AIML - CS 337

Topic : Predicting Gender Based on Voice

Autumn 2022

Members: Kartik, Advait, Nikhil, Harshvardhan *Project Report*

1 Introduction

Gender identification is one of the major problems in speech analysis today. Tracing the gender from acoustic data i.e., pitch, median, frequency etc and being able to do this with reasonable accuracy. Machine learning gives promising results for classification problem in all the research domains. There are several performance metrics to evaluate algorithms of an area. Here, we explore several models and learning methods to be able to predict gender based on voice. We analyse techniques involving feature engineering as well as those which operate on an already existing feature vectors. The main parameter in evaluating any algorithms is its performance. Misclassification rate must be less in classification problems, which says that the accuracy rate must be high. Thus, we tune the models with respect to the various hyper-parameters to obtain maximal validation accuracy.

2 Data

We use 2 datasets for this task.

- VoiceGender Dataset
- VoxCeleb Dataset

2.1 VoiceGender Dataset

Some salient points about this dataset.

- This dataset consists of roughly 3000 feature vectors which have 20 features each.
- There are roughly 1600 male and 1400 female samples signifying a good representation of both classes.
- Each sample has a unique label, either male or female
- The variation in features across each feature space is considerably large, thus providing an unbiased sample of each class

This dataset is optimal for training and testing classification models since no feature engineering is required.

2.2 VoxCeleb Dataset

VoxCeleb is an audio-visual dataset consisting of short clips of human speech, extracted from interview videos uploaded to YouTube[2]. Some salient points are

- 7000+ speakers with over a million utterances and over 2000 hours of audio samples.
- 61% samples are male while the rest are female
- The samples are collected from individuals across the world to ensure a fair spread of samples from each class.

This dataset provides raw audio files, which must be encoded into feature vectors using feature engineering, before further classification.

3 Models and Analysis

Our final goal is to predict gender on the basis of voice. Mathematically, speaking, let X be an audio sample whose gender is Y. Then, our job is to predict function f, such that

$$Y = f(X) \tag{1}$$

We break down this problem into 2 parts. First, we need to model X mathematically. X, currently being an audio sample sampled from the space of all audio samples, is not a valid input to a mathematical function. Hence, we need to convert this into a sample which can be represented by a feature vector. Secondly, we need to classify the feature representation of X, say X' as male or female. That is, find g such that

$$Y = g(X') \tag{2}$$

3.1 Modelling g

Since we assume that $X' \in R^d$, we can model g as a function $R^d \mapsto 0, 1$ along with a binary cross entropy loss function to model the mis-classification loss of function g. We do this in the following ways

$$L = \sum_i (1 - y_i) \log(1 - g(X')) + y_i \log g(X')$$

3.1.1 Multilayer Perceptron

We use the architecture shown in Figure 1 with 2 hidden layers for the purpose of this. The final output is subjected to a sigmoid for the purpose of 0-1 classification.

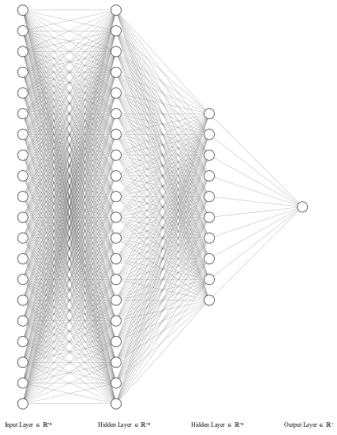

Figure 1: Multilayer Perceptron Architecture

3.1.2 SVM and Kernels

We implemented 4 kernels, namely RBF, Laplace, Gaussian and Polynomial for the purpose of transforming and linearly separating the data using an SVM. We achieved maximal accuracy using the Laplace kernel.

3.1.3 Decision Trees

We used decision trees to identify the key features in voice that help us seperate out gender

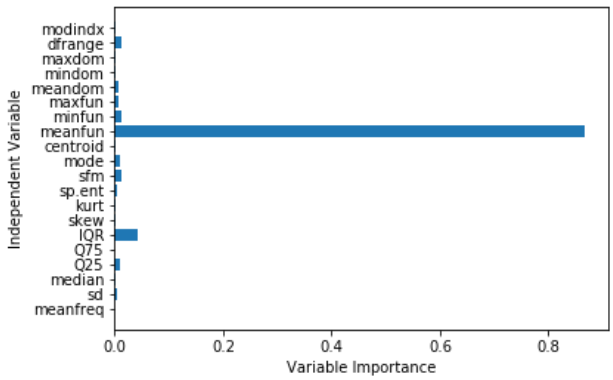

Figure 2: Decision Trees

3.1.4 Random Forests

We are able to identify more key features using this method

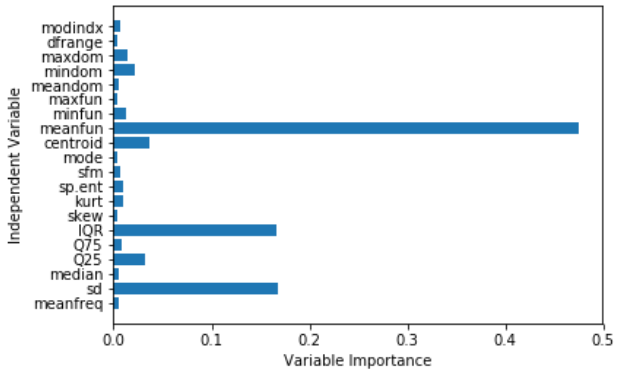

Figure 3: Random Forests

3.2 Obtaining X'

To obtain X' from X, we need to do feature engineering

3.2.1 Mel Frequency Cepstral Coefficients

The first step in any automatic speech recognition system is to extract features i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc.

The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope.

Mel Frequency Cepstral Coefficents (MFCCs) are a feature widely used in automatic speech and speaker recognition.[1]

3.2.2 Steps at a Glance

We will give a high level intro to the implementation steps, then go in depth why we do the things we do. Towards the end we will go into a more detailed description of how to calculate MFCCs.

- Frame the signal into short frames.
- For each frame calculate the periodogram estimate of the power spectrum.

- Apply the mel filterbank to the power spectra, sum the energy in each filter.
- Take the logarithm of all filterbank energies.
- Take the DCT of the log filterbank energies.
- Keep DCT coefficients 2-13, discard the rest.
- There are a few more things commonly done, sometimes the frame energy is appended to each feature vector. Delta and Delta-Delta features are usually also appended. Liftering is also commonly applied to the final features.

Why do we do these things?

An audio signal is constantly changing, so to simplify things we assume that on short time scales the audio signal doesn't change much (when we say it doesn't change, we mean statistically i.e. statistically stationary, obviously the samples are constantly changing on even short time scales). This is why we frame the signal into 20-40ms frames. If the frame is much shorter we don't have enough samples to get a reliable spectral estimate, if it is longer the signal changes too much throughout the frame.

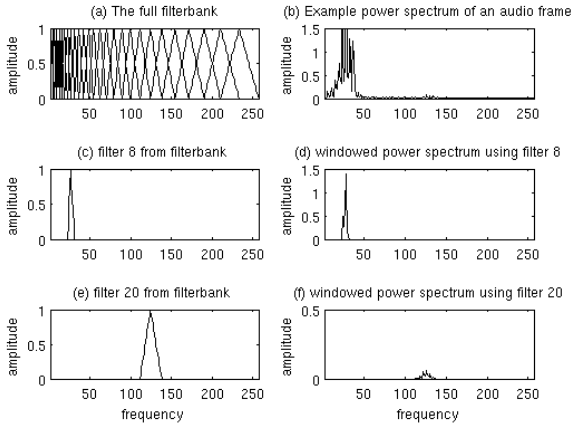

Figure 4: Plot of Mel Filterbank and windowed power spectrum

The next step is to calculate the power spectrum of each frame. This is motivated by the human cochlea (an organ in the ear) which vibrates at different spots depending on the frequency of the incoming sounds. Depending on the location in the cochlea that vibrates (which wobbles small hairs), different nerves fire informing the brain that certain frequencies are present. Our periodogram estimate performs a similar job for us, identifying which frequencies are present in the frame. The periodogram spectral estimate still contains a lot of information not required for Automatic Speech Recognition (ASR).

Once we have the filterbank energies, we take the logarithm of them. This is also motivated by human hearing: we don't hear loudness on a linear scale. The final step is to compute the DCT of the log filterbank energies. But only 12 of the 26 DCT coefficients are kept. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade ASR performance, so we get a small improvement by dropping them.

Since these series can get quite long as one new data point is created every 20ms, one can always extract the mean, variance, quartiles, min, max and median as a descriptive statistic at the end of an audio sample, and compare several audio samples on this basis. We now finally, have our X', to represent our feature space.

This has been implemented using the librosa library in preprocess.py in our code folder.

4 Hyperparameter Tuning

We tuned the various hyperparameters such as learning rate, the various parameters of the kernels and the perceptron architecture. Below are some of our observations and graphs

4.1 Tuning the Kernels

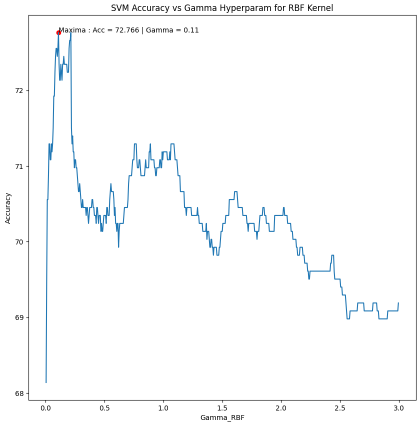

Figure 5: Tuning the RBF Kernel

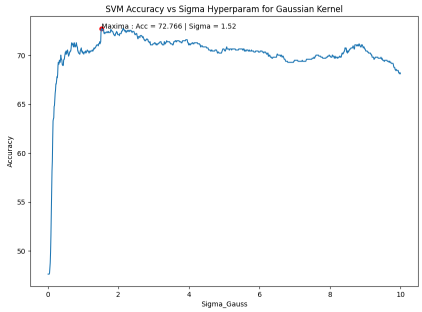

Figure 6: Tuning the Gaussian Kernel

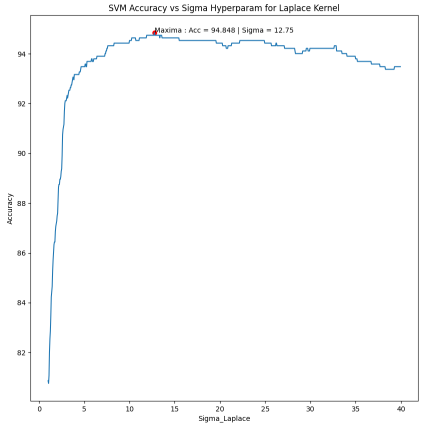

Figure 7: Tuning the Laplace Kernel

4.2 Tuning the Learning Rate

Tuning the Learning Rate is important as With low learning rates the improvements will be linear. With high learning rates they will start to look more exponential. Higher learning rates will decay the loss faster, but they get stuck at worse values of loss.

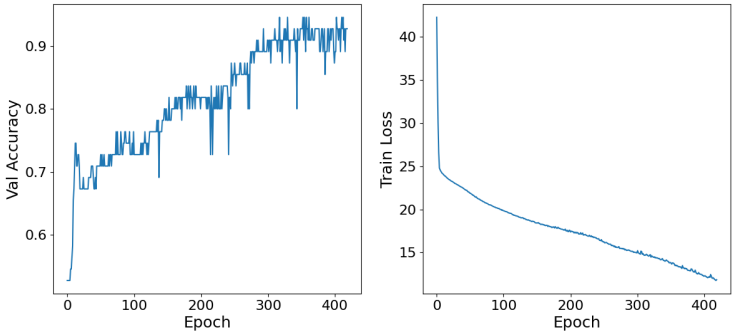

Figure 8: Tuning the Learning Rate

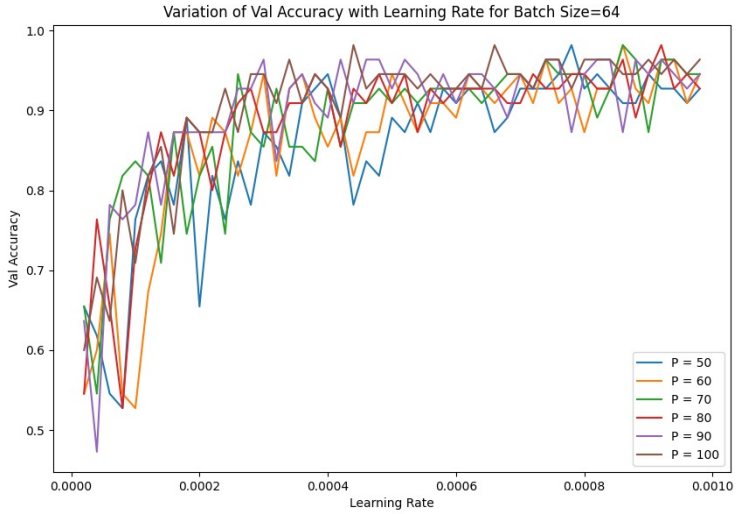

Figure 9: Tuning the Learning Rate

5 Results

Here we summarise the results of the overall model on both the datasets. To summarise, we used 2 datasets, one of which had provided us the extracted features on which we simply trained and tested classification models and another on which we performed feature extraction on our own.

5.1 VoiceGender

On this dataset, our classification models had the following results

- Multilayer Perceptron: 0.93
- Support Vector Machine: 0.94
- Decision Trees: 0.96
- Random Forests: 0.97

5.2 VoxCeleb

On this dataset, our classification models had the following results

- Decision tree accuracy: 0.76
- Logistic regression accuracy: 0.89
- Hard voting accuracy: 0.89
- K Nearest Neighbors accuracy: 0.73
- Random forest accuracy: 0.88
- Support Vector Machine accuracy: 0.79

Note that all results have been measured on the validation set.

6 Conclusion

The results obtained shows that SVM algorithm performs better in classification and with reduced error rate. These results obtained using this comparative algorithm are only for this voice gender dataset and it may vary for another dataset. SVM tends to have more accuracy over another algorithm in classifying gender in spite of variations in pitch and frequency. Future work to add more algorithms to this Comparative model and to compare the performance with this work and to identify which algorithm in linear and Non-Linear performs better in the classification of gender in voice gender dataset.[3]

The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and MFCCs represent this very accurately, making them the state of the art technique for feature engineering.

All in all, the voice of a person is a complicated data sample, which is affected by many factors. However, there is a distinct correlation between gender and certain extractable features of voice which makes it possible for us, with reasonable accuracy, to predict gender on the basis of voice. This ability to predict gender becomes increasingly accurate with more number of features and the model's ability to learn and generalise the provided data, as seen in both datasets.

References

[1] B. Logan. Mel frequency cepstral coefficients for music modeling. *Proc. 1st Int. Symposium Music Information Retrieval*, 11 2000.

[2] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. *CoRR*, 2017.

[3] A. Raahul, R. Sapthagiri, K. Pankaj, and V. Vijayarajan. Voice based gender classification using machine learning. *IOP Conference Series: Materials Science and Engineering*, 263:042083, 11 2017.