1 Introduction

Dynamic Time Warping(DTW) algorithm is based on the idea of dynamic programming, which can measure the similarity of two time series with inconsistent length. It is known to all that speech signal has strong randomness due to the fact that different pronunciation habits, different environment and different mood will lead to the phenomenon of different pronunciation duration. Dynamic Time Warping algorithm can solve the template matching problem of different pronunciation lengths. It is an earlier and classic algorithm in speech recognition (Senin,2008).

2 How it work

2.1 The sequence correspondence

According to the principle of nearest distance, DTW algorithm constructs the corresponding relationship between two sequence elements and evaluates the similarity of the two sequences (Müller, 2007). There are three rules: (1) one-way correspondence, no turning back, (2) one-to-one correspondence, no space, and (3) after correspondence, the distance is the nearest.

For example, suppose we have two sequences with unequal lengths. We need to find the corresponding relationship. The sequence takes a point every 10 milliseconds to form an array, as shown in the figure. The correspondence at the top meets the three requirements listed above but the following two correspondences are wrong. The blue third point in the lower left corner is connected with the first point, which does not follow the first requirement. The blue fourth point in the lower right corner directly skips the red fifth point to connect the sixth point, which does not follow the first requirement.

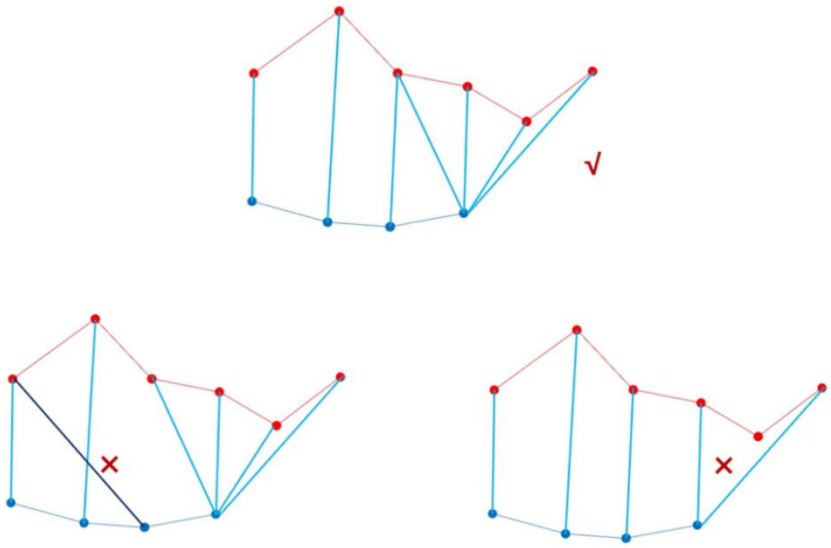


Figure1 Example of sequence correspondence

2.2 The calculation of distance

Suppose we have two sequences, which are A and B whose length and feature number are shown in the table below. This table can be expressed as the cumulative distance matrix that we need to fill in.

(1) For the leftmost column, D[i,0] = dis(A_i,B_0) + D[i-1,0]. For example, D[5,0] = dis(2,1) + D[4,0] = |2-1| + 20 = 21

	3	33								
	7	31								
	5	25								
	1	21								
	2	**21**								
Sequence A	**8**	20								
	9	13								
	4	5								
	3	2								
	1	0								
		1	**6**	**2**	**3**	**0**	**9**	**4**	**1**	**6**	**3**
	Sequence B										

(2) For the bottom column, D[0,j] = dis(A_0,B_j) + D[0,j-1]. For example, D[0,5] = dis(1,7) + D[0,4] = |1-9| + 9 = 17

	3	33								
	7	31								
	5	25								
	1	21								
	2	21								
Sequence A	**8**	20								
	9	13	7	11	11					
	4	5	4	5	5	8	12	12	15	17	18
	3	2	3	4	4	7	13	14	16	19	19
	1	0	5	6	8	9	**17**	20	20	25	27
		1	**6**	**2**	**3**	**0**	**9**	**4**	**1**	**6**	**3**
	Sequence B										

(3) For the other columns, D[i,j] = dis(A_i,B_j) + min(D[i-1,j], D[i,j-1], D[i-1,j-1]). For example, D[3,3]=dis(A3,B3)+min(D[3,2],D[2,3],D[2,2]) = |9-3| + min(11,5,5) = 6+5 =11

	3	33								
	7	31								
	5	25								
	1	21								
	2	21								
Sequence A	**8**	20								
	9	13	7	11	**11**					
	4	5	4	5	5	8	12	12	15	17	18
	3	2	3	4	4	7	13	14	16	19	19

	1	0	5	6	8	9	17	20	20	25	27
		1	**6**	**2**	**3**	**0**	**9**	**4**	**1**	**6**	**3**
		Sequence B									

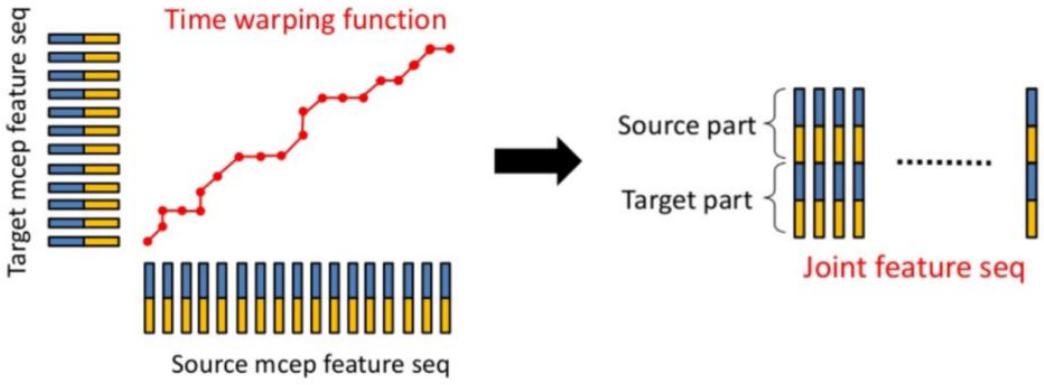
(4) After filling in the cumulative distance matrix according to the above rules, we need to start from the upper right corner and find the path from the lower left corner. The rule is to find the lower left three fill in the blanks with the smallest value as the next node.

Sequence A	3	33	23	19	16	19	23	18	20	16	23
	7	31	20	18	16	19	17	18	21	13	16
	5	25	19	13	12	16	15	15	15	12	14
	1	21	18	10	11	11	19	14	11	16	18
	2	21	13	9	10	12	16	11	12	16	17
	8	20	9	13	16	19	9	12	19	20	23
	9	13	7	11	11	14	8	13	20	18	23
	4	5	4	5	5	8	12	12	15	17	18
	3	2	3	4	4	7	13	14	16	19	19
	1	0	5	6	8	9	17	20	20	25	27
		1	6	2	3	0	9	4	1	6	3
		Sequence B									

Based on the above table, the result is [(A0,B0), (A1,B1), (A1,B2), (A1,B3), (A2,B4), ...(A9,B9)]

3.3 Apply to the speech emotion recognition

We can prepare data sets to obtain speech features under various emotions, which as the source feature. The voice input by the user is used as the target feature. According to the similarity, we can construct the corresponding relationship between source features and target features to form feature pairs. Based on the pairs, we can find the most consistent emotion, as shown in the following figure.



3.4 Package in python

pip install dtw
pip install dtw_c

pip install fastdtw
pip install librosa

References

Senin, P. (2008). Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, *855*(1-23), 40.
Müller, M. (2007). Dynamic time warping. *Information retrieval for music and motion*, 69-84.