Introduction To Hurst Exponent

The Hurst exponent, so named after British hydrologist Harold Edwin Hurst, is a primary measure for the study of the long-term memory and self-similarity of time series data. It is a crucial measure across numerous fields such as geophysics, economics, finance, and even environmental science, which offers insights into the persistence, trend, and volatility of the data over time.

In this project, the **Hurst exponent** is determined through two methods: the **Multifractal Detrended Fluctuation Analysis (MF-DFA)** and the **Rescaled Range (R/S) method**. The two methods are commonly used to determine the fractality of a time series, with the Hurst exponent determining if the time series is long-range dependent or random.

- **MF-DFA** is a technique that specializes in studying the multifractal behavior of time series through detrending and investigating the fluctuations across scales. The technique is especially handy when dealing with data that shows multifractality or intricate self-similar structures.
- **R/S method**, however, is a more straightforward method founded on rescaling the range of partial sums of the time series. It is a straightforward estimate of the Hurst exponent and is extensively employed in time series data with long-range dependencies.

The objective of this project is to compute the Hurst exponent employing both approaches and compare them. In doing this, we are able to ensure the validity and accuracy of these approaches in determining the Hurst exponent in real-world time series data.

Subsequent sections will outline the theory behind the Hurst exponent, describe how each approach used to compute the Hurst exponent is done, and show the outcomes derived using MF-DFA and the R/S method.

The Hurst Exponent

The **Hurst exponent (H)** is a statistical parameter that is employed to describe the long-memory of time series data. The Hurst exponent gives the amount of persistence or anti-persistence present in the data. The Hurst exponent can lie within the interval from 0 to 1, and it informs about the type of the time series:

- **H <0.5**: Represents an anti-persistent time series. This is a series that, if it is rising at some point, will fall in the future, and vice versa.
- **H = 0.5**: Is a random walk, where the time series has no correlation between the past and the future. This is the behavior of a Brownian motion.
- **H >0.5**: Indicates persistence or trend-following behavior. If the series is going up, it will likely keep going up, and the reverse for declining trends.

The Hurst exponent is used in hydrology, geophysics, economics, and finance to analyze the fractal nature of data and to forecast trends in the future.

Multifractal Detrended Fluctuation Analysis (MF-DFA)

The **Multifractal Detrended Fluctuation Analysis (MF-DFA)** is a detection method for multifractality in time series data. It is an extension of the standard **DFA** (Detrended Fluctuation Analysis) and is best suited for time series with multifractal and complex structures.

The MF-DFA method goes through the following steps:

1. **Detrending**: A polynomial fit (typically of order 1 or 2) is subtracted from non-overlapping windows of the time series to eliminate trends in each window.
2. **Fluctuation Analysis**: The detrended time series is analyzed for fluctuations across a range of scales or window sizes.
3. **Log-Log Plot**: A log-log plot of the fluctuation function is drawn, and the slope of the plot is the value of the Hurst exponent. The MF-DFA technique provides for the determination of various scaling exponents over various regions and hence is particularly suitable for the analysis of multifractal behavior.

The MF-DFA method is robust against trends and is capable of handling a broad variety of time series data, even with noise or non-stationary behavior.

Rescaled Range (R/S) Method

The **Rescaled Range (R/S)** technique is one of the oldest methods to compute the Hurst exponent and is based on the idea of rescaling the range of partial sums of a time series. The technique is simple and efficient for identifying long-range dependence in a time series.

The technique includes the following steps:

1. **Cumulative Sum**: The data of the time series is transformed into a cumulative sum sequence. This process converts the original data into a sequence of partial sums.

2. **Rescaling**: The range of the cumulative sum is computed within a moving window. The range is rescaled by dividing it by the standard deviation of the series within the same window.

3. **Analysis**: The rescaled range is then plotted versus the window size, and a log-log plot is made. The slope of this plot is equivalent to the Hurst exponent.

The R/S method is simple to use and gives a straightforward estimation of the Hurst exponent. It is, however, sensitive to data non-stationarity and might not work as well as more sophisticated techniques such as MF-DFA for highly complicated or multifractal series.

Comparison of MF-DFA and R/S Methods

MF-DFA and the R/S approach both try to estimate the Hurst exponent, but they differ in terms of complexity and which type of time series they are capable of processing:

- **MF-DFA** is better for multifractal or non-stationary time series, since it has been designed to identify multiple scaling exponents and is trend-robust.

- **R/S method** is easier to understand and use but could be less effective with multifractal series or series with strong trends.

Although there are these distinctions, both techniques can give very close values for the Hurst exponent when used on well-behaved time series, and thus it is useful to compare the output of both methods in order to gauge their dependability and stability.

MF-DFA Method - Step-by-Step Explanation

Input Time Series:

- Receiving the time series data (signal) as input is the first step. The signal usually corresponds to a series of observations over time.

Calculating the Profile:

- The time series data is shifted by subtracting its mean value. This is performed to center the data around zero. Then, the cumulative sum of the mean-shifted data is calculated, which is known as the **profile** of the signal. This profile assists in capturing the overall trend or behavior of the signal over time.

Defining the Scales:

- To process the data at various levels, we specify a number of **scales**. A scale is basically a window size for which the data will be processed. Here, the scales are selected logarithmically from a minimum scale to a maximum scale. This enables the method to pick up fine and coarse variations in the data. A fixed number of scales is chosen between these boundaries to compare various time window sizes.

Segmenting the Signal:

- The signal is split into non-overlapping **segments** for every scale. The number of segments varies with the scale and the overall length of the signal. Each segment corresponds to a portion of the signal that will be examined for fluctuations and trends.

Detrending the Data in Each Segment:

- A **detrending** is performed in each segment. This is done by fitting a straight line (linear trend) to the segment's data. The linear fit is then subtracted from the segment, and the **root mean square (RMS)** of the difference (residuals) is computed. The RMS value measures the fluctuation (or deviation from the trend) for the segment. RMS error assists in quantifying the level of randomness or noise within the segment with respect to the fitted trend.

Calculating the Fluctuation Function:

- After the RMS values for all segments are computed, a **fluctuation function** is obtained. This function gives a summary of the fluctuation behavior at each scale. If a particular value of $\mathbf{q}$ is chosen, the fluctuation function is obtained by averaging the RMS values to the power of $\mathbf{q}$. In certain situations, when $\mathbf{q = 0}$, the logarithmic mean of the RMS values squared is employed instead.

Logarithmic Transformation:

- To more easily visualize the scaling relationship between the fluctuation function and the scale, the values of the scale and the fluctuation function are converted to logarithms. This converts the data into a format where a linear relationship can be determined, making it simpler to estimate the scaling exponent.

Performing Linear Regression:

- The next step is to perform a **linear regression** between the logarithms of the scale and the fluctuation function. This regression helps to determine the slope of the relationship. The slope of the log-log plot represents the scaling exponent, which is related to the **Hurst exponent**.

Calculating the Hurst Exponent:

- The slope from the regression is utilized to calculate the **Hurst exponent (H)**. The Hurst exponent is actually the value of the slope minus 1. If slope minus 1 is negative, the Hurst exponent is set as the slope. If the slope minus 1 is positive, it is adjusted by setting slope minus 1 as slope minus 1.

Output:

- The calculated Hurst exponent is the ultimate result of the MF-DFA procedure. This exponent gives information on the long-term memory or persistence of the time series. The Hurst exponent is usually printed and can be saved to a file for later analysis or reporting.

MF-DFA Python Code

```
import numpy as np
from scipy import stats
import os
import glob

def calculate_hurst_exponent(signal):
    N = len(signal)
    Y = np.cumsum(signal - np.mean(signal))  # Profile

    # Define scales
    scale_min = 16
    scale_max = N // 4
    num_scales = 20
    scale_list = np.unique(np.logspace(np.log2(scale_min), np.log2(scale_max),
                                        num=num_scales, base=2, dtype=int))

    F_s = []
    used_scales = []

    m = 1  # Linear detrending
    q = 2  # q=2 for Hurst

    for scale in scale_list:
        Ns = N // scale
        if Ns < 2:
            continue

        rms_list = []

        # Forward segments
        for v in range(0, Ns):
```

```
        start = v * scale
        end = start + scale
        segment = Y[start:end]
        x = np.arange(scale)
        coeffs = np.polyfit(x, segment, m)
        fit = np.polyval(coeffs, x)
        rms = np.sqrt(np.mean((segment - fit)**2))
        rms_list.append(rms)

    # Backward segments
    for v in range(0, Ns):
        start = N - (v+1)*scale
        end = start + scale
        segment = Y[start:end]
        x = np.arange(scale)
        coeffs = np.polyfit(x, segment, m)
        fit = np.polyval(coeffs, x)
        rms = np.sqrt(np.mean((segment - fit)**2))
        rms_list.append(rms)

    rms_list = np.array(rms_list)

    # Calculate fluctuation function
    if q == 0:
        Fq = np.exp(0.5 * np.mean(np.log(rms_list**2)))
    else:
        Fq = (np.mean(rms_list**q))**(1/q)

    F_s.append(Fq)
    used_scales.append(scale)

log_scale = np.log2(used_scales)
```

```
    log_Fq = np.log2(F_s)

    slope, _, _, _, _ = stats.linregress(log_scale, log_Fq)

    if slope-1 < 0:
        return slope
    else:
        return slope-1

# Example usage
folder_path = "input_signal"
text_files = glob.glob(os.path.join(folder_path, "*.txt"))

for file_path in text_files:
    signal_req = np.loadtxt(file_path)
    hurst = calculate_hurst_exponent(signal_req)
    print("Hurst Exponent:", hurst)
    with open("output_signal.txt", "a") as f:
        f.write(f"Hurst Exponent: {hurst}\n")
```

R/S Method - Step-by-Step Description

Input Time Series:

- The process starts by accepting a time series signal as input. The signal usually indicates a collection of data points recorded over time.

Setting the Scales:

- The process establishes a range of scales on which the analysis is done. These ranges from a minimum scale (16) to a maximum scale (N // 4, where N is the size of the signal).
- A total of 20 scales are selected logarithmically between these limits. This logarithmic selection allows for a wide range of scales to be tested, capturing both short-term and long-term behavior in the data.

Dividing the Signal into Segments:

- For every scale, the signal is broken down into non-overlapping segments of corresponding length. At scale 16, for instance, the signal is divided into segments of 16 units of length. For each segment, the following computations are performed.

Detrending the Data:

- For every segment, the segment mean is computed and subtracted from every data point within the segment. This step brings the segment to zero and makes it simple to examine the fluctuations around the mean.

Cumulative Sum:

- The sum of the mean-adjusted segment is then accumulated. The cumulative sum is the running total of deviations from the mean, which can be used to determine the overall trend and fluctuations in the segment.

Determining the Range (R):

- The range (R) of the cumulative sum is calculated by determining the highest and lowest values of the cumulative sum within each segment. The range is obtained by subtracting the lowest from the highest value, which is a measure of the level of changes in the segment.

Calculating the Standard Deviation (S):

- The standard deviation (S) of the initial segment is calculated. The standard deviation calculates how much the points vary from the mean.

Rescaled Range (R/S):

- The rescaled range (R/S) can be determined as a ratio between the range (R) and the standard deviation (S) of each segment. The resultant ratio is a measure of normalized fluctuation for the segment.

Average Rescaled Range:

- For every scale, the mean of the R/S values across all segments is calculated. This mean yields an estimate of the fluctuation across the entire signal at the scale.

Logarithmic Transformation:

- The scale values and the mean rescaled range values are both transformed by using logarithms (log base 2). This transformation linearizes the relationship between the scale and the rescaled range, making it simpler to estimate the scaling exponent.

Linear Regression:

- A linear regression is conducted between the logarithms of the scale values and the logarithms of the average rescaled range values. The slope of the regression line is the scaling exponent, which is equivalent to the Hurst exponent.

Hurst Exponent Calculation:

- The Hurst exponent (H) is calculated from the slope of the regression line. The formula used to calculate the Hurst exponent is:

$$H = \text{slope} - 1$$

- If the slope minus 1 is negative, the Hurst exponent is used as the slope itself.

- If the slope minus 1 is positive, the Hurst exponent is shifted by subtracting 1 from the slope.

Output:

- The last Hurst exponent is the result of the R/S method. This number gives a measure of the long-term memory or persistence of the time series. The Hurst exponent is usually printed and can also be written to a file for later analysis.

R/S Python Code

```
import numpy as np

import matplotlib.pyplot as plt

from numpy.polynomial.polynomial import Polynomial

import os

import glob


def calculate_hurst_exponent_rs(signal):

    N = len(signal)


    # Define scales

    scale_min = 16

    scale_max = N // 4

    num_scales = 20

    scale_list = np.unique(np.logspace(np.log2(scale_min), np.log2(scale_max), num=nu


    RS = []


    for scale in scale_list:

        if N // scale < 2:

            continue


        rs_values = []

        for start in range(0, N - scale + 1, scale):

            segment = signal[start:start+scale]

            mean_seg = np.mean(segment)

            Y = np.cumsum(segment - mean_seg)

            R = np.max(Y) - np.min(Y)

            S = np.std(segment, ddof=1)

            if S != 0:

                rs_values.append(R / S)
```

```
        RS.append(np.mean(rs_values))

    log_scale = np.log2(scale_list)
    log_RS = np.log2(RS)

    slope, _, _, _, _ = stats.linregress(log_scale, log_RS)

    if slope-1 < 0:
        return slope
    else:
        return slope-1

# Example usage
folder_path = "input_signal"
text_files = glob.glob(os.path.join(folder_path, "*.txt"))

for file_path in text_files:
    signal_req = np.loadtxt(file_path)
    hurst = calculate_hurst_exponent(signal_req)
    print("Hurst Exponent:", hurst)
    with open("output_signal.txt", "a") as f:
        f.write(f"Hurst Exponent: {hurst}\n")
```

Results

The Hurst exponents were calculated for four different signals using two different methods: Multifractal Detrended Fluctuation Analysis (MF-DFA) and Rescaled Range (R/S) analysis. Each signal was divided into multiple segments, and for each segment, the Hurst exponent was computed.

MF-DFA Method

Signal 1:

- Hurst Values: 0.59, 0.52, 0.54, 0.55, 0.59, 0.60, 0.55, 0.53, 0.53, 0.62, 0.66, 0.58, 0.53, 0.54, 0.53, 0.55, 0.53, 0.52, 0.57, 0.56, 0.54, 0.61, 0.62, 0.54, 0.57, 0.52, 0.56, 0.58, 0.60, 0.56, 0.51, 0.52, 0.57, 0.57, 0.57, 0.55, 0.55, 0.52, 0.51

Signal 2:

- Hurst Values: 0.39, 0.39, 0.41, 0.41, 0.38, 0.38, 0.40, 0.37, 0.39, 0.38, 0.37, 0.39, 0.41, 0.37, 0.42, 0.41, 0.42, 0.39, 0.41, 0.40, 0.38, 0.40, 0.38

Signal 3:

- Hurst Values: 0.21, 0.25, 0.21, 0.20, 0.23, 0.22, 0.26, 0.26, 0.25, 0.23, 0.24, 0.26, 0.22, 0.25, 0.29, 0.26, 0.25, 0.24, 0.25, 0.27, 0.23, 0.25, 0.24, 0.26, 0.24, 0.30, 0.23, 0.28, 0.25, 0.28, 0.27, 0.25, 0.27, 0.23, 0.22, 0.23, 0.21, 0.24, 0.28

Signal 4:

- Hurst Values: 0.41, 0.36, 0.41, 0.32, 0.39, 0.42, 0.39, 0.40, 0.40, 0.37, 0.33, 0.42, 0.37, 0.32, 0.34, 0.36, 0.41, 0.33, 0.42, 0.43, 0.41, 0.39, 0.40

R/S Analysis Method

Signal 1:

- Hurst Values: 0.59, 0.52, 0.54, 0.55, 0.59, 0.60, 0.55, 0.53, 0.53, 0.62, 0.66, 0.58, 0.53, 0.54, 0.53, 0.55, 0.53, 0.52, 0.57, 0.56, 0.54, 0.61, 0.62, 0.54, 0.57, 0.52, 0.56, 0.58, 0.60, 0.56, 0.51, 0.52, 0.57, 0.57, 0.57, 0.55, 0.55, 0.52, 0.51

Signal 2:

- Hurst Values: 0.39, 0.39, 0.41, 0.41, 0.38, 0.38, 0.40, 0.37, 0.39, 0.38, 0.37, 0.39, 0.41, 0.37, 0.42, 0.41, 0.42, 0.39, 0.41, 0.40, 0.38, 0.40, 0.38

Signal 3:

- Hurst Values: 0.21, 0.25, 0.21, 0.20, 0.23, 0.22, 0.26, 0.26, 0.25, 0.23, 0.24, 0.26, 0.22, 0.25, 0.29, 0.26, 0.25, 0.24, 0.25, 0.27, 0.23, 0.25, 0.24, 0.26, 0.24, 0.30, 0.23, 0.28, 0.25, 0.28, 0.27, 0.25, 0.27, 0.23, 0.22, 0.23, 0.21, 0.24, 0.28

Signal 4:

- Hurst Values: 0.41, 0.36, 0.41, 0.32, 0.39, 0.42, 0.39, 0.40, 0.40, 0.37, 0.33, 0.42, 0.37, 0.32, 0.34, 0.36, 0.41, 0.33, 0.42, 0.43, 0.41, 0.39, 0.40

Observations And Inference

Signal 1: The values of Signal 1 exhibit moderate variation across the sequence, with values generally ranging from 0.51 to 0.66. Notably, the highest values are concentrated around 0.60-0.62, while the values at the lower end remain between 0.51-0.55. This indicates a relatively stable signal with slight fluctuations.

Signal 2: Signal 2 shows relatively consistent values, mostly between 0.37 and 0.42, with occasional slight peaks at 0.41 and 0.42. The variation is minimal compared to Signal 1, which suggests this signal is more stable with fewer large fluctuations.

Signal 3: Signal 3 fluctuates more significantly, with values ranging from 0.20 to 0.30. There are higher frequencies of values between 0.22 and 0.26, with notable spikes at 0.29 and 0.30. The variability in this signal suggests it might have a more dynamic behavior than the first two signals.

Signal 4: Signal 4 displays values between 0.32 and 0.43, with a slightly higher concentration around 0.39 and 0.41. The overall pattern is similar to Signal 2, with small deviations indicating moderate fluctuations in the signal.