Entropy and Lorenz Model

November 2021

1 Entropy

1.1 Theory svd

We can find online the documentation of the function svd-entropy [[1]].

$$S_{svd} = S(\mathbf{x(t)}, order, delay) \tag{1}$$

We denote with $\mathbf{x}$ the dataset that we want to study.

$$\begin{aligned} \mathbf{x(t_0)} &= \mathbf{x}_0, \\ \mathbf{x(t)} &= (x_1, x_2, x_3, \ldots, x_N) \end{aligned} \tag{2}$$

This function creates a matrix Y with the dataset.

$$\mathbf{y}(i) = (x_i, x_{i+\text{delay}}, ..., x_{i+(\text{order}-1)\text{delay}}) \tag{3}$$

$$Y = \begin{pmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \\ . \\ . \\ . \\ \mathbf{y}(N-(ord-1)del) \end{pmatrix}$$

If we take as parameters $order = 3$ and $delay = 1$ we obtain the following matrix:

$$Y = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \\ ... & ... & ... \\ ... & x_{N-1} & x_N \end{pmatrix}$$

The next step is to write this matrix using the SVD decomposition.

$$Y = U\Sigma V^* \tag{4}$$

[1] `https://raphaelvallat.com/entropy/build/html/generated/entropy.svd_entropy.html`

Where U is an unitary matrix, V is also an unitary matrix, and Σ is a diagonal matrix. This is linked with PCA. Now we denotes the eigenvalues of the matrix Σ with σ_i. Now we can compute the average eigenvalues:

$$\bar{\sigma}_k = \frac{\sigma_k}{\sum_j^M \sigma_j} \tag{5}$$

Where M is the number of eigenvalues.

After this we can compute the SVD Entropy:

$$S_{svd} = -\sum_k^M \bar{\sigma}_k log_2(\bar{\sigma}_k) \tag{6}$$

1.2 Shannon Entropy

We analyze the Shannon Entropy function [2].

```python
def shannon_entropy(time_series):
    """Return the Shannon Entropy of the sample data.
    Args:
        time_series: Vector or string of the sample data
    Returns:
        The Shannon Entropy as float value
    """

    # Check if string
    if not isinstance(time_series, str):
        time_series = list(time_series)

    # Create a frequency data
    data_set = list(set(time_series))
    freq_list = []
    for entry in data_set:
        counter = 0.
        for i in time_series:
            if i == entry:
                counter += 1
        freq_list.append(float(counter) / len(time_series))

    # Shannon entropy
    ent = 0.0
    for freq in freq_list:
        ent += freq * np.log2(freq)
    ent = -ent
    return ent
```

$$S_{sh} = S(\mathbf{x}) \tag{7}$$

Here we find the definition of Shannon Entropy:

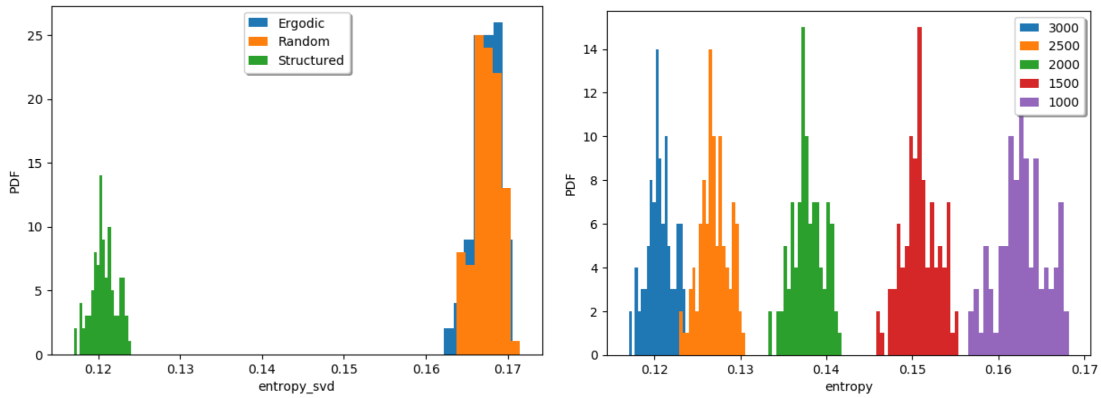
$$S_{sh} = -\sum_i P(x_i) log_2(P(x_i)) \tag{8}$$

Where $P(x_i)$ is the frequency of x_i in the dataset $\mathbf{x}$.

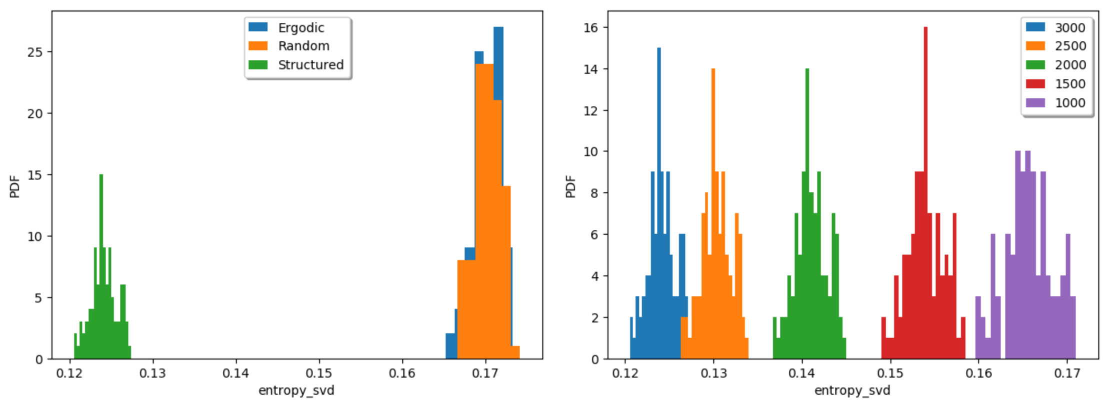
[2] https://github.com/nikdon/pyEntropy/blob/master/pyentrp/entropy.py

1.3 Plots Entropy

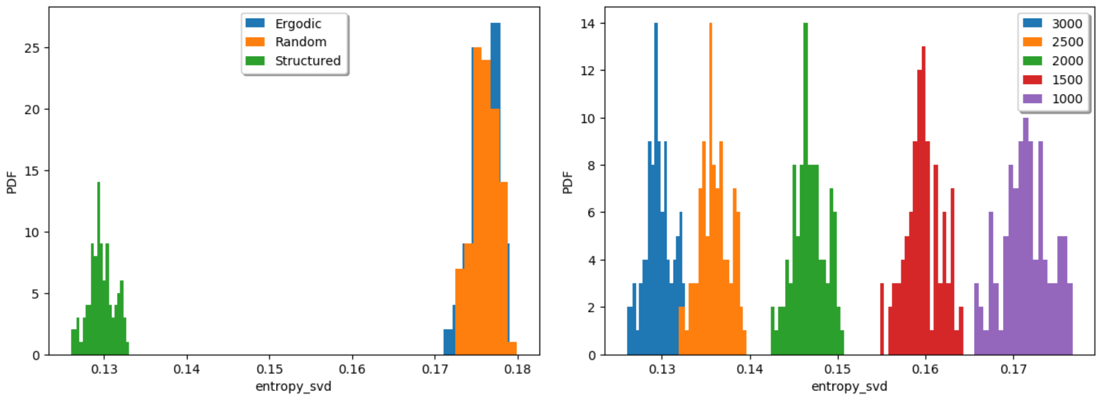
1.3.1 SVD with order 3



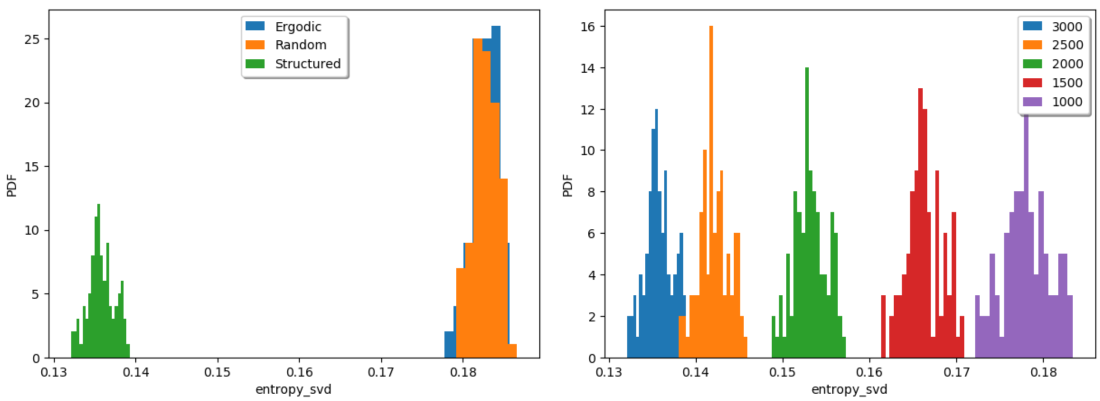
1.3.2 SVD with order 4



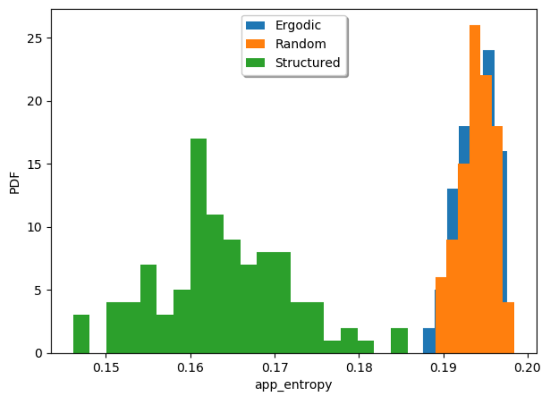
1.3.3 SVD with order 5

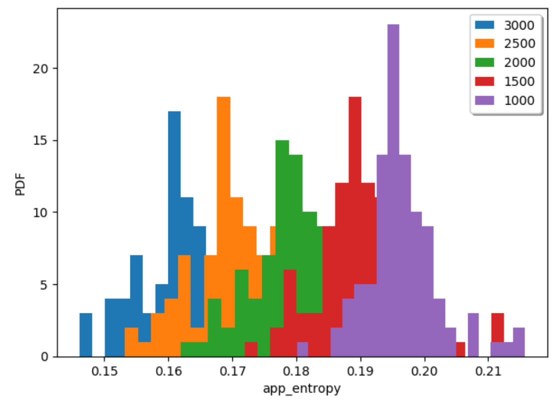


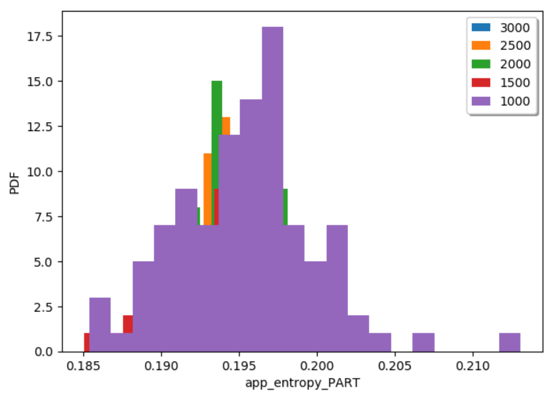
1.3.4 SVD with order 6



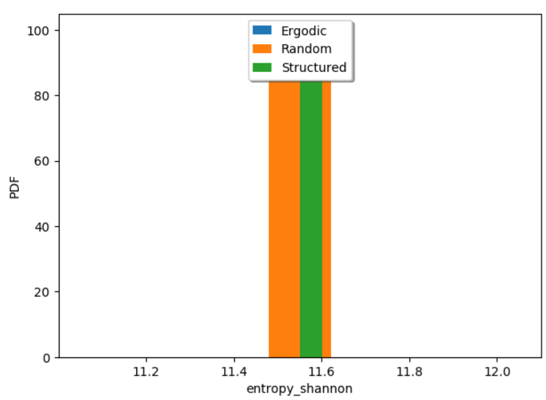
1.3.5 Approx Entropy

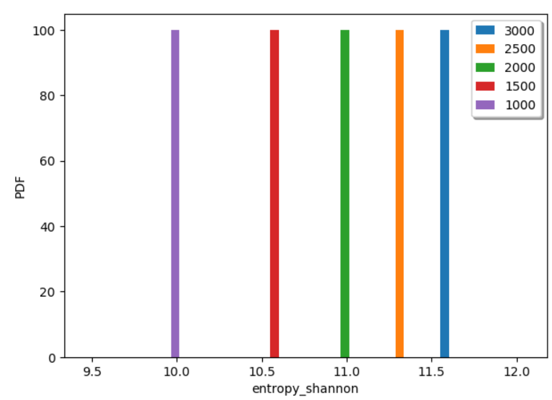






1.3.6 Shannon Entropy





Shannon Entropy value for each dataset= 11.550746785383529.