Pattern Matching in Time Series

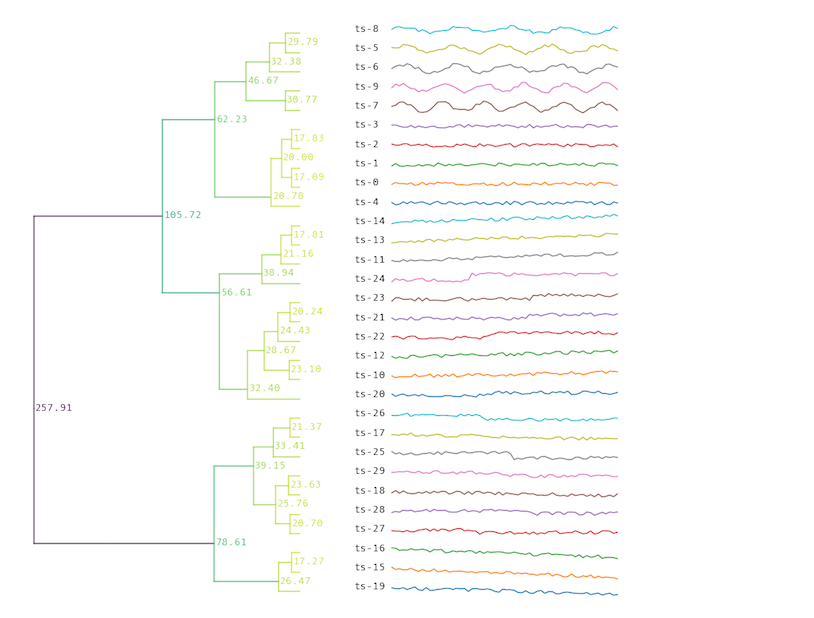
Problem statement: Find out the different pattern in time series data and show wherever it is repeating in data.

* **DTW (dynamic time wrapping):**
* Calculate distance between two time series using DTW by converting the series in vertical stack. Plot the result according to their pixel density.
* Calculate distance between two time series from distance formula and then apply DTW on it
* Steps to follow using DTW:
* Logic 1: Already discussed
* Step 1 : Using DTW we are getting the similarity of time series based upon minimum distance.
* Step 2: Applying Hierarchical Clustering on similarity output table.
* Logic 2: -
* Step 1 : Using DTW we are getting the similarity of time series based upon minimum distance.
* Step 2 : Decomposition of Time Series Forecasting P1 and P2 based upon seasonal & trend pattern
* Step 3 : Correlation of seasonal pattern of P1 and Seasonal Pattern of P2 should exist
* Step 4 : Correlation of trend pattern of P1 and trend Pattern of P2 should exist
* Step 5 : Clustering of Seasonal component of P1 and P2 based upon the seasonal index using SVM ( Radial Basis function) and k-Means Clustering
* Step 6 : Clustering of trend component of P1 and P2 based upon the seasonal index using SVM ( Radial Basis function) and k-Means Clustering
* Step 7 : From this we can come to a conclusion which week they are having a same seasonal or trend pattern for P1 and P2.

Results - not able to match the correct part, not able to detect all parts .

* **Hierarchical clustering:**

This can be used with DTW results. We get the results as follow

****

This technique won’t be suitable for our use case. As we want part of time series to be matched with other part of same or different time series.

* **Kmeans clustering to get two most similar series.**
* **Algorithm 1: Fuzzy clustering**

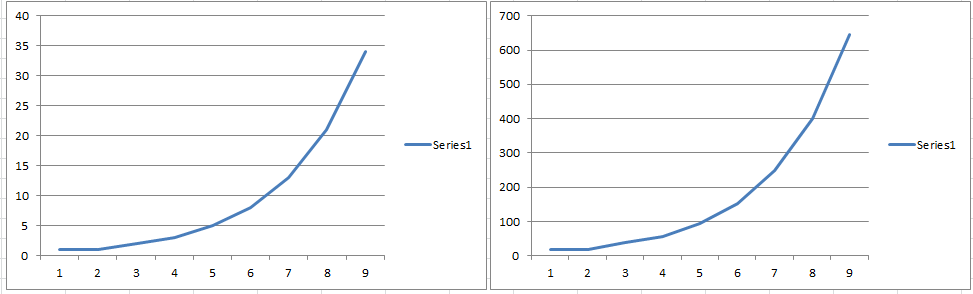
This is the clustering method in which we cluster similar points of data together and plot them in different color. We give two time series data of same length as input. Fuzzy cluster does the grouping of two time series. It assigns group number for each point. Two time series data points having same cluster number can be treated as similar data points.

Eg. There are two time series 19, 19, 38, 57, 95, 152, 247 , 399, 646 and 1 , 1, 2, 3, 5, 8, 13 , 21, 34 fuzzy clustering group these time series data as

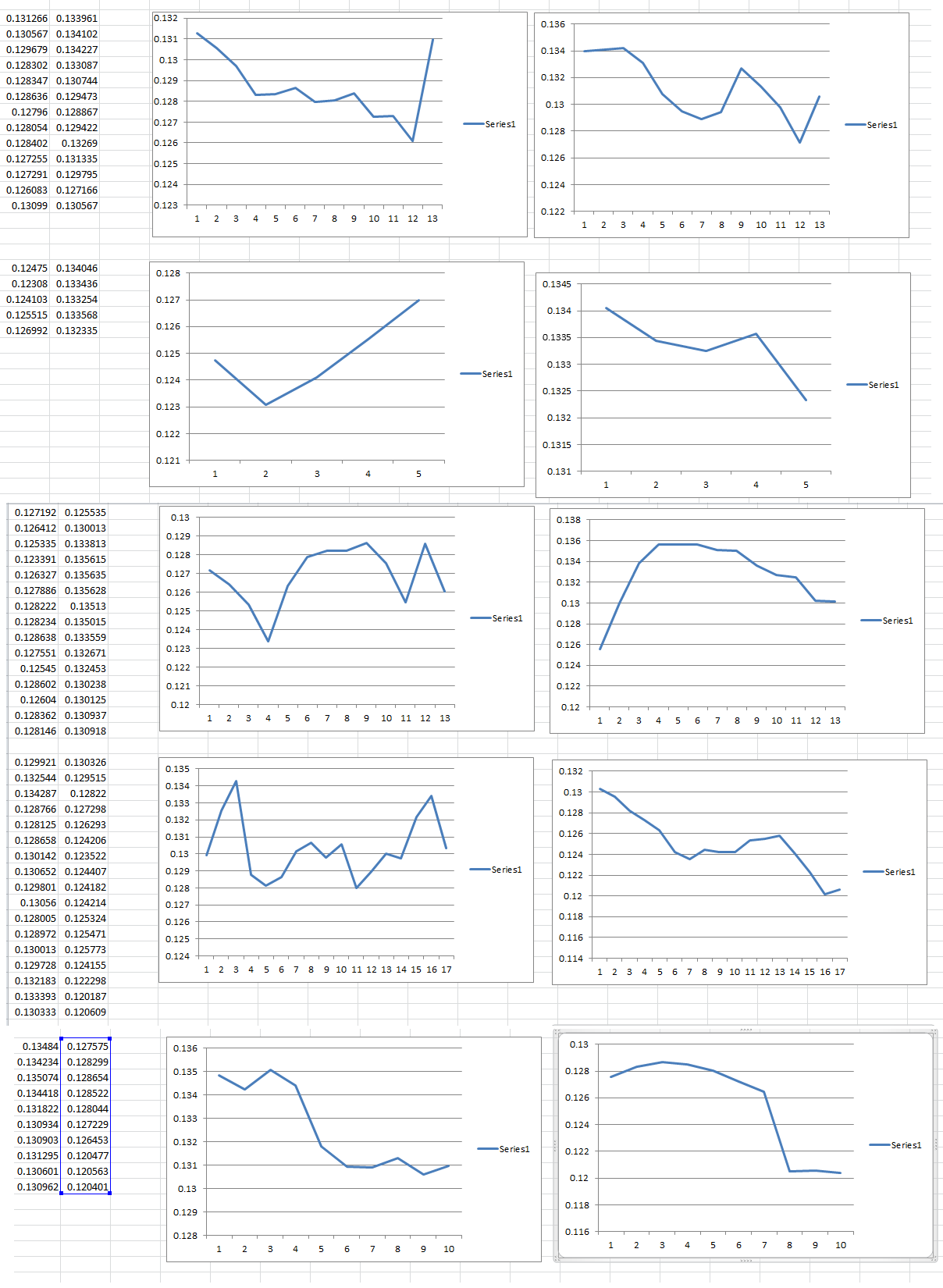
Cluster number: [1 1 1 1 1 1 1 0 0]

cluster 1 :  (array([399, 646]), array([21, 34]))

cluster 2 :  (array([ 19,  19,  38,  57,  95, 152, 247]), array([ 1,  1,  2,  3,  5,  8, 13]))



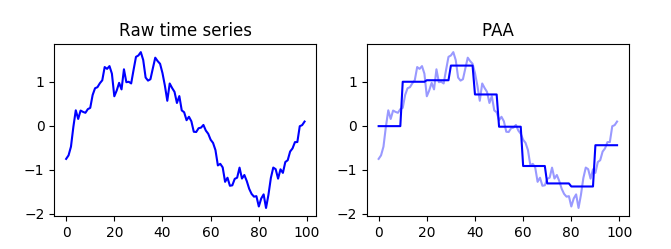
Drawback : it work well for similar data but doesn’t work well in un matched data. we can see in below image matched patterns are given. Some of the patterns are very slightly similar hence we cannot consider fuzzy clustering for our use case



* **Find elliot wave using rule based :**

Rules for Elliot wave are:

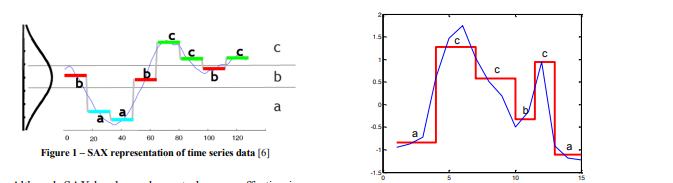
* Complete eight wave cycle has rules that govern its shape. It has been listed below.
* The motive phase is composed of five waves, three advancing (1, 3, 5) and two counter trend waves, 2 and 4.
* The corrective phase is composed of three waves, two receding (A and C) and one counter trend wave, B.
* Motive waves can head up or down.
* Corrective waves can head up or down.
* The motive phase aligns with the larger trend i.e Wave 3 is the longest wave
* The corrective phase is a counter trend move against the larger trend.
* Wave 2 never moves beyond the start of wave 1.
* Wave 3 is never the shortest wave.
* Wave 4 never overlaps wave 1.
* This is an observation: one of the waves, 1, 3, or 5, will often (but not always) be much longer (extended) than the other two.
* We haven’t considered rule based method to find the patter because there was no ML used in it.
* **SAX and PAA:**
* These are two methods used together to find the pattern in time series.
* At first the time series data is represented in simpler form using PAA. This will reduce the huge data into smaller representation and easier to use further.
* We take one input parameter say w(as window size) which is the number of points in which you want to divide your data.
* Based on that value we divide the data and then we calculate the average for each group and that average value will represent that group

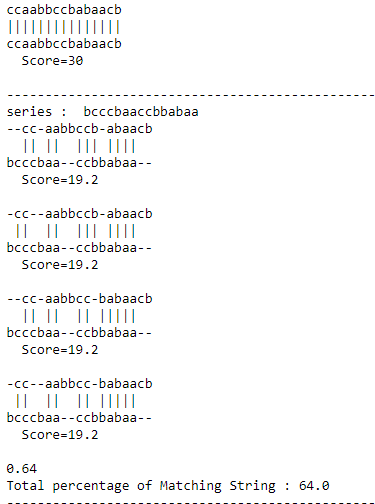


* Result or PAA will be taken as input for SAX. In this method we divide the data points into number of groups any m. Each group will be then represented by one letter .
* SAX use z normalize to divide the data points into different groups.
* Using elbow method we can decide the number of clusters to be used.

Using moving average we can calculate the error rate for each window size and window size with less error rat can be then used as window size for PAA.

Bio pairwise match for finding the pattern and matching same pattern in another series



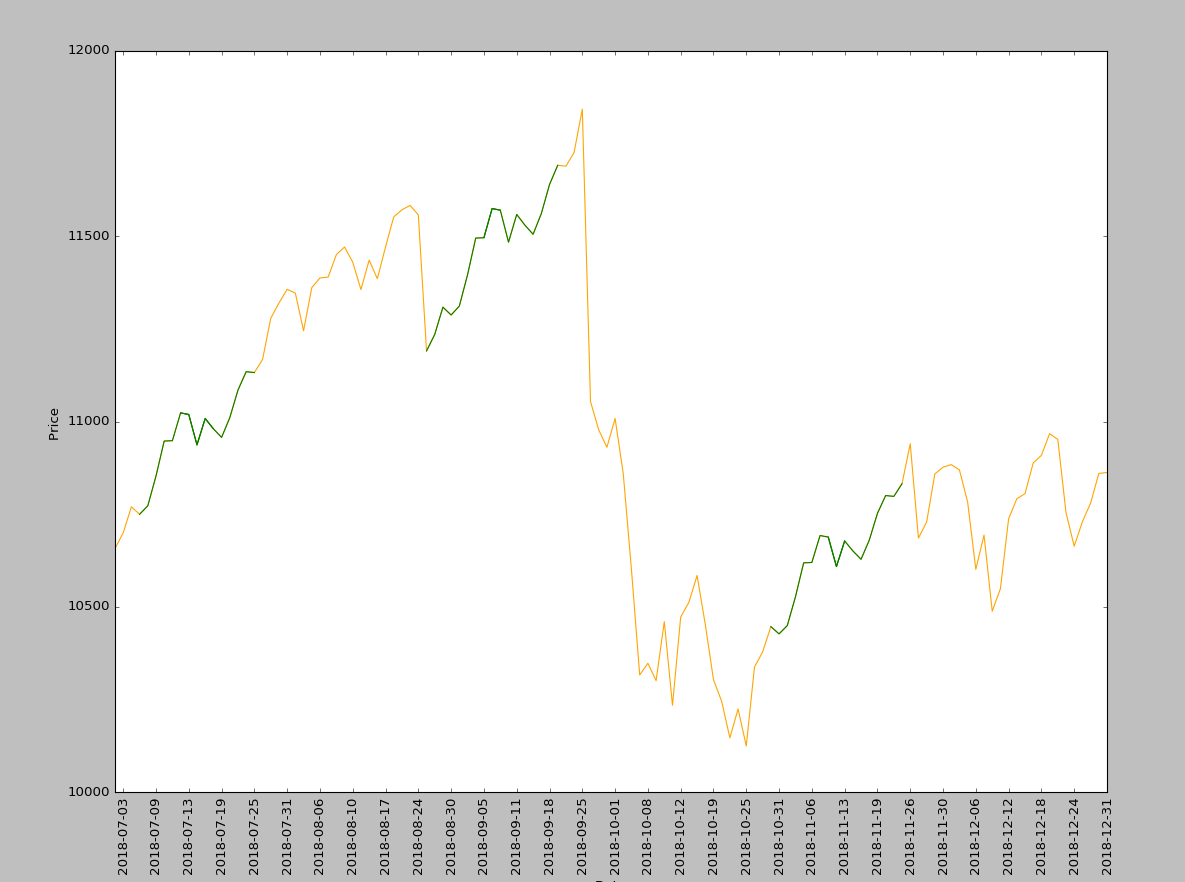
* After converting the data into alphabetic wave form we can then match the pattern using bio pairwise library. It gives you the output as 
* As there are more patterns within the alphabetic representation of time series and there is manipulation (converting data using PAA) in original data and we want to match the pattern/ subpart of data this algorithm is not suitable.
* **Matrix Profile :**

In this method we partition the data of size w (window size), then we compare each partition with all the other partitions.

We get the matrix profile value for every pair based on distance between them. for each pattern from all pairs, only the smallest matrix profile value will be considered. And the location of that matching pattern will be stored.

If we plot the matrix profile values, the largest value can be considered as anomaly in data.

If we plot the matching pattern it will show as below



* **TS fresh:**

Links to refer :

DTW : <https://nipunbatra.github.io/blog/2014/dtw.html>