Analysis of similarity between artificially simulated time series with Dynamic Time Warping

Dmytro Krukovets

Abstract

Paper presents a suite of the model that finds similarity in dynamics between time series and groups them by this property; and an artificial data generator that builds those time series that have issues, close to the real ones. These two parts open a rich field for the further analysis of both real-life data and new algorithms that are able to find and distinguish these real-life issues for the more comprehensive analysis.

Keywords: Dynamic Time Warping, clustering, distance matrix, artificial dataset

MSC 2010: 37M10, 91B84, 62H30, 51K05.

1 Introduction

In this paper, I would like to introduce a model that helps to split time series into several groups and helps with an analysis of the underlying relationship that puts these series into similar groups. This is about a Dynamic Time Warping algorithm that is widely used to find distances between series. Distance is a measure of similarity to some extent, thus series that are akin by dynamics would have a low distance between them. This algorithm captures situations where there are some distortions between series: lags, stretching, contractions and other.

Among major use-cases for the algorithm, I'd like to mention sound recognition and sound motion analysis, where you'd compare audio by several dynamic properties [5]. Another example is an analysis of cardiograms, where there are some “healthy” patterns and “problematic” ones. They can be compared and splited into healthy and unhealthy for further ability to recognize the latter ones fast [1]. One more example is financial pattern recognition that is highly used to analyze stock prices [4].

A common characteristic of the abovementioned use-cases is high-frequency data and a decent amount of observations. But what if these time series are rather short? The major example of such a situation is macroeconomics and corresponding forecasting. Recently, the Data Science algorithms start to infiltrate into the field and become a popular tool despite the fact that series are short [7]. This invasion was accompanied by modern ways to enlarge current macroeconomic datasets with support of web-scraping techniques, Google Trends and other. But also, another way was to adapt traditional Data Science algorithms to fit the case of shorter series without much loss of efficiency.

The paper will be focused on the development of a model that creates some artificial data with several parameters, that represent real data and then the other part of the model that divides this dataset into several groups. There will be a comparison of DTW with simpler benchmarks (Euclidean distance, correlation-based distance) to prove a better performance of the former one in terms of capturing the real dynamics.

2 Data

The research is highly dependent on the data quality because we would like to find the similarity of the series and subjectively evaluate whether results correspond to the initial hypotheses. Also, the research is not bound by some particular topic and has no requirements for the data except to be realistic. There are several options of what kind of dataset to choose in order to get the most appropriate result: 1) open dataset with short time series; 2) artificially created set of series. I have decided to follow the second approach for several reasons: 1) It allows a full control over issues that should be analyzed; 2) We are not forced to control for endogeneity; 3) We can formulate the hypothesis with a relative ease. In the case of real data, these problems become a subject of higher amount of biases and errors, based on subjective judging of the data dynamics.

The dataset is created according to the TimeSynth project by [8]. This is a tool for simpler creating the data series that might be easily replicated in the Excel or whatever other application that allows working with the data series. In the case of the paper, I've created a set of forty series with ninety observations each.

These series have got several properties. First of all – it is a basic function. It could be a sine, a cosine, AR(1), AR(12), ARMA(1,1) or ARMA(12,1) process. Then the white noise, generated as a normally distributed variable with mean 0, is added. Noise can have standard deviation 0.1 (small), 1 (medium) and 5 (large). Also, there is an option to do not add the noise. The last property is an outlier that can take multiple forms: small outlier (1), large outlier (5) and a set of two small and one large outlier. All of them can be both positive and negative.

This routine helps to create twenty-five series. Fifteen more are created in a similar manner, but they have got no outliers and have a structural break, which means a new basic function starting from the thirty-first observation. A full set of variables is given in Appendix A.

3 Model

As for a model, the pipeline is as follows: to find distances between time series, to build a distance matrix and represent series as a point on the two-dimensional plane with corresponding distances between them, to cluster these series into groups with the corresponding algorithm.

3.1 Distance algorithm

3.1.1 Euclidean distance

As a first and the easiest algorithm, we'll use simple Euclidean distance between series. The algorithm finds distances as follows. If we have time-series S_1 and S_2, $S_1(t) = p_t$ and $S_2(t) = q_t$, then:

$$dist(S_1, S_2) = \frac{\sqrt{\sum_{t=1}^{n}(p_t - q_t)^2}}{n}. \tag{1}$$

In other words, it's an average of squared deviations of one series from another. Obviously, this approach does not count for any distortions such as lagged series (those, who has similar functional form, but one of them with a lag). The approach is not of great use for real-life time series analysis as long as it cannot count popular properties of series, but it is a good benchmark to compare with.

3.1.2 Correlation

The next step is a correlation-based measure that allows representing high correlation as a short distance and vice versa. A distance between two series is given as a

$$dist(S_1, S_2) = 1 - |corr(S_1, S_2)|\,. \tag{2}$$

The design is given in such a form because a high absolute correlation (whenever it is positive or negative) means a high level of similarity [3]. There are a plethora of other forms for correlation-based distance, but we will stop on the simplest one as a great benchmark and also because an investigation of these sub-methods is out of the scope for the current paper.

3.1.3 Dynamic Time Warping

Then we'll go with a simple Dynamic Time Warping (DTW) algorithm [2]. In the original form, the DTW builds a matrix with distances from each point of one series to each of another. The left bottom cell is a distance between the first points, while the right upper is between last points. Then, the path between these cells is built in a way to minimize the sum of distances. It helps to produce no intersections between correspondences and connect all points between each other. Such a design gives an opportunity to deal with lagged reaction (correspondence might be with a time shift), stretching (one point might correspond to several) and contraction (inverse to the previous one).

The most famous expansion of the algorithm, FastDTW [10], is called to speed it up. The original algorithm complexity is $o(n^2)$, thus the time to perform it grows quadratically with increased number of observations, because there is a necessity to build an n by n matrix of distances. FastDTW shows that there is no necessity to calculate the whole matrix, but only a part of it. This comes from the fact that path lies in the central area mostly. Thus, few ”masks” are used to ”shadow” the area that has relatively low chances to contain a part of the path, thus should not be calculated at all. FastDTW utilizes few more additions, however abovementioned one is the most important for the further work because it gives the idea with limitations of the distance matrix. Moreover, despite the overwhelming usage of the FastDTW approach, it is not necessary in our case because this work is concentrated on a relatively small series that does not hold even a thousand observations.

3.2 Distance Matrix

After finding distances with all abovementioned methods, we can put them into corresponding matrices where a cell in a row p and column q means a distance between series p and q. This matrix is symmetric because the distance between p and q is equal to the distance between q and p. A theorem [6] suggests that we can build a unique (up to the rotation) two-dimensional set of points, distances between which are equal to those in the distance matrix. It is important in order not to obtain several different groupings for a set.

3.3 Clustering

During the previous stage, we obtained a two-dimensional plane with a set of points and now we can use a clustering technique to group similar series into a single cluster. An algorithm of choice was a simple K-Means clustering [9]. The reason is the simplicity and interpretability that helps with hypothesis checking, corresponding to the basic visual analysis results etc.

4 Results

As for the result, the main one is a plane of points, that correspond to series, grouped via K-Means algorithm. We'll do this for all cases and analyze corresponding results.

Let us start with the simple Euclidean distance case. In Figure 1, we can see two main clusters (black and blue), so as several pairs of relatively close series like in the ltblue and green clusters. The black cluster consists mainly of AR-type series with different noises, while blue is a cluster of Sine/Cosine. Green cluster is AR(12) with large noise and outlier, while ltblue is a combination of cosine and AR(12) with large noise. The major plus is that algorithm is able to find the basic difference between series (divides on AR-type and sine-type functions), despite its straightforward nature. On the other hand, it gives no better insight into the data, which is unfavourable for the future applying.

The next will be a sub-exercise with standardized data rather than the raw one. As we can see in Figure 2, this case has less strongly marked clusters. The strongest one is blue, which contains ARMA(1,1) processes with different noises. The ltblue cluster seems to be not bad too, but it contains sine-based functions (even without cosine). Other clusters are not that tight and do not give much of the additional information. This exercise repeats the previous conclusion that there are relatively no conclusions, it seems like it works worse than the previous one, but on the other hand, standardized data removes ”levels” of series that plays a role as long different series have a different constant term.

Figure 3 shows a correlation-based distance case, which is already more interesting as long as it has several pretty nice groups. The first one is black, which consists of ARMA(1,1) models and several combinations of ARMA with AR (structural break). The red one seems

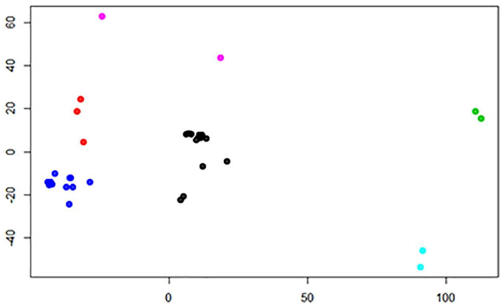

Figure 1. Euclidean distance

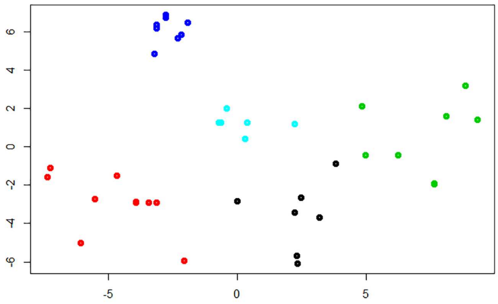

Figure 2. Scaled Euclidean distance

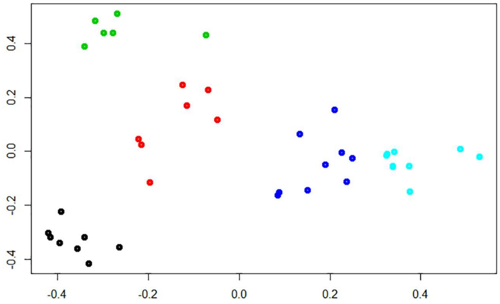

Figure 3. Correlation-based distance

to be slightly divided into two sub-parts and it consists of Sine/Cosine plus one ARMA(12,1) with large noise. The green cluster is a bunch of sine series with small-to-medium noises. The blue one (that seems to be united with the ltblue) is full of series with structural breaks, where one part is AR-based and the other is Sine-based. It is quite interesting as long as previous methods were unable to find this correspondence. The ltblue cluster continues the tradition of blue and consists of many broken series as long as several AR(1) with relatively little noise. The case with scaled data leads to the same result as long as standardization does not affect correlation.

Finally, we're coming to the Dynamic Time Warping algorithm, which case is depicted in Figure 4. Here we can observe several extremely tight clusters (ltblue, part of the red and part of the violet). Green and black clusters seem to be outliers. All these series are fully or partially (due to the structural break) are AR(12) with large noise. As for the red cluster, the tightest part contains sines with small noise, another are cosines or combinations of sine and cosine. It is interesting because in this case sines and cosines became closer to each other, which was not observed earlier. And it is a correct move because basically sine and cosine dynamics are the same. The ltblue cluster is basically very tight and consists of most of the AR-type series with

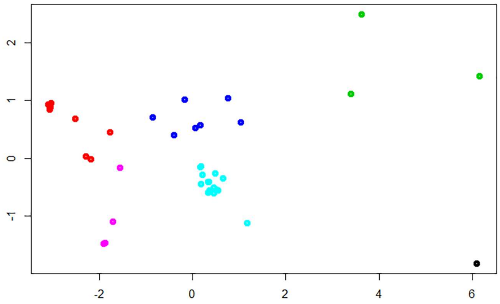

Figure 4. Dynamic Time Warping distance

no-to-medium noise. It also seems better than in the previous cases because DTW captures the AR-type dynamics despite the noise (if it's not too large and leading in the total dynamics). Moreover, even AR-type with structural break takes its place here. Blue cluster is mostly about very noisy series, so as a violet one, that contains mostly sine-based functions with or without a structural break.

5 Conclusion

The model, developed and described in the paper, supports a deep investigation of the relationship between time series, especially those that imitate structural breaks, outliers, a small number of observations etc. These particular issues are rather popular in the macroeconomics, where the exercise finds its good use as one of the possibilities to investigate economic relationship based on the data.

Results show us that the more sophisticated algorithm, the more information we can obtain out of it. For example, simple Euclidean case so as the correlation-based method could not understand that sine and cosine are basically the same dynamics, while the DTW was much more able to do this despite the noise. However, the large noise plays its role and distract the algorithm results. As for the outlier, there was no decent sign that algorithm was able to find it and it could not make any difference, but the result regarding the structural break is more interesting. In the case of DTW, there is more tendency of those series to be with other simple series, especially when the break is homogeneous (sine to cosine or AR(1) to ARMA(12,1)). That seems very promising for a further and deeper investigation.

As for further development, there are two major ways. The first one is to focus on the development of DTW extensions and a richer set of clustering models or even more advanced standardization in order to obtain a better result and squeeze more information from the data. The second one is to move forward in the development of the actual tool for this comparison and artificial data generation process, make it even more automatic, consider other possible basic function and issues that arise in the real data and should be modelled in the artificial case properly.

References

[1] Annam J. R., Mittapalli S. S., and Bapi R. S. *Time series Clustering and Analysis of ECG heart-beats using Dynamic Time Warping*, Annual IEEE India Conference, Hyderabad, 2011, pp. 1–3, DOI: 10.1109/INDCON.2011.6139394. Available at: https://ieeexplore.ieee.org/abstract/document/6139394.

[2] Berndt D. and Clifford J. *Using Dynamic Time Warping to Find Patterns in Time Series*, 1994, AAAI Technical Report WS-94-03. Available at: https://www.aaai.org/Papers/Workshops/1994/WS-94-03/WS94-03-031.pdf.

[3] Chatterjee D. *Log Book – Guide to Distance Measuring Approaches for K-Means Clustering*, 2019. Available at: https://towardsdatascience.com/log-book-guide-to-distance-measuring-approaches-for-k-means-clustering-f137807e8e21.

[4] Coelho M. *Patterns in Financial Markets: Dynamic Time Warping*, 2011. Available at: https://run.unl.pt/bitstream/10362/9539/1/Coelho_2012.pdf.

[5] Cowling M. and Sitte R. *Comparison of techniques for environmental sound recognition*, Pattern Recognition Letters, Volume 24, Issue 15, November 2003, pp. 2895–2907. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0167865503001478?via%3Dihub.

[6] Dokmanic I., Parhizkar R., Ranieri J., and Vetterli M. *Euclidean Distance Matrices. Essential Theory, Algorithms and Applications*, 2015. Available at: https://arxiv.org/pdf/1502.07541.pdf.

[7] Krukovets D. D*ata Science Opportunities at Central Banks: Overview*, Visnyk of the National Bank of Ukraine, 249, 13-24, 2020. https://doi.org/10.26531/vnbu2020.249.02.

[8] Maat J. R., Malali A., and Protopapas P. *TimeSynth: A Multipurpose Library for Synthetic Time Series in Python*, 2017. Available at: http://github.com/TimeSynth/TimeSynth.

[9] MacQueen J. *Some methods for classification and analysis of multivariate observations*, 1967. Available at: https://projecteuclid.org/euclid.bsmsp/1200512992.

[10] Salvador S. and Chan P. *FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space*, 2004. Available at: https://www.semanticscholar.org/paper/FastDTW%3A-Toward-Accurate-Dynamic-Time-Warping-in-Salvador-Chan/05a20cde15e172fc82f32774dd0cf4fe5827cad2.

Appendix A

Without structural break

AR(1)_Small_noise_Small_up
SINE_Large_noise_SLS_up
COSINE_Small_noise_Small_up
AR(1)_Medium_noise_Small_down
ARMA(1,1)_Medium_noise_SLS_up
SINE_Large_noise_Large_up
ARMA(1,1)_Small_noise_SLS_up
SINE_No_noise_SLS_down
COSINE_Large_noise_Large_up
SINE_Small_noise_SLS_down
COSINE_Large_noise_Small_up
AR(12)_Large_noise_Small_down
ARMA(12,1)_Large_noise_Small_up
ARMA(1,1)_Medium_noise_No_outlier
AR(12)_Large_noise_SLS_up
AR(1)_Small_noise_SLS_up
ARMA(1,1)_No_noise_SLS_down
AR(1)_No_noise_No_outlier
ARMA(1,1)_No_noise_Small_down
SINE_No_noise_Small_up
COSINE_Medium_noise_Large_up
ARMA(12,1)_No_noise_No_outlier
ARMA(1,1)_Medium_noise_Small_up
ARMA(12,1)_Small_noise_Small_down
SINE_Small_noise_Large_down

With structural break

ARMA(12,1)_COSINE_Medium_noise
AR(12)_AR(1)_No_noise
COSINE_AR(12)_Large_noise
ARMA(1,1)_SINE_Large_noise
AR(1)_ARMA(1,1)_Medium_noise
AR(1)_SINE_No_noise
COSINE_AR(12)_Large_noise
AR(12)_AR(1)_No_noise
AR(1)_ARMA(1,1)_No_noise
ARMA(1,1)_SINE_No_noise
AR(12)_SINE_Large_noise
COSINE_ARMA(1,1)_No_noise
SINE_COSINE_Medium_noise
COSINE_ARMA(1,1)_Large_noise
COSINE_ARMA(1,1)_Large_noise

Dmytro Krukovets

Received October 24, 2020

Institution: Taras Shevchenko National University of Kyiv
Address: Akademika Hlushkova Ave, 4D, Kyiv, Ukraine, 03680
E–mail: dkrukovets@kse.org.ua