0.305(0.024)*	***0.300(0.022)***
	2048	*0.298(0.019)*	*0.351(0.026)*	*0.302(0.029)*	*0.313(0.021)*	*0.285(0.017)*	***0.280(0.016)***
	4096	*0.277(0.012)*	*0.326(0.016)*	*0.288(0.048)*	*0.290(0.014)*	*0.267(0.011)*	***0.264(0.011)***
N(0,1)	*256*	*0.418(0.031)*	*0.390(0.040)*	*0.400(0.037)*	*0.384(0.038)*	***0.377(0.037)***	*0.384(0.036)*
Heavisine	*512*	*0.370(0.023)*	*0.354(0.028)*	*0.360(0.026)*	*0.347(0.029)*	***0.344(0.028)***	*0.353(0.026)*
S8	*1024*	*0.322(0.019)*	*0.309(0.021)*	*0.312(0.020)*	*0.303(0.021)*	***0.298(0.021)***	*0.303(0.021)*
	2048	*0.284(0.015)*	*0.281(0.015)*	*0.281(0.015)*	*0.282(0.016)*	***0.276(0.016)***	*0.278(0.015)*
	4096	*0.267(0.011)*	*0.266(0.011)*	*0.265(0.011)*	*0.268(0.011)*	***0.263(0.011)***	*0.264(0.011)*
E(0,1)	*256*	*1.058(0.103)*	***0.779(0.075)***	*0.792(0.099)*	*0.829(0.073)*	*0.998(0.066)*	*1.105(0.061)*
Doppler	*512*	*0.741(0.046)*	***0.604(0.045)***	*0.631(0.057)*	*0.607(0.039)*	*0.668(0.032)*	*0.725(0.034)*
S8	*1024*	*0.585(0.026)*	*0.517(0.029)*	*0.522(0.044)*	***0.476(0.026)***	*0.519(0.022)*	*0.560(0.020)*
	2048	*0.442(0.019)*	*0.429(0.024)*	*0.411(0.038)*	***0.377(0.020)***	*0.382(0.017)*	*0.404(0.017)*
	4096	*0.347(0.014)*	*0.365(0.017)*	*0.344(0.032)*	*0.317(0.015)*	***0.311(0.013)***	*0.323(0.013)*
N(0,1)	*256*	*1.179(0.086)*	***0.812(0.059)***	*0.861(0.093)*	*0.885(0.057)*	*1.062(0.045)*	*1.166(0.046)*
Doppler	*512*	*0.833(0.042)*	***0.624(0.035)***	*0.643(0.026)*	*0.632(0.026)*	*0.713(0.027)*	*0.785(0.030)*
S8	*1024*	*0.641(0.029)*	*0.506(0.026)*	*0.528(0.032)*	*0.485(0.023)*	*0.551(0.022)*	*0.596(0.021)*
	2048	*0.472(0.014)*	*0.400(0.014)*	*0.403(0.013)*	***0.369(0.014)***	*0.401(0.013)*	*0.433(0.013)*
	4096	*0.357(0.010)*	*0.322(0.011)*	*0.332(0.018)*	***0.304(0.011)***	*0.321(0.011)*	*0.337(0.010)*

Table 2. Average root mean square errors of estimation with SNR=5

6. APPENDIX

Proof of theorem 1:

It is easy to see that

$$\begin{aligned} Z_{j,k} &= 2^{\frac{j-J}{2}}[(\epsilon_{k2^{J-j}+1} + \epsilon_{k2^{J-j}+2} + \ldots + \epsilon_{(2k+1)2^{J-j-1}}) \\ &\quad -(\epsilon_{(2k+1)2^{J-j-1}+1} + \epsilon_{(2k+1)2^{J-j-1}+2} + \ldots + \epsilon_{(k+1)2^{J-j}})], \end{aligned}$$

are identical and independently distributed for a given j because of the i.i.d. assumption on $(\epsilon_i)_{i=1}^n$. Now we want to compute the distribution of the coefficient noise at the finest level. For the Haar wavelet transformation,

$$Z_{J-1,k} = \frac{1}{\sqrt{2}}(\epsilon_{2k+1} - \epsilon_{2k+2}) \quad , k = 0\ldots, 2^{J-1} - 1.$$

Obviously they are i.i.d. and have the same density function. Let

$$\begin{aligned} Z_1 &= \frac{1}{\sqrt{2}}(\epsilon_1 - \epsilon_2) \\ Z_2 &= \epsilon_2 \end{aligned}$$

then the joint density function of $Z = (Z_1, Z_2)$ can be derived from the exponential distribution of ϵ_1 and ϵ_2. After integrating with respect to Z_2, we get our desired density function.

$$\begin{aligned} f_{Z_1}(z_1) &= \frac{\sqrt{2}}{\sigma^2}\int e^{-(\frac{\sqrt{2}z_1+2z_2}{\sigma}+2)} 1_{\sqrt{2}z_1+z_2+\sigma>0} 1_{z_2+\sigma>0}\, dz_2 \\ &= \frac{1}{\sqrt{2}\sigma}(e^{-\frac{\sqrt{2}z_1}{\sigma}} 1_{z_1>0} + e^{\frac{\sqrt{2}z_1}{\sigma}} 1_{z_1<0}) \end{aligned}$$

We can easily get the cumulative distribution function of $Z_{J-1,k}$, for $k = 0\ldots, 2^{J-1} - 1$:

$$F(z) = \frac{e^{\frac{\sqrt{2}z}{\sigma}}}{2} 1_{z<0} + (1 - \frac{e^{-\frac{\sqrt{2}z}{\sigma}}}{2}) 1_{z>0}.$$

The following lemma says that $\frac{\sqrt{2}}{2}\sigma log(n_1)$ is the smallest level that all the noise in the finest resolution level cannot cross over. Therefore, following the lemma, the proof of theorem 2 becomes obvious.

Lemma 1. If $n_1 = \frac{n}{2} \to \infty$, then

$$Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} > \frac{\sqrt{2}}{2}\sigma(log(n_1)+log(log(n_1))-log(2))) \to 0.$$

and

$$Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} > \tfrac{\sqrt{2}}{2}\sigma(log(n_1)-log(log(n_1))-log(2))) \to 1.$$

Proof:

$$Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} \le \frac{\sqrt{2}}{2}\sigma(log(n_1)+log(log(n_1))-log(2)))$$

$$= \left[F(\frac{\sqrt{2}}{2}\sigma(log(n_1)+log(log(n_1))-log(2))\right]^{n_1}$$

$$= \left[1-\frac{1}{2}e^{-\frac{\sqrt{2}}{\sigma}\frac{\sigma}{\sqrt{2}}(log(n_1)+log(log(n_1))-log(2))}\right]^{n_1}$$

$$= \left(1-\frac{1}{n_1 log(n_1)}\right)^{n_1} \sim e^{-\frac{1}{log(n_1)}} \to 1.$$

So

$$Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} > \frac{\sqrt{2}}{2}\sigma(log(n_1)+log(log(n_1))-log(2)))$$

$$= 1 - Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} \le \frac{\sqrt{2}}{2}\sigma(log(n_1)+log(log(n_1))-log(2)))$$

$$\to 0$$

Similarly,

$$Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} \le \frac{\sqrt{2}}{2}\sigma(log(n_1)-log(log(n_1))-log(2)))$$

$$= \left[F(\frac{\sqrt{2}}{2}\sigma(log(n_1)-log(log(n_1))-log(2))\right]^{n_1}$$

$$= \left[1-\frac{1}{2}e^{-\frac{\sqrt{2}}{\sigma}\frac{\sigma}{\sqrt{2}}(log(n_1)-log(log(n_1))-log(2))}\right]^{n_1}$$

$$= \left(1-\frac{log(n_1)}{n_1}\right)^{n_1} \sim e^{-log(n_1)} \to \frac{1}{n_1} \to 0$$

Hence,

$$Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} > \frac{\sqrt{2}}{2}\sigma(log(n_1) - log(log(n_1)) - log(2)))$$

$$= 1 - Pr(\max_{0\le k\le n_1-1}\{Z_{J-1,k}\} \le \frac{\sqrt{2}}{2}\sigma(log(n_1) - log(log(n_1)) - log(2)))$$

$$\to 1$$

Proof of theorem 3:

It is enough to consider the univariate case. Let $w = \theta + \delta Z$, where $f_z(x) = \frac{e^{-\sqrt{2}x}}{\sqrt{2}}1_{x\ge 0} + \frac{e^{\sqrt{2}x}}{\sqrt{2}}1_{x<0}$. The thresholding estimator of θ is $\hat{\theta} = sgn(w)(|w| - \lambda)_+$. We want to compute its L_2 loss:

$$\begin{aligned}
E(\hat{\theta} - \theta)^2 &= E((w - \lambda - \theta)^2 1_{w>\lambda} + (w + \lambda - \theta)^2 1_{w<-\lambda} + \theta^2 1_{w^2\le\lambda^2}) \\
&= \int_{w>\lambda}(w - \lambda - \theta)^2 \frac{1}{\sqrt{2}\delta}[e^{\sqrt{2}\frac{w-\theta}{\delta}}1_{w<\theta} + e^{-\sqrt{2}\frac{w-\theta}{\delta}}1_{w>\theta}]\,dw \\
&\quad + \int_{w<-\lambda}(w + \lambda - \theta)^2 \frac{1}{\sqrt{2}\delta}[e^{\sqrt{2}\frac{w-\theta}{\delta}}1_{w<\theta} + e^{-\sqrt{2}\frac{w-\theta}{\delta}}1_{w>\theta}]\,dw + \theta^2 E 1_{w^2\le\lambda^2} \\
&= (\theta^2 + (\frac{\delta^2}{2} - \frac{\sqrt{2}}{2}\theta\delta)e^{-\frac{\sqrt{2}}{\delta}(\lambda-\theta)} + (\frac{\delta^2}{2} + \frac{\sqrt{2}}{2}\theta\delta)e^{-\frac{\sqrt{2}}{\delta}(\lambda+\theta)})1_{|\theta|\le\lambda} \\
&+ \left[(\lambda^2 + \delta^2) - (\frac{\delta^2}{2} + \frac{\sqrt{2}}{2}\theta\delta)(e^{\frac{\sqrt{2}}{\delta}(\lambda-\theta)} - e^{-\frac{\sqrt{2}}{\delta}(\lambda+\theta)})\right]1_{\theta>\lambda} \\
&+ \left[(\lambda^2 + \delta^2) - (\frac{\delta^2}{2} - \frac{\sqrt{2}}{2}\theta\delta)(e^{\frac{\sqrt{2}}{\delta}(\lambda+\theta)} - e^{-\frac{\sqrt{2}}{\delta}(\lambda-\theta)})\right]1_{\theta<-\lambda} \\
&= \theta^2 1_{\theta^2\le\lambda^2} + \lambda^2 1_{\theta^2>\lambda^2} + \delta^2 + ((\frac{\delta^2}{2} - \frac{\sqrt{2}}{2}\theta\delta)e^{-\frac{\sqrt{2}}{\delta}(\lambda-\theta)} \\
&\quad + (\frac{\delta^2}{2} + \frac{\sqrt{2}}{2}\theta\delta)e^{-\frac{\sqrt{2}}{\delta}(\lambda+\theta)} - \delta^2)1_{\theta^2\le\lambda^2} - (\frac{\delta^2}{2} + \frac{\sqrt{2}}{2}\theta\delta)\left(e^{\frac{\sqrt{2}}{\delta}(\lambda-\theta)} - e^{-\frac{\sqrt{2}}{\delta}(\lambda+\theta)}\right)1_{\theta>\lambda} \\
&\quad - (\frac{\delta^2}{2} - \frac{\sqrt{2}}{2}\theta\delta)\left(e^{\frac{\sqrt{2}}{\delta}(\lambda+\theta)} - e^{-\frac{\sqrt{2}}{\delta}(\lambda-\theta)}\right)1_{\theta<-\lambda} \\
&\le \min(\theta^2, \lambda^2) + \delta^2.
\end{aligned}$$

Therefore,

$$\sum_{i=1}^{n_1}(\hat{\theta}_i - \theta_i)^2 \le \sum_{i=1}^{n_1}\left[\min(\theta^2, \lambda^2) + \delta^2\right]$$

By taking $\lambda = \frac{log(n_1)}{\sqrt{2}}(1+o(1))$ and $\delta = \frac{\sigma}{\sqrt{n}}$, we finish the proof of theorem 3.

Before we start to prove Corollary 1, first we introduce a lemma. Its proof is straightforward according to the property that the sum of i.i.d. exponential rv's has a gamma distribution.

Lemma 2. If $(X_i)_{i=1}^k$ are i.i.d. and exponentially distributed with mean 0 and variance 1, then $X = \sum_{i=1}^k X_i$ has the density function

$$f_X(x) = \frac{(\frac{x}{\sigma}+k)^{k-1}}{(k-1)!}e^{-(\frac{x}{\sigma}+k)}1_{\frac{x}{\sigma}+k>0} \tag{4}$$

Outline of the proof of Corollary 1

By using the result of Lemma 2, we can easily prove that the $Z_{j,k}$ have the following distribution:

$$F_{Z_{j,k}}(x) = 1 - \frac{1}{2}\sum_{m=0}^{L-1}\binom{L+m-1}{m}(\frac{1}{2})^{L+m-1}\int_{\sqrt{2L}x}^{\infty}\frac{y^{L-m-1}e^{-y}}{(L-m-1)!}dy,$$

where $L = 2^{J-1-j}$, $k = 0, 1, 2, \ldots, 2^j - 1$ and $j = J-2, J-3, \ldots$. Therefore, as $n \to \infty$, it is easy to show that

$$pr\left(\max_k Z_{j,k} > \frac{log(n) + Llog(log(n))}{\sqrt{2L}}\right) \to 0$$

and

$$pr\left(\max_k Z_{j,k} > \frac{log(n) - log(log(n))}{\sqrt{2L}}\right) \to 1$$

Similarly, we can finish the proof of theorem 4 through the following lemma:

Lemma 3. At the finest resolution level, $Z_{J-1,k}$ have the following density function :

$$\begin{aligned} f_{Z_{J-1,k}}(x) &= \sum_{j=1}^{S_1} c_j^1 \sum_{l=1}^{S_2} c_l^2 \frac{\sqrt{2}}{a_j + b_l} \\ &\quad (exp\{-\frac{\sqrt{2}x + \sum_{i=1}^{S_1} a_i - \sum_{p=1}^{S_2} b_p}{a_j}\}1_{\sqrt{2}x+\sum_{i=1}^{S_1} a_i - \sum_{p=1}^{S_2} b_p > 0} \\ &+ exp\{\frac{\sqrt{2}x + \sum_{i=1}^{S_1} a_i - \sum_{p=1}^{S_2} b_p}{b_l}\}1_{\sqrt{2}x+\sum_{i=1}^{S_1} a_i - \sum_{p=1}^{S_2} b_p < 0}), \end{aligned}$$

where a_j are positive components and b_l are absolute value of negative components of the underlied mother wavelet, and

$$c_j^1 = \frac{a_j^{S_1-1}}{\prod_{i=1}^{S_1}(a_j - a_i)} \text{ where } i \neq j$$

$$c_2^1 = \frac{b_l^{S_2-1}}{\prod_{p=1}^{S_2}(b_l - b_p)} \text{ where } p \neq l$$

and we assume that $a_j \neq a_i$ for $i \neq j$, $1 \leq i, j \leq S_1$ and $b_l \neq b_p$ for $p \neq l, 1 \leq p, l \leq S_2$.

Remark: When there exists one pair or more pairs of a_j, a_i or b_l, b_p such that $a_j = a_i$ or $b_l = b_p$, the distribution of $Z_{J-1,k}$ will be slight different, but it will not affect our final results.

We omit the tedious proof (Wu 1997a). The key fact is that for any positive constants $w_1, w_2, \ldots, w_k$ with the assumption that $w_j \neq w_i$ for $i \neq j$, where $1 \leq i, j \leq S_1$, the weighted sum of exponential random variables $y_i \sim E(0,1)$, $Z = \sum_{j=1}^{k} w_j y_j$ has the following distribution function:

$$F_Z(z) = \sum_{j=1}^{k} \frac{w_j^{k-2}}{\prod_{i \neq j}(w_j - w_i)} exp\{-\frac{z + \sum_{i=1}^{k} w_i}{w_j}\} 1_{z+\sum_{i=1}^{k} w_i > 0}$$

The proof can also be found in Wu(1997b).

[Received February 28, 1998. .]

REFERENCES

Felix Abramovich and Yoav Benjamini (1996) Adaptive thesholding of wavelet coefficients. *Computational Statistics & Data Analysis***22** 351-361.

Cai, T. (1996) Minimax Wavelet Estimation via Block Thresholding, Technical Report 96-41, Department of Statistics, Purdue University.

Cohen, A., Daubechies, I., Jawerth, B., and Vial, P. (1993). Multiresolution analysis, wavelets, and fast algorithms on an interval. *Comptes Rendus Acad. Sci. Paris A*,**316**, 417-421.

Daubechies, I. (1991) Ten Lectures on Wavelets SIAM: Philadelphia.

Donoho, D.L. (1995) De-Noising by Soft Thresholding, *IEEE Trans. Info. Thry.*, **41**, pp. 613-627.

Donoho,D.L. (1993) *Unconditional bases are optimal bases for data compression and for statistical estimation*, Applied and Computational Harmonic Analysis, 1, 100-115.

Donoho, D.L. and Johnstone, I.M.(1994) "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, **81** 425-455.

Donoho, D. L. and Johnstone, I. M (1995) Adapting to unknown smoothness via Wavelet shrinkage. *J. Amer. Statist. Assoc.*,**432** 1200-1224.

Gao, H. (1993) Choice of thresholds for wavelet estimation of the log spectrum. Technical Report, Department of Statistics, Stanford University.

Johnstone, I.M. and Silverman, B.W. (1994) Wavelet threshold estimators for data with correlatd noise. Technical Report, Department of Statistics, Stanford University.

Leadbetter, M. R., Lindgren, G., Rootzen, Holger (1983) *Extremes and Related Properties of Random Sequences and Processes.* New York: Springer-Verlag.

Nason, G.P. (1996) "Wavelet Shrinkage using Crossing-validation", *J. R. Statist. Soc.* Ser **B, 58**, 463-479.

Neumann, M.H. and Sachs R.V. (1995) "Wavelet Thresholding: Beyond the Gaussian I.I.D. Situation", *Lect. Notes Statist.*, **103**, 301-329.

Ogden, T. and Parzen, E. (1996) Data dependent wavelet thresholding in nonparametric regression with change-point applications. *Computational Statistics & Data Analysis***22** 53-70.

Wang, Y.(1996) Function estimation via wavelet shrinkage for long memory data. *Ann. Statist.* **24** 466-484.

Wu, Y.(1997a) A Case Study of Wavelet Estimation for Non-Gaussian Error in Nonparametric Regression IMS Annual Meeting in Park City, Utah, 1997, Poster session.

Wu, Y.(1997b) Wavelet Estimation for Nonparametric Regression: Beyond Gaussian Noise II. In progress.