* **Introduction**

Entropy of a signal is a technique used to quantify the amount of regularity and the unpredictability of fluctuations over time-series data. It is used to numerically quantify the amount of irregularity or fluctuations in a given signal. Numerous methods exist to calculate the entropy of a given time-series. Some of these have been already applied to EEG data in the past. The entropies used in this project are Shannon, Tsallis, Renyi and Permutation entropy.

* **Time Dependent Entropy**

Time Dependent Entropy corresponds to the trend of entropy of a given signal over time. To get a signal’s Time Dependent Entropy, the signal is first split into windows. This done using to parameters namely the window size (w) and step size (delta). The first window is from first to wth value of the discrete-time signal while the second window starts from (1+delta) to (1+w+delta) values of the signal and so on. Thus number of windows(M) created are, considering length of signal as K:

M= K-w/delta

The time dependent entropy is then calculated using each of these windows and thus M such values are obtained. These M values form a time-series called as the time-dependent entropy of the given time-series.

To find the entropy of a given window, various methods can be used. We employ some of these methods. First the minimum and maximum values of the signal in that window is noted. The amplitude interval between these two values is then divided into L intervals where L is a variable parameter. Then each of these slots/intervals are assigned a probability value. The probability value corresponds to the probability of the signal being in that interval which is equal to number of points in that interval divided by the total number of points in the interval. These probabilities are then used to calculate different entropies. We have used four such entropies:

1. Shannon Entropy
2. Tsallis Entropy
3. Renyi Entropy
4. Permutation Entropy

* **Theory**

1. Shannon Entropy

In development of foundations of classical information theory, a rigorous proof of Shannon’s entropy based on additivity law was presented. Suppose a system is divided into two subsystems A and B, and pij(A, B) is joint probability of finding A in ith state and B in jth state then:

pij(A, B) = pi(A) \* pij(B|A)

Shannon’s entropy of this system is defined as

S(A,B) = -kb∑ pij(A,B) \* ln(pij(A,B))

Similarly , for a discrete time signal, the probabilities can be calculated as mentioned above. Let these values be pi where i belongs to (1, L). The Shannon entropy of this window will be:

Sh\_ent = -∑pi \* ln(pi)

1. Tsallis Entropy and Renyi Entropy

Tsallis Entropy and Renyi entropy are modification of the well-known Shannon entropy. Both of them are non-additive and have been used to deal with EEG or EEG type discrete type signals before. They too rely on the same probability measures as Shannon entropy but have an additional parameter each, which can be used to bring out signal characteristics that Shannon entropy cannot highlight. Tsallis entropy can be modified by using its q value whereas Renyi entropy has a parameter known as alpha. Their formulas are:

Te\_ent = -∑(piq-pi)

RE\_ent = -∑piα/(1-α)

Thus both entropies can be modified by changing their parameters. Tsallis entropy can be used to estimate Shannon entropy as they tend to it at q tends to zero.

1. Permutation Entropy

* **Files containing all functions used to generate and analyse TDE:**

1. entropy.py
2. sig\_entropy:

This function is used to find the time-dependent entropy of an entire time series with varying entropy parameters and types. It takes the discrete time signal and all the parameters as an input and outputs the TDE as a list. It also takes parametric inputs for the different types of entropies with a string that decides which entropy to calculate.

Arguments:

***L***: Number of partitions of amplitude

***w***: Width of the moving window

***delta***: Step size of the window

***sig***: Time series whose entropy is to be found

***entropy\_name***: Name of entropy. Currently Shannon / renyi / tsallis / permutation are available

***q***: q for tsallis. Only if entropy is tsallis

***alpha***: alpha for renyi

***d, delay***: d, delay for permutation

Returns:

***entropy***: An array with the entropy values

1. partition\_entropy:

This function returns a single entropy value while taking as input a complete or part of a discrete-time signal and other parametric inputs. It considers the entire signal to be divided into one partition and hence returns only one value. This function is used by the sig\_entropy function.

Arguments:

***partition***: The signal/partition whose entropy is to be found

***L***: Number of partitions of amplitude

***w***: Width of the moving window

***delta***: Step size of the window

***entropy\_name:*** Name of entropy. Currently Shannon / renyi / tsallis / permutation are available

***q***: q for tsallis. Only if entropy is tsallis

***alpha***: alpha for renyi

***d, delay:*** d, delay for permutation

Returns:

***per\_ent***: A single float value containing the entropy the signal

1. P\_m\_l:

This function calculates the probability that a signal is present in one specific bin of the amplitude interval being considered. As discussed before the amplitude interval is defined as interval between minimum and maximum values of the signal. This is then split into different partitions. This function finds probability of signal being in one of these partitions.

Arguments:

***partition*** : input signal

***k***: Defines the part of the partition. Should be between 0 and L-1

***slot\_height***: Height of each slot/part of the partition

***maxp, minp***: Maximum and Minimum values in the signal

Return:

Single float value containing probability of signal being in that slot of the partition

1. shannon\_entropy:

This function returns a single value containing Shannon entropy of the given signal.

Arguments:

***L***: Number of partitions of amplitude

***partition***: Time series whose entropy is to be found

***slot\_height***: Height of each slot/part of the partition

***maxp, minp***: Maximum and Minimum values in the signal

Returns:

***sh\_ent***: Float value containing Shannon entropy

1. Tsallis entropy:

This function returns Tsallis entropy of given signal. Note: q value of tsallis should be none zero. If you require entropy at q tending to zero, use the Shannon entropy function.

Arguments:

***L***: Number of partitions of amplitude

***partition***: Time series whose entropy is to be found

***slot\_height***: Height of each slot/part of the partition

***maxp, minp***: Maximum and Minimum values in the signal

***q***: q parameter for tsallis entropy

Returns:

Float value containing tsallis entropy

1. renyi\_entropy:

This function returns Renyi entropy of a given signal. Note: alpha value for renyi entropy must not be equal to 1. For alpha equal to 1, use Shannon entropy instead as renyi entropy tends to Shannon entropy when alpha tends to 1.

L: Number of partitions of amplitude

***partition***: Time series whose entropy is to be found

***slot\_height***: Height of each slot/part of the partition

***maxp, minp:*** Maximum and Minimum values in the signal

***alpha***: alpha parameter for renyi entropy

Returns:

Float value containing renyi entropy

1. permutation\_entropy:

This function returns permutation entropy of a given signal.

Arguments:

***time\_series***: Time series for analysis

***m:*** Order of permutation entropy

***delay***: Time delay

Returns:

Vector containing Permutation Entropy

1. read\_write.py

The entire project was done by taking inputs from and writing outputs to respective text files. All inputs and outputs were of array format and hence were stored in ‘.txt’ files with each new line containing a new in order element of the array. This file has two important functions namely the reading and writing functions to the files.

1. read\_file:

This function is used to read an array from a locally stored file. Before this, ensure the file has been given appropriate permissions and is in same directory as the working directory. If not, change the working directory by using the *os.chdir()* function.

Arguments:

***file\_name***: Name of file.

Returns:

***vec***: Vector with values as in file

1. write\_to\_file:

This function is used to write an array to a particular file in the working directory. If this file is not present, then it creates a new file with the given name. If it is present it appends the current file. To change this, change the parameter of line 25 from ‘a+’ to ‘w’ or ‘w+’.

Arguments:

***file\_name***: Name of file.

***vector\_output***: The vector to be written to the file

1. process\_and\_plot.py:

This file is used in the monitoring task of mental fatigue or mental workload monitoring. It contains functions used for analysing the change in entropies over time.

1. rolling\_mean:

This is used to calculate rolling\_mean of any discrete time signal. If length of input is l, then this returns an array of length l-n.

Arguments:

***input\_sig***: List/array whose mean is to be calculated

***n***: The window size to be considered while calculating rolling mean. Should be greater than 0 and less than or equal to the length of signal

Returns:

Array with rolling mean of input signal

1. sorted\_array\_with\_indices:

This is an optional function. This was used only to see which channels had a more prominent trend than the others. This function takes an array as an input and returns a sorted array along with another array denoting the original positions of the sorted elements in the new array. This function uses a copy of the passed array and hence can be used on an array which will be required later.

Arguments:

***b*** = input\_array

Returns:

Tuple with first element as sorted array and second containing array with indices of original positions

1. gui.py

This file contains all codes used to make the GUI required for time dependent entropy estimations. This GUI can be used to edit the parameters of the TDE and also plot multiple entities like the EEG signal, a part of it, its TDE or TDE of a specific part of that signal.

For more information, look into the file or follow this reference:

<https://pythonspot.com/pyqt5>