Summer Training Project Colloquium Report

Gender Recognition Based on Voice.

Laxman Singh Tomar

June 2019-July 2019

Summer Training Project Colloquium Report

Gender Recognition Based on Voice.

Laxman Singh Tomar

June 2019-July 2019

Contents

Report submitted to the Department of Computer Science and Engineering, Hindustan College of Science and Technology in fulfilment of the requirement for:

INDUSTRIAL SUMMER TRAINING

Dedication

I would like to dedicate this work to teachers whom I've never met in real but their valuable contribution have made this work possible and my family who have always been supportive in my work.

Acknowledgement

(I encountered several resources which helped me in getting through this work)

I would like to express my sincere gratitude to:

- Hassam Ullah Sheikh
- Reynolds et. al.
- Akshat Gupta

Abstract

Analyzing Speech Analytics leads to improvements in applications working as per human-preferences. One of the most foundational tasks in speech analytics is Identifying the Gender with the help of Voice. In this project, I'll analyze and cover the workflow of how to detect the gender of the speaker using MFCC (Mel-Frequency and Cepstral Coefficients) and GMM (Gaussian Mixture Models). I'll make use of the mentioned techniques to achieve noteworthy performance.

Keywords: MFCC; Delta-Cepstral Features; GMMs

Chapter 1

Introduction

Large amounts of computing power available along with Artificial Intelligent systems has resulted in dramatic enhancement into capability of machines to recognize the voices. Faster Processing and large amount of Speech Data available makes the performance of these systems roughly on par with humans. From Audrey a speech recognizing system which could recognize a single voice speaking digits aloud at Bell Labs in 1952; we've reached to having day to day conversations with voice assistants like Google Assistant and Siri in our smartphones.

But most of these systems are usually neutral to the gender of the speaker and results being given. Having systems which can respond as per the user's gender is indeed an amazing capability. A large amount of tasks which are based on gender preferences can be handled by them. It results into better customer service and enhances user experience.

Every speech system that is available today has its own drawbacks and continuous work is being done to increase the performance of such systems. To increase the performance of speech systems pre-processing like Gender Recognition and Language Identication is the need of the hour. In this project, I focus on detecting the gender of the speaker based on input voice samples. I'm going to make use of MFCC (Mel-Frequency and Cepstral Coefficients) and GMM (Gaussian Mixture Models) to achieve commendably accuracy.

1.1 Motivation

Modern Speech Systems have drawbacks and cannot provide acceptable performance in noisy environments, multilingual speakers, compressed speech and silence.

1.2 Objectives

1.2.1 Project

Creating a Gender Detection system that can be used to improve the existing systems' performance in detecting gender from the input voice samples.

1.2.2 Company

Machine Learning has been referred as the most in-demand skill of the 21st century with offering more than 10000+ jobs providing Rs. 4 million+ CTC. Robofied is one of the few start-ups who've recently emerged as leaders in the terrain of Machine Learning and AI in India. At the moment, they provide solutions in Healthcare, Business and other domains and slowly expanding themselves. I simply couldn't refrain myself from joining them!

Chapter 2

Methods

2.1 Dataset

Data about voice samples of males and females is The Free ST American English Corpus dataset which can be downloaded from here!. It contains utterances from 10 speakers, 5 from each gender. Each speaker has about 350 utterances.

2.2 Acoustic Features

Acoustic features can be referred as the acoustic property of sound that is used to find distinctive features of a class of speech sounds. In Natural Language Processing one feature named Mel-Frequency Cepstral Coefficients (MFCC) dominates all other features. MFCCs are not only used in Gender Identication but they are also used in Speech Recognition and Language Identication Systems.

2.2.1 Mel-Frequency Cepstral Coefficients (MFCC)

MFCC are coefficients that collectively form Mel frequency cepstrum which is a power spectrum of a short window of a speech signal i.e. 25 milliseconds etc. MFCC tries to represent the shape of the vocal tract using the short term power spectrum thus trying to approximate human auditory system responses. As the MFCC represents a short term power spectrum so they are considered to be short term features. Generally, MFCCs are derived as follows:

1. Take the Fourier transform of (a windowed excerpt of) a signal. It transforms the time domain signal into spectral domain signal where source and filter part are now in multiplication.

2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.

3. Take the logs of the powers at each of the mel frequencies. It helps in separating source and filter.

4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

5. The MFCCs are the amplitudes of the resulting spectrum.

A better set of features is shifted delta cepstral coefficients which are derived from MFCC and have proven to be better than MFCC in natural language processing tasks.

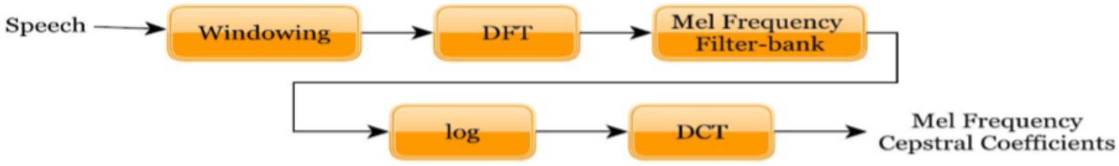

Figure 1: MFCC

2.2.2 Delta-Cepstral Features

Delta-cepstral features were proposed to add dynamic information to the static cepstral features. They also improve recognition accuracy by adding a characterization of temporal dependencies to the hidden-markov models (HMM) frames,which are nominally assumed to be statistically independent of one another. For a short-time cepstral sequence $C[n]$,the delta-cepstral features are typically defined as:

$$D[n] = C[n+m] - C[n-m]$$

where n is the index of the analysis frames and in practice m is approximately 2 or 3. Similarly, double-delta cepstral features are defined in terms of a subsequent delta-operation on the delta-cepstral features. In addition of delta-cepstral features to the static MFCC features strongly improves speech recognition accuracy, and a further(smaller) improvement is provided by the addition of double-delta cepstral. For these reasons some form of delta and double-delta cepstral features are part of nearly all speech recognition systems.

2.3 Gaussian Mixture Models (GMMs)

> A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori(MAP) estimation from a well-trained prior model[1].

A Gaussian Mixture Model popularly known as GMM is a probabilistic clustering model for representing a certain data distribution as a sum of Gaussian Density Functions. These densities forming a GMM are also known as components of GMM. The likelihood of a data point is given by the following equation:

$$P(X|\lambda) = \sum_{k=1}^{K} w_k P_k(X|\mu_k, \Sigma_k)$$

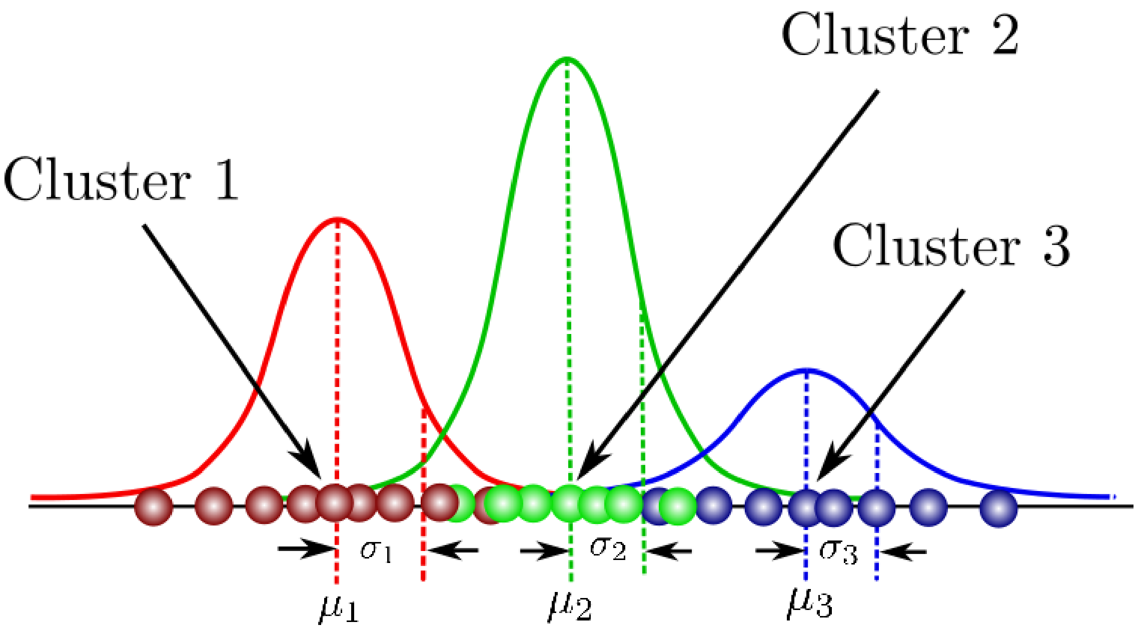

Figure 2: Gaussian Mixture Models.

where $P_k(X|\mu_k, \Sigma_k)$ is the Gaussian Distribution:

$$P_k(X|\mu_k, \Sigma_k) = \frac{1}{\sqrt{2\pi|\Sigma_k|}} e^{\frac{1}{2}(X-\mu_k)^T\Sigma^{-1}(X-\mu_k)}$$

where:

λ : It represents training data.

μ: It represents the mean.

Σ: It represents the co-variance matrices.

w_k: It represents the weights.

k: It represents the index of the components.

Initially, it identifies k clusters in the data by the K-means algorithm and assigns equal weight $w = \frac{1}{k}$ to each cluster. k Gaussian distributions are then fitted to these k clusters. The parameters μ , σ and w of all the clusters are updated in iterations until the converge. The most popularly used method for this estimation is the Expectation Maximization (EM) algorithm.

2.4 Project Workflow

The idea here is to recognize the gender of the speaker based on pre-generated Gaussian mixture models (GMM). Once the data is properly formatted, we train our Gaussian mixture models for each gender by gathering Mel-frequency cepstrum coefficients (MFCC) from their associated training wave files. Now that we have generated the models, we identify the speakers genders by extracting their MFCCs from the testing wave files and scoring them against the models. These scores represent the likelihood that user MFCCs belong to one of the two models. The gender models with the highest score represents the probable gender of the speaker. In the following table, we summarize the previous main steps, as for a detailed modeling of the processing steps, you can refer to the Workflow graph in Figure 1.

Table 1: Main Steps of Gender Identification System

A. Training Phase
1. Data formatting and Management
2. Extracting MFCC features from the Training Data
3. Training gender GMMs
B. Testing Phase
4. Extracting MFCC features from the Testing Data
5. Scoring the extracted MFCCs against the GMMs
6. Recognizing the speaker's gender based on the scores

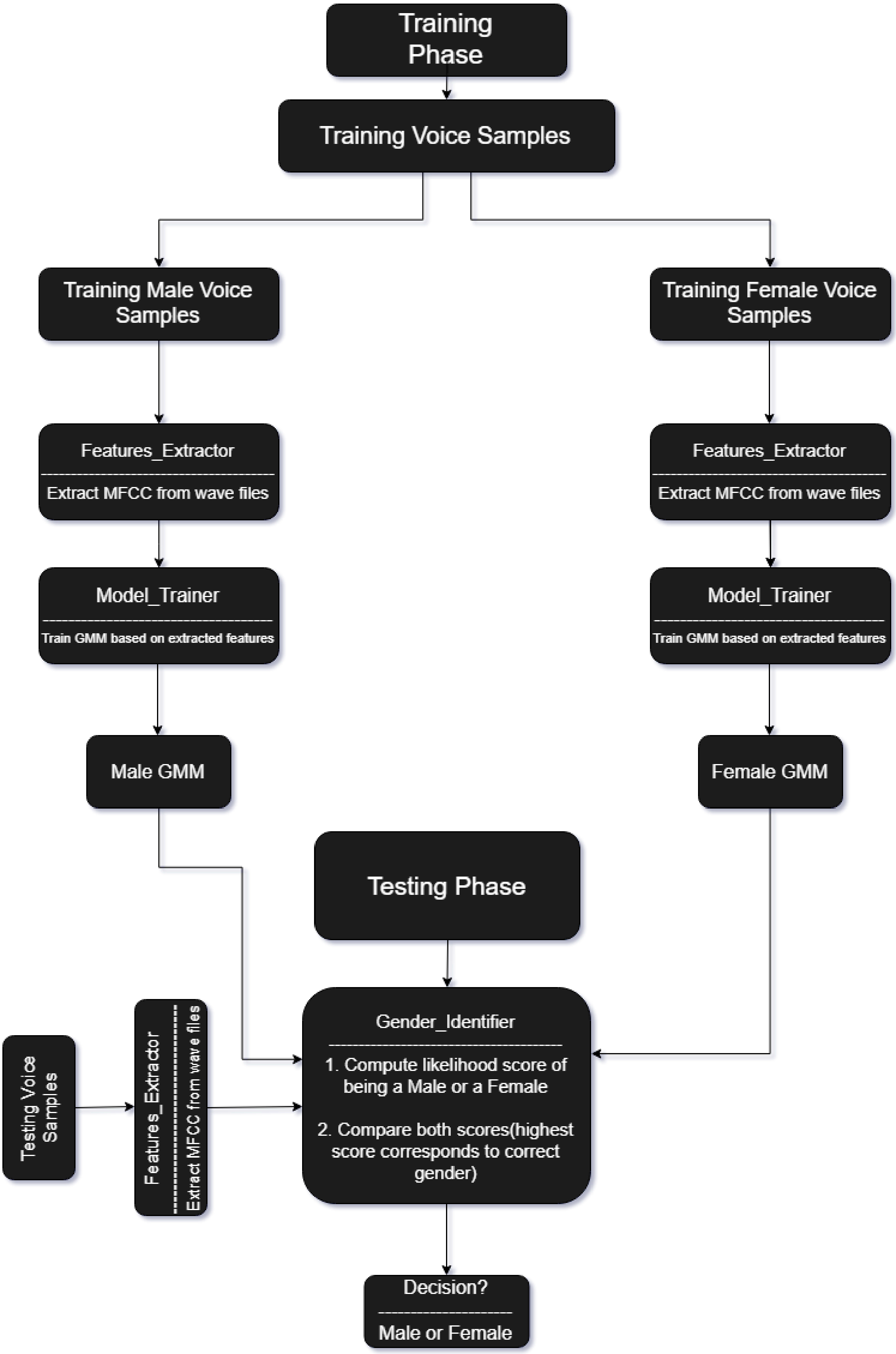
Training Phase
Training Voice Samples
Training Male Voice Samples
Training Female Voice Samples
Features_Extractor
Extract MFCC from wave files
Features_Extractor
Extract MFCC from wave files
Model_Trainer
Train GMM based on extracted features
Model_Trainer
Train GMM based on extracted features
Male GMM
Female GMM
Testing Phase
Testing Voice Samples
Features_Extractor
Extract MFCC from wave files
Gender_Identifier
1. Compute likelihood score of being a Male or a Female
2. Compare both scores(highest score corresponds to correct gender)
Decision?
Male or Female

Figure 3: Project Workflow Graph explaining steps to be carried out during Training and Testing Phase.

Chapter 3

Results

3.1 Empirical Results

Splitting criterion for training and testing sets depends on the user's choice. I took $^2/_3$ for the Training Set and the rest for the Testing Set. For extracting features from audio samples, I extracted MFCCs and later performed CMS Normalization. Later, I appended the Delta Cepstral Features and the Double Delta Cepstral Features to the static MFCC features for better speech recognition.

python_speech_features is a library which has built in module for MFCC. It has several hyperparameters and values being plugged in have a crucial impact. Hyperparameters and their values that I plugged for extracting MFCCS:

Table 2: Hyperparameters and their plugged values for MFCCs

Hyperparameters	**Values**
Length of the analysis window	0.05 seconds
Step between successive windows	0.01 seconds
Number of Cepstrum to return	5
Number of filters in FilterBank	30
Size of the FFTs	800
Append Energy	True

Note: Append Energy if set to True implies that, the zeroth cepstral coefficient is replaced with log of total frame energy.

I've used Gaussian Mixture Models present in Sci-kit learn which is a package developed in python for Machine Learning & Predictive Modeling. It has few hyperparameters whose values that I used are as follows:

Table 3: Hyperparameters and their plugged values for GMMs

Hyperparameters	**Values**
Number of Mixture Components	16
Number of EM iterations	200
Type of Co-variance parameters to use	diag
Number of Initializations	3

Note: diag means each component has its own diagonal covariance matrix

3.2 Performance

Once the training is finished, features aka MFCCs are extracted from the testing dataset. Later these MFCCs are scored against the GMMs by computing the per-sample average log-likelihood of the given data samples. Once scores were obtained then the score which is maximum-that gender will be returned back.

Following the above described procedure, I was able to get 95.749% accuracy which is acceptable by any benchmark you measure against.

Chapter 4

Discussion

4.1 Discussion

Initially, I tried extracting voice features using specan function of WarbleR package of R language which helps in extracting 22 different acoustic features from the voice samples, later any popular classification algorithm like Decision Trees or XGBoost would have done a good job by achieving 90+% accuracy. Due to time not being enough, I had to look for a pythonic way to perform this task and ended up choosing the approach we just discussed.

4.2 Alternative Approaches

In general, there are three main approaches to building an automatic gender identication system: The first approach uses pitch as a discriminating factor and use labelled data to identify the gender of the speaker. The second approach deals with acoustic features like MFCC and unlabeled data to identify the gender. In this approach relevant features are extracted then the model is trained. In this case generally a GMM is trained for each gender and results from one model are subtracted from the other model to find the gender. The third approach is quite commonly used after year 2005, in which pitch models are combined with acoustic models to form a fused model.

4.3 Technology Stack

Programming Languages: Python

Frameworks: python_speech_features, Numpy, Pandas, Sci-kit Learn

4.4 Future Enhancements

1. As of now, it works only on audio waveforms whereas popular audio formats like .mp3 still required to be converted into .wav prior to try model out.

2. As of now, it's a working model which can be deployed into a web-app. It's ability can be enhanced in noisy environments too for better results.

3. The accuracy can be further improved using GMM Normalization also known as UBM-GMM system.

4.5 Efforts

Hours: 225 hours approx.

Monetary Value: USD $15

Value Gained: Learned various skills including Data Gathering, Data Pre-processing, Model Building, Testing.

4.6 Certification

My certification can be viewed at: `http://bit.ly/2miy1kt`

4.7 Conclusions

Results are clearly depicting how crucial it's to tune the hyperparameters for both feature extraction and GMMs correctly to gain good performance. Further increase in the data will lead to better results.

List of Figures

List of Tables

Bibliography

[1] Reynolds, D. A. Gaussian mixture models (2009).

Appendix A

Implementation in Python

`http://bit.ly/2npLCGA`

This Jupyter notebook created by me contains the entire workflow for the project implemented in Python.

Appendix B

Guide to MFCCs

`http://bit.ly/2Naarkv`

If you really want to explore more about MFCCs, I'd suggest you to read this blog.