Time Series Clustering

CS4801 - Machine Learning Project Report

Arko Sharma

10.11.2018

INTRODUCTION

Time series are an ordered sequence of values of a variable at equally spaced time intervals. Time series analysis is often associated with the discovery and use of patterns (such as periodicity, seasonality, or cycles), and prediction of future values (specifically termed forecasting in the time series context). One key difference between time series analysis and time series data mining is the large number of series involved in time series data mining. As shown in a vast volume of time series literature, traditional time series analysis and modeling tend to be based on non-automatic and trial-and-error approaches. It is very difficult to develop time series models using a non-automatic approach when a large number of time series are involved.

The general theme in a time series analysis problem is to determine the value of a parameter over time and understanding the pattern with which it changes. The two important classes of such analysis are whole analysis and subsequence analysis. Whole clustering methods are used to determine how sequences are related to each other. Subsequence analysis on the other hand finds out how a particular sequence evolves over time and uses this data for prediction and control tasks.

In practical scenarios, a combination of both methods are used. Applications are manifold and find lot of similarities with streaming data applications :

- Sensors in transportation vehicles, industrial equipment, and farm machinery send data to a streaming application. The application monitors performance, detects any potential defects in advance, and places a spare part order automatically preventing equipment down time.
- A financial institution tracks changes in the stock market in real time, computes value-at-risk, and automatically rebalances portfolios based on stock price movements.
- A real-estate website tracks a subset of data from consumers' mobile devices and makes real-time property recommendations of properties to visit based on their geo-location.

Here we deal with whole - clustering , ie given a set of control chart series , we perform a clustering task to group sequences which are of the same trend in mathematical terms.

The Dataset

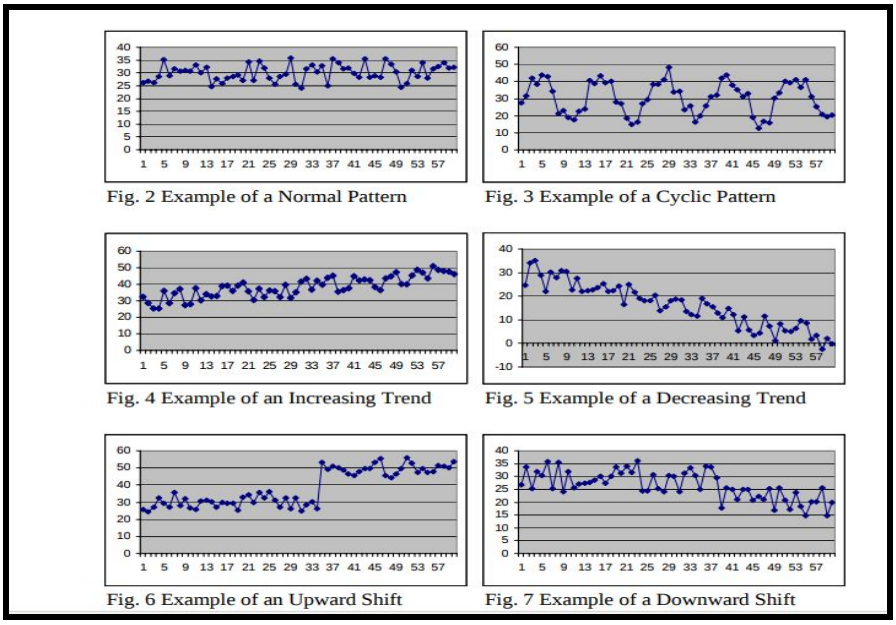


Figure showing the types of sequences of the control-chart data.

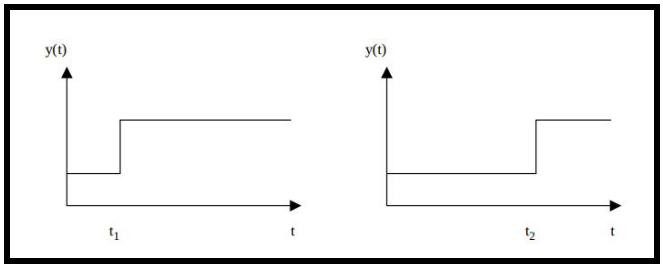
The dataset is a set of 600 time-series sequences indexed by time from 1 to 60. Since we are dealing with whole-clustering , we have 600 samples and each of them may be thought of as having 60 features which are the value of the y - parameter for discrete instances of time from 1 to 60. So the number of features is 60.

The series need to be grouped according to the mathematical formulae. The geometric plots of the formulae can be seen in the above plot.

Clustering Task

The task at hand is to group similar sequences together based on their mathematical formulae. It is quite different from a normal clustering task in the sense that there is a

dependence on both previous and forthcoming values of the feature vectors. In other words, we need to know the entire sequence to get an idea of the type of sequence it is, rather than the values of the features themselves. Only then would we able to establish a viable pattern for the entire series. Also, a subtlety is that the nature of the series is temporal - so any simple transformation along the time axis such as shifting and scaling needs to be accounted for. The following figure demonstrates this .



Here we can see that the two series are quite different if we consider them in absolute values of the y - parameter. But actually they are both samples of an upward shift, where the shift itself is occurring at later point in time in the second case.

It is only becoming clear because we have a visual verification to our aid. It is difficult to incorporate these types of intuitions into a machine learning algorithm because most of them are dependent on the values of the feature vectors or some transformation of these vectors . In this simple case, the transformation may seem trivial to implement but it also raises the question that what types of transformations should be used for improving the accuracy while not grouping dissimilar sequences together. Here, if we use an upward transformation, then our algorithm will be confused between increasing sequences and upward shift sequences. So we also need to take into account some sort of normalization that uses the entire range of the data values.

The Models

For performing the clustering task, the foremost measure that comes to mind is the euclidean distance. The most general purpose K - means algorithm can be applied by taking random means as initial values and computing the similarities using the euclidean distance. It seems suitable because there are only 6 clusters and it could be really efficient. The problem with K - means though is quite logical - the dependence on the initial means is quite strong and we lose some sensitiveness while calculating the means

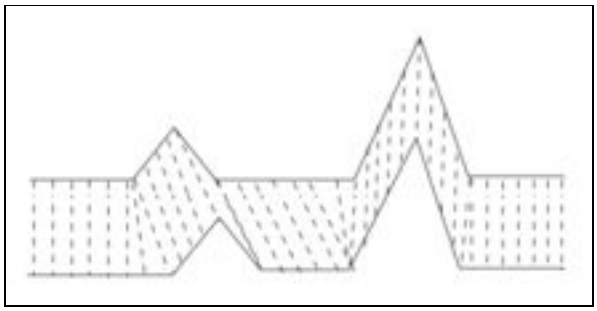
- which could imply the failure in noticing apparent trends in the patterns. If we have really varied shifts and breaks in the same time - series samples, then K-means may treat these as outliers and neglect their effect on the trends.

So, we need a model in which the order of the data needs to be important and the variations in the trends need to be accounted for. Hierarchical clustering with a proper distance measure seems appropriate for this task as the structure in the data is more important (grouping in global classes) . Since in applications of streaming data such as bioinformatics and financial investment decisions, the overall similarities between different trends play an important role , these similarities can be identified by looking at the hierarchical structure of the data and even the overall paradigms of the patterns can be approximated for prediction. Being independent of initial conditions is a welcome change from K - means but the tradeoff lies in complexity. For this task we have relatively less number of samples and hence we can try out agglomerative clustering after we decide the similarity measure.

Similarity Measure

The use of euclidean distance and other norm - based distances poses a problem to time series analysis problems because of neglecting the temporal nature of the sequences. Generally a suitable transformation is required to look at the time - based trends of evolving data. Also since the applications of time series analysis are often used in real world scenario, a dynamic notion of similarity is required whereas norm - based methods look at only the current values of the coordinates and are therefore static in nature.

Dynamic time warping



In time series analysis, dynamic time warping (DTW) is one of the algorithms for measuring similarity between two temporal sequences, which may vary in speed. For instance, similarities in walking could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data — indeed, any data that can be turned into a linear sequence can be analyzed with DTW. A well known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. Also it is seen that it can be used in partial shape matching application.

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restriction and rules:

- **Every index from the first sequence must be matched with one or more indices from the other sequence, and vice versa**
- **The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match)**
- **The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match)**
- **The mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing, and vice versa .he optimal match is denoted by the match that satisfies all the restrictions and the rules and that has the minimal cost, where the cost is computed as the sum of absolute differences, for each matched pair of indices, between their values.**

The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification.

These restrictions make DTW appropriate for our clustering problem for agglomerative

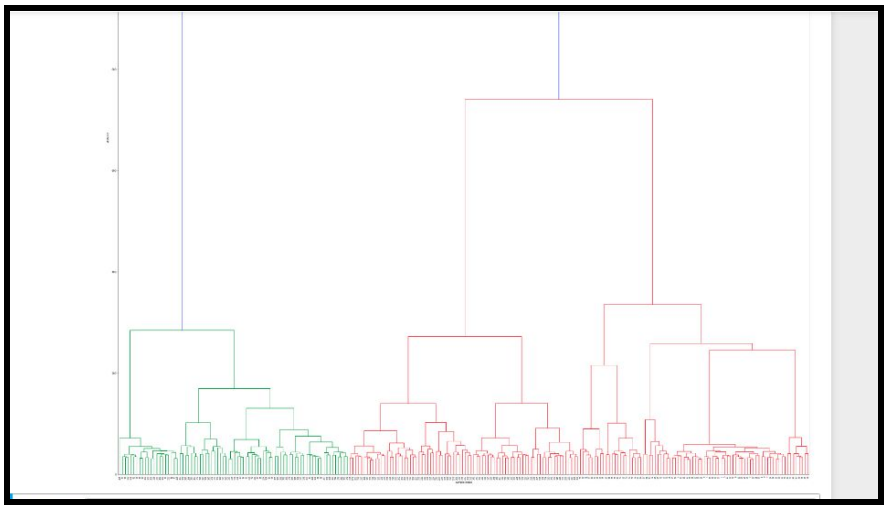
clustering. Once we decide the model we need to evaluate it using a proper metric. We could also use density based , graph - cut based and evolutionary methods for understanding clustering but it is not very clear as to what type of transformation would be suitable for these types of clustering algorithms.

IMPLEMENTATION

Experimental results confirm our faith in hierarchical clustering over K - means clustering.

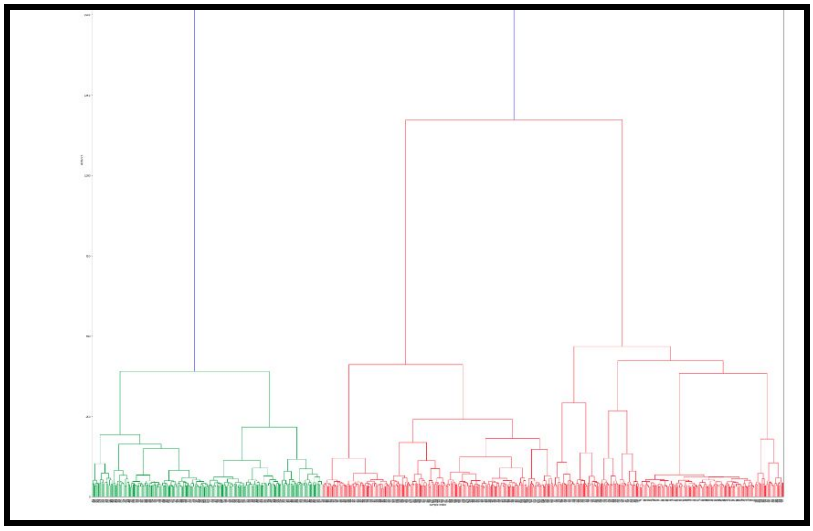
For agglomerative clustering, we look at several types of distances and compare their scores with DTW distance measures.

A look at the dendograms gives the general picture :

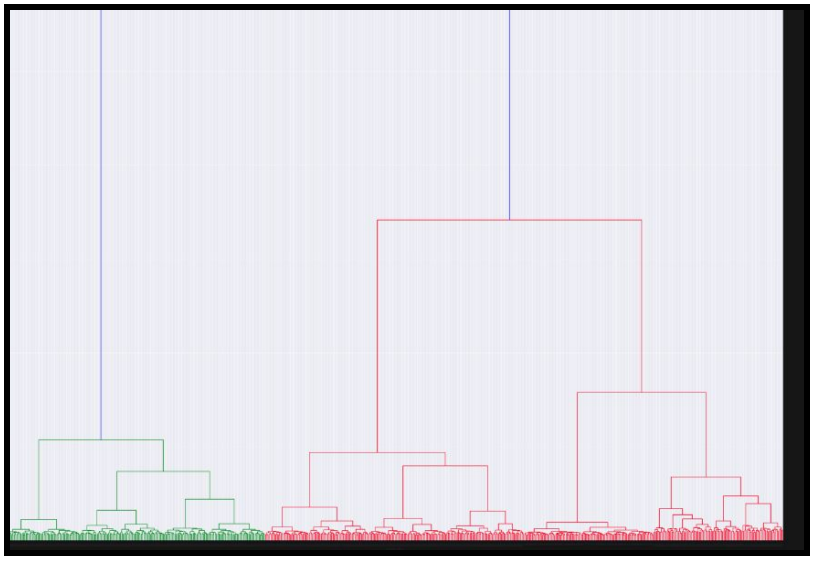


1. Using Euclidean distances, we get too many subclasses. Dissimilarities in the class -

labels are also high. The classes are of unequal sizes. This is the plot where there are only 40 samples per class.

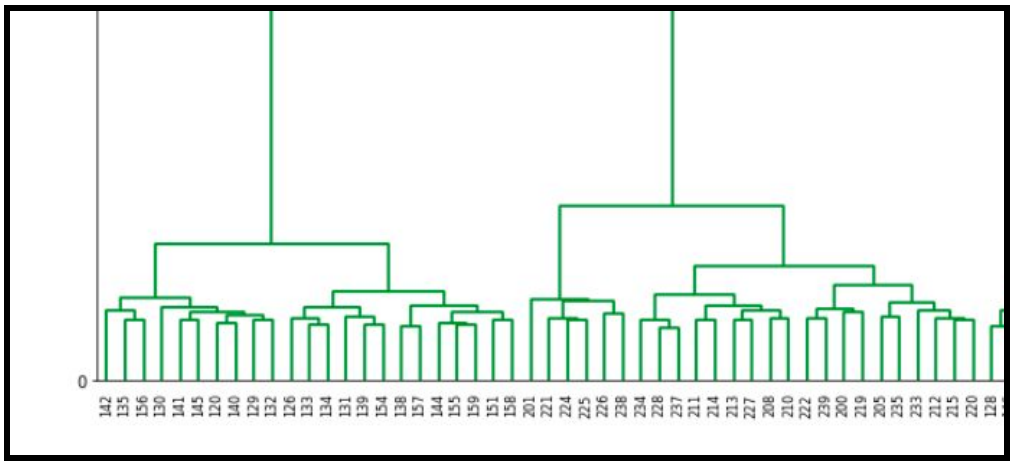


2. Euclidean distance - with all samples. Again a similar trend is seen.

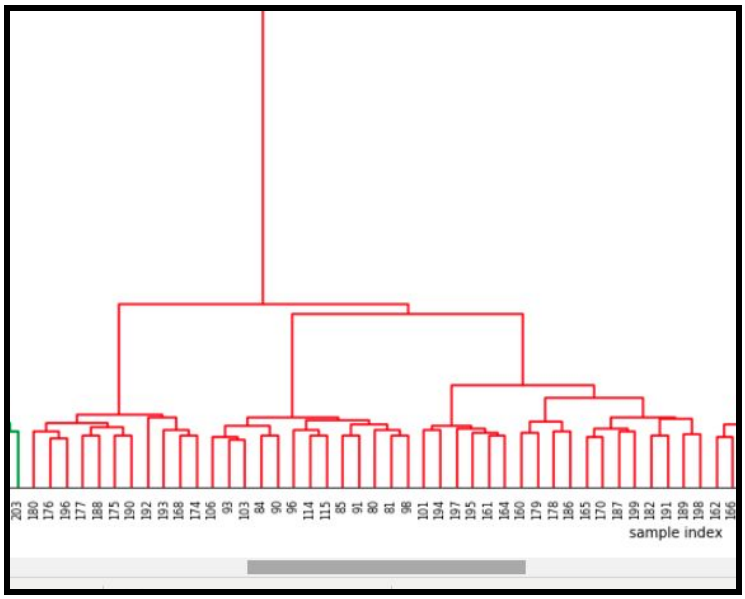


3. The above figure shows the dendogram (100 samples per class) using DTW distances with agglomerative clustering. As is seen, the number of subclasses closely resemble the actual number of classes(cutting at the proper point).

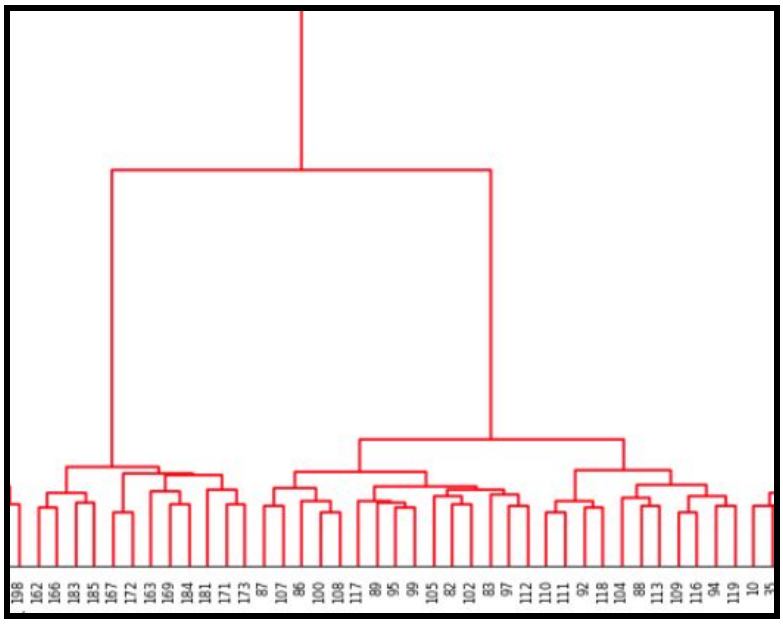
Closer look reveals the actual results (again for 40 samples per class) :



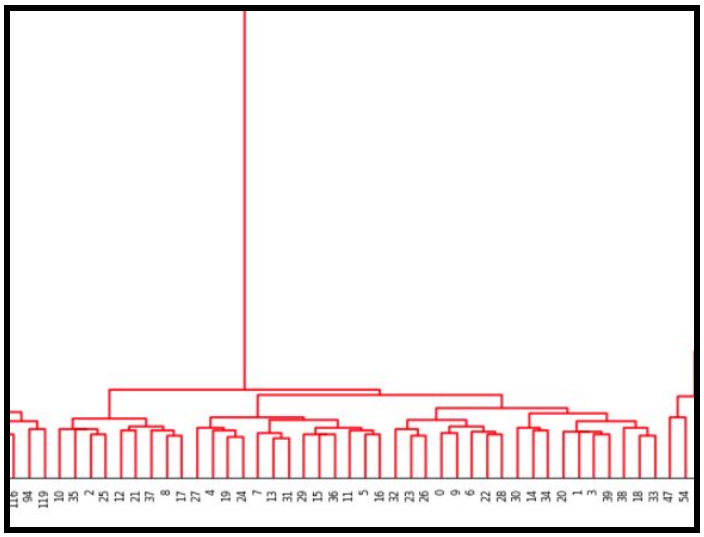
a. Clusters 1 and 2 - resembling (120 - 160) and (200 - 240).



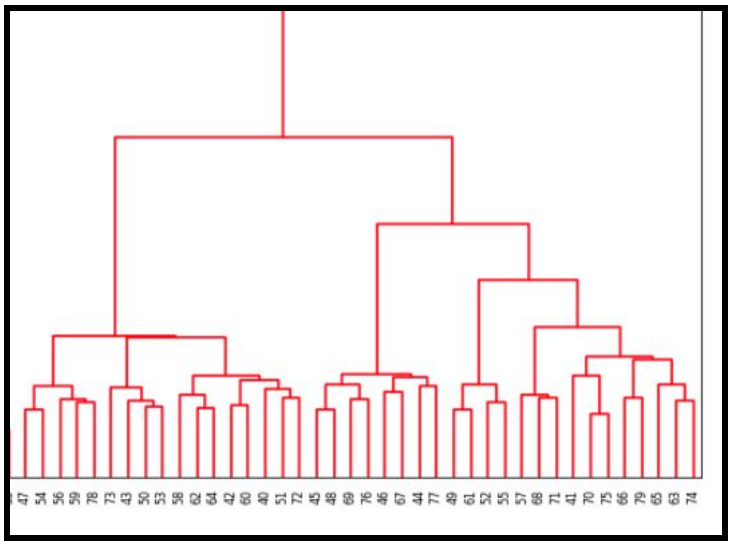
b. Third cluster - resembling (160 - 200)



c. The above two clusters are somewhat similar (80 - 120) and previously (160 - 200). This is expected as these two are really similar in their trends (upward shift and increasing).



d . The cluster resembling class (1 - 40).



e. Last cluster resembling class (40 - 80).

EVALUATION

As is apparent, visual verification is the best measure of the clustering task because of the temporal natures of the time series and presence of various types of allowed temporal transformations.

But since the applications of time - series analysis are widespread and mostly critical in taking decisions, we must have some sort of mathematical verification of our predicted clusters.

In particular any evaluation metric should not take the absolute values of the cluster labels into account but rather if this clustering defines separations of the data similar to the ground truth set of classes or satisfying some assumption such that members belong to the same class are more similar that members of different classes according to the similarity metric.

In this problem, we exploit the knowledge of the true classes to set up apriori knowledge and use sklearn's completeness score which is a measure of how well the members of the same class get grouped together , regardless of the order of the grouping.

This means that data appearing earlier in the dataset may be assigned a higher class - label but the completeness score still be high if many of them get the same class label and

this is exactly what we want.

K - means algorithm gives score of 0.75 with random initialization of the mean points. The performance of hierarchical clustering is superior, rarely falling below 0.8 even after lot of modifications. The euclidean distance and cosine distance parameters both resulted in 0.81 score with agglomerative technique.

Even though the expected maximum was for DTW, agglomerative clustering with L1 - distance actually gave a score of 0.881 which is a little more than DTW's 0.879 . This is a really counterintuitive result taking into account the nature of the problem. The reason of this is probably the less number of samples per class (100) and very less variance in the y - parameter (0 - 60) , which is why simple one - to - one matching of the feature values as done in L1 norm yields quite a good result.

Nevertheless L1 is not expected to scale well . Both increase in sample size as well increase in range of values would lead simple L1 distance to go astray. For better asymptotics, we would probably want to stick with DTW.

Just for completeness , spectral and density based methods were also tried with default parameters but the performance of these were found to be very low. Some modifications in the preprocessing is required for these methods and they need to be manipulated according to the model at hand.

CONCLUSION

In this project, various well known clustering algorithms for the task time series analysis were compared. A very special case of time series data was handled where the labels were known apriori and the challenge was to determine how to modify these algorithms to work well with time series data.

The most challenging part was computing an appropriate similarity notion to deal with the extravagance of temporal sequences. DTW was found to be superior in performance both intuitively as well as experimentally, when used with hierarchical methods.

Another problem was the curse of dimensionality - which slowed progress and also made visualisation tough. Since the time -steps could not be compromised, the number of samples was periodically reduced to get insights about the performances of the algorithms and sometimes, online GPU - accelerated environments were used for

training.

The last challenge was for evaluation because of the unsupervised nature of clustering and less variance in the range of the series. The number of features was high from a dimensionality point of view but the number of time steps (60) is quite low when we think about the nature of the problem. This led to less confidence - margins between models which were very different mathematically and visual resolution was significant.

Overall, the hierarchical model with DTW metric shows seems to be the most promising in terms of asymptotic performance. One idea for future introspection is the use of LSTMs and neural networks and modifying them to perform clustering on time series.

REFERENCES

1. *Time-Series Similarity Queries Employing a Feature-Based Approach .R. J. ALCOCK and Y. MANOLOPOULOS*
2. *Dynamic Time Warping - wikipedia.com.*
3. *Scikit-learn documentation.*