Automatic Speech Recognition with MFCC and DTW

João Borges
Telecommunications, Automation and Electronics
Research and Development Center (LASSE)
Federal University of Pará
Belém, Brazil
joao.tavares.borges@itec.ufpa.br

***Abstract*—In this work, the Mel-frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) algorithms are implemented in Python to develop an isolated word recognition pipeline. The system currently shows around 60% accuracy in word identification.**

***Index Terms*—MFCC, DTW, Speech Recognition**

I. Introduction

The use of Mel-frequency Cepstral Coefficients (MFCC) [1] to extract audio feature and Dynamic Time Warping (DTW) [2] to compare the obtained values is a classical method of Automatic Speech Recognition (ASR) for simple isolated word detection [3]. This method leverages the Mel frequency scale, that is designed to be a perceptually relevant scale for pitch, meaning that equal frequency distances have the same perceptual difference from a human listener point of view.

II. Data Processing Pipeline

This work uses a dataset composed of 105,000 WAVE audio files, with 30 different classes, obtained from Tensorflow[1], from which a subset of 5 classes, with 25 samples each, will be used in the experiments. The dataset is divided into 80% for training and 20% for validation. The remaining process can be split into 4 steps:

- Calculate MFCC for all signals
- Perform DTW between sample and reference signals to find the most similar ones
- Check if the most similar sample have the same label as the reference (i.e. both are ”cat”). If yes, then increment the recognition score.
- After finishing the process, obtain the mean between successful recognitions versus number of samples, which is the recognition rate.

In the following subsections each of these steps will be described in details.

A. Obtaining the MFCCs

The MFCCs of all audio files are calculated with the following steps:

- The signal goes through a process of Short-Time Fourier Transform (STFT)
- After that, the Mel filter banks are calculated
- Then, a dot product between the previous results, followed by a dB conversion, are performed to obtain the melspectrogram
- Finally, the MFCC is obtained by applying Discrete Cosine Transform (DCT) to the melspectrogram

1) Short-Time Fourier Transform Step: Due to the highly non stationary nature of speech data, composed of different phonemes with their own frequencies distributed along the time, it is necessary to visualize how the signal spectrum changes over time, a feature absent in the traditional Fourier transform technique. This motivates the method known as STFT, that divides the signal into frames through a process of windowing and then applies the Fourier transform in each one of them. These frames usually need to be overlapped in order to avoid loss of signal. In the current implementation, the STFT adopts the widely used *Hann* window for the windowing process, as it smooths the discontinuities, avoiding the artifacts over the spectrum seen when rectangular windows are used.

2) Mel filter banks: To calculate the Mel filter banks, first is necessary to determine the number of Mel bands that will be used, then obtain the lowest and highest frequencies of the signal in the Mel scale. After that the interval between the lowest and the highest frequencies must be divided into a number of evenly separated points, equal to the number of Mel bands. These points are then converted back to the Hertz scale and rounded to the nearest bin. Finally, the triangular filters are created.

3) The melspectrogram and the MFCC: In order to obtain the melspectrogram first we need the spectrogram, which is obtained by squaring the magnitude of the STFT result. After that, a dot product is executed between the spectrogram and the Mel filter banks, followed by a power to dB conversion, resulting in the melspectrogram. To obtain the MFCC, this spectrogram goes through a DCT. This last transformation could be a Fourier transform, but in order to discard the complex valued component, the DCT is used instead.

[1]https://storage.cloud.google.com/download.tensorflow.org/data/speech_commands_v0.02.tar.gz

B. Performing the DTW and Evaluating the System Performance

After obtaining the MFCCs of all the dataset, 20% of it is separated to act as validation set. The next step is to evaluate if the system can successfully compare the MFCCs of the sample and validation sets to point out similar words. This similarity is calculated using DTW, which is a method used to compare two sequences of numbers with different lengths [2]. The implementation utilized for the DTW is publicly available in an online repository[2].

The procedure to obtain the recognition rate is the following:

- First, a sample from the validation set is chosen
- After that, the system performs DTW between this sample and all the others in the training set, to get the one that has the MFCC with the highest degree of similarity, in other words, the one that has the minimum distance calculated via DTW
- The system then checks if this sample, with minimum distance from the reference, corresponds to the same word. If yes, then the system correctly recognized the word by checking MFCC similarity with DTW, increasing one point in the recognition score
- Finally, after performing the same process for all samples in the validation set, the system calculates a mean score by dividing it with the amount of samples in the validation set.

III. Conclusion

The ASR performed by the described pipeline resulted in a 60% accuracy rate for the subset utilized in the experiments. This performance can be improved by tuning several hyperparameters from the processes described, such as the number of FFT points and window overlap and stride in the STFT, or the number of Mel bands in the Mel filter bank design etc. The code used is available publicly in an online repository[3]

References

[1] K. Dash, D. Padhi, B. Panda, and S. Mohanty, "Speaker identification using mel frequency cepstralcoefficient and bpnn," *International Journal of Advanced Research in Computer Science and Software Engineering Research Paper*, vol. 2, 2012.

[2] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[3] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *arXiv preprint arXiv:1003.4083*, 2010.

[2]https://github.com/pierre-rouanet/dtw

[3]https://github.com/joaotavares43/asr-jupyter/blob/main/asr.ipynb