Multivariate permutation entropy: implementation and its application for complexity analysis of biomedical signals

Abstract

To measure the complexity of multivariate systems, the multivariate permutation entropy (MvPE) algorithm is proposed. It is employed to measure complexity of multivariate system in the phase space. As an application, MvPE is applied to analyze the complexity of chaotic systems, including hyperchaotic Henon map, fractional-order simplified Lorenz system and financial chaotic ´ system. Results show that MvPE algorithm is effective for analyzing the complexity of the multivariate systems. Compared with PE, MvPE has better robustness for noise and sampling interval, and the results are not affected by different normalization methods [1]. In this article, we are going to implement MvPE in Python and analyze complexity of MEG signals using written function.

Introduction

1.MEG: Magnetoencephalography (MEG) is a non-invasive technique for investigating human brain activity. It allows the measurement of ongoing brain activity on a millisecond-by-millisecond basis, and it shows where in the brain activity is produced.
At the cellular level, individual neurons in the brain have electrochemical properties that result in the flow of electrically charged ions through a cell. Electromagnetic fields are generated by the net effect of this slow ionic current flow. While the magnitude of fields associated with an individual neuron is negligible, the effect of multiple neurons (for example, 50,000 – 100,000) excited together in a specific area generates a measureable magnetic field outside the head. These neuromagnetic signals generated by the brain are extremely small—a billionth of the strength of the earth's magnetic field. Therefore, MEG scanners require superconducting sensors (SQUID, superconducting quantum interference device). The SQUID sensors are bathed in a large liquid helium cooling unit at approximately -269 degrees C. Due to low impedance at this temperature, the SQUID device can detect and amplify magnetic fields generated by neurons a few centimeters away from the sensors. A magnetically shielded room houses the equipment, and mitigates interference. MEG has advantages over both fMRI and EEG. The technologies complement each other, but only MEG provides timing as well as spatial information about brain activity. fMRI signals reflect brain activity indirectly, by measuring the oxygenation of blood flowing near active neurons. MEG signals are obtained directly from neuronal electrical activity. MEG signals are able to show absolute neuronal activity whereas the fMRI signals show relative neuronal activity, meaning that the fMRI signal analysis always be compared to reference neuronal activity. This means that MEG can be recorded in sleeping subjects. MEG does not make any operational noise, unlike fMRI. While fMRI measurement requires the complete absence of subject movement during recording, MEG measurement does not, so children can move their heads within the MEG helmet.

2.Permutatin entropy: For a given time series $\{x(n), n = 1, 2, 3, \cdots, N\}$ and reconstruction dimension d, the reconstructed series is denoted by
$X(i) = \{x(i), x(i + 1), \cdots, x(i + d - 1)\}$,

where $i = 1, 2, \cdots, N - d + 1$. Then X(i) can be arranged in an increasing order $\pi = (r_0, r_1, \cdots, r_{d-1})$, where $x_i + r_0 \leq x_{i+r1} \leq \cdots \leq x_{i+rd-1}$

Obviously, there are d! possible order patterns. Taking d = 3 as an example, there are 6 possible patters: $\{\pi_1, x_1 \leq x_2 \leq x_3\}$, $\{\pi_2, x_1 \leq x_3 \leq x_2\}$, $\{\pi_3, x_2 \leq x_1 \leq x_3\}$, $\{\pi_4, x_3 \leq x_1 \leq x_2\}$, $\{\pi_5, x_2 \leq x_3 \leq x_1\}$, and $\{\pi_6, x_3 \leq x_2 \leq x_1\}$, as shown in Fig.1. If we let πj = j, j = 1, 2, · · · , d!, we can get a patter series {s(i), i = 1, 2, · · · , N − d + 1}. If the order patter of X(i) is π_j , then s(i) = j. The Bandt-Pompe probability distribution p(πj) is denoted by

$$p(\pi_j) = \frac{\#\{s|i \leq N - d + 1; s = j\}}{N - d + 1}$$

where the symbol # stands for ”number”. According to the definition of Shannon entropy, the normalized PE is defined as

$$PE(x, d) = S[p]/S_{max} = -\sum_{j=1}^{d!} p(\pi j) \ln p(\pi j)]/Smax$$

where $S_{max} = S[P_e] = \ln(d!)$, and $P_e = \{1/d!, \cdots, 1/d!\}$. The range of d is {2, 3, · · · , 7} [7,8]. Generally, larger PE value means the time series is more complex.{He, 2016 #1}

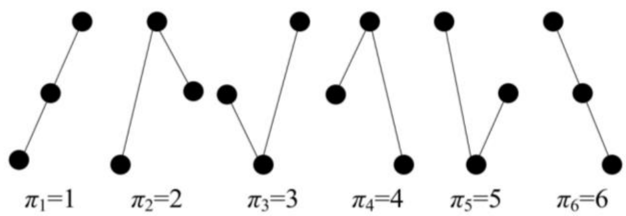


Figure 1: Bandt-Pompe order patterns for embedding dimension $d = 3$ (π^6)

3. Multivariate permutation entropy

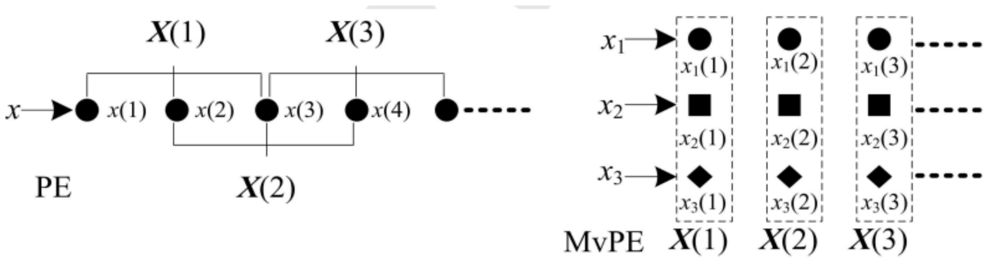
Suppose that sequences generated by a multivariate system are presented as $\{x_j(n), n = 1, 2, 3, \cdots, N, j = 1, 2, ..., \sigma\}$, where σ is the dimension of the system or the number of chosen time series. As the fluctuation range of each time series is different, we should normalize each series. Here the min-max scaling is employed, which is defined as

$$\tilde{x}(n) = \frac{xj(n) - \min(xj)}{\max(xj) - \min(xj)}$$

where max(·) is the maximum function and min(·) is the minimum function. Then the new reconstructed series is defined as

$$X(n) = \{\tilde{x}_1(n), \tilde{x}_2(n), \ldots, \tilde{x}_\sigma(n)\}$$

for MvPE calculation. Obviously, σ time series of the system are used for complexity measure, and the difference between PE and MvPE to obtain the reconstructed series is visualized in Fig.2 for better understanding of the two algorithms.



For each X(n), a certain order pattern π can be obtained. So we have the corresponding order patter series {s(n), n = 1, 2, · · · , N} according to the reconstructed series {X(n), n = 1, 2, 3, · · · , N}. The probability distribution $p(\pi_j)$ is obtained according to

$$p(\pi_j) = \frac{1}{N} \# \{s | i \leq N; s = j\}$$

where j=1, 2, · · · , M, and M is the number of the possible patterns. Thus, MvPE is defined as

$$MvPE(x) = S[p]/S_{max} = [-\sum_{j=1}^{M} p(\pi j) \ln p(\pi j)]/Smax.$$

where S max = S $[P_e]$ = ln(M). {He, 2016 #1}

Database: our database consist of MEG signal of 19 subjects, each containing tow classified 6-variable signal. Two mentioned classes are standard and deviant.

Implementation: In this section we implement function using Python to calculate MvPE for a multivariate signal.

As we see in part 3 of Introduction, first thing that we should do is normalized the multivariate signal using min, max method i.e., we should normalized each row of our data matrix in a way that minimum of that row became zero and maximum became one, and we do it using the code below:

```
def minmaxscaler(mat):

    l=[]
    for i in range(len(mat)):
        Min = mat[i,:].min()
        Max = mat[i,:].max()
        for k in mat[i,:]:

            l.append((k-Min)/(Max-Min))

    return(np.array(l).reshape(mat.shape))
```

Then we implement algorithm that mentioned in part 3 of introduction using code below:

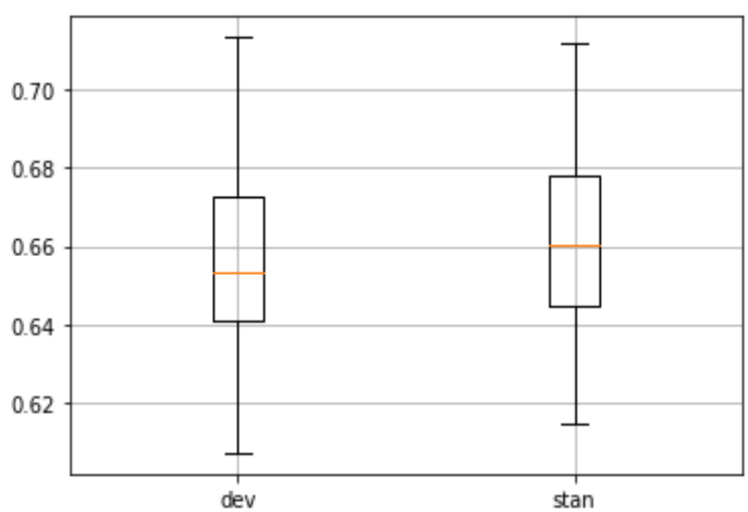
```python
def MVPE(mvts):

    ScaledMvts = minmaxscaler(mvts)
    permutations = np.array(list(it.permutations(range(6))))
    patmat = permutations.dot(ScaledMvts) #pattern matrix
    Plist = patmat.argmax(0) #list consist of pattern for each column
    Pdict = dict()
    for i in Plist:
        if i in Pdict:
            Pdict[i] += 1
        else:
            Pdict[i] = 1
    for i in Pdict:
        Pdict[i] = Pdict[i]/501
    MvPE=0
    for i in Pdict:
        MvPE += -Pdict[i]*log(Pdict[i])
    MvPEN = MvPE/(log(factorial(6)))

    return MvPEN
```

Then we use MVPE function to calculate Multivariate permutation entropy of each signal.

Conclusion: After calculating MvPE for standard and deviant signals we plot the boxplot of each class as follows:



Boxplot shows that mean of MvPE for standard signal is bigger than mean of deviant signal, So it is more complex.

On the other hand variance of deviant signal is bigger than standard signal, so it is more chaotic.

[1] S. He, K. Sun, H. J. P. A. S. M. Wang, and i. Applications, "Multivariate permutation entropy and its application for complexity analysis of chaotic systems," vol. 461, pp. 812-823, 2016.