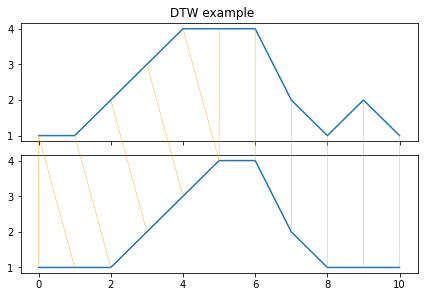
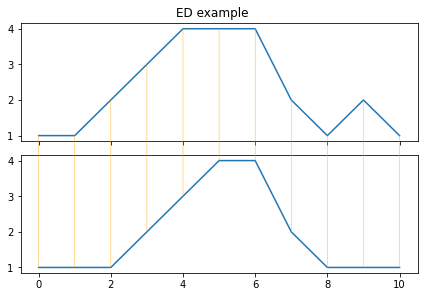
# Clustering with dynamic time wraping

## Why DTW as a metric for time series analysis?

The dynamic time warping (DTW) algorithm is a frequently used method for measuring the distance between time series. As opposed to the classical Euclidean distance (ED) metric, which measures the distance between each two points of the same timestep, the DTW distance measure is less sensitive to distortion in the time axis by tolerating phase shifts and prolongations.

Two curves are assessed similar not by having similar values at each point, but by having a similar “shape” given the DTW distance measure (sum of the vertical lines). If, for example, the influence of an external event arrives with a delay at time series B a week later as compared to time series A, this causes a shift in the shape of the two time series (compare point 1 and 2 of upper and lower curve). Such shifts are not detected by the step-wise comparison (ED example), but in the DTW example. This flexibility allows for an improved shape-based comparison of time series, that can be used for classification or clustering problems. To perform a cluster analysis the distance measures for all possible pairs of a number of time series must be calculated. Depending on the chosen clustering algorithm and its distance method, the time series are then grouped together based on that single pairwise measures.

## Why is it useful for our project?

(This popped into my mind as a possible line of argumentation. But needs a serious discussion, these claims feel quite shaky)

The objective is a comparison of forecast practices and performance across the domains. This raises the question of the difference between the domains in terms of predictability. One way of comparing the complexity of domains is by defining features based on which they can assessed. We can imagine different classes for such features, e.g.:

* *Simple quantitative measures* such as standard deviation, curve length, etc.
* *Advanced quantitative measure* such as accuracy of benchmark forecast (naïve forecast), degree of a fitted polynomial function, etc.
* *Qualitative measure* such as degree of abstraction, data availability etc. (these are just wild guesses).

A preceding definition of relevant features is also the standard procedure for most clustering problems. However, this poses the problem of being able to define, in our case, “predictability” by an explicit set of features which often comes with a broad set of assumptions and the risk of missing relevant features. By using DTW as similarity measure, we try to bypass that problem by selecting only one explicit but abstract feature of “shape”. This feature still contains statistical characteristics implicitly, which can be calculated for each cluster individually, but we do not assume them as differentiating characteristics beforehand.

## How it’s done?

1. **Scaling:** First the data need to be scaled that the similarity measure is not influenced by the different scales of the domains. I selected a simple scaling between 0 and 1 to for each domain to have as little impact on the shape of the time series as possible. I used the [MinMaxScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html) where the transformation is given by:  
      
   See the appendix in the code for other scaling methods I also tried. The results of the MinMaxScaler are the following:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | negaffect | posaffect | eafric | easian | egend | iafric | iasian | igend | lifesat | ideoldem | ideolrep | polar |
| 0 | 0,19 | 0,77 | 0,92 | 0,53 | 0,87 | 0,86 | 0,16 | 0,18 | 0,10 | 0,93 | 0,85 | 0,29 |
| 1 | 0,10 | 0,79 | 0,74 | 0,62 | 0,62 | 0,73 | 0,50 | 0,69 | 0,08 | 0,91 | 0,91 | 0,31 |
| 2 | 0,28 | 0,44 | 0,71 | 0,74 | 0,65 | 0,94 | 0,68 | 0,88 | 0,04 | 0,90 | 0,86 | 0,27 |
| 3 | 0,50 | 0,29 | 0,79 | 0,29 | 0,91 | 0,94 | 0,47 | 0,74 | 0,22 | 0,87 | 0,83 | 0,32 |
| 4 | 0,13 | 1,00 | 1,00 | 0,58 | 0,82 | 1,00 | 0,54 | 0,83 | 0,43 | 0,82 | 0,73 | 0,26 |
| 5 | 0,14 | 0,73 | 0,76 | 0,67 | 0,58 | 0,82 | 0,39 | 0,50 | 0,23 | 0,82 | 0,70 | 0,30 |
| 6 | 0,16 | 0,52 | 0,73 | 0,54 | 1,00 | 0,81 | 0,63 | 0,49 | 0,00 | 0,79 | 0,69 | 0,28 |
| 7 | 0,41 | 0,36 | 0,63 | 0,36 | 0,87 | 0,61 | 0,53 | 0,75 | 0,11 | 0,82 | 0,69 | 0,00 |
| 8 | 0,13 | 0,98 | 0,71 | 0,59 | 0,68 | 0,78 | 0,85 | 0,39 | 0,28 | 0,81 | 0,69 | 0,06 |
| 9 | 0,05 | 0,81 | 0,62 | 0,74 | 0,62 | 0,85 | 0,71 | 0,57 | 0,40 | 0,82 | 0,68 | 0,07 |
| 10 | 0,20 | 0,44 | 0,67 | 0,55 | 0,47 | 0,77 | 0,49 | 0,60 | 0,52 | 0,82 | 0,68 | 0,09 |
| 11 | 0,40 | 0,21 | 0,78 | 0,74 | 0,67 | 0,75 | 0,25 | 0,50 | 0,64 | 0,85 | 0,67 | 0,10 |
| 12 | 0,02 | 0,88 | 0,59 | 0,48 | 0,77 | 0,71 | 0,55 | 0,64 | 0,64 | 0,84 | 0,72 | 0,42 |
| 13 | 0,04 | 0,82 | 0,51 | 0,86 | 0,59 | 0,66 | 0,78 | 0,60 | 1,00 | 0,82 | 0,72 | 0,34 |
| 14 | 0,16 | 0,50 | 0,50 | 0,52 | 0,87 | 0,70 | 0,46 | 0,26 | 0,96 | 0,83 | 0,72 | 0,32 |
| 15 | 0,00 | 0,92 | 0,56 | 0,45 | 0,59 | 0,76 | 0,34 | 0,45 | 0,84 | 0,84 | 0,72 | 0,30 |
| 16 | 0,12 | 0,68 | 0,66 | 0,41 | 0,55 | 0,61 | 0,27 | 0,51 | 0,80 | 0,79 | 0,75 | 0,25 |
| 17 | 0,23 | 0,48 | 0,78 | 0,38 | 0,74 | 0,90 | 0,38 | 0,43 | 0,57 | 0,85 | 0,74 | 0,46 |
| 18 | 0,12 | 0,79 | 0,39 | 0,36 | 0,58 | 0,70 | 0,35 | 0,44 | 0,57 | 0,85 | 0,74 | 0,39 |
| 19 | 0,24 | 0,60 | 0,40 | 0,54 | 0,59 | 0,49 | 0,24 | 0,20 | 0,68 | 0,86 | 0,76 | 0,27 |
| 20 | 0,24 | 0,42 | 0,35 | 0,57 | 0,72 | 0,68 | 0,41 | 0,49 | 0,61 | 0,90 | 0,77 | 0,43 |
| 21 | 0,15 | 0,71 | 0,34 | 0,66 | 0,69 | 0,62 | 0,63 | 0,44 | 0,69 | 0,92 | 0,80 | 0,50 |
| 22 | 0,23 | 0,59 | 0,40 | 0,44 | 0,46 | 0,65 | 0,35 | 0,39 | 0,76 | 0,92 | 0,80 | 0,57 |
| 23 | 0,30 | 0,39 | 0,50 | 0,64 | 0,38 | 0,68 | 0,36 | 0,43 | 0,73 | 0,87 | 0,77 | 0,48 |
| 24 | 0,19 | 0,66 | 0,55 | 0,65 | 0,29 | 0,76 | 0,69 | 0,33 | 0,74 | 0,86 | 0,75 | 0,57 |
| 25 | 0,20 | 0,65 | 0,52 | 0,53 | 0,67 | 0,83 | 1,00 | 0,72 | 0,78 | 0,85 | 0,77 | 0,65 |
| 26 | 0,30 | 0,37 | 0,56 | 0,49 | 0,72 | 0,68 | 0,35 | 0,59 | 0,78 | 0,85 | 0,76 | 0,75 |
| 27 | 0,15 | 0,68 | 0,65 | 0,54 | 0,49 | 0,79 | 0,56 | 0,08 | 0,82 | 0,84 | 0,75 | 0,44 |
| 28 | 0,21 | 0,66 | 0,62 | 0,86 | 0,85 | 0,73 | 0,62 | 0,37 | 0,83 | 0,84 | 0,76 | 0,44 |
| 29 | 0,26 | 0,37 | 0,40 | 0,38 | 0,82 | 0,75 | 0,75 | 0,52 | 0,62 | 0,86 | 0,77 | 0,65 |
| 30 | 0,16 | 0,74 | 0,43 | 0,64 | 0,63 | 0,73 | 0,45 | 1,00 | 0,48 | 0,85 | 0,76 | 0,57 |
| 31 | 0,25 | 0,59 | 0,36 | 0,59 | 0,41 | 0,73 | 0,43 | 0,15 | 0,50 | 0,85 | 0,76 | 0,67 |
| 32 | 0,36 | 0,35 | 0,51 | 0,51 | 0,62 | 0,74 | 0,71 | 0,42 | 0,66 | 0,86 | 0,77 | 0,65 |
| 33 | 0,11 | 0,79 | 0,48 | 0,51 | 0,51 | 0,72 | 0,47 | 0,46 | 0,68 | 0,88 | 0,80 | 0,54 |
| 34 | 0,24 | 0,61 | 0,55 | 0,49 | 0,42 | 0,75 | 0,18 | 0,25 | 0,80 | 0,94 | 0,82 | 0,75 |
| 35 | 0,27 | 0,37 | 0,45 | 0,49 | 0,00 | 0,81 | 0,32 | 0,07 | 0,70 | 0,96 | 0,86 | 0,49 |
| 36 | 0,13 | 0,78 | 0,53 | 0,62 | 0,29 | 0,71 | 0,73 | 0,37 | 0,68 | 0,92 | 0,85 | 0,57 |
| 37 | 0,19 | 0,61 | 0,44 | 0,74 | 0,30 | 0,66 | 0,40 | 0,35 | 0,77 | 0,95 | 0,85 | 0,72 |
| 38 | 0,36 | 0,34 | 0,47 | 1,00 | 0,42 | 0,77 | 0,72 | 0,65 | 0,91 | 0,94 | 0,80 | 0,52 |
| 39 | 0,44 | 0,24 | 0,38 | 0,52 | 0,62 | 0,67 | 0,36 | 0,70 | 0,58 | 0,87 | 0,74 | 0,72 |
| 40 | 0,55 | 0,22 | 0,15 | 0,15 | 0,67 | 0,23 | 0,29 | 0,15 | 0,49 | 0,95 | 0,86 | 0,34 |
| 41 | 1,00 | 0,00 | 0,02 | 0,00 | 0,49 | 0,00 | 0,00 | 0,27 | 0,10 | 0,96 | 0,86 | 0,67 |
| 42 | 0,81 | 0,20 | 0,00 | 0,15 | 0,44 | 0,20 | 0,15 | 0,31 | 0,39 | 0,96 | 0,86 | 0,80 |
| 43 | 0,84 | 0,21 | 0,09 | 0,27 | 0,42 | 0,19 | 0,21 | 0,25 | 0,46 | 0,98 | 0,87 | 0,70 |
| 44 | 0,88 | 0,19 | 0,05 | 0,20 | 0,47 | 0,26 | 0,17 | 0,31 | 0,49 | 0,98 | 0,90 | 0,82 |
| 45 | 0,97 | 0,17 | 0,10 | 0,20 | 0,55 | 0,36 | 0,20 | 0,10 | 0,50 | 1,00 | 0,89 | 1,00 |
| 46 | 1,00 | 0,16 | 0,07 | 0,19 | 0,38 | 0,29 | 0,23 | 0,11 | 0,48 | 0,91 | 0,95 | 0,80 |
| 47 | 0,76 | 0,26 | 0,15 | 0,30 | 0,24 | 0,28 | 0,20 | 0,00 | 0,37 | 0,92 | 1,00 | 0,49 |
| 48 | 0,83 | 0,22 | 0,12 | 0,22 | 0,41 | 0,25 | 0,34 | 0,16 | 0,54 | 0,00 | 0,00 | 0,57 |
| 49 | 0,48 | 0,42 | 0,04 | 0,25 | 0,20 | 0,31 | 0,18 | 0,36 | 0,53 | 0,89 | 0,85 | 0,65 |
| 50 | 0,42 | 0,43 | 0,06 | 0,17 | 0,26 | 0,31 | 0,15 | 0,29 | 0,47 | 0,92 | 0,88 | 0,75 |
| 51 | 0,32 | 0,46 | 0,03 | 0,03 | 0,43 | 0,31 | 0,07 | 0,27 | 0,51 | 0,88 | 0,84 | 0,59 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Pairwise comparison:** As previously mentioned each of the time series must be compared to measure their similarity. This is essentially done by creating a cost matrix and finding the optimal path (see [here](https://medium.com/walmartglobaltech/time-series-similarity-using-dynamic-time-warping-explained-9d09119e48ec) for detailed description). The final distance measure is calculated as the sum of the optimal path.
2. **Distance matrix:** The distance matrix represents the distance measures of every possible pair and is used as input for the clustering algorithm.

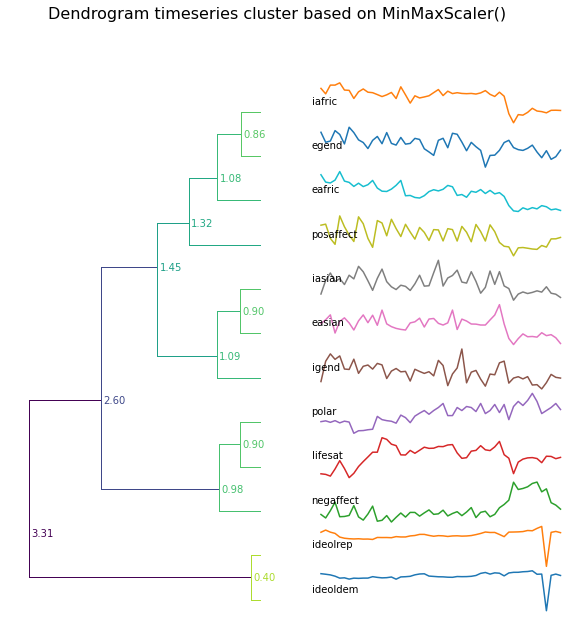
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | negaffect | posaffect | eafric | easian | egend | iafric | iasian | igend | lifesat | ideoldem | ideolrep | polar |
| negaffect | 0,000 | 1,731 | 2,070 | 1,469 | 1,815 | 2,508 | 1,334 | 1,372 | 0,939 | 3,312 | 2,809 | 0,982 |
| posaffect | 1,731 | 0,000 | 1,323 | 1,145 | 0,952 | 1,105 | 1,281 | 1,331 | 1,570 | 2,112 | 1,718 | 1,706 |
| eafric | 2,070 | 1,323 | 0,000 | 0,968 | 0,963 | 1,084 | 1,275 | 1,345 | 2,244 | 2,671 | 2,229 | 2,599 |
| easian | 1,469 | 1,145 | 0,968 | 0,000 | 1,117 | 1,018 | 0,897 | 1,094 | 1,486 | 2,326 | 1,903 | 1,667 |
| egend | 1,815 | 0,952 | 0,963 | 1,117 | 0,000 | 0,862 | 1,229 | 1,231 | 1,723 | 2,324 | 2,031 | 2,077 |
| iafric | 2,508 | 1,105 | 1,084 | 1,018 | 0,862 | 0,000 | 1,448 | 1,364 | 1,956 | 1,575 | 1,360 | 2,455 |
| iasian | 1,334 | 1,281 | 1,275 | 0,897 | 1,229 | 1,448 | 0,000 | 0,924 | 1,316 | 2,891 | 2,348 | 1,653 |
| igend | 1,372 | 1,331 | 1,345 | 1,094 | 1,231 | 1,364 | 0,924 | 0,000 | 1,352 | 2,689 | 2,280 | 1,691 |
| lifesat | 0,939 | 1,570 | 2,244 | 1,486 | 1,723 | 1,956 | 1,316 | 1,352 | 0,000 | 2,677 | 2,423 | 0,895 |
| ideoldem | 3,312 | 2,112 | 2,671 | 2,326 | 2,324 | 1,575 | 2,891 | 2,689 | 2,677 | 0,000 | 0,404 | 2,917 |
| ideolrep | 2,809 | 1,718 | 2,229 | 1,903 | 2,031 | 1,360 | 2,348 | 2,280 | 2,423 | 0,404 | 0,000 | 2,386 |
| polar | 0,982 | 1,706 | 2,599 | 1,667 | 2,077 | 2,455 | 1,653 | 1,691 | 0,895 | 2,917 | 2,386 | 0,000 |

1. **Linkage-based clustering:** Among the many possible clustering methods I chose a [linkage-based algorithm](https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage) using a [complete distance measure](https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage) to calculate the distance between clusters. Other cluster distance measures can also be tested in code implementation.

For many of the steps I used an existing library, [DTAIDistance](https://dtaidistance.readthedocs.io/en/latest/). However, since I could not reconstruct the exact mechanisms of some of the functions used, I also successfully reproduced the results using the clustering package of [SciPy](https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html) directly, which is used by DTAIDistance. Check the code for more on that.

## Results

Following the preliminary results.



Cluster 1

Cluster 2

Cluster 3

Cluster 4

I decided for a cutoff value that results in 4 clusters. This could be further consolidated by using measures for cluster quality. It also depends on other parameter settings that might need further robustness checks.

The following table shows the cluster results and some statistical methods for each domain.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Domain cluster | Cluster 1 | |  | Cluster 2 | | |  | Cluster 3 | | |  | Cluster 4 | | | |
|  | ideolrep | ideoldem |  | polar | lifesat | negaffect |  | iasian | easian | igend |  | iafric | egend | eafric | posaffect |
| Curve length | 64,79 | 67,64 |  | 57,46 | 59,28 | 55,13 |  | 56,48 | 57,56 | 56,39 |  | 61,36 | 59,33 | 57,46 | 58,49 |
| Variance | 0,017 | 0,017 |  | 0,049 | 0,059 | 0,071 |  | 0,048 | 0,046 | 0,049 |  | 0,053 | 0,041 | 0,064 | 0,058 |
| SD | 0,131 | 0,132 |  | 0,221 | 0,243 | 0,266 |  | 0,218 | 0,214 | 0,221 |  | 0,230 | 0,203 | 0,253 | 0,241 |

## Limitations/Further considerations

* Argument for selection of maximal allowed shift to be considered when comparing two time series. Currently no limit. Does not really seem reasonable. Typically a window of 10% of the time series length is chosen ([Ref](https://www.researchgate.net/profile/Chotirat-Ratanamahatana/publication/216301292_Everything_you_know_about_dynamic_time_warping_is_wrong/links/02bfe5118865acbad4000000/Everything-you-know-about-dynamic-time-warping-is-wrong.pdf)).
* Robustness of result depending on setting of parameter (allowed shift window, cluster distance measure, cluster quality measures)
  + Changing these values has influence on the final cluster. I did not check the degree of change in detail yet. But for the shift window it seems not to be much. Selected method for cluster distance measure might have a bigger influence, but I think the complete method is a good choice as it does not use any kind of averaging.
* Method (DTW plus clustering) to difficult? It can appear like a black box. We can counteract that appearance by showing based on simple statistics that the clusters make sense or that they are bring about useful results.
* Currently used package does not allow for much editing of the output graphs. This would be something I need to get back to if we decide this analysis is relevant.