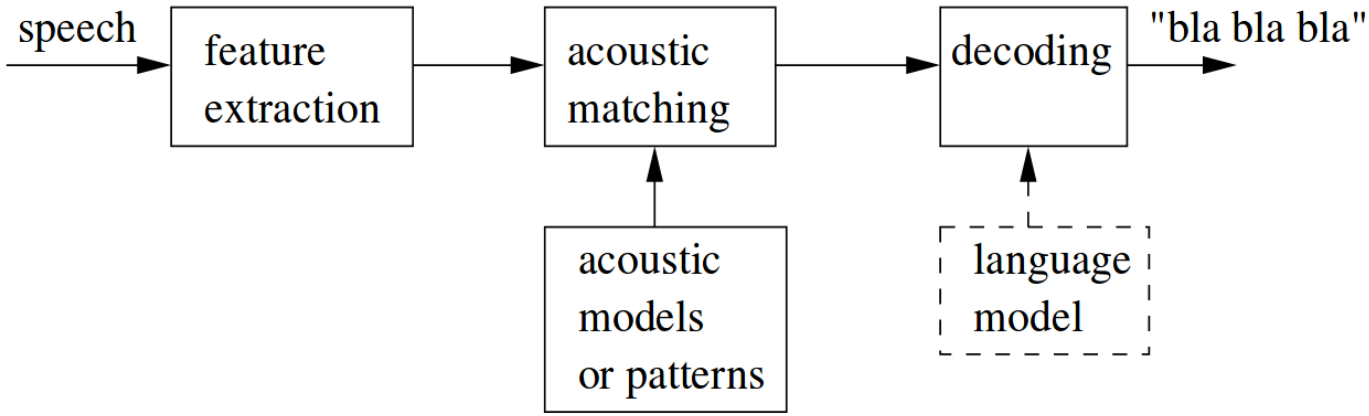
Spoken digit recognition with dynamic time warping and K nearest neighbours

National Institute of Technology Karnataka

30th January 2020

Basic speech recognition system

Structure of a recognizer



Acoustic matching

A human never says one thing twice in exactly the same way. Parameter vectors are always different ! Steps involved in acoustic matching are :

- Calculating distance between two vectors
- Use of a classifier / statistic modeling

Dynamic Time Warping

- In time series analysis, dynamic time warping (DTW) is one of the algorithms for measuring similarity between two temporal sequences, which may vary in speed.
- Since speech signals are almost always distinct in terms of speed, we use DTW.
- In general, DTW is a method that calculates an optimal match between two given sequences.
- The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension.

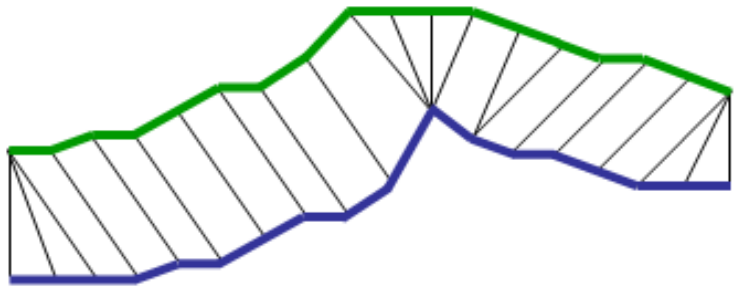


Figure: Similar sequences varying in speed

General steps

- The two sequences are arranged on the sides of a grid
- Inside each cell of the grid a distance measure can be placed, comparing the corresponding elements of the two sequences. This matrix is called Cost matrix.
- To find the best alignment between these two sequences one need to find a path through the grid which minimizes the total distance between them. Overall distance is found for all possible paths. This overall distance matrix is called Cost accumulation matrix.
- The overall distance is the minimum of the sum of the distances between the individual elements on the path from the start to the end.

Pseudo code

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
    DTW := array [0..n, 0..m]

    for i := 1 to n
        for j := 1 to m
            DTW[i, j] := infinity
    DTW[0, 0] := 0

    for i := 1 to n
        for j := 1 to m
            cost := d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                        DTW[i  , j-1],    // deletion
                                        DTW[i-1, j-1])    // match

    return DTW[n, m]
}
```

Let's try it out !

Consider two sequences :
X=1,1,2,4,3,5,3,2,3,2
Y=1,2,4,3,5,3,2,3,2,5
Construct the cost accumulation matrix for the above two sequences

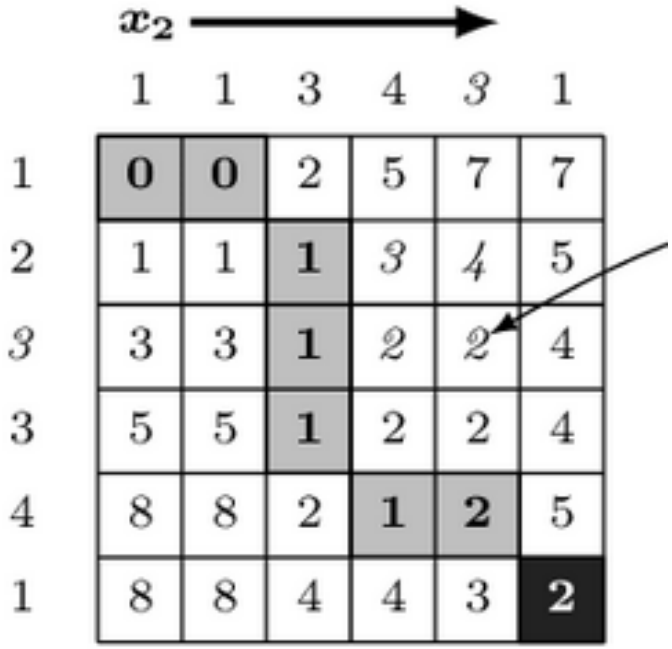
Answer

2	16	8	8	4	6	2	1	1	0	3
3	15	8	6	3	5	1	1	0	1	3
2	13	7	5	3	5	1	0	1	1	4
3	12	7	3	2	2	0	1	1	2	4
5	10	6	2	2	0	2	5	5	6	4
3	6	3	1	0	2	2	3	3	4	6
4	4	2	0	1	2	3	5	6	8	9
2	1	0	2	3	6	7	7	8	8	11
1	0	1	4	6	10	12	13	15	16	20
1	0	1	4	6	10	12	13	15	16	20
	1	2	4	3	5	3	2	3	2	5

Figure: Answer

Warping path

Backtracking to find optimal path



K Nearest neighbours

- Similar things exist in close proximity. "Birds of the same feather flock together".
- K-NN - Used for both classification and regression.
- Lazy learning - Brute force method

Steps involved

- Load the data
- Initialize K to your chosen number of neighbors
- Calculate the distance between the query example and the current example from the data.
- Add the distance and the index of the example to an ordered collection
- Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
- Pick the first K entries from the sorted collection
- Get the labels of the selected K entries
- The output is the label with the highest number of votes

Effect of changing K

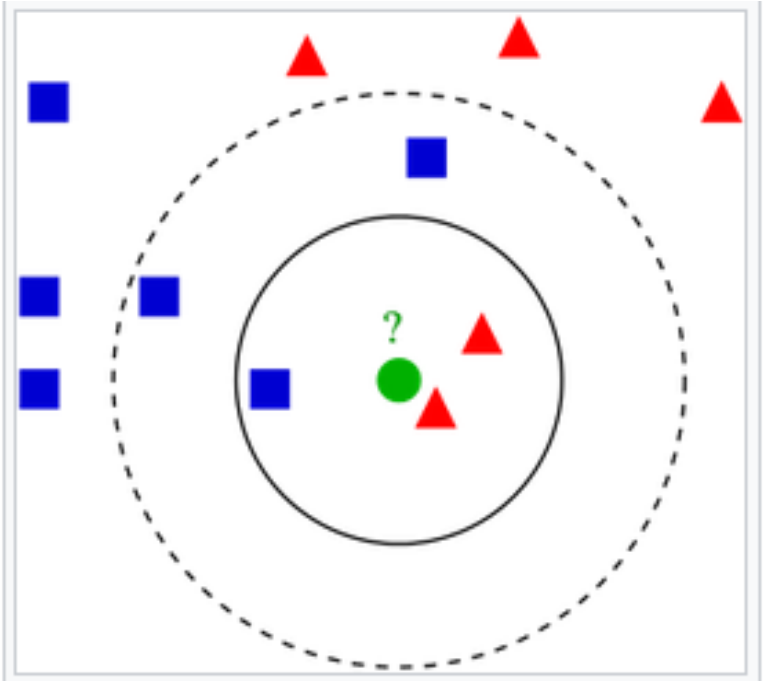


Figure: Answer