Speech Detection and Noise Compression Based on Singularity

Haitao Luo

School of Informatics, Guangdong University of Foreign Studies(GDUFS)

Guangzhou, China, 510420

Email: htluo@gdufs.edu.cn

Keywords: noise compression; wavelet packet; singularity; speech detect; algorithm.

Abstract. Frequencies of traffic noise are near to or even the same as speech signal. It is hard to filter noise from speech. Wavelet packet decomposition coefficient reveals singularity of signal. Constructed a wavelet according to Daubechies' method, and derived a wavelet packet from the constructed scaling and wavelet functions. Analyzed singularity measurement of speech signal and noise. Decomposed the noisy speech signal by wavelet packet. Developed algorithm to detect starting and ending point of speech; Developed algorithm to compress noise by wavelet packet. Reconstructed the wavelet packet tree. Re-built audio file using reconstructed data and the result is acceptable.

Introduction

Usually, the process of Chinese speech recognition is as follows: First, extract identity parameters from speech signal; Second, create a model for each Chinese character according to these parameters; Third, when process real speech recognition, extract parameters from speech signal and compare them to model, search for the nearest model as the recognition result. The speech signal using for model can be produced in laboratory, in quiet environment without noise. However, sometimes the speech signal for recognition is recorded in noisy environment, where resulted signal contains noise. The noisy speech signal is difficult to recognize. It is even impossible to detect starting and ending point of speech. So, it will be helpful to compress noise before processing it.

Antoniadis[1] contributed to the methodology available for estimating smooth regression functions from noisy data. He gave the condition for an estimator to attain optimal convergence rates in the integrated mean square sense as the sample size increase to infinity. Donoho and Johnstone[2] showed that a certain method for empirically selecting a basis in which to adaptively denoise attains near-ideal performance, in a precise sense. The result extended to more general orthogonal basis libraries, basically to any libraries constructed from an at-most polynomially-growing number of coefficient functionals. Donoho[3] proposed a formal interpretation of the term "denoising" and show how wavelet transforms may be used to optimally "de-noise" in the interpretation. Moreover, he showed that the "denoising" property signals success in a range of situations where many previous nonwavelets methods have met only partial success. While they focus theoretically on de-noising, He[4] processed speech signal by wavelet. He extracted pitched periods and separated voiced/unvoiced section of speech signal.

I have some sample speech signals mixed with traffic noise. The noise frequency is close or even identical to speech. While it is impossible to filter out noise from signal, it is feasible to detect terminal points, starting and ending points, of speech and compress noise based on function singularity.

In my previous paper[5], I applied three different methods to de-noise the signal. These methods are soft thresholding, hard thresholding, and polynomial function thresholding. For the threshold selection rule, I used principle of Stein's Unbiased Risk Estimate (SURE) rule. Since each terminal node coefficients has different amplitude of data, I applied locally thresholding strategy. That is, for each terminal node other than approximation node, within the time span corresponding to speech, I calculated the threshold value according to the SURE rules. However, all these methods haven't considered singularity of signal. Therefore, the compressed and reconstructed signal may lose some

speech information, and the rebuild wave audio files are a little infidelity. When process speech signal mixed with noise, singularity measurement reveals unique different feature between speech and noise. Based on this, we can process speech and noise in different ways.

Construct wavelet packet

When function *f*(t) or its any order of differential coefficients discontinues at a point, we know that function is singular at that point. Wavelet and wavelet packet are effective way to analyze singularity of a function, and to locate the position of singular point. According to Daubechies' method, to construct a wavelet packet, a scaling filter h_k, also known as low reconstruction filter, is needed first. In frequency domain, the filter is as follows:

$$H(\omega)=\frac{1}{\sqrt{2}}\sum_{k\in z}h_k e^{-i\omega k} \tag{1}$$

Meanwhile, $H(\omega)=(\frac{1+e^{-i\omega}}{2})^N\cdot L(e^{-i\omega})$ (2)

Where, $\left|L(e^{-i\omega})\right|^2$ is an even function of ω. So it is a polynomial of cosω, or $\sin^2\frac{\omega}{2}$.

Suppose that $\left|L(e^{-i\omega})\right|^2=f(\sin^2\frac{\omega}{2})=f(y)$, where $y=\sin^2\frac{\omega}{2}$. Then we have:

$$|H(\omega)|^2=|(\frac{1+e^{-i\omega}}{2})|^{2N}\cdot|L(e^{-i\omega})|^2=(\cos^2\frac{\omega}{2})^N\cdot|L(e^{-i\omega})|^2=(1-y)^N\cdot f(y)$$

Actually, $L(e^{-i\omega})$ is a polynomial of the (N-1)th power of $e^{-i\omega}$, and according to Daubechies, following equation holds:

$$\left|L(e^{-i\omega})\right|^2=f(\sin^2\frac{\omega}{2})=f(y)=\sum_{k=0}^{N-1}\binom{N-1+k}{k}\sin^{2k}\frac{\omega}{2} \tag{3}$$

Now, let N equals to 2, and for simplifying let $z=e^{-i\omega}$, then $L(e^{-i\omega})=L(z)$. Since $L(z)$ is a polynomial of the (N-1)th power of z, I suppose that $L(z)=a_0+a_1z$, then I have:

$$|L(z)|^2=L(z)\cdot\overline{L(z)}=(a_0+a_1z)(a_0+a_1z^{-1})=a_0^2+a_1^2+a_0a_1(z+z^{-1})$$

$$=a_0^2+a_1^2+a_0a_1(2-4\sin^2\frac{\omega}{2})$$

While N=2, from (3), we also have: $|L(z)|^2=f(y)=1+2y=1+2\sin^2\frac{\omega}{2}$. The last two equations are identical, so are their corresponding coefficients, therefore I have

$$\begin{cases}(a_0+a_1)^2=1\\ 2a_0a_1=-1\end{cases}$$

Pick up one pair of solutions of above equations as: $a_0=\frac{1}{2}(1+\sqrt{3})$, and $a_1=\frac{1}{2}(1-\sqrt{3})$. According to equation (2), I have:

$$H(\omega)=(\frac{1+z}{2})^2\cdot L(z)=\frac{1}{4}[a_0+(2a_0+a_1)z+(a_0+2a_1)z^2+a_1z^3]$$

From (1): $H(\omega)=\frac{1}{\sqrt{2}}\sum_{n=0}^{3}h_nz^n=\frac{1}{\sqrt{2}}(h_0+h_1z+h_2z^2+h_3z^3)$

The above two equations are identical, so are their corresponding coefficients, therefore:

$h_0=\frac{1}{8}(\sqrt{2}+\sqrt{6})$, $h_1=\frac{1}{8}(3\sqrt{2}+\sqrt{6})$, $h_2=\frac{1}{8}(3\sqrt{2}-\sqrt{6})$, $h_3=\frac{1}{8}(\sqrt{2}-\sqrt{6})$

Then I have scaling function:

$\varphi(t)=\sqrt{2}[h_0\varphi(2t)+h_1\varphi(2t-1)+h_2\varphi(2t-2)+h_3\varphi(2t-3)]$, where $t\in[0,3]$.

According to this equation, enough discrete point function value can be calculated, and its figure can be drawn as figure1. The corresponding wavelet function is:

$$\psi(t)=\frac{\sqrt{3}-1}{4}\varphi(2t+2)+\frac{3-\sqrt{3}}{4}\varphi(2t+1)-\frac{3+\sqrt{3}}{4}\varphi(2t)+\frac{1+\sqrt{3}}{4}\varphi(2t-1)$$

Figure 2 shows this wavelet.

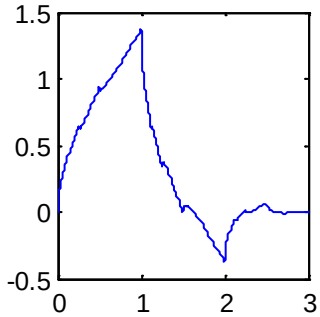

Fig. 1 Scaling function φ(t)

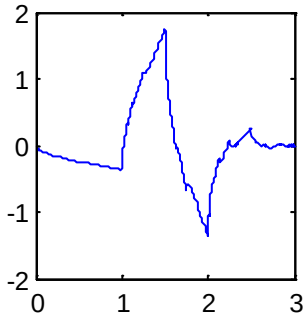

Fig. 2 Wavelet function ψ(t)

From above two functions, φ(t) and ψ(t), we can derive wavelet packet, whose bases are both orthonormal and compactly support.

Starting and ending points of speech detect

I recorded Chinese speech “0”, “1”,…… ”10”, ”100”, “1000” and “10000” into “.wav” audio files beside a busy road. The signals which are mixed with loud traffic noise are all sampled in 44100 Hz and normalized. Following figures show two of them. In each figure, only the middle one continuous period with high magnitude and high density is speech mixed with noise, other parts are pure noise.

Fig. 3 Chinese speech “0”

Fig. 4 Chinese speech “1”

I decompose signal to the 1st level using derived wavelet packet. Following two figures show leaf nodes' coefficients of wavelet packet tree. According to wavelet decomposition, figure 5 is low frequency part of original signal, and figure 6 is high frequency part. In both figures, only the middle one continuous period with high magnitude and high density corresponds to speech mixed with noise, other parts are simply noises. Some impulse points in figure 6 result from singular points in the original noisy signal. These singular points include starting and ending points of speech. However, after checking several speeches and their decomposition figure, I found that not every starting and ending point of speech appears impulse in figure of node 2.

Fig. 5 Node 1 of speech “0”

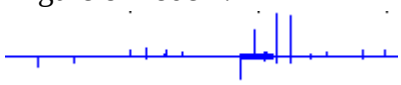

Fig. 6 Node 2 of speech “0”

Apparently, figure of node 2 is relatively “clean” for starting and ending point detection. I divide it into a series of overlapped frames, calculate the “energy” of each frame. Instead of magnitude square, frame energy is represented by the sum of absolute magnitude of points of each frame. Figure 7 shows frame energy of node 2. Figure 8 is consecutive frame energy differences of figure 7, each value of point is calculated by next frame energy minus current frame energy.

Fig. 7 Frame energy of node 2

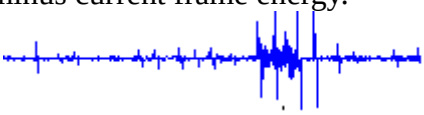

Fig. 8 Energy difference

In figure 7, protuberant period with high magnitude represents speech mixed with noise, and its starting and ending points with low enough magnitude represent speech starting and ending point respectively. In order to find out these two points, I set up a threshold. When the magnitude of figure 7 is lower than the threshold, they are the points. Notice that if figure 7 were continuous function or curve, theoretically, the point which has local minimum magnitude should have zero differential coefficient, which figure 8 will approximately reveal, because figure 8 is consecutive

point magnitude difference of figure 7. Therefore, we have the algorithm to decide starting point as follows.

a. In figure 7, set up a threshold,

b. Departs from the highest magnitude point, search leftward(backward in time domain) point by point

c. When both energy and energy difference are below the threshold, get the point number (here is the frame number in figure 7)

d. Pick up one point of the frame and map it to the original speech signal, get the starting point of speech.

In step d, I simply pick up the middle point of the frame. Any other points can also be considered. When I map the point to the original speech signal, I simply double the point number. The reason behind this is that when a signal is decomposed by wavelet to level 1, it is actually filtered by a low pass and a high pass filter, and then a point is got every two points. So the length of leaf node is nearly half of original signal, and their position numbers correspond to each other. The way to decide the ending point of speech is similar to above algorithm, except that search direction is rightward (forward in time domain) in figure 7.

Compress noise based on singularity

When I continue to decompose the original speech signal mixed with noise to deeper level, such as level 7, I got its terminal leaf nodes. Following figures show some terminal nodes coefficients of speech signal "0".

Fig. 9 node (7,0) or 127

Fig. 10 node(7,1) or 128

In the above figures, the period with high magnitude corresponds to speech mixed with noise, all other parts corresponds to pure noise. These figures also tell us that the frequencies of noise are close to speech, or even the same as speech. Therefore, it is almost impossible to filter noise from speech. Other ways are needed to compress noise.

According to Zhang[6], point singularity measurement of function *f*(t) can be represented by its Lipschitz coefficient α. Following three figures show three kinds of function *f*(t) and their α.

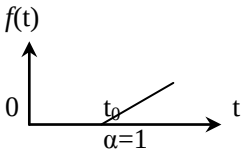

Fig. 11 Slope

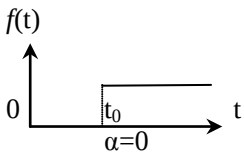

Fig. 12 Unit ladder

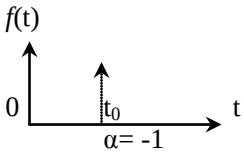

Fig. 13 Impulse

Above figures tell us that, the bigger the α, the more smoother the function *f*(t) at the point t_0 is, and the less singular the same point is. The smaller the α, the more singular the function *f*(t) at the point t_0 is. To our concerns, the function *f*(t) can be expressed as:

$$f(t) = x(t) + s(t) \tag{4}$$

Where *x*(t) is speech signal, and *s*(t) is noise. It is summed noise, and the noise is added to speech signal. When we sample *f*(t), we have

$$f(k) = x(k) + s(k) \tag{5}$$

Where k=0,1,2,…,n. n is length of $f(k)$. When function *f*(t) is decomposed, it is actually transformed by binary wavelet packet. Since wavelet transformation is linear, the transformed coefficients of $f(k)$ are the sum of coefficients of $x(k)$ and $s(k)$ for each k. The following equation stands

$$|Coeffs| \le c \times 2^{L \times \alpha} \tag{6}$$

Where, |*Coeffs*| are absolute value of leaf-node transformed coefficients of $f(k)$, it summates coefficients of $x(k)$ and $s(k)$. c is a positive constant, L is level number $f(t)$ is decomposed to by wavelet packet, and α is consistent Lipschitz coefficient at every point of the time span where $f(t)$ defines. To the normal speech signal without noise, α is positive. To the noise, α is negative. For example, Gauss white noise has negative α, which can be expressed as -0.5-ε, where ε>0, because Gauss white noise is almost singular at every point. Therefore, we conclude that, while we increase decomposition level, the maximum absolute node coefficients of speech signal should increase, and the maximum absolute node coefficients of noise should decrease.

To compress noise is to subtract noise from signal. It is difficult to determine how many amounts should be subtracted. It complicates in our concerns, because the signal mixes with noise. And it is impossible to discriminate signal from noise at each point of $f(k)$. While the signal is being decomposed to deeper level, |*Coeffs*| of $f(k)$ may increase or decrease, depending on which one of coefficients of $x(k)$ and $s(k)$ dominates. If the former dominates, |*Coeffs*| may increase, otherwise |*Coeffs*| may decrease. However, when |*Coeffs*| decreases, we should compress this value most likely. Following noise compression algorithm approximately reflects this conclusion.

a. setup the deepest level L to which signal $f(k)$ is decomposed by wavelet packet, L is an integer bigger than 1, setup a ratio r, r is a real number and $0<r<1$

b. input audio file $f(k)$, decompose it into level L, produce a full binary wavelet packet tree where each non-leaf node has two offspring (brothers)

c. in level L, calculate threshold as $r/2^L$; compare each absolute coefficient of every leaf node of the decomposed wavelet packet tree to threshold, if the former is less than the threshold, set it to zero, otherwise, remain it unchanged

d. reconstruct the tree to upper one level, and set $L=L-1$

e. if $L=0$, go to next step, otherwise go to step c

f. reconstruct wavelet packet tree to root, get the root coefficients, rebuild audio file using root coefficients

Results and evaluation

When above algorithm is applied to signal mixed with noise, the signal is compressed and noise is partly subtracted. In order to describe and evaluate the result, I need one definition as follow:

Definition: Point Compression Ratio (PCR), In the sense of fidelity or minimum infidelity when signal is being compressed, summate the absolute point magnitude difference of original signal and compressed signal, then divided by the number of points of signal.

PCR indicates the average amount of compression or subtraction in point wise, and therefore it is independent of length of speech signal. Following tables show some of results.

Table 1, compression result when L=4 and r=0.4

Chinese speech	0	1	2	3	4	5	6	7	8	9
PCR	2.18%	2.11%	2.62%	3.43%	3.79%	1.95%	2.31%	3.31%	2.90%	2.18%

Table 2, compression result when L=4 and r=0.45

Chinese speech	0	1	2	3	4	5	6	7	8	9
PCR	2.52%	2.50%	3.05%	3.92%	4.30%	2.22%	2.64%	3.84%	3.35%	2.56%

Table 3, compression result when L=5 and r=0.4

Chinese speech	0	1	2	3	4	5	6	7	8	9
PCR	2.19%	2.11%	2.62%	3.43%	3.80%	1.95%	2.30%	3.32%	2.92%	2.18%

All above resulted compressed signal are fidelity. The results are more sensitive to the ratio r than the level L to which $f(k)$ is decomposed. The reason is that when ratio r increases, the threshold increases also, and therefore more amounts of signal are compressed or subtracted, and PCRs are getting bigger. When r gets to 0.5~0.6, PCRs will be getting more big, and more amounts of signal

are subtracted, and the resulted signals begin to appear a little infidel. This indicates that some speech information begins to lose. Continue to increase *r*, speech signal infidelity increases. When *r* approaches 1.0, some of compressed speech signal become obscure completely.

When level *L* varies, some of PCRs vary a very small amount, others remain unchanged. The reason behind it is that singularity measurement of every point of the signal is fixed, and therefore won't be affected by decomposition level.

Conclusion

Different from traditional de-noising process where noise is in much higher frequency than signal, most of traffic noise energy concentrates in low frequency near to speech signal, or even in the same frequency as speech. It is difficult to filter noise from signal in this case. From figure 6, it is easy to detect starting and ending point of speech. The detection algorithm works well. After numbers of times of testing with numbers of various threshold, I found that when the threshold equals to 0.1, more than 93% of detection is nearly exact. The noise compressing algorithm also works well when I set the ratio r=0.4. All rebuilding audio files sound clear and clean.

References

[1] Antoniadis, A., Smoothing noisy data with coiflets[J], Statistica Sinica 4 (2), 651-678.

[2] Donoho, D.L.; I.M. Johnstone, Ideal de-noising in an orthonormal basis chosen from a library of bases[J], CRAS Paris, Ser I, t. 319, 1317-1322.

[3] Donoho, D.L., De-Noising by soft-thresholding[J], IEEE Trans. on Inf. Theory, vol. 41, 3, 613-627.

[4] He, Qichao; Long, Jianzhong; Zhou, Jiliu, Discrete Wavelet Transform(DWT) and Application in Speech Signal Processing[J], Journal of Sichuan University(Natural Science Edition), Vol. 32 No.3, 289-294.

[5] Luo, Haitao, Local thresholding de-noise speech signal[J], ICDIP 2013 Beijing, Proceedings of SPIE, Volume 8878 88780E-1

[6] Zhang, Guohua; Zhang, wenjuan; Xue, Pengxiang; Wavelet Analyses and Application Basis, Publishing company of Northwest University of China, 2006, 111-114.