Dynamic Time Warping with Python
Data Science Perspective

Empowering Data Scientists to unravel patterns in time-series data

Muhammad Adeel Sultan Khan

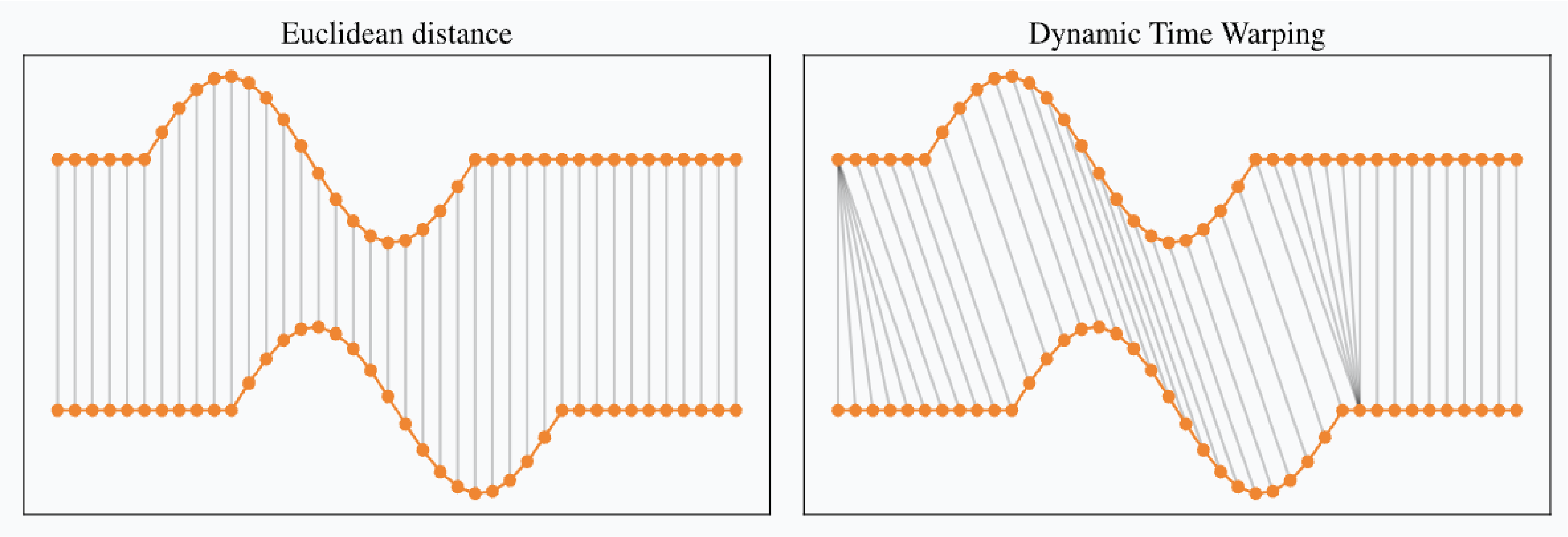
Agenda

- Dynamic Time Warping in TimeSeries Data
- Alignment based metrics
- What is Dynamic Time Warping
- Understanding Dynamic Time Warping
- Principles of Dynamic Time Warping
- Applications of Dynamic Time Warping
- How does Dynamic Time Warping work?
- DTW Mathematical Formulation
- DTW Optimization Problem
- Summary of DTW Steps
- Key Features of DTW
- Advantages of Dynamic Time Warping
- Limitations of DTW
- Enhancements and Variations of DTW
- DTW Algorithm – Key Features

Dynamic Time Warping in TimeSeries Data

- Dynamic time warping (DTW) is a way of comparing two, temporal sequences that don't perfectly sync up through mathematics. The process is commonly used in data mining to measure the distance between two time series. It's also a useful method in fields like financial markets and speech recognition.
- Dynamic Time Warping is used to compare similarity or calculate distance between two arrays or time series with different length.
- Dynamic Time Warping (DTW) is an algorithm designed to compare two sequences and measure their similarity by finding an optimal alignment between them. This is achieved by warping the time axis of the sequences to align them in a way that minimizes the distance between corresponding points.
- For instance, consider two sequences representing the same speech sound but spoken at different speeds. DTW can stretch or compress the time axis to align the sequences, allowing for a meaningful comparison.
- Dynamic Time Warping (DTW) is a powerful algorithm used in time series analysis to measure the similarity between two temporal sequences. Unlike traditional distance metrics like Euclidean distance, DTW can handle sequences of different lengths. It can align sequences that may be out of sync, making it particularly useful in fields such as speech recognition, gesture analysis, and finance.

Alignment based metrics



Here, we are computing similarity between two time series using either Euclidean distance (left) or Dynamic Time Warping (DTW, right), which is an instance of alignment-based metric that we will present in more details later in this tutorial. In both cases, the returned similarity is the sum of distances between matched features. Here, matches are represented by gray lines and the distance associated to a match between i-th feature in time series x and j-th feature in time series x′ is d(xi,xj′). Note how DTW matches distinctive patterns of the time series, which is likely to result in a more sound similarity assessment than when using Euclidean distance that matches timestamps regardless of the feature values.

Dynamic Time Warping is equivalent to minimizing Euclidean distance between aligned time series under all admissible temporal alignments.

https://rtavenar.github.io/blog/dtw.html

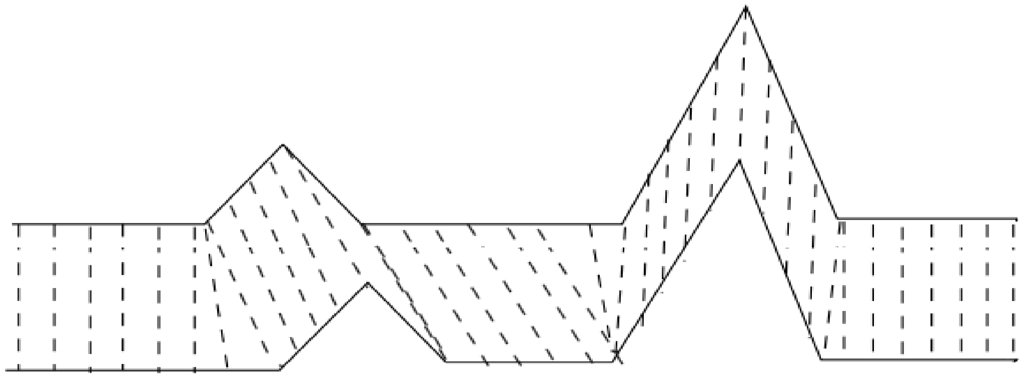
What is Dynamic Time Warping

- In time series analysis, **dynamic time warping** (**DTW**) is an algorithm 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.

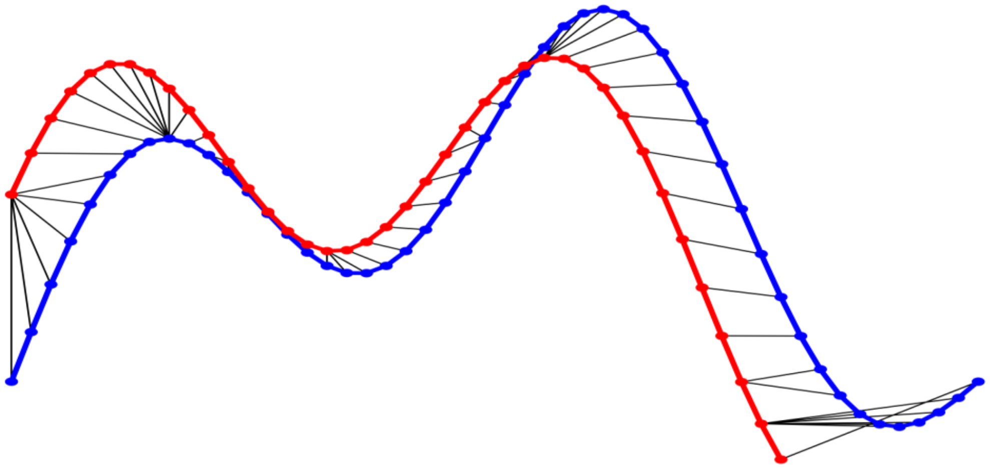
- DTW has been applied to temporal sequences of video, audio, and graphics data — indeed, any data that can be turned into a one-dimensional sequence can be analyzed with DTW. A well-known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. It can also be used in partial shape matching applications.

- Dynamic Time Warping is used to compare similarity or calculate distance between two arrays or time series with different length.

- DTW is a family of algorithms which compute local stretch or compression to apply to the time axes of two timeseries in order to optimally map one (query) onto the other (reference). DTW outputs the remaining cumulative distance between the two and the mapping itself (warping function). DTW is widely used e.g. for classification and clustering tasks in econometrics, chemometrics and general timeseries mining.



Dynamic time warping between two piecewise linear functions. The dotted line illustrates the time-warp relation. Notice that several points in the lower function are mapped to one point in the upper function, and *vice versa*.



Understanding Dynamic Time Warping

- One of the most salient uses of dynamic time warping is in speech recognition - determining whether one phrase matches another, even if it the phrase is spoken faster or slower than its comparison. You can imagine that this comes in handy to identify the “wake words” used to activate your Google Home or Amazon Alexa device

- Dynamic time warping is a useful, powerful technique that can be applied across many different domains. Once you understand the concept of dynamic time warping, it’s easy to see examples of its applications in daily life, and its exciting future applications. Consider the following uses:

  - ***Financial markets*** - comparing stock trading data over similar time frames, even if they do not match up perfectly. For example, comparing monthly trading data for February (28 days) and March (31 days).

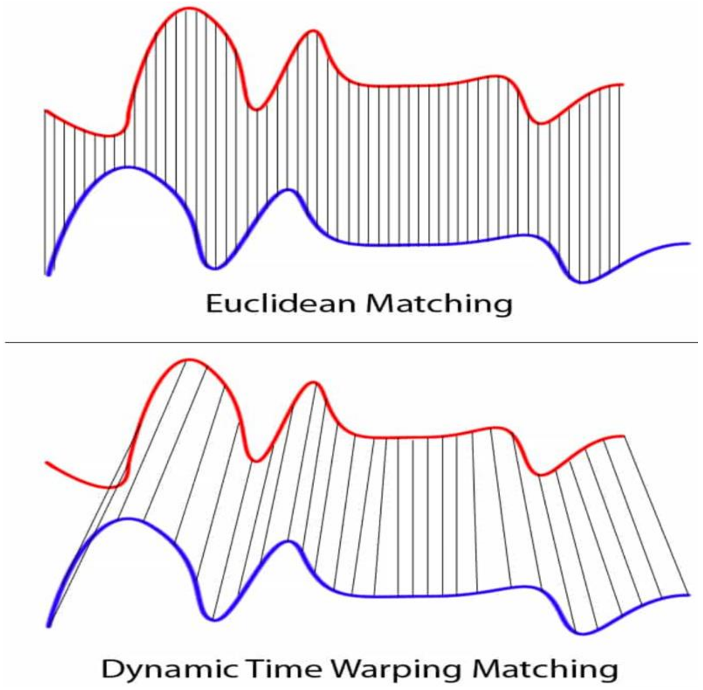
  - ***Wearable fitness trackers*** - more accurately calculating a walker’s speed and the number of steps, even if their speed varied over time.

  - ***Route calculation*** - calculating more accurate information about a driver’s ETA, if we know something about their driving habits (for example, they drive quickly on straightaways but take more time than average to make left turns).

https://www.databricks.com/blog/2019/04/30/understanding-dynamic-time-warping.html

Principles of Dynamic Time Warping

- The objective of time series comparison methods is to produce a *distance metric* between two input time series. The similarity or dissimilarity of two-time series is typically calculated by converting the data into vectors and calculating the Euclidean distance between those points in vector space.
- Dynamic time warping is a seminal time series comparison technique that has been used for speech and word recognition since the 1970s with sound waves as the source
- 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, i.e. if j > i are indices from the first sequence, then there must not be two indices l > k in the other sequence, such that index i is matched with index l and index j is matched with index k, and vice versa



Applications of Dynamic Time Warping

- **Speech recognition**

  DTW can determine if one phrase matches another, even if the phrases are spoken at different speeds. This is useful for identifying wake words for devices like Google Home and Amazon Alexa.

- **Financial markets**

  DTW can compare stock trading data over similar time frames, even if the data doesn't match up perfectly.

- **Video, audio, and graphics data**

  DTW can analyze temporal sequences of any data that can be turned into a one-dimensional sequence.

- **Satellite image time series classification**

  DTW can be used to provide an overview of the state of rangelands and cropland in semi-arid regions.

- **Medical Diagnosis**
- In the medical field, DTW is used to align and compare biological signals, such as ECG or EEG readings, helping in the diagnosis of conditions by comparing patient data with reference models.
- **Gesture Recognition**
- In gesture recognition, DTW helps align different gestures for comparison, even when performed at different speeds or with slight variations.
- DTW has several advantages over other methods, including:
  - It's accurate for two sequences with different lengths.
  - It's fast to calculate.
  - It's not sensitive to time delays or uneven sampling time

How does Dynamic Time Warping work?

- **Step 1: Distance Matrix Construction**

The first step in DTW involves constructing a distance matrix between the two sequences. Each element of the matrix represents the distance (typically Euclidean) between corresponding points in the two sequences.

- **Step 2: Cost Matrix and Accumulated Cost**

Next, a cost matrix is created by accumulating the minimum distances from the start of the sequences to the current point. This accumulated cost represents the optimal path's cumulative distance up to that point.

- **Step 3: Optimal Path Finding**

The optimal alignment path is found by tracing back from the last element in the cost matrix to the first element. This path represents the best alignment between the two sequences, minimizing the total distance.

- **Step 4: Warping Path**

The warping path shows how one sequence can be warped (stretched or compressed) along the time axis to best match the other sequence.

DTW Mathematical Formulation

- Given two sequences $A = \{a_1, a_2, \dots, a_n\}$ and $B = \{b_1, b_2, \dots, b_m\}$, where a_i and b_j are elements of the sequences A and B respectively, DTW computes the minimum cumulative distance between them.

1. **Cost Matrix**: Define a cost matrix C of size $n \times m$, where $C(i, j)$ represents the cost (or distance) of aligning a_i with b_j. The cost is typically calculated using a distance metric, such as the Euclidean distance:
    1. $C(i, j) = \text{distance}(a_i, b_j) = |a_i – b_j|$
2. **Accumulated Cost Matrix**: Construct an accumulated cost matrix D where each element $D(i, j)$ represents the minimum cumulative cost to align the first iii elements of A with the first j elements of B:
    2. $D(i, j) = C(i, j) + \min \begin{cases} D(i-1, j) \\ D(i, j-1) \\ D(i-1, j-1) \end{cases}$
    3. Here,
        2. $D(i-1, j)$ corresponds to an insertion,
        3. $D(i, j-1)$ corresponds to a deletion
        4. $D(i-1, j-1)$ corresponds to a match (or diagonal move).
3. **Boundary Conditions**: The boundary conditions are initialized as follows: $D(1,1)=C(1,1) \\ D(i,1)=D(i−1,1)+C(i,1) \quad \text{for} i=2,…,n \\ D(1,j)=D(1,j−1)+C(1,j) \quad \text{for} j=2,…,m$
4. **Optimal Warping Path**: The optimal warping path $W = \{(i_1, j_1), (i_2, j_2), \dots, (i_L, j_L)\}$ is a sequence of matrix indices that minimizes the cumulative distance. This path is found by backtracking from $D(n, m)$ to $D(1, 1)$ by following the minimum cost direction at each step. The overall DTW distance is given by: $\text{DTW}(A, B) = D(n, m)$

DTW Optimization Problem

- Dynamic Time Warping (DTW) is a similarity measure between time series. Let us consider two time series x=(x0,…,xn−1) and y=(y0,…,ym−1) of respective lengths n and m. Here, all elements x_i and y_j are assumed to lie in the same d-dimensional space. In tslearn, such time series would be represented as arrays of respective shapes *(n, d)* and *(m, d)* and DTW can be computed using the following code:

```
from tslearn.metrics import dtw, dtw_path

dtw_score = dtw(x, y)
# Or, if the path is also an important information:
optimal_path, dtw_score = dtw_path(x, y)
```

DTW Optimization Problem

DTW between x and y is formulated as the following optimization problem:

$$DTW(x, y) = \min_{\pi} \sqrt{\sum_{(i,j)\in\pi} d(x_i, y_j)^2}$$

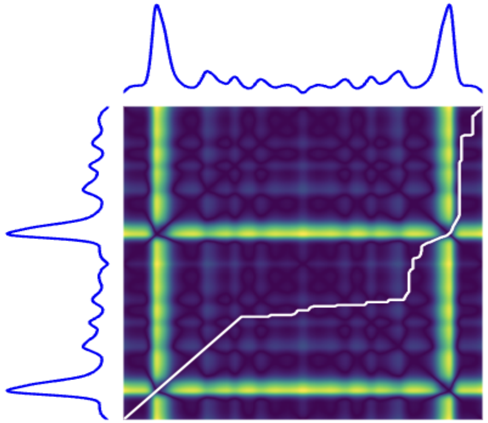
where 'd(x_i, yi)' is the distance between points 'x_i' from series A and 'y_j' from series B , and the sum is taken over all points (i, j) in the optimal alignment path.

where $\pi = [\pi_0, \dots, \pi_K]$ is a path that satisfies the following properties:

- it is a list of index pairs $\pi_k = (i_k, j_k)$ with $0 \leq i_k < n$ and $0 \leq j_k < m$
- $\pi_0 = (0, 0)$ and $\pi_K = (n-1, m-1)$
- for all $k > 0$, $\pi_k = (i_k, j_k)$ is related to $\pi_{k-1} = (i_{k-1}, j_{k-1})$ as follows:
  - $i_{k-1} \leq i_k \leq i_{k-1} + 1$
  - $j_{k-1} \leq j_k \leq j_{k-1} + 1$

Here, a path can be seen as a temporal alignment of time series such that Euclidean distance between aligned (ie. resampled) time series is minimal.

The following image exhibits the DTW path (in white) for a given pair of time series, on top of the cross-similarity matrix that stores $d(x_i, y_j)$ values.



Summary of DTW Steps

1. Calculate the cost matrix C(i,j).

2. Compute the accumulated cost matrix D(i,j).

3. Find the optimal warping path by backtracking from D(n,m) to D(1,1).

4. The final DTW distance is D(n, m).

Key Features of DTW

- **Handling of Non-Linear Alignments:** DTW can handle non-linear alignments between sequences, which is crucial when comparing sequences that may be out of sync or have different lengths.
- **Robustness to Temporal Variations:** Unlike simple distance measures, DTW is robust to variations in speed or timing within the sequences, making it ideal for applications like speech or motion analysis.
- **Flexibility:** DTW can be applied to any time series data, regardless of the domain, as long as the sequences can be represented numerically.

Advantages of Dynamic Time Warping

- **Handling Different Lengths:** DTW can compare sequences of different lengths, making it versatile for various applications where time series data may not be perfectly aligned.
- **Alignment of Complex Sequences:** DTW is particularly useful in aligning complex sequences that may have local shifts in time, such as varying speech rates or irregular patterns in financial data.
- **Effective in Noisy Environments:** DTW's ability to warp time allows it to effectively compare sequences even in noisy environments where traditional methods might fail.

Limitations of DTW

- **Computational Complexity:** DTW can be computationally intensive, especially with long sequences, as the algorithm's complexity is quadratic in the length of the sequences.
- **Over-Warping:** Without constraints, DTW might over-warp sequences, leading to unnatural alignments. Adding constraints like Sakoe-Chiba bands can mitigate this issue.
- **Sensitivity to Scaling:** DTW does not inherently handle differences in amplitude or scaling between sequences, so pre-processing like normalization is often required.

Enhancements and Variations of DTW

- **Global Path Constraints:** Constraints like the Sakoe-Chiba band or Itakura parallelogram can be applied to limit the warping path, reducing computational cost and preventing over-warping.
- **Multidimensional DTW (MD-DTW):** For complex data with multiple dimensions (e.g., motion capture data), MD-DTW can be used to align and compare multidimensional time series.
- **Weighted DTW:** Weighted DTW applies different weights to different parts of the sequence, allowing for more flexible and context-aware comparisons.

DTW Algorithm – Key Learnings

- The Dynamic Time Warping is a good non-parametric algorithm, therefore it is simple to be used because it does not require any parameter tuning.
- It is a robust benchmark for Time Series Classification problems.
- It is computationally expensive.
- In case of known characteristic time scale of a phenomenon (e.g., periodicity) and in case of a specific pattern to classify, the Dynamic Time Warping can be used as starting point for a Classification task.

Demo Jupyter notebook

- https://tslearn.readthedocs.io/en/stable/user_guide/dtw.html
- https://dtaidistance.readthedocs.io/en/latest/usage/dtw.html
- https://pypi.org/project/dtw-python/

References

- https://pypi.org/project/dtw-python/
- https://tslearn.readthedocs.io/en/stable/user_guide/dtw.html#dtw
- https://www.geeksforgeeks.org/dynamic-time-warping-dtw-in-time-series/