Study and implementation of clustering algorithms in time series. Application and testing of methods on economic data.

University of Crete
Department of Applied Mathematics

Author : Aikaterini Lefkaditi
Supervisor : Ioannis Pantazis

July 23, 2021

Abstract

The spread of devices that produce large amounts of data requires new improved clustering algorithms by Computer Science. This data needs to be organized into compact structures, so that it is easy to use and requires less storage space. An approach to the solution of this classification and optimization problem is the Warped K-Means (WKM) method, a clustering algorithm of sequentially-distributed data, which is based on the well-known K-Means Algorithm and is focused on solving its sequential originated problem. The main objective of this project is to extend WKM, in order to include piecewise linear functions as clusters.

1 Introduction

Clustering is the task of grouping a set of objects in such a way that objects in the same group, which is called cluster, are more similar to each other, than to those in other clusters. It is an important technique for statistical data analysis, used in many fields such as pattern recognition, information retrieval and machine learning. Cluster analysis is an iterative multi-objective optimization. This is the reason that clustering techniques continue to evolve.

Warped K-Means [1] is applied on sequentially-distributed data, time series, tracking subtrajectories within a single trajectory, aiming to get a simplified data structure, while preserving the data sequetentiality. This algorithm uses a closed-form solution having a low computational cost, which provides really fast convergence times and high accuracy.

As we noticed while applying WKM source code on sequentially-distributed time series , WKM creates straight horizontal lines to describe every cluster. Nevertheless, the residual error can be even more reduced, if it could use a linear form instead of these lines, in cases that it is considered beneficial. Therefore, the main purpose of this project is to create an extended version of the algorithm in order to attain that.

During the first months, it was necessary to fully comprehend the method and how it works. For this reason, a few simple examples were created and then their results were optimized. Subsequently, we applied the same method on real time series.

This paper is organized as follows. First, we describe the clustering method of the original WKM and the mathematical background that it is used for. Secondly, we demonstrate the new extended version of WKM and the final algorithm in the form of pseudocode and we highlight the changes that we made. In the end, we apply the two methods to a few simple examples that we created in order to test them and finally to real data, economic time series, and compare them to summarise and demonstrate the results.

2 WKM - Original Version

In this section, we initialize the equations that are used by the algorithm and explain in detail how they are combined in order to minimize the Sum of Quadratic Error. After that, we explain the methodology and present the algorithm in pseudocode.

2.1 Background

Partitional clustering divides a dataset $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ of a n-dimensional features vectors into a set $\Pi = \{\boldsymbol{c}_1, ..., \boldsymbol{c}_k\}$ of k disjoint homogeneous classes with $1 < k \ll n$. One way to tackle this problem is to define a criterion function that measures the quality of the clustering partition and then find a partition Π that optimizes such a criterion function. For finding the partition WKM tries to minimize the Sum Of Quadratic Error(SQE), denoted simple as J:

$$J = \sum_{j=1}^{k} H_j \tag{1}$$

where

$$H_j = \sum_{i=b_j}^{b_{j+1}-1} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 \tag{2}$$

represents the heterogeneity (or distortion) of cluster c_j, and

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{i=b_j}^{b_{j+1}-1} \boldsymbol{x}_i \tag{3}$$

is the cluster mean, with $n_j = \|c_j\|$ being the number of samples in cluster j and having a dataset X sequentially distributed:

$$X = \boldsymbol{x}_1, ..., \boldsymbol{x}_n \tag{4}$$

Defined a sequential clustering into k classes as the mapping:

$$b : \{1, ..., k\} \hookrightarrow \{1, ..., n\}$$

Then

$$c_j = \{\boldsymbol{x}_{b_j}, ..., \boldsymbol{x}_{b_j+n_j-1}\} \tag{5}$$

and

$$n_j = b_{j+1} - b_j \tag{6}$$

It also counted the variation in the SQE produced when moving a sample $\boldsymbol{x}$ from cluster j to cluster j' as:

$$\Delta J(x, i, j, j') = \frac{n_j}{n_j + 1} \|\boldsymbol{x} - \boldsymbol{\mu}_j\|^2 - \frac{n_{j'}}{n_{j'} - 1} \|\boldsymbol{x} - \boldsymbol{\mu}_{j'}\|^2 \tag{7}$$

If this increment is negative, the new means, $\mu_j^{(new)}$, $\mu_{j'}^{(new)}$ and the SQE, J', can then be incrementally computed as follows:

$$\boldsymbol{\mu}_j^{(new)} = \boldsymbol{\mu_j} + \frac{\boldsymbol{x} - \boldsymbol{\mu}_j}{n_j + 1} \tag{8}$$

$$\boldsymbol{\mu}_{j'}^{(new)} = \boldsymbol{\mu}_{j'} - \frac{\boldsymbol{x} - \boldsymbol{\mu}_{j'}}{n_{j'} - 1} \tag{9}$$

$$J' = J + \Delta J(\boldsymbol{x}, j, j') \tag{10}$$

2.2 Methodology

First of all, we use Trace Segmentation (TS), a boundary initialization algorithm, which uses the trajectory of data and a number of clusters as input and defines an initial partition. By this technique, each boundary is evenly allocated according to a piecewise linear interpolation on accumulated distances, resulting in a non-linearly distributed boundary allocation.

Following this, we use these boundaries in the second algorithm, Warped K-Means (WKM), which is the main algorithm. The first half of samples in cluster j are only allowed to move to cluster j - 1 and the last half of samples are only allowed to move to cluster j + 1. These reallocations are happening only in case SQE has a negative increment and the process is done when no more transfers are performed.

As a result, only the samples close to the cluster boundaries get reallocated, the sequentiality of the clustering procedure is preserved and there is no need to check every sample of each cluster. The computational cost of the algorithm is O(nd), where n is the number of the samples and d is the sample vector dimension. On the following page, we present the complete form of the two algorithms in pseudocode.

Algorithm1. *TS Boundary Initialization Algorithm*

Input : Trajectory $X = x_1, ...x_n; No.Clusters k \geq 2$
Output : $Boundaries\ b_1, ...b_k$
  $L_1 = 0$
    for $i = 2$ **to** n **do**
      $L_i = L_{i-1} + ||\boldsymbol{x}_i - \boldsymbol{x}_{i-1}||$
    $\lambda = \frac{L_n}{k}$
    $i = 1$
    for $j = 1$ **to** k **do**
      while $\lambda(j-1) \geq L_i$ **do**
        $i++$
      $b_j = i$

Algorithm2. $Warped\ K - Means\ (Original)$

Input : $Trajectory\ X;\ No.\ Clusters\ k \geq 2$
Onput : $Boundaries\ b_1, ...b_k;\ Centroids\ \mu_1, ..., \mu_k;\ Distortion\ J$
  $Initialize\ Boundaries\ b_1, ...b_k$
    for $j = 1$ **to** k **do**
      $Compute\ \mu_j,\ n_j,\ J$
      repeat
      $transfer = false$
      for $j = 1$ **to** k **do**
        if $j > 1$ **then**
          $first = b_j; last = first + \frac{n_j}{2}(1 - \delta)$
          for $i = first$ **to** $last$ **do**
            if $n_j > 1$ **and** $\Delta J(x_i, j, j-1) < 0$ **then**
              $transfers = true$
              $b_j += 1;\ n_j -= 1;\ n_{j-1} += 1$
              $Update\ \mu_j,\ \mu_{j-1},\ J$
            else
              break
        if $j < k$ **then**
          $last = b_{j+1} - 1;\ first = last - \frac{n_j}{2}(1 - \delta)$
          for $i = last$ **to** $first$ **do**
            if $n_j > 1$ **and** $\Delta J(x_i, j, j+1) < 0$ **then**
              $transfers = true$
              $b_{j+1} -= 1;\ n_j -= 1;\ n_{j+1} += 1$
              $Update\ \mu_j,\ \mu_{j+1},\ J$
            else
              break
    until $\neg transfers$

---

3 WKM - Extended Version

In order to include piecewise linear functions as clusters, it was necessary to solve the new differential equations which were involved in the slopes. Having checked one by one the equations, it was decided which of them should be changed. The next step was to check them by using a few simple examples to confirm that everything was right and insert all these equations in the algorithm. Once again, using a few simple examples confirmed that the new equations took into account the slopes and the results were verified as expected.

3.1 New equations

According to the previous equations the SQE counted as:

$$J = \sum_{j=1}^{k} H_j \tag{11}$$

where

$$H_j = \sum_{i=b_j}^{b_{j+1}-1} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j - \boldsymbol{\alpha}_j(i - \boldsymbol{m}_j)\|^2 \tag{12}$$

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{i=b_j}^{b_{j+1}-1} \boldsymbol{x}_i \tag{13}$$

$$\boldsymbol{a}_j = \frac{12}{n_j(n_j-1)(n_j+1)} \sum_{i=b_j}^{b_{j+1}-1} (i - m_j)\boldsymbol{x}_i \tag{14}$$

$$\boldsymbol{m}_j = \frac{1}{n_j} \sum_{i=b_j}^{b_{j+1}-1} i \tag{15}$$

$$n_j = b_{j+1} - b_j \tag{16}$$

Finally, Mean Squared Error (MSE) measures the average of the squares of errors, it is used to emphasize the error regardless of the sample's length and it is counted as :

$$\hat{J} = \frac{J}{\text{n}} \tag{17}$$

As before μ_j is the cluster mean, a_j is the slope and m_j is the center of the points' location inside the cluster.

The new variation in the SQE is:

$$\Delta J(x, j, j') = H_j^{(new)} + H_{j'}^{(new)} - H_j^{(old)} - H_{j'}^{(old)} \tag{18}$$

Where $J^{(new)}$ is the SQE in case we make a transfer. At first, we update the boundaries $b_j^{(new)}$, $b_{j'}^{(new)}$ and the length of the clusters $n_j^{(new)}$, $n_{j'}^{(new)}$ and then compute $\mu_j^{(new)}$, $\mu_{j'}^{(new)}$, $m_j^{(new)}$,$m_{j'}^{(new)}$,$a_j^{(new)}$,$a_{j'}^{(new)}$ in this order and the $J^{(new)}$, can then be incrementally computed as follows:

$$b_j^{(new)} = b_j - 1 \tag{19}$$

$$b_{j'}^{(new)} = b_{j'} + 1 \tag{20}$$

$$n_j^{(new)} = n_j - 1 \tag{21}$$

$$n_{j'}^{(new)} = n_{j'} + 1 \tag{22}$$

$$\boldsymbol{\mu}_j^{(new)} = \boldsymbol{\mu}_{\boldsymbol{j}}^{(old)} - \frac{\boldsymbol{x} - \boldsymbol{\mu}_j^{(old)}}{n_j^{(new)}} \tag{23}$$

$$\boldsymbol{\mu}_{j'}^{(new)} = \boldsymbol{\mu}_{j'}^{(old)} + \frac{\boldsymbol{x} - \boldsymbol{\mu}_{j'}^{(old)}}{n_{j'}^{(new)}} \tag{24}$$

$$\boldsymbol{m}_{\boldsymbol{j}}^{(new)} = \frac{1}{n_j^{(new)}} \sum_{i=b_j}^{b_{j+1}^{(new)}-1} i$$

$$= \frac{n_j^{(new)} + 1}{n_j^{(new)}} \boldsymbol{m}_j^{(old)} - \frac{b_{j+1}}{n_j^{(new)}} \tag{25}$$

$$\boldsymbol{m}_{j'}^{(new)} = \frac{1}{n_j^{(new)}} \sum_{i=b_{j'}}^{b_{j'+1}^{(new)}-1} i$$

$$= \frac{n_{j'}^{(new)} - 1}{n_{j'}^{(new)}} \boldsymbol{m}_{j'}^{(old)} + \frac{b_{j'}}{n_{j'}^{(new)}} \tag{26}$$

$$\boldsymbol{\alpha}_j^{(new)} = \frac{12}{n_j^{(new)}(n_j^{(new)} + 1)(n_j^{(new)} + 2)} \sum_{i=b_j^{(new)}}^{b_{j+1}^{(new)}-1} (i - \boldsymbol{m}_j^{(new)})\boldsymbol{x}_i \tag{27}$$

$$\boldsymbol{\alpha}_{j'}^{(new)} = \frac{12}{n_{j'}^{(new)}(n_{j'}^{(new)} + 1)(n_{j'}^{(new)} + 2)} \sum_{i=b_{j'}^{(new)}}^{b_{j'+1}^{(new)}-1} (i - \boldsymbol{m}_{j'}^{(new)})\boldsymbol{x}_i \tag{28}$$

$$J^{(new)} = J + \Delta J(\boldsymbol{x}, j, j') \tag{29}$$

3.2 The New Algorithm

The following part is the extended version of the Warped K-Means algorithm, in the form of pseudocode. There are a few changes in both Algorithm 1 and 2 (for example the existence of the parameters α and m), based on the new equations. In this project, we use the parameter δ as 0.

***Algorithm*1.** *TS Boundary Initialization Algorithm*

***Input* :** Trajectory $X = x_1, ...x_n$; $No.Clusters k \geq 2$
***Output* :** $Boundaries\ b_1, ...b_k$

$L_1 = 0$
for $i = 2$ **to** n **do**
  $L_i = L_{i-1} + ||\boldsymbol{x}_i + \boldsymbol{x}_{i-2} - 2x_{i-1}||$
$\lambda = \frac{L_n}{k}$
$i = 1$
for $j = 1$ **to** k **do**
  while $\lambda(j-1) \geq L_i$ **do**
    $i++$
  $b_j = i$

***Algorithm*2.** $Warped\ K - Means\ (Extended)$

***Input* :** $Trajectory\ X$; $No.\ Clusters\ k \geq 2$
***Onput* :** $Boundaries\ b_1, ...b_k$; $Centroids\ \mu_1, ..., \mu_k$; $Distortion\ J$

$Initialize\ Boundaries\ b_1, ...b_k$
for $j = 1$ **to** k **do**
  $Compute\ \mu_j,\ n_j,\ J\ ,\ m_j,\ a_j$
  repeat
  $transfer = false$
  for $j = 1$ **to** k **do**
    if $j > 1$ **then**
      $first = b_j$; $last = first + \frac{n_j}{2}(1-\delta)$
      for $i = first$ **to** $last$ **do**
        if $n_j > 1$ **and** $\Delta J(x_i, j, j-1) < 0$ **then**
          $transfers = true$
          $b_j += 1$; $n_j -= 1$; $n_{j-1} += 1$
          $Update\ \mu_j,\ \mu_{j-1},\ J,\ \alpha_j,\ \alpha_{j-1},\ m_j,\ m_{j-1}$
        else
          break
    if $j < k$ **then**
      $last = b_{j+1} - 1$; $first = last - \frac{n_j}{2}(1-\delta)$
      for $i = last$ **to** $first$ **do**
        if $n_j > 1$ **and** $\Delta J(x_i, j, j+1) < 0$ **then**
          $transfers = true$
          $b_{j+1} -= 1$; $n_j -= 1$; $n_{j+1} += 1$
          $Update\ \mu_j,\ \mu_{j+1},\ J,\ \alpha_j,\ \alpha_{j+1},\ m_j,\ m_{j+1}$
        else
          break
until $\neg transfers$

---

4 Examples - Results

4.1 Synthetic time series

Both the Original and the Extended algorithm of WKM were executed (in Python) by using simple examples to confirm that they are correct. The following part of this subsection is a presentation of these examples. Moreover, under every figure, the Sum of Quadratic Error (SQE) and the Mean Squared Error (MSE) are noted, in order to compare the efficiency of the algorithm. The colorful lines symbolize the samples of each cluster and the red dashed lines symbolize the mean value of it. In this way, we observe how far the elements are from the cluster's mean value, which also creates the error.

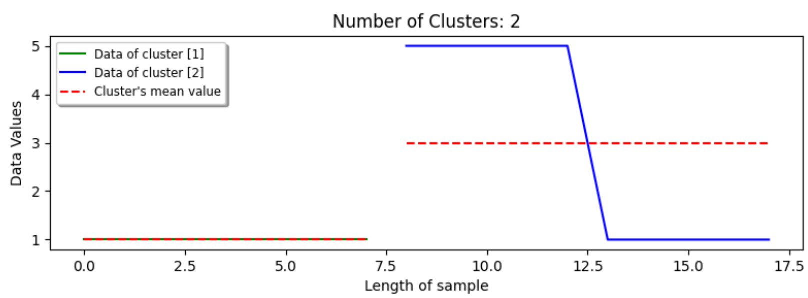


Figure 1: Piecewise constant time series consists of 3 different parts. Applying the Original WKM with 2 clusters, the results were SQE = 40.0 and MSE = 2.22 . An important observation is that the first cluster consists of samples that have the same value, which is also the mean value, therefore the lines are mixed.

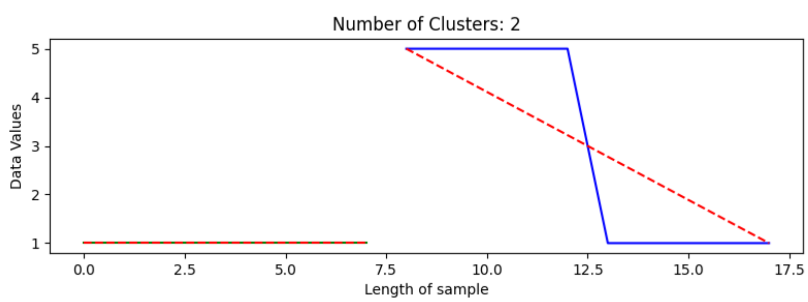


Figure 2: Applying the Extended WKM with 2 clusters on piecewise constant time series, the results were SQE = 9.69 and MSE = 0.55 . In the second cluster, we notice that the mean value appears with slope, in contrast with the first cluster, where the slope is equal to 0 and the lines are mixed once again.

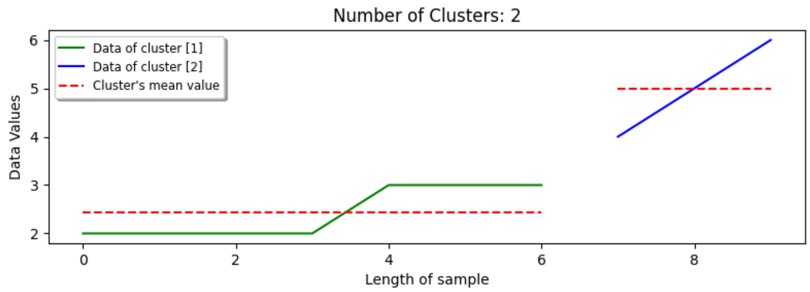


Figure 3: Piecewise linear time series consists of 3 linear parts. Applying the Original WKM with 2 clusters, the results were SQE = 3.71 and MSE = 0.37 .

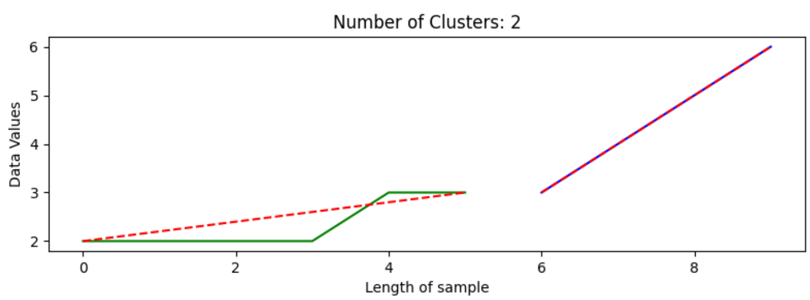


Figure 4: Applying the Extended WKM with 2 clusters on the piecewise linear time series, the results were SQE = 0.41 and MSE = 0.04 and the second cluster's mean value follows the line of the samples as they are increasing at a steady rate.

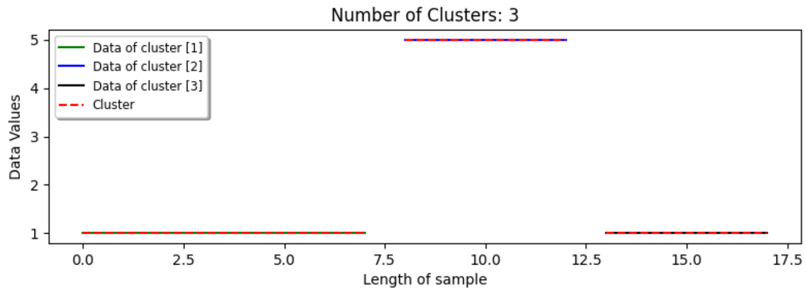


Figure 5: In contrast with Figure 1, applying the Original WKM using 3 clusters, on the piecewise constant time series, which consists of 3 parts, the results were SQE = 0, MSE = 0.

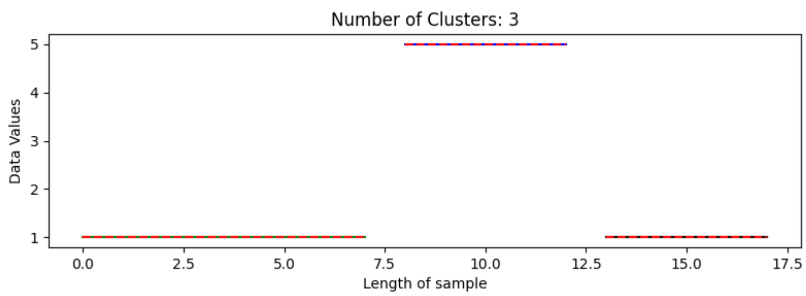


Figure 6: Applying the Extended WKM using 3 clusters, on the piecewise constant time series, which consists of 3 parts, the results were SQE = 0, MSE = 0, similar to the previous figure.

In Figures 1 and 2 we noticed that the intervals are the same, although the SQE and MSE are significantly reduced in the case of the extended WKM. After that, according to the Figures 3 and 4, the two algorithms create different intervals, and once more the extended version is more effective in minimizing the errors. The last two Figures (7 and 8) present the best case, where the samples follow precisely the mean value's line.

4.2 Economic time series

The same process was used for real data, economic series. In Figures 7 and 8 we present the Effective Federal Funds Rate (FEDFUNDS)[1],which is the interest rate at which depository institutions trade federal funds with each other. Afterwards, we present the 10-Year Treasury Constant Maturity Rate (DGS10)[2].

Similar to the figures of the previous paragraph, the colorful lines symbolize the samples of each cluster and the red dashed lines symbolize the mean values. As expected, the extended version minimizes error more effectively than the original version.

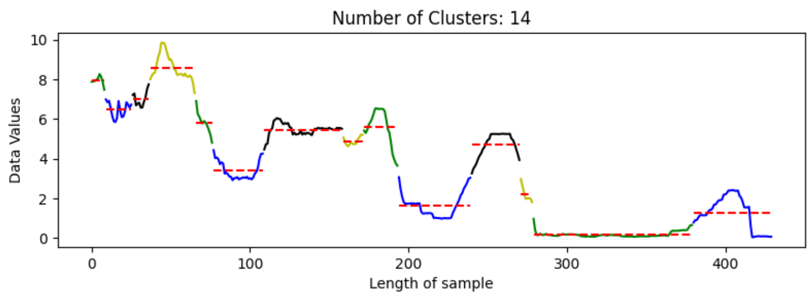


Figure 7: Applying the Original WKM using 14 clusters on FEDFUNDS, the results were SQE = 116.88 and MSE = 0.27 .

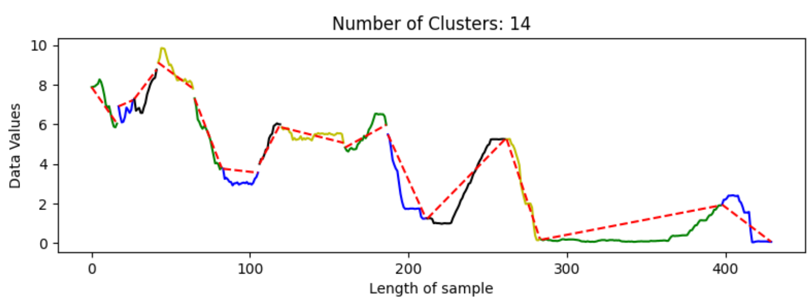


Figure 8: Applying the Extended WKM using 14 clusters on FEDFUNDS, the results were SQE = 10.23 and MSE = 0.02 .

[1] https://fred.stlouisfed.org/series/FEDFUNDS
[2] https://fred.stlouisfed.org/series/DGS10

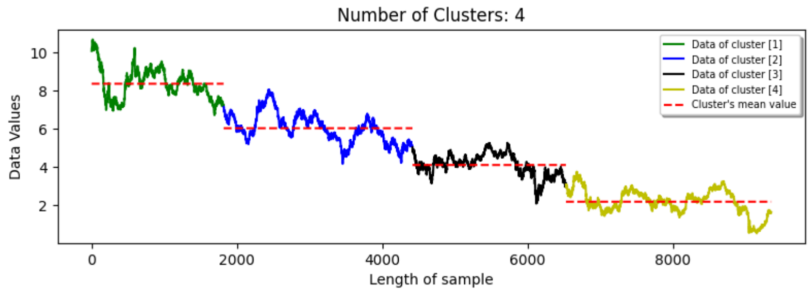


Figure 9: Applying the Original WKM using 4 clusters on DGS, the results were SQE = 4679.64 and MSE = 0.5 .

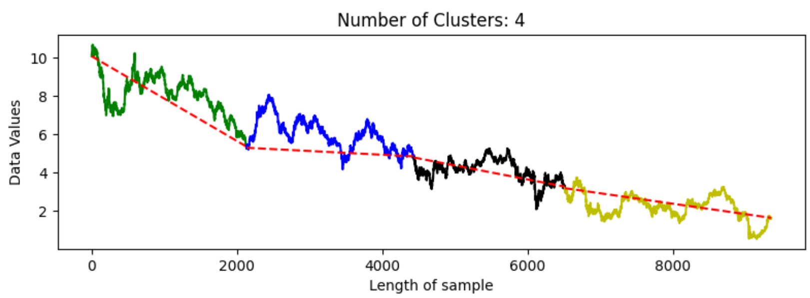
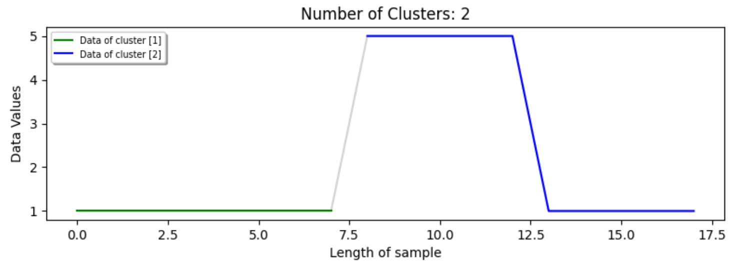


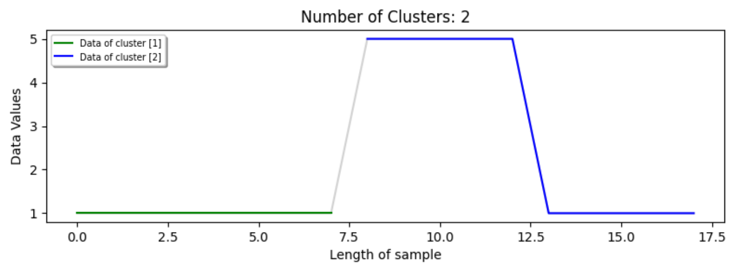
Figure 10: Applying the Extended WKM using 4 clusters on DGS, the results were with SQE = 2765.65 and MSE = 0.29 .

4.3 Comparison between initial and final clustering

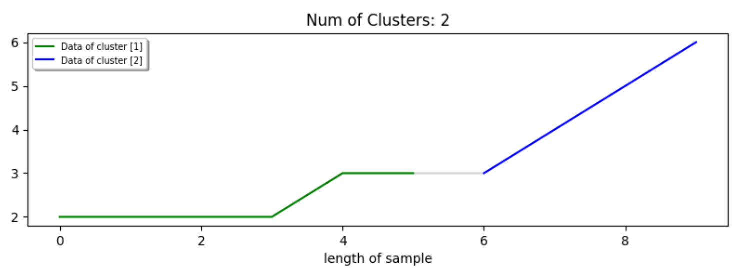
As it is explained in the previous paragraphs, WKM uses Trace Segmentation Algorithm (TS), to initialize the clusters' boundaries. The following graphs emphasize the effectiveness of WKM, compared to TS, as clustering methods. According to the examples, TS works sufficiently as there is a simple case (Figures 11a and 11b) where the clusters' boundaries remained the same after the application of WKM and there was no need of transferring samples between the clusters. Contrariwise, as we can see in the rest of the Figures (12a, 12b, 13a, 13b), where the time series are larger and in a linear form, there are many changes between the final clusters.



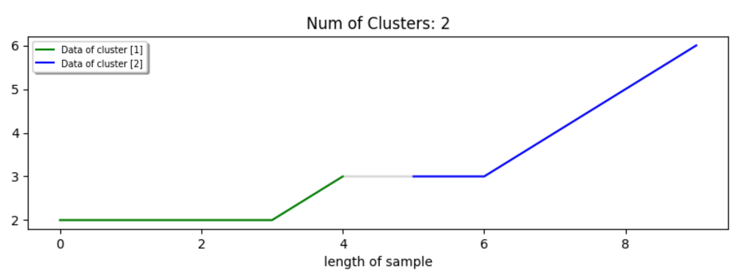
(a) Piecewise constant time series plotted in two clusters according to the extended version of WKM.



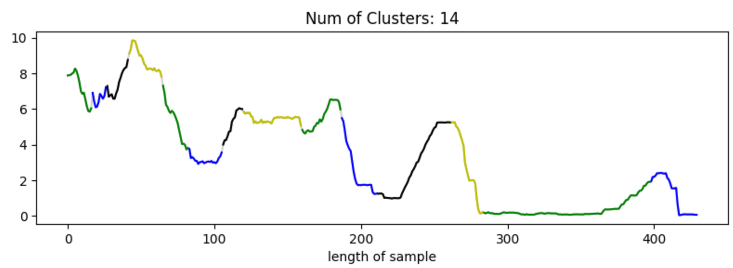
(b) Piecewise constant time series plotted in two clusters according to the (Extended) TS Boundary Initialization Algorithm.



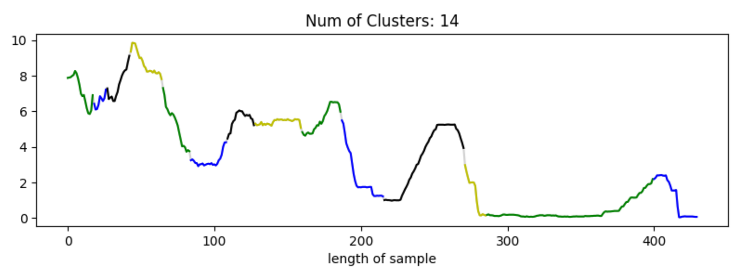
(a) Piecewise linear time series plotted in two clusters according to the extended version of WKM.



(b) Piecewise linear time series plotted in two clusters according to the (Extended) TS Boundary Initialization Algorithm.



(a) FEDFUNDS plotted in 14 clusters according to the extended version of WKM.



(b) FEDFUNDS plotted in 14 clusters according to the (Extended) TS Boundary Initialization Algorithm.

5 Conclusion

Warped K-Means is a significant algorithm between other clustering methods on sequentially distributed data. As it was proved through various examples, adding slopes to the Warped K-Means method was constructive, as the linear cases were successfully organized in clusters, while the Sum of Quadratic Error was notably reduced, which was the aim of this project.

References

[1] [LLV13] Leiva, Luis A., and Enrique Vidal. "Warped k-means: An algorithm to cluster sequentially-distributed data." Information Sciences 237 (2013): 196-210.