**Alignment of Two Time Series Using Dynamic Time Warping (DTW)**

Ruhul Amin

Department of Electrical and Computer Engineering

Texas A&M University

**Abstract**

Here I am implementing the Dynamic Time Warping (DTW) to align two time sequence and two random signals. This algorithm is popular to pattern detection and find the similarity between two signals. In this report, I implement the Dynamic Time Warping to align two arbitrarily selected sequences, and to align two random ECG signals from the MIT arithmetic database.

**Introduction**

Dynamic Time Warping is a very powerful algorithm and it has been used for a long time in various research field. This report contains three major parts. In the first part I write the basic DTW codes and describe their implementations. In the second part, I implement my code from the first part to align two arbitrary signals and visualize the original and aligned signals. In the third part, I implement the DTW code to align two ECG signals.

**Part one:**

In this part the DTW algorithm is created using Python Programming Language. Firs two arbitrary sequences have been created. Since time series analysis, dynamic time warping is an algorithm for measuring similarity between two temporal sequences which may vary in time. So, the two time series which have been created are almost similar. After that the time series are visualized using the appropriate python command. From the graph, it is clear that both the signals show similar behavior. The both have a pick and around the peak both the slopes are downwards.

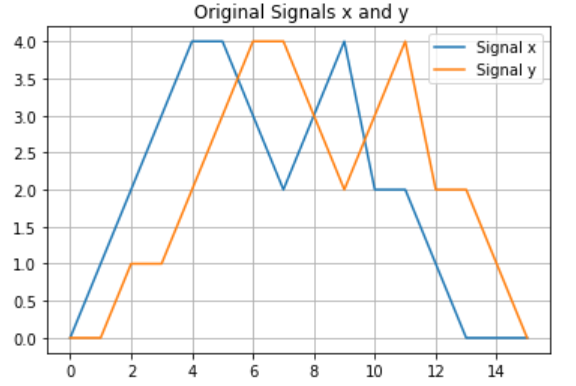


Figure: Two similar arbitrary signals

The goal is to find the distance between all pair of points in the two signals. The lesser distance implies that these points may be candidates to be matched together. The distance used here is Euclidean distance. A simple function is used to visualize the distance matrix.

In order to create a mapping between the two signals, it needs to create a path which should started from (0, 0) and should end at the point (m, n) where (m, n) is the end point of two signals. Now to find the accumulated value i use three technique: move along with the row, move along with the column and move diagonally. After finding the cumulative value matrix, now it needs to find the optimal path using the backtracking technique.

**Part Two:**

In this part two arbitrary signal is aligned using the code from part one. Here, I caret an array reverse\_y to align the two signals using the values found in the path which are the differences between the two arbitrary signals. After running the code the aligned signal is:

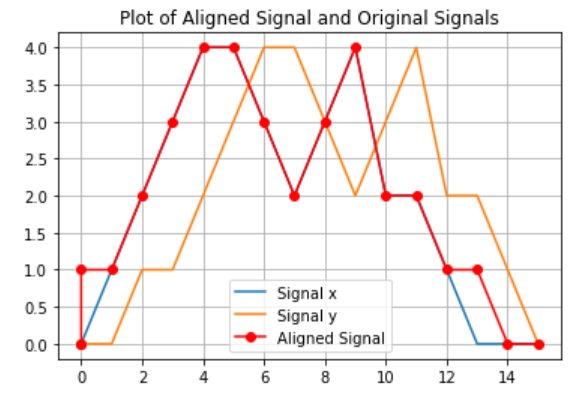


Figure: Two similar arbitrary signals and aligned signa

**Part three:**

In this part two random ECG signals are downloaded from the MIT algorithm database. Using my code I created in part I can align this two ECG signals.

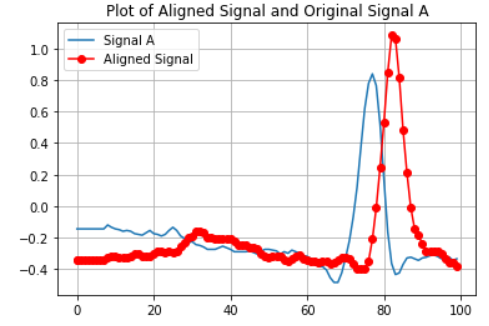


Figure: ECG signals and their aligned signal

**Conclusion:**

This project completely describes how two arbitrary signal can be aligned using the Python programming language. I tested my code with different arbitrary signals and it is capable to align any two signals perfectly and correctly.