# Summary

This report gives a brief summary on how a speech recognition system can use DTW algorithm and HMM to recognize digits uttered by a speaker.

# Introduction

Speech recognition system basically does extraction of features of an utterance and classify it using any algorithm (here, we use DTW or HMM). In this case, MFCC was used to extract feature vectors from speech signals. The paper is divided into four sections. Section 3 deals with isolated digit recognition using DTW algorithm. Section 4 deals with recognition of first digit where unknown number of digits are uttered by a speaker. Section 5 deals with isolated digit recognition using HMM.

# Isolated Digit Recognition Using Dynamic Time Warping Algorithm

The input to this speech recognition system is the digits uttered by a speaker. The utterance can be a single digit or continuous digits, in this case we consider single digits (isolated digits). We extract the mfcc feature vectors and use it for recognition algorithms. Since same digit uttered by speakers may have different duration (i.e., speaking speeds), we use DTW to find out the similarity between different temporal sequences. We use DTW to measure a distance like quantity between two sequences of feature vectors, using this score between an unknown data and known data, we classify the unknown data to the class with which it has least score.

**Input:** Sequence of feature vectors for each digit uttered by a speaker are stored in a file in the following format: (mfcc file)

38 98

-1.129163e+01 8.201429e+00 -4.876589e+00 ....

-1.112325e+01 8.435195e+00 -4.208691e+00 ....

-1.074037e+01 7.301194e+00 -5.066467e+00 ....

First Line represents the dimension of each feature vector and number of feature vectors.

Here, 38 is dimension of feature vector and 98 is the number of feature vectors extracted for a digit uttered.

**Train Data**: List of mfcc files of recognized digit spoken by different speakers.

**Test Data**: List of mfcc files of unrecognized digit spoken by different speakers.

**3.1 Algorithm:** Using the input we find out distance matrix and cost matrix as follows:

For a test file of digit to be recognized, we first find out the score/similarity between the test file and each train file.

**Distance Matrix**: This matrix contains Euclidian distance between each feature vector of test file and each feature vector of train file. So it will be M\*N matrix where M is number of feature vectors of test file while N is the number of feature vectors of train file.

Lesser distance between two feature vectors indicates that these two feature vectors can be matched together. Diagonal entries are likely to have low distances.

Let X = {x1, x2, x3 ..xm } be the test feature vector, Y = {y1,y2,y3..yn } be the train feature vector.

Distance (i, j) = |xi-yj|

**Cost Matrix:** Obtaining cost matrix implicitly means warping the path. We try to find out the path of minimum distance from (1, 1) to (M, N).

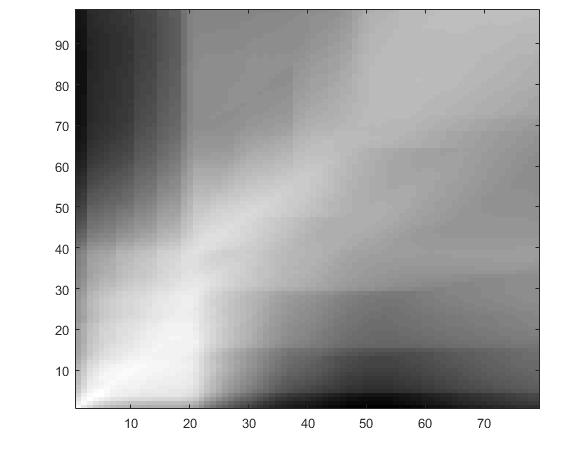
Since the sequence is temporal, we cannot go back and hence the path only moves in forward direction. Either (i+1, j) or (i, j+1) or (i+1, j+1).

Cost (1, 1) = distance (1, 1)

Cost (i, 1) = distance (i, 1) + distance(i-1, 1)

Cost (1, j) = distance (1, j) + distance(1, j-1)

Cost (i, j) = distance (i, j) + minimum{cost(i-1,j),cost(i,j-1), cost(i-1,j-1)}



Plot of cost matrix between test file: ft\_1a.mfcc and train file: st\_1b.mfcc

It is clear from the picture that minimum path lies along the diagonal.

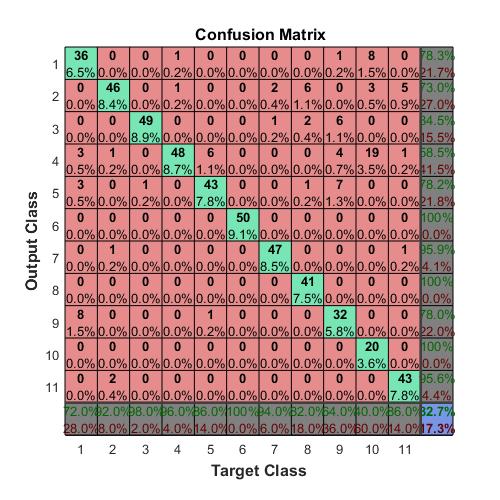
**Score**: Score between test file and train file is taken as: cost(M,N)/(M+N)

Find out the score between test file and each train file of digit uttered by a speaker. The test file is classified into a digit for which score obtained was minimum. If we have train files of many speakers, we conduct voting between each digit recognized against each speaker.

**3.2 RESULTS:**

Scores were calculated between several test files and train files of different speakers. Using these scores test files were recognized as digits. Confusion matrix has been plotted and the results were as follows:

Number of test files: 550 Number of Digits: 11 (includes letter ‘O’) There are 50 test cases for each digit

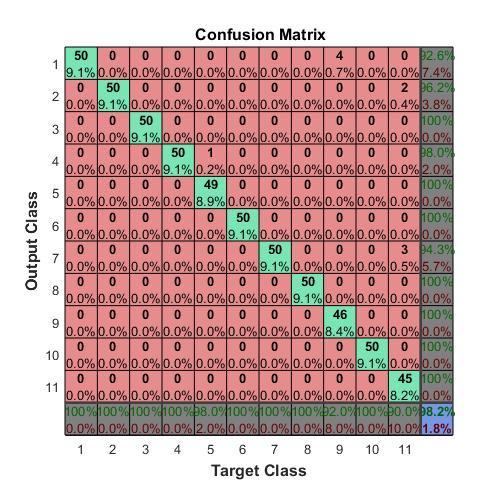


Plot of confusion matrix where Number of train files: 11 and Number of speakers: 1

Element (i,j) represents number of test cases that were recognized as ‘i’ which are actually ‘j’. Diagonal entries indicate how many test cases were recognized correctly.

(1,1):36 indicates that out of 50,36 were recognized correctly as digit-1. Here digit 10 was confused mostly with 4 (19) and 1(8). **Accuracy in this case is 82.7%**

On increasing the number of train files and speakers, digit recognition becomes more accurate.



Plot of confusion matrix where Number of train files: 550 and Number of speakers: 500

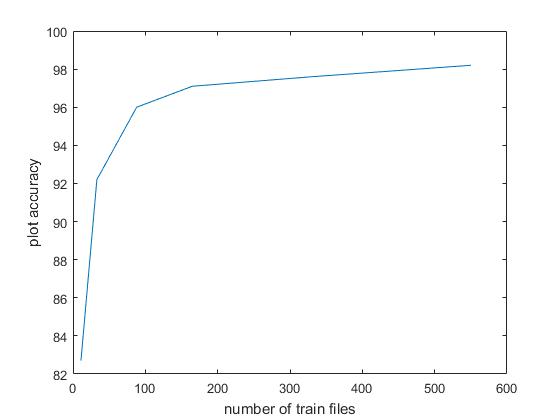
In this case, most of the diagonal entries are 50 which indicates almost all the test files have been recognized correctly.

Digit 9 has been confused with digit 1 and Digit 11 has been confused with 7.

This happens because the phoneme sequence for digits 9 and 1 are similar which reduces the distance between their feature vectors, thereby minimizing the cost.

Here **accuracy is 98.2%**

Thus, more the number of train files, more is the accuracy.

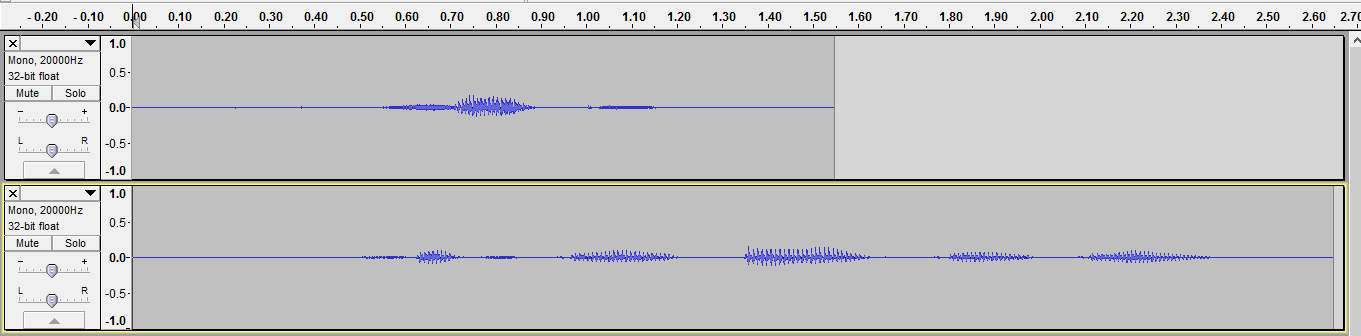


# Recognition of first digit uttered by speaker using dtw

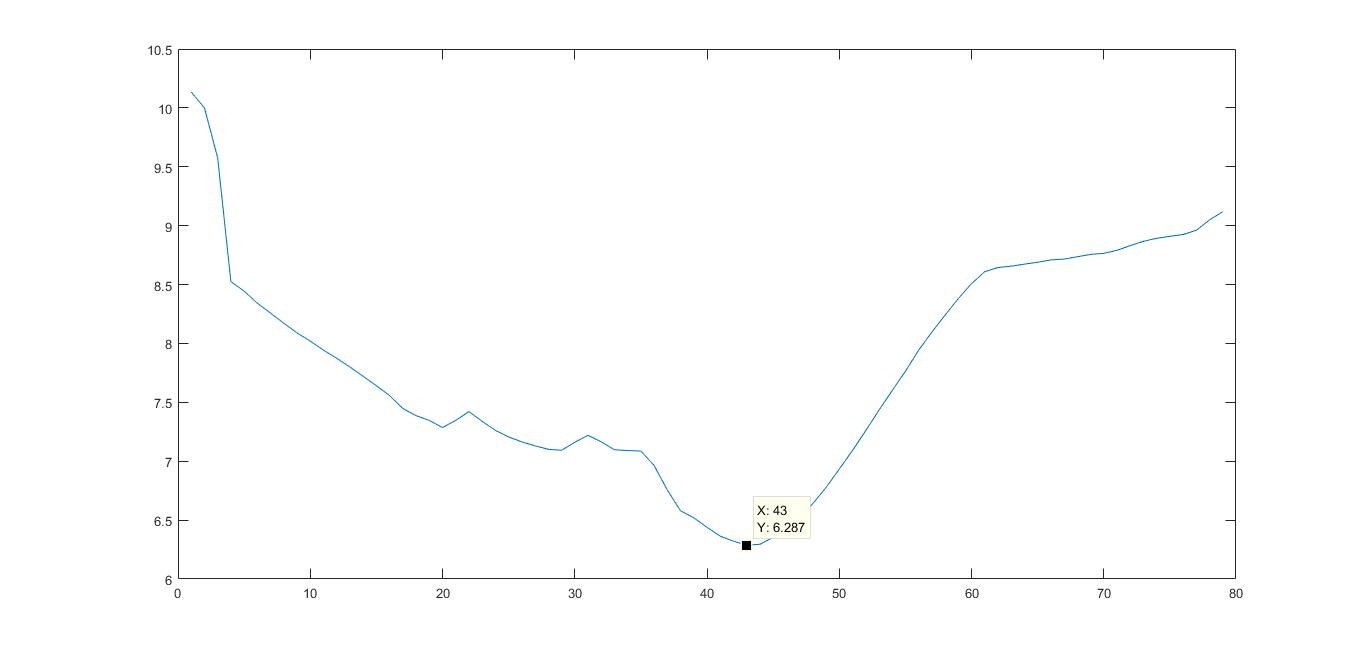
This section deals with the case when multiple digits have been uttered by a speaker in a row. Here, we try to identify the first digit uttered by the speaker. The algorithm we use is same as in the above case. In this case, we don’t know the exact point where the first digit ends in the sequence of feature vectors of test file. So, we try to find out cost matrix of size N\*N where N is the number of feature vectors of train file. Using this cost file, we find out the point where score is the least of all. We consider this point as the end point of first digit. Then we use these scores to identify the class to which the test file belongs to using the method as in above case.

Both the test and train files have silent regions. The train files happen to have silent region for long duration. Using the DTW algorithm straightforward, it would try to match the silent region of test file and train file which gives minimum score even for files belonging to different digits.

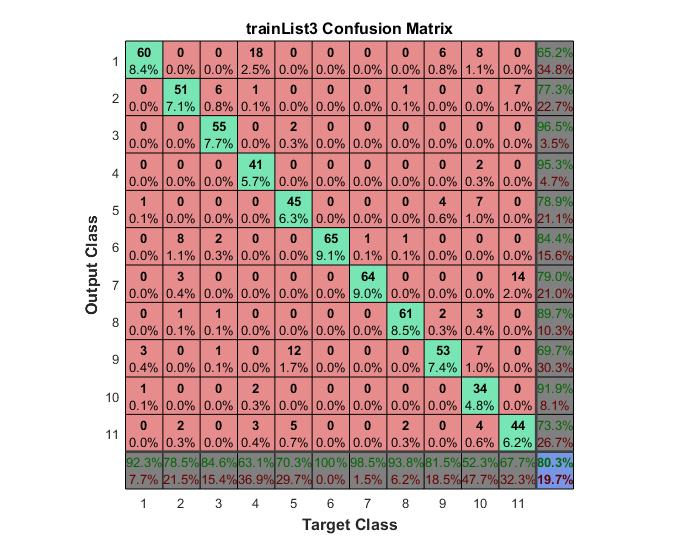
The picture below shows the wav files of train file: pb\_6b.wav and test file: 63533a.wav. The silent region extends till 0.50. The DTW algorithm will try to match these silent regions.



The picture below shows the plot of N against cost matrix (X,N) where N : number of feature vectors of train file. The score obtained is minimum at around 43 which belongs to silent region in both train and test files.



Hence, we cut the silent regions in both the wav files and find out the scores between files. The results were as follows:



Plot of confusion matrix where Number of train files -88 and Number of test files - 714

Plot accuracy: 80.3%

Digit - 4 has been confused for digit - 1 eighteen times. The reason for this confusion is the existence of test files of the form 41\*.wav. The DTW algorithm tries to match the train files of digit-1 with the digit-1 in the test files of these forms.

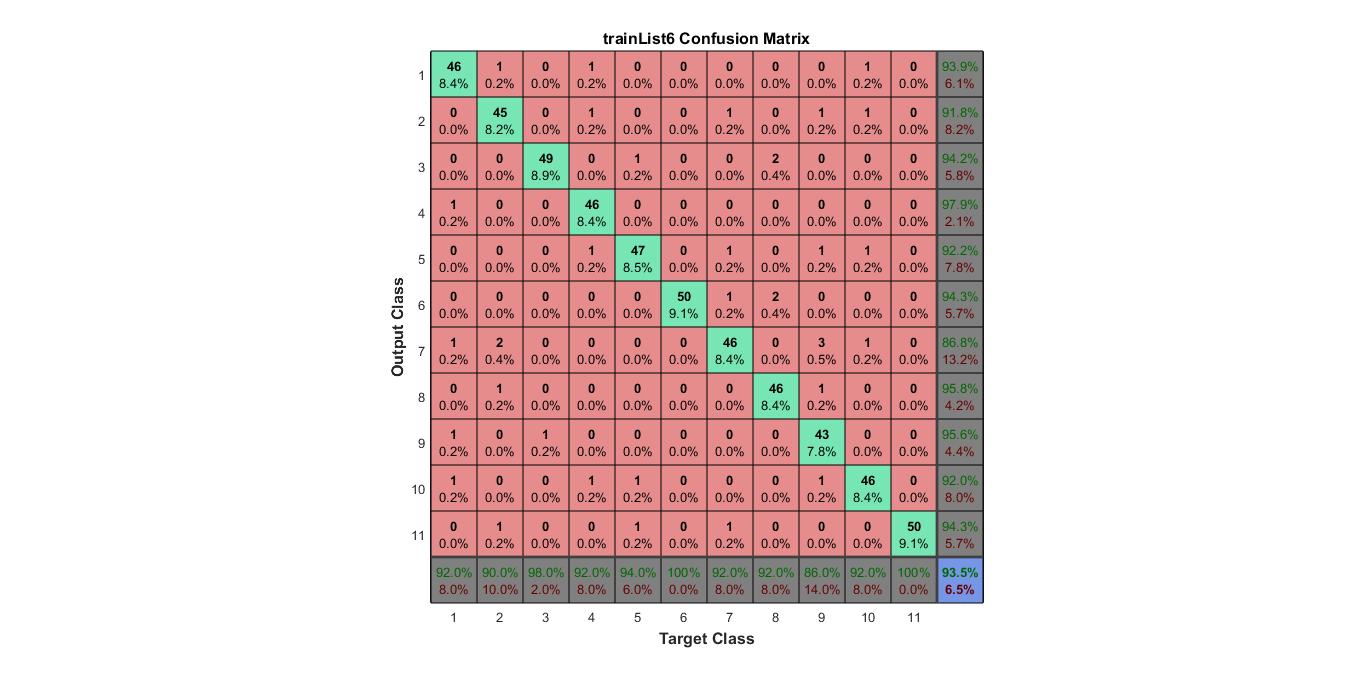
Other reasons for confusion can be the similarity in phoneme sequence, for example digit -7 and digit - 11

# Isolated Digit Recognition By HMM

Another most commonly used method in speech recognition is Hidden Markov Model (HMM) which is a statistical model. Each digit is modelled by a distinct HMM. The states in the model represent each letter in a digit. The observations are the feature vectors that we are able to extract. We cluster the data using unsupervised clustering (k-means). Using this clustered train data, we train the model for each digit.

For each unknown digit to be recognized, we calculate the log likelihood for each model and choose the highest one.

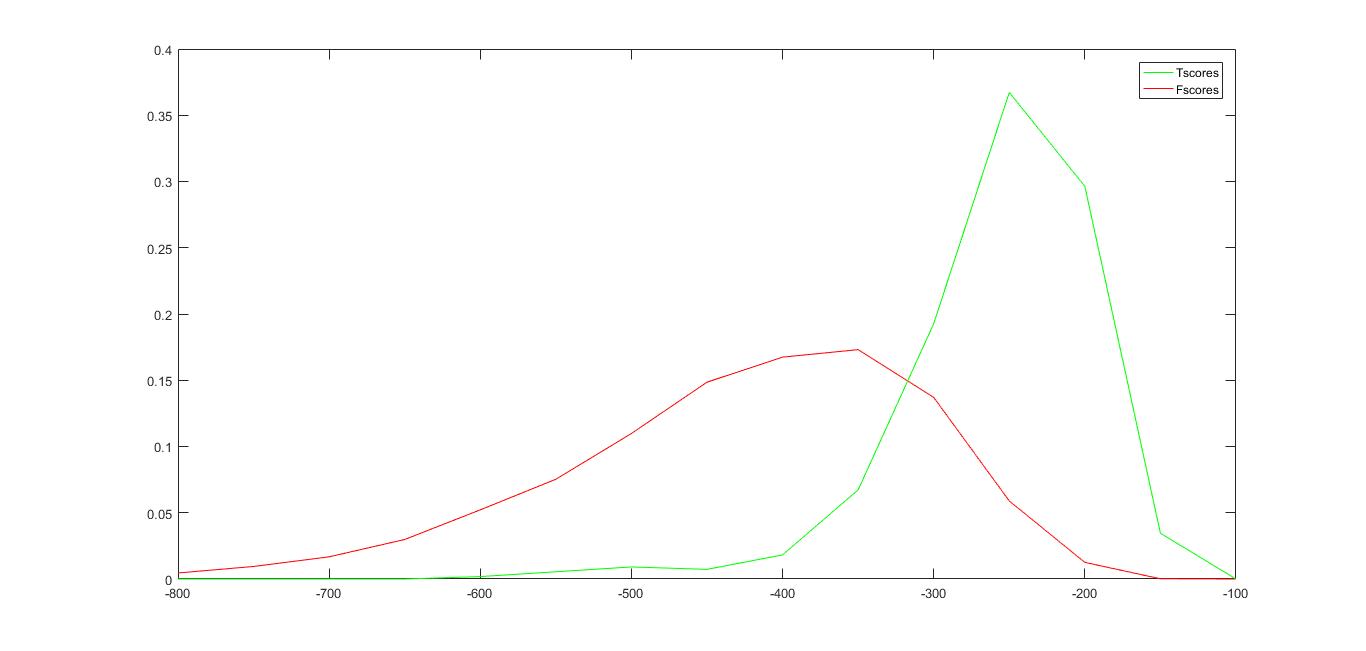
## RESULTS:

****

Number of test files: 550 Number of train files: 550 Number of cluster: 50 Number of states: 6 Plot accuracy: 93.5%

## EQUAL ERROR RATE PLOT (EER):

The EER is obtained by plotting the density distribution of true scores and impostor scores. True scores are the scores that are obtained between the test file and model of the same digit as test file. Impostor scores are scores other than true scores.



Number of test files: 550 Number of train files: 550 Number of cluster: 50 Number of states: 6 EER: 0.1495 Threshold score: -317.2295

# Comparison of dtw and hmm for digit recognition

Dtw better than hmm