# Speech Signal Processing

## Classification of Pathological speech using Mel Frequency Cepstral Coefficients on LPResidual

Niharika V

Varsha R

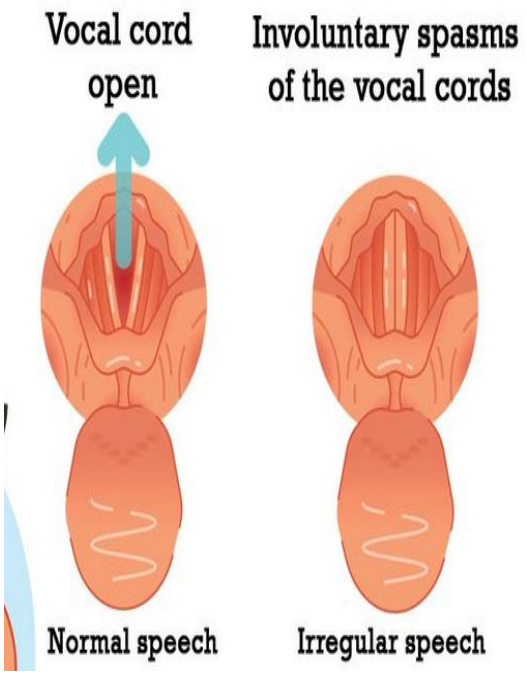
# Contents

# Introduction

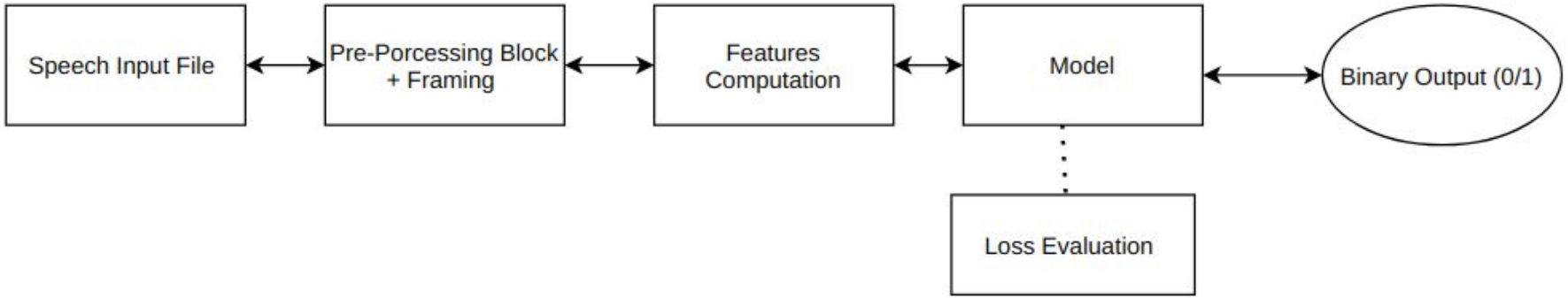
- The project aims at classification of voice-disorder data using Machine Learning techniques .  In specific , we are looking at Spasmodic Dysphonia -- a vocal cord disorder .
- Identification of voice disorders are expensive and time-consuming process --and it requires endoscopic instruments equipments.
- For solving this , we propose automated machine learning model approach extract various features of vocal cords system which differentiates the variation in normal human speech & disfigured speech .

# Understanding Spasmodic Dysphonia

- The common impairments to the voice are structural lesions, neoplasms, and neurogenic disorders.
- Spasmodic dysphonia, or laryngeal dystonia, is a disorder affects the voice muscles in the larynx, also called the voice box where the muscles inside the vocal folds spasm (make sudden, involuntary movements), interfering with vocal fold vibrations.
- Spasmodic dysphonia causes voice breaks during speaking and can make the voice sound tight, strained, or breathy.Symptoms of spasmodic dysphonia can come on suddenly or gradually appear over the span of years.
- Gradual onset can begin with the manifestation of a hoarse voice quality, which may later transform into a voice quality described as strained with breaks in phonation.



# Our Approach : Block Diagram

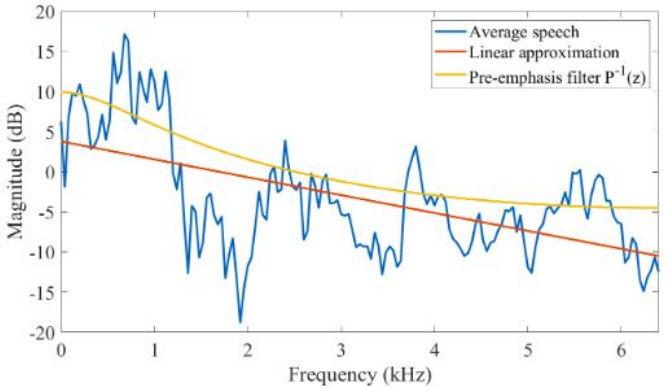


# Data Statistics

- Dataset contains samples of healthy people and people suffering from Spasmodic dysphonia.
- Consists of 42 samples of healthy people and 30 samples of spasmodic dysphonia.
- Since the data is skewed, we can try modifying our loss function, as there is class imbalance.
- The main idea is to have more penalty to the misclassification in the minority class and lesser penalty to the majority class misclassification.The weightage will be proportional to the ratio of samples in each class.

# Pre-Processing Block

- The preprocessing block is crucial , since the data is noisy and contains background speaker (whispers) .
- **Pre-emphasis Filter** : The use of this block is to highlight the high frequency components of the voice sample .
- The filter is applied as time-domain FIR filter with one free parameter (a)




F(z) = 1 - a* (z^(-1))  [ 0<a<=1 ]

- For speech signals , with sampling rate of 8 -12.6 KHz , a can be set as 0.68 .

# Features Engineering

- The performance of the machine learning model depends on its input features as well as the data present . Hence , we picked up few global as well as local features for model , to represent the vocal cords better . These are divided into local and global features .
- **MFCC** : Mel-frequency Cepstrum is a representation of a sound signal, based on the linear discrete cosine transform of a log power spectrum on a nonlinear mel scale of frequency. It helps in analysing the vocal tract independently of the vocal folds. The choice MFCC coefficients can be decided empirically , usually 13-15 are preferred .
- **LP Residual :** In LP analysis, we compute the source and excitation components from time domain analysis . Here , the prediction of current sample is based on linear combination of past p samples , where p is the order of prediction.The difference between the actual value and the approximated value is called the prediction error signal or the LP residual.The LP residual extracted using LP order in the range 8–20 best represents the speaker-specific excitation information.
- **Harmonic to noise ratio (HNR)** : Defined as the log ratio of the energies of the periodic and the aperiodic components.This quantifies the ratio of signal information over noise due to turbulent air flow, resulting from an incomplete vocal fold closure in speech pathologies.

- **Jitter** :Jitter is defined as the parameter of frequency variation from cycle to cycle.This describes the instabilities of the oscillating pattern of the vocal folds, quantifying the cycle-to-cycle changes in fundamental frequency.In the formula below Ti are the consecutive periods and N is the number of extracted F0 periods.

$$Jitter(\%) = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}(|T_i - T_{i+1}|)}{\frac{1}{N}\sum_{i=1}^{N}(T_i)}$$

- **Shimmer** :This relates to the amplitude variation of the sound wave.This indicates the instabilities of the oscillating pattern of the vocal folds, quantifying the cycle-to-cycle changes in amplitude.The shimmer is estimated as the average absolute base-10 logarithm of the difference between the amplitudes of consecutive periods, multiplied by 20. It is expressed in decibels.In the formula below Ai are the extracted peak-to-peak amplitudes and N is the number of extracted F0 periods :

$$Shimmer(dB) = \frac{1}{N-1}\sum_{i=1}^{N-1}|20log\frac{A_{i+1}}{A_i}|$$

# Machine Learning Model

- Since ,the problem involves classification , we can use techniques such as Support Vector Machine, Logistic regression, GMMs, KNN, Random forest, XGBoost, etc.
- We have experimented with Radial Basis Function (RBF) kernel in case of SVM.
- As explained in the previous slides, we have experimented with two modes of loss calculation: balancing the loss value across classes to handle the class imbalance and the normal mode. We call the former mode, “class balanced mode”.
- SVM deals with hinge loss and is a max margin classifier, while logistic is based on cross entropy loss. GMM is based on the EM algorithm. Random forest and XGBoost have similar roots, they are tree based ensembling methods. KNN is based on simple majority voting amongst the K nearest feature vectors. We have tried to accommodate different types of classification algorithms.
- XGBoost is a special tree boosting based algorithm which is designed to take lesser amount of resources to train on, but produces good results. It is a gradient boosting based tree ensemble model. Reference (For more details) : https://arxiv.org/pdf/1603.02754.pdf

# Evaluation metrics

- **Precision** -This tells us how many, out of all instances that were predicted to belong to class X, actually belonged to class X. The precision for class X is calculated as: TP/( TP + FP )
  TP = the number of true positives for class X
  FP = the number of false positives for class X
- **Recall -** This expresses how many instances of class X were predicted correctly. The recall is calculated as: TP/( TP + FN )
  TP = the number of true positives for class X
  FN = the number of false negatives for class X
- **F1 score -** This is the harmonic mean of precision and recall.
- **Accuracy score:** This denotes the number of samples predicted correctly divided by the total number of samples.

We report the macro precision, recall and F1 values in our results.

# Experiments

Things we tried as part of this project:

- Concatenating all per frame MFCC feature vectors and feeding them to classifiers
- Building a classifier which predicts at frame level and taking majority vote of all frames during inference with MFCC features.
- Varying the number of MFCC feature values from 13-20 and studying its effect.
- Adding Jitta, Shimmer & HNR features and studying their effect.
- Trying out different classifiers and studying how various algorithms are able to capture the nuances in speech signals. Trying out two modes : class balanced, normal modes wherever relevant.

# Results

We have the detailed list of results here:
https://docs.google.com/spreadsheets/d/13xdr6rXQnn15iVOiMZEflnoxoT0dqECf8gPF75C4G3g/edit?usp=sharing

**General Observations:**

a. Concatenation vs Frame by frame : There is almost **27% difference in accuracy** between frame by frame analysis and simple concatenation (90%, 63% accuracy respectively). Huge difference might be because the number of samples is quite low when we approach at signal level. (There are only 72 files), whereas there are around 55K frames in total.
b. There is not much of a difference in end result when we vary the number of MFCC feature values from 13-20. There is one peculiar case though, of SVM, for which 15 feature values gave around 86% accuracy but 13 feature values gave 90% accuracy. In all other cases there is no notable change.
c. Adding Jitta, Shimmer, HNR feature values indeed helped **improve the accuracy by 1-2.5%**. Since the feature vector size is very small, adding rich features might have helped.
d. Out of all classifiers, **class balanced versions of Random forest and SVM** performed well on the whole. We can clearly observe that the class balanced mode is outperforming the normal mode in all the cases. This is expected as the dataset is imbalanced.

Thank You