A fast algorithm for sample entropy computation by neighborhood matrix iteration

Guiqi Sun, Xiaodong Zhuang

***Abstract*—**In order to reduce the computational complexity of traditional sample entropy, a fast sample entropy algorithm is proposed based on neighborhood matrix. The fast algorithm does the calculations in binary, and the neighborhood matrix of the high-dimensional signal vector is deduced from the neighborhood matrix of the lower dimensional signal vector. Compared with the traditional algorithm, the computation time has been significantly reduced by using the fast sample entropy algorithm.

***Keywords*—**Sample Entropy, neighborhood Matrix, fast Algorithm

I. Introduction

SAMPLE entropy is a measure of the complexity of time series, which was proposed by several nonlinear dynamics researchers at the end of last century [1-2]. The sample entropy is more consistent than the approximate entropy, and has been successfully applied in the analysis of the complexity of the signal sequence. The sample entropy has the following properties: (1) The sample entropy does not compare its own data segment [3-4]. It is the exact value of the negative mean natural logarithm of the conditional probability. So, the calculation of the sample entropy does not depend on the data length. (2) The sample entropy has a better consistency [5]. (3) The sample entropy is insensitive to lost data [6]. Even if the data is lost as much as 1/3, the effect on the sample entropy is still small.

At present, the faster algorithm for calculating the sample entropy is presented by using a skip-list [7-9] and k-d tree [10-12] data structure, which counts the number of the matched pairs of the input time series. The effectiveness of this approach depends on the efficient programming of k-d tree, so it can’t be widely used subject to the programming complexity. A fast sample entropy iterative algorithm based on neighborhood matrix is proposed because the traditional sample entropy cost is high and the computation time is long. In this paper, the traditional sample entropy and fast sample entropy principle are introduced. Finally, the experimental results show that the fast sample entropy algorithm is more efficient than the traditional method.

This work was supported in part by supported by the project of Shandong Province College Science and Technology Plan (project number: J15LN41), and the project of Natural Science Foundation of Shandong Province (project number: ZR2016FM11)..

Guiqi Sun is with the Electronics & Information College, Qingdao University, Qingdao 266071, China (e-mail: guiqisun@163.com).

XIaodong Zhuang is with the Electronics & Information College, Qingdao University, Qingdao 266071, China

II. The Sample Entropy

The Sample Entropy is proposed by Richman [13]. It is an improved method of complexity, which provides a dimensionless index representing signal characteristics by measuring the complexity of the sequence [13,14]. The more complex signal sequence, the greater the corresponding sample entropy. The Sample Entropy is similar to the approximate entropy [15-17], but it solves the problem of statistical inconsistency in approximate entropy, and reduces the error of approximate entropy though its principle of calculation.

The algorithm of computing sample entropy is as blow [18]:

Step 1. Form a series of vector $x(1)$~$x(N\text{-}m+1)$, whose length is m, defined by:

$$x(i)=\left[x(i),x(i+1),...,x(i+m-1)\right] \quad i=1\sim N-m+1 \tag{1}$$

Step 2. The distance $d[X(i), X(j)]$ between two vectors $X(i)$ and $X(j)$ is defined as the maximum absolute difference in the scalar components of $X(i)$ and $X(j)$.

$$d\left[X(i),X(j)\right]=\max_{k=0\sim m-1}\left[x(i+k)-x(j+k)\right] \tag{2}$$

Step 3. Define $N^m(i)$, the number of vector $X(j)$ (j=1~$N-m+1$, $j\neq i$) such that the distance between the vectors $X(j)$ and the generic vector $X(i)$ is lower than r.

Define $B_i^m(r)$ as:

$$B_i^m(r)=\frac{N^m(i)}{N-m+1},\ i=1\sim N-m+1 \tag{3}$$

Step 4. The actual logarithmic average over all the vectors of the $B_i^m(r)$ probability is computed as:

$$B^m(r)=\frac{1}{N-m+1}\sum_{i=1}^{N-m+1}B_i^m(r) \tag{4}$$

Step 5. Increase the length of vector to m+1. Repeat steps 1 to 4 and find $B_i^{m+1}(r)$.

Step 6. Theoretically, the sample entropy is defined as:

$$SampEn(m,r)=\lim_{N\to\infty}\left\{-\ln\left[B^{m+1}(r)/B^m(r)\right]\right\} \tag{5}$$

In practice, the length of time series N is finite, therefore, we can only get the estimate result instead of precise result. When the length is N, estimation of SampEn denoted as:

$$SampEn(m,r)=-\ln\left[B^{m+1}(r)/B^m(r)\right] \tag{6}$$

It is important for the calculation of sample entropy to determine the value of m and r, because the sample entropy values directly relate to the value of m and r. In this paper, we take the dimension m=2 and 3, and the threshold value is set as r=0.12Std, where Std is the standard deviation of the time series $x(i)$.

It can be seen from the above that, for a sequence whose length is N, there are two cases when calculating $B_i^m(r)$. When m=2, it needs $2\times C_N^2$ difference-comparison operations; when m=3, it requires $3\times C_N^2$ difference-comparison operations. Totally, the traditional sample entropy requires $5\times C_N^2$ difference-comparison operations. This method has a large amount of calculation and a long running time, so it needs to be improved.

III. A Fast Algorithm for Sample Entropy Computation

According to step 2 and step 3 of the traditional sample entropy algorithm, the vector distance can be transformed into binary form: for i, j, if $|x(i+k)-x(j+k)|<r$, ($i\neq j$, k=0,…,m-1) , the result is 1, otherwise, the result is 0. It compares the distance between the sequence points in the signal sequence and the size of the similarity tolerance, and the result is transformed to binary [19]. Then, neighborhood matrix is formed, which contains all the comparison result between X_i and X_j. And the neighborhood matrix of the high-dimensional signal vector can be deduced recursively from the neighborhood matrix of the low-dimensional signal vector. In this way, floating point calculation can be translated into logical and integer operations, which can greatly speed up the computation [20]. The steps of the proposed fast algorithm are as follows:

(1) Form a series of vector $x(1)$~$x(N-m+1)$, where m=1.

(2) Defining $d_1(i,j)$ as the element of the neighborhood matrix between $X(i)$ and $X(j)$:

$$d_1(i,j)=\begin{cases}1, |x(i)-x(j)|<r\\0, |x(i)-x(j)|\geq r\end{cases} \tag{7}$$

(3) Obtain $B_i^1(r)$ as the neighborhood matrix of the one-dimensional signal vector.

(4) When m=2, the neighborhood matrix of the two-dimensional signal vector can be deduced through the one-dimensional signal vector:

$$d_2(i,j)=d_1(i,j)\cap d_1(i+1,j+1) \tag{8}$$

$$B_i^2(r)=\sum_{j=1}^{N-1} d_2(i,j) \tag{9}$$

(5) When m=3, the neighborhood matrix of the three-dimensional signal vector can be deduced through the two-dimensional signal vector:

$$d_3(i,j)=d_2(i,j)\cap d_2(i+1,j+1) \tag{10}$$

$$B_i^3(r)=\sum_{j=1}^{N-2} d_3(i,j) \tag{11}$$

The general formula for the neighborhood matrix of the m-dimensional signal vector deduced the neighborhood matrix of the (m-1)-dimensional signal vector is as follows:

$$d_m(i,j)=d_{m-1}(i,j)\cap d_{m-1}(i+1,j+1) \tag{12}$$

$$B_i^m(r)=\sum_{j=1}^{N-1} d_m(i,j) \tag{13}$$

(6) Average $B_i^m(r)$:

$$B^m(r)=\frac{1}{(N-m)(N-m+1)}\sum_{i=1}^{N-m+1} B_i^m(r) \tag{14}$$

(7) Sample entropy is computed as:

$$SampEn(m,r)=-\ln\left[B^{m+1}(r)/B^m(r)\right] \tag{15}$$

In the fast algorithm for a sequence with length N, it only needs C_N^2 difference-comparison operations to get the neighborhood matrix. The neighborhood matrix of the multidimensional signal vector can be deduced through the (m-1)-dimensional vector of the neighborhood matrix. So, the fast algorithm reduces the difference-comparison operations to 1/5 times of the original method. Compared with the traditional sample entropy calculation, the fast method obviously improves the computation efficiency.

IV. Experimental Result Analysis

In this study, MATLAB R2016a is used as the programming environment. The operating system is Windows 10, ultimate (64 bit), the processor is Intel(R) Core(TM) i7-7500U Processor, and the installed memory is 8.00GB. The speech signals are divided into frames, and the sample entropy of each frame is calculated. The frame length is 500 and the frame shift is 50. For each single phoneme signal, we have calculated the sample entropy of each frame. The experimental results are shown from Fig. 1 to Fig. 7.

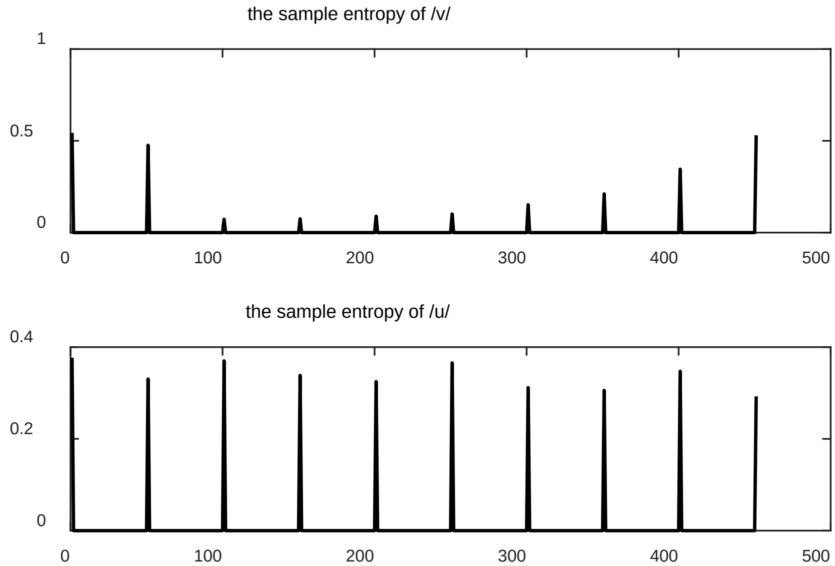

Fig. 1. The results of /v/ and /u/

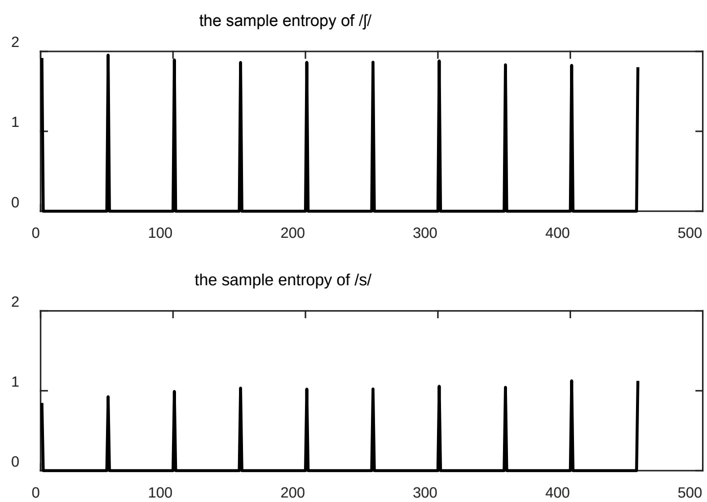

Fig. 2. The results of /ʃ/ and /s/

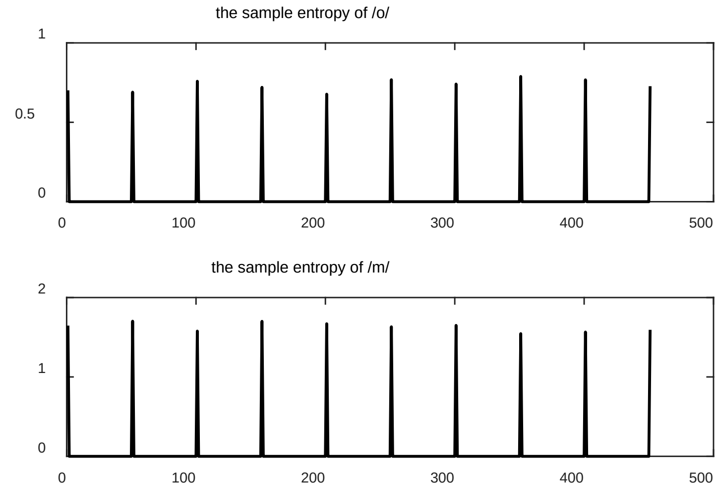

Fig. 3. The results of /o/ and /m/

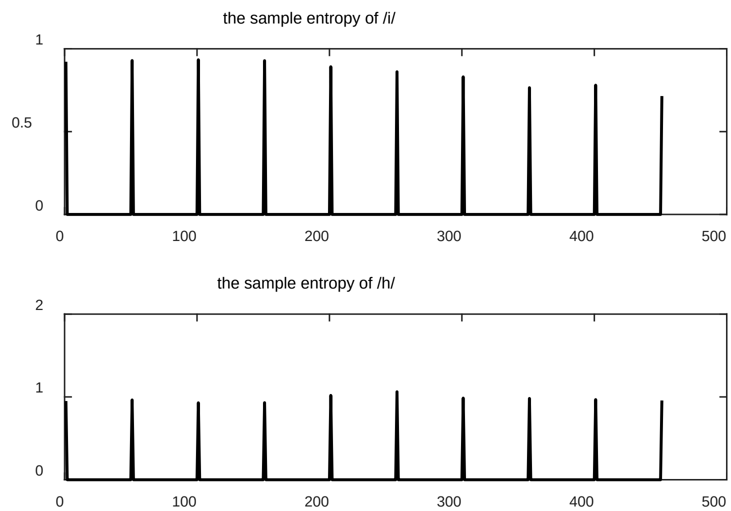

Fig. 4. The results of /i/ and /h/

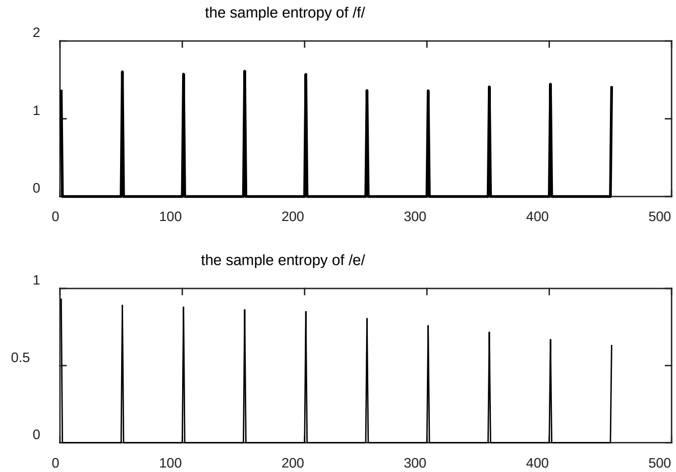

Fig. 5. The results of /f/ and /u/

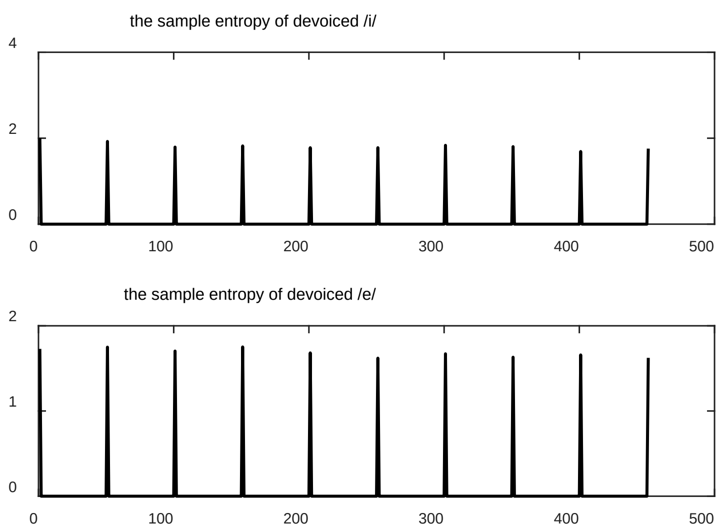

Fig. 6. The results of devoiced /i/ and devoiced /e/

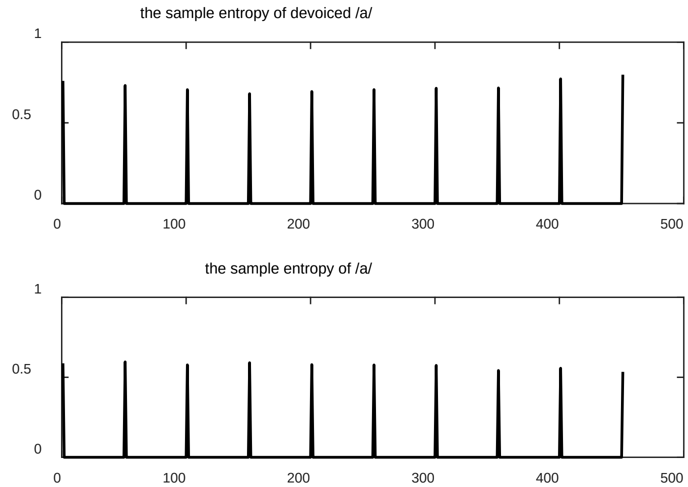

Fig. 7. The results of /a/ and devoiced /a/

Here is a Summary of the running time of each of the above sounds in Table 1 for comparison.

Table 1. The running time of each signal speech

Speech signal	traditional sample entropy algorithm (seconds)	fast sample entropy algorithm (seconds)
/v/	3.873711	0.133045
/u/	3.765491	0.132110
/f/	3.724067	0.152851
/ʃ/	3.800903	0.146402
/o/	3.742444	0.129946
/m/	3.650981	0.145057
/i/	3.684277	0.137326
/e/	3.698717	0.125836
/a/	3.715260	0.129309
/s/	3.741394	0.133130
/h/	3.718233	0.127741
/f/	3.723706	0.127774
Devoiced-/i/	3.837166	0.132441
Devoiced-/a/	3.709344	0.124831
Devoiced-/e/	3.727959	0.127638

The following conclusions can be drawn from the experimental results: the fast sample entropy algorithm runs each single phoneme signal is about 0.13 seconds, and runs “philosophy” for 14.888 seconds. Under the same environment, the running time of the traditional algorithm is about 3.8 seconds. The fast sample entropy algorithm is about 30 times faster than the traditional algorithm.

Then the experimental results of the English word “mathematics” and the word “philosophy” run by the fast sample entropy algorithm are shown in the following figures, where the frame length is 500 but the frame shift is 1:

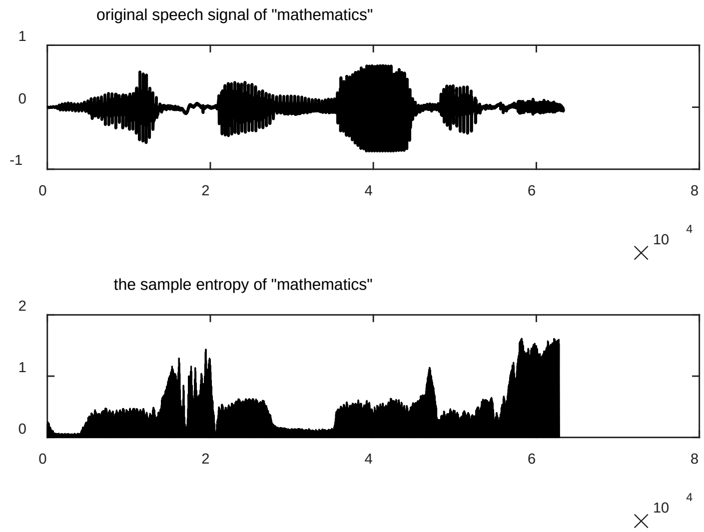

Fig. 8. The result of the “mathematics”

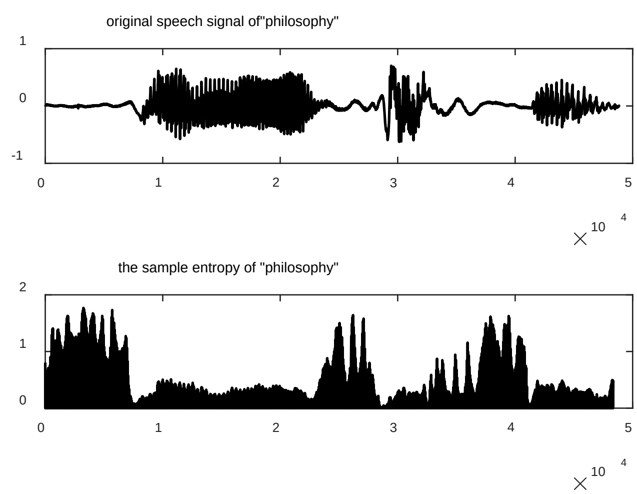

Fig. 9. The result of the “philosophy”

The fast sample entropy algorithm runs the English word “mathematics” for 20.505 seconds, and runs “philosophy” for 14.888 seconds. The running time of each frame is 0.041 and 0.029 seconds. Under the same environment, the running time of the traditional algorithm is 1643.501821 seconds and 1245.634468 seconds. From the above results, it can be seen that the fast algorithm reduces the computation time by 80 times and improves the computation efficiency.

V. Conclusion

Numerical calculation is transformed to binary in the fast algorithm, and the neighborhood matrix of the high-dimensional signal vector is deduced from the neighborhood matrix of the low-dimensional signal vector. The algorithm analysis and experimental results show that the improved algorithm can shorten the computation time and improve the efficiency of the calculation, so that the sample entropy can be used in real-time processing applications.

References

[1] X. H. Zhang, W. Wang. “Synchronization between different hyperchaotic systems and dimensions and its application in secret communication,” Computer Science. Chongqing, vol. 39,pp. 220-222, April 2012.

[2] C. H. Huang, C. H. Lin, C. L. Kuo. “Chaos Synchronization-based detector for power-quality disturbances classification in a power system,” IEEE Tranxactions on Power Delivery. United States, vol. 26, pp. 944-935, April 2011.

[3] T Deng, J H Lin, C G Huang, et al. “AE signal extraction of a high speed train roller based on modified EEMD and segment sample entropy”, China. vol. 16, pp. 148-154, October, 2017.

[4] L Zhang, V Chan. “Scalable Fast Scheduling for Optical Flow Switching Using Sampled Entropy and Mutual Information Broadcast,” Journal of Optical Communications & Networking, United States. vol. 5, pp. 459-475, December, 2014.

[5] T Peng, X H Chen, C B Zhuang. “Analysis on complexity of monthly runoff series based on sample entropy in the Dongjiang river”,Ecology and Environmental Sciences, China. vol. 18, pp. 1379-1382, November, 2009.

[6] J M Yentes, N Hunt, K K Schmid, et al. “The appropriate use of approximate entropy and sample entropy with short data sets,” Annals of Biomedical Engineering, United States. vol. 41, pp. 349-365, February, 2013

[7] M G Lamoureux, B G Nickerson. “A deterministic skip list for k-dimensional range search,” Acta Informatica, Germany. Vol. 4, pp. 221-255, December, 2005.

[8] J Xu, Q Xiao, F Zhou. “An Improved Authenticated Skip List for Relational Query Authentication,,” International Conference on Broadband & Wireless Computing. Guangzhou, pp.229-232, November 2014.

[9] S Mandal, S Chakraborty, S Karmakar. “Distributed deterministic 1–2 skip list for peer-to-peer system,” Peer-to-Peer Networking and Applications. America, vol 1, pp. 63-86, January, 2015

[10] F W Zhu, S J Lv, Z P Yuan. “Small point cloud matching based on bi-direction random KD-tree,” Asia International Symposium on Mechatronics, China, pp.5, October, 2016.

[11] I. Wald, H. Friedrich, G. Marmitt, et al. “Fast isosurface ray tracing using implicit KD-trees,” IEEE Transactions on Visualization and Computer Graphics, United States. vol. 11, pp. 562-572, July, 2005.

[12] X X Bai, X S Dong, Y Q Su. “Edge Propagation KD-Trees: Computing Approximate Nearest Neighbor Fields,” United States. vol. 22, pp. 2209-2213, August, 2015.

[13] J. S. Richman, J. R. Moorman. “Physiological timeseries analysis using approximate entropy and sample entropy,” American Journal of Physiology Heart & Circulatory Physiology. United States, vol. 278, pp. 2039-2049,2000.

[14] X. Q. Zhang, J. Liang, X. Zhang, et al. “Combined model for ultra short-term wind power prediction based on Sample Entropy and extreme learning machine,” Proceedings of the Csee. Beijing, vol.33, pp. 33-40, September, 2013.

[15] S Pincus. “Approximate entropy (ApEn) as a complexity measure,” Chaos, America, vol. 1, pp.110, December, 1995.

[16] Y H An, J P Ou. “Structural damage localisation for a frame structure from changes in curvature of approximate entropy feature vectors,” Nondestructive Testing & Evaluation, England. vol. 1, pp. :80-97, December, 2014.

[17] Y H Pan, Y H Wang, S F Liang, et al. “Fast computation of sample entropy and approximate entropy in biomedicine,” Computer Methods & Programs in Biomedicine, Netherlands, vol. 3, pp. 382-396, December, 2011.

[18] Z. H. Zhao, S. P. Yang. “Sample entropy-based roller bearing fault diagnosis method,” Journal of Vibration & Shock. Beijing, vol.31. pp. 136-140, October, 2012.

[19] L. Wang, G. Z. Xu, J. Wang, et al. “Motor Imagery BCI Research Based on Sample Entropy and SVM,” Sixth International Conference on Electromagnetic Field Problems & Applications. Dalian, pp.1-4, June, 2012.

[20] T. Q. Chang, W. Li, J. W. Chen and Y. Zhang. “Wavelet threshold de-noising method based on maximum energy matching and fast sample entropy,” Computer Engineering & Applications. Beijing, vol. 50, pp. 210-213, November 2014.