Introduction

The Dynamic Time Warping (DTW) algorithm measures the similarity between two time series that might be of different length and do not synchronise perfectly. The algorithm minimises the effects of shifting and distortion in time by dynamically transforming the time series in order to detect similar shapes with different phases. The algorithm has first been introduced by Bellman and Kalaba in 1959 [1] and applied to speech recognition in the 1970s [2] and [3]. The algorithm has been adapted to statistical arbitrage to detect and exploit the lead-lag structure between two time series [4].

Algorithm

Let $A = (a_1, a_2, ..., a_N), N \in \mathbb{N}$ and $B = (b_1, b_2, ..., b_M), M \in \mathbb{N}$ be two time series sampled at equidistant points in time. In order to compare A and B on some feature space Φ, one needs to use a local distance measure $\Phi \times \Phi \rightarrow \mathbb{R}^+$, called **distance function** or **cost function**. The DTW algorithm arranges all sequence points by minimising the distance function. The algorithm starts by building the distance matrix $C \in \mathbb{R}^{N \times M}$ representing all pairwise distances between A and B. This distance matrix called the **local cost matrix** for the alignment of two sequences A and B is defined as:

$$C_l \in \mathbb{R}^{N \times M} : c_{i,j} = ||a_i - b_j||, i \in [1..N], j \in [1..M] \tag{1}$$

Given C_l, the algorithm then constructs the **alignment path**, called **warping path** as a sequence of points $p = (p_1, p_2, ..., p_I)$ with $p_l = (p_i, p_j) \in [1..N] \times [1..M]$ for $l \in [1..I]$ satisfying the following three criteria:

- **Boundary condition:** The starting and ending points of the warping path must be the first and last points of the aligned sequence, i.e. $p_1 = (1,1)$ and $p_I = (N, M)$.
- **Monotonicity condition:** $n_1 \leq n_2 \leq ... \leq n_I$ and $m_1 \leq m_2 \leq ... \leq m_I$, which preserves the time-ordering of points.
- **Step-size condition:** It limits the changes in the points of the aligned sequence to $p_{l+1} - p_l = \{(1,1), (0,1), (1,0)\}$.

The **distance** or **cost function** associated with the **alignment** or **warping path** computed with respect to the local cost matrix C is:

$$c_p(A, B) = \sum_{l=1}^{I} c(a_{n_l}, b_{m_l}) \tag{2}$$

The **warping path** has a minimum cost (distance) associated with the alignment called the **optimal warping path**, called P^*. The DTW algorithm finds the optimal warping path

$$DTW(A, B) = c_{p^*}(A, B) = \min_{p \in P^{N \times M}} c_p(A, B) \tag{3}$$

recursively by dynamic programming using the step-size and bounday conditions.

$$DTW(i, j) = cost + \min(DTW(i-1, j), DTW(i, j-1), DTW(i-1, j-1)) \tag{4}$$

One of the limitation of the above algorithm is to allow one element of the array A to match an unlimited number of elements of the array B. We can define a window constraint to restrict the number of elements one can match.

Implementation

A simple implementation of the DTW algorithm is implemented in the `.quantQ.dtw` namespace.

- The cost function (1) is implemented in the function `.quantQ.dtw.cost`. It takes two arrays of floating numbers as parameter.

- The warping path (2) is implemented in the function `.quantQ.dtw.warpingPath`.
- The optimal warping path (3) is implemented in the function `.quantQ.dtw.optimalWarpingPath`.
- The optimal warping path with a window constraint is implemented in the function `.quantQ.dtw.optimalWarpingPathWindow`.

Bibliography

[1] R. Bellman and R. Kalaba, “On adaptive control processes”, *Automatic Control, IRE Transactions on*, vol. 4, no. 2, pp. 1–9,1959. [Online]. Available: http://ieeexplore.ieee.org/xpls/absall.jsp?arnumber=1104847

[2] C. Myers, L. Rabiner, and A. Rosenberg, “Performance tradeoffs in dynamic time warping algorithms for isolated word recognition”, *Acoustics, Speech, and Signal Processing* [see also *IEEE Transactions on Signal Processing*], *IEEE Transactions on*, vol. 28, no. 6, pp. 623–635, 1980. [Online]. Available: http://ieeexplore.ieee.org/xpls/absall.jsp?arnumber=1163491

[3] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition”, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, 1978. [Online]. Available: http://ieeexplore.ieee.org/xpls/absall.jsp?arnumber=1163055

[4] Stübinger, Johannes. “Statistical arbitrage with optimal causal paths on high-frequency data of the S&P 500.” *Quantitative Finance 19.6* (2019): 921-935.