BRTA Proof

1 Dynamic Time Warping

Dynamic Time Warping (DTW) [1] is a well known algorithm to find an optimal alignment (or match) in order to measure similarity between two sequences such as time-series data.

Algorithm 1 DTW Pseudo Code

function DTW($s[1..n], t[1..m]$)
  $DTW \leftarrow [0..n, 0..m]$
  for $i \leftarrow 1$ to n **do**
    $DTW[i, 0] \leftarrow \infty$
  end for
  for $i \leftarrow 1$ to m **do**
    $DTW[0, i] \leftarrow \infty$
  end for
  $DTW[0, 0] \leftarrow 0$
  for $i \leftarrow 1$ to n **do**
    for $j \leftarrow 1$ to m **do**
      $cost \leftarrow cost(s[i], t[j])$
      $DTW[i, j] \leftarrow cost + minimum(DTW[i-1, j], DTW[i, j-1], DTW[i-1, j-1])$
    end for
  end for
  return $DTW[n, m]$
end function

The above pseudo-code demonstrates the implementation of DTW for two sequences s and t which are strings of discrete symbols. DTW[i, j] ($DTW_{i,j}$ in short) is the distance between s[1:i] and t[1:j] with the best alignment.

2 Proof

Let t be a time-series and $\hat{s}$ denote its local optimal. Define s as a constant time-series whose value is always $\hat{s}$, i.e., $s_i = \hat{s}$ for all i. We implemented DTW

[1] https://en.wikipedia.org/wiki/Dynamic_time_warping

where the inputs are t and s (as explained above in Section 1). For the cost function, which is intentionally under-specified as $cost(s[i], t[j])$ in the pseudo code in Section 1, we use the absolute-value function $cost(x, y) = |x - y|$. Furthermore, we define $c_j = cost(\hat{s}, t_j) = |\hat{s} - t_j|$ as the "point-wise" cost. Also, let

$$\bar{c}_j = \sum_{k=1}^{j} c_k = c_1 + \ldots + c_j$$

be the cumulative point-wise cost.

Lemma 2.1. *For all $1 \leq i \leq n$ and $1 \leq j \leq m$,*

$$DTW_{i,j} \geq \bar{c}_j.$$

In particular, if $i \leq j$, then

$$DTW_{i,j} = \bar{c}_j.$$

Proof. We prove Lemma 2.1 by induction. We start with proving the first part of the statement and then, we show that whenever $i \leq j$, the second part holds as well.

Base case: By the definition of the DTW algorithm and in particular, using the fact that $DTW_{0,j} = DTW_{i,0} = \infty$, (as initialized in the DTW cost matrix D to handle the base-case cost computation) we have the following:

- $DTW_{1,j} = \bar{c}_j$ for all j, and
- $DTW_{i,1} = i \cdot \bar{c}_1 \geq \bar{c}_1$ for all i.

To complete the induction, assume that the statement of the lemma holds for every i and j with $i + j < i_0 + j_0$. We prove that the statement holds for DTW_{i_0,j_0}, as well. By the definition of the algorithm, DTW_{i_0,j_0} is the minimum of the following three values.

- $DTW_{i_0-1,j_0} + c_{j_0} \geq \bar{c}_{j_0} + c_{j_0} \geq \bar{c}_{j_0}$.
- $DTW_{i_0,j_0-1} + c_{j_0} \geq \bar{c}_{j_0-1} + c_{j_0} = \bar{c}_{j_0}$.
- $DTW_{i_0-1,j_0-1} + c_{j_0} \geq \bar{c}_{j_0-1} + c_{j_0} = \bar{c}_{j_0}$.

Hence, in every of the mentioned cases, $DTW_{i_0,j_0} \geq \bar{c}_{j_0}$. This proves the first part of the lemma.

For the second part of the statement of the theorem, notice that if $i_0 \leq j_0$, then $i_0 - 1 \leq j_0 - 1$. By the induction hypothesis, $DTW_{i_0-1,j_0-1} = \bar{c}_{j_0-1}$. The algorithm takes the minimum of three values to compute DTW_{i_0,j_0}. One of these three values—the third one in the above-mentioned list—is $DTW_{i_0-1,j_0-1} + c_{j_0} = \bar{c}_{j_0}$. Thus, DTW_{i_0,j_0} is no more than $\bar{c}_{j_0}$. This completes the proof. $\square$

2.1 BRTA Formula

In Lemma 2.1, we showed the relationship between $DTW_{i,j}$ (the best alignment distance as computed by DTW between two time series s[1:i] and t[1:j]) and $\overline{c}_j$ (the cumulative point-wise cost as defined above). Note that in BRTA, we construct s_l as a constant time-series with local optimum value. To arrive at the BRTA formula, for the direct case (the inverse case can be proven similarly), we can calculate D_l, the distance to the local optimum, as:

$$D_l = \overline{c}_j = \sum_{k=1}^{j} c_k = c_1 + \ldots + c_j = \sum_{k=1}^{j} \hat{s}_l - \sum_{k=1}^{j} t_1 + \ldots + t_j \tag{1}$$

By definition, for the global optimum time-series s_g, $\hat{s_g} \geq \hat{s}_l$. Therefore to calculate the distance between the time-series t and globally optimum time-series s_g, we should calculate the following:

$$D_g = \sum_{k=1}^{j} \hat{s_g} - \sum_{k=1}^{j} t_1 + \ldots + t_j = D_l + \sum_{k=1}^{j} \hat{s_g} - \sum_{k=1}^{j} \hat{s}_l \tag{2}$$

In equation 2, D_l is computed once locally; j, and $\hat{s}_l$ are values that are reported as part of the summaries; and $\hat{s_g}$ is the maximum of all the received local optimums. Therefore, the need to report back the time-series t is eliminated in its entirety. We can now write D_g as :

$$D_g = D_l + \sum_{k=1}^{j} \hat{s_g} - \sum_{k=1}^{j} \hat{s}_l = D_l + Max_g - O_l \tag{3}$$

Equation 3 arrives at the BRTA formula for the distance to the global optimum time-series in the direct case. Here, $D_l = \overline{c}_j$ as computed from equation 1. Max_g is the maximum value of all the time-series (for a Non Functional Property) and O_l is the local optimum. The inverse case can be proven similarly. Please note that, as explained in the paper, for the final comparison we normalize the distances by dividing them by j, to level the effect of the number of points in each time-series. The reasoning is that the number of observations is not the goal of the optimization process, rather the goal is "closeness" to the optimum.