1. What is the EWMA Algorithm?

EWMA (Exponentially Weighted Moving Average) continuously calculates the average of a time series, giving more weight to recent data while gradually reducing the weight of older data. This allows us to quickly respond to changes in the data, unlike traditional averages that treat all data equally.

2. How Does EWMA Work?

The formula is:

EWMA□ = α * X□ + (1 - α) * EWMA□₋₁

- **X□**: The most recent data point.
- **EWMA□₋₁**: The previous EWMA (average of past data).
- **α**: The weight factor (0 ≤ α ≤ 1). A higher α gives more weight to recent data, while a lower α considers past data more.

3. Why Use EWMA for Anomaly Detection?

EWMA helps detect anomalies by comparing the current value with the calculated moving average. If the value deviates significantly (more than a threshold of standard deviations), it is flagged as an anomaly.

4. Key Parameters in EWMA:

- **Alpha (α)**: Determines the importance of new data. A higher α means faster reaction to changes, while a lower α smooths the data over time.
- **Threshold**: Defines how far from the average a data point must be to be considered an anomaly. A lower threshold catches smaller deviations, while a higher threshold only flags major changes.

5. Advantages and Disadvantages of EWMA:

- **Advantages**: Simple, fast, reduces noise, and adapts to real-time data changes.
- **Disadvantages**: Can be too sensitive (with a low α) or too slow to detect anomalies (with a high α).

Conclusion

EWMA is a practical algorithm for detecting anomalies in time series data. It adjusts to changes quickly and can be applied in fields like financial transactions or system monitoring to detect unusual patterns efficiently.

1. Anomaly Detection Function: detect_anomaly

This function detects unusual events in the data stream.

```
def detect_anomaly(data, alpha=0.3, threshold=3):
    if len(data) < 2:
        return False, None
```

First, it checks if there's enough data. If there are fewer than two data points, it can't detect anomalies, so it returns "no anomaly.

```
   try:
        ewma = data[0]
        ewma_values = []
        for point in data:
            ewma = alpha * point + (1 - alpha) * ewma
            ewma_values.append(ewma)
```

Here, we calculate the moving average (EWMA) by combining the new data with past averages. The parameter `alpha` controls how much weight is given to new data.

```
       last_value = data[-1]
        moving_avg = ewma_values[-2]
        std_dev = np.std(ewma_values)
```

We then compare the latest value with the previous moving average and calculate the standard deviation, which tells us how much the data varies.

```
        if abs(last_value - moving_avg) > threshold * std_dev:
            return True, moving_avg
        return False, moving_avg
```

If the difference between the latest value and the average exceeds a set threshold, it detects an anomaly. Otherwise, the data is considered normal.

Using Matplotlib, this function creates a graph showing the data stream, detected anomalies (red lines), and the moving average (dashed line).

We use a `deque` structure to store the latest 20 data points. Simulated data is generated, and anomalies are checked.

In the end;

```
if __name__ == "__main__":
    real_time_anomaly_detection()
```

When the script is run, it launches the real-time anomaly detection process.

SOURCE;

https://medium.com/aimonks/understanding-exponentially-weighted-moving-averages-ewma-cc8e78b01184

https://medium.com/codex/simple-moving-average-and-exponentially-weighted-moving-average-with-pandas-57d4a457d363

https://gregorygundersen.com/blog/2022/06/04/moving-averages/

Ferit Talha GUN

Jr. Software Developer