Dynamic Time Warping

Uri Shaham

January 20, 2025

1 Similarity of time series

Let $s = (s[1], \ldots, s[m]), t(t[1], \ldots, t[n])$ be two discrete time series of lengths m, n, respectively. If $m = n$ we could use Euclidean distance to measure the similarity between them. However, what if the sequences are, for example $s =$ "*hello*" and $t =$ "*hheeeeellllllllllllllllllllloo*"? clearly, they are similar in a temporal sense. However, they correspond to different (and varying) speeds. DTW is an alignment algorithm that returns the best alignment between any two discrete temporal sequences.

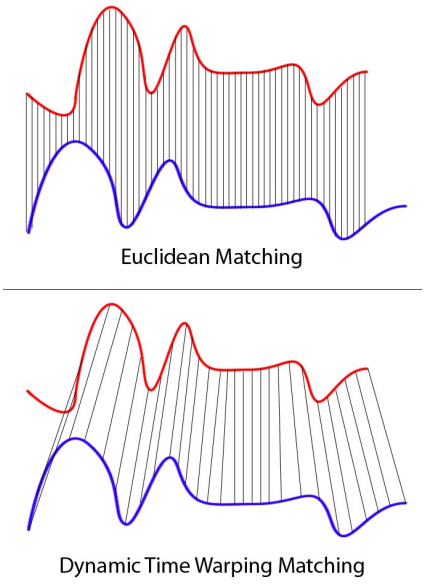

Figure 1: Euclidean similarity (top) DTW similarity.

2 Dynamic time warping

DTW computes an optimal alignment between s, t, under the following restrictions:

- Continuity of time: any index in s is matched with at least one index in t and vice versa.

- Boundary conditions: $s[1]$ is matched with $t[1]$, $s[m]$ is matched with $t[n]$.
- Monotonicity: if $s[i]$ and $s[j]$ are matched with $t[k]$ and $t[l]$, respectively, and $i \leq j$ then $k \leq l$ (i.e., the alignment does not go back in time)

The warped sequences are s', t' of respective length $m' = n'$, and $m' \geq m$, $n' \geq n$.

The alignment is performed using dynamic programming, using a $(m+1) \times (n+1)$ matrix, DTW, such that $\text{DTW}_{i,j}$ is the alignment cost between $(s[1], \ldots, s[i])$ and $(t[1], \ldots, t[j])$.

Algorithm 1 Dynamic time warping

Require: s, t, cost measure $c(\cdot, \cdot)$, (e.g., $c(x, y) = (x - y)^2$)
  Initialize:
  DTW = array[0...m, 0...n]
  for $i = 1$ to m **do**
    $\text{DTW}_{i,0} = \infty$
  end for
  for $j = 1$ to n **do**
    $\text{DTW}_{0,j} = \infty$
  end for
  Compute alignment:
  for $i = 1$ to m **do**
    for $j = 1$ to n **do**
      $\text{DTW}_{i,j} = c(s[i], t[j]) + \min\{\text{DTW}_{i,j-1}, \text{DTW}_{i-1,j}, \text{DTW}_{i-1,j-1}\}$
    end for
  end for

Example:

	0	1	1	2	2	3	5
0	0	inf	inf	inf	inf	inf	inf
1	inf	0	0	1	2	4	8
2	inf	1	1	0	0	1	4
3	inf	3	3	1	1	0	2
5	inf	7	7	4	4	2	0
5	inf	11	11	7	7	4	0
5	inf	15	15	10	10	6	0
6	inf	20	20	14	14	9	1

Figure 2: DTW between $s =$ "1235556" and $t =$ "12335"

Interpretation:

- A horizontal move represents deletion. That means t accelerated during this interval.
- A horizontal move represents insertion. That means s accelerated during this interval.
- A diagonal move represents match. That means s and t had the same pace during this interval.

2.1 Adding a locality constraint

It is often desired to add a locality constraint, allowing to match $s[i]$ and $t[j]$ only if $|i-j|$ is at most w, a window parameter (of course, w needs to be at least $|n-m|$, otherwise an alignment is not possible). The modified algorithm is left as a homework exercise.

Homework

1. Write the Pseudo code of DTW with locality constraint, implement it and provide the alignment on some interesting case.
2. What is the time and space complexity of DTW with and without locality constraint?
3. Find a small time series classification dataset, and implement a nearest neighbor classifier with DTW as the distance measure.