**Introduction**

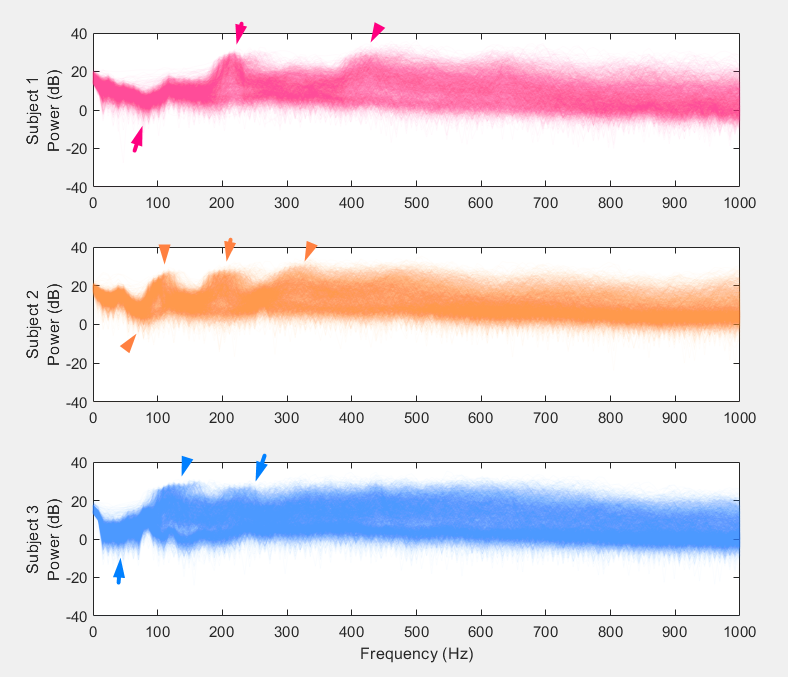
Speech recognition is widely used in current smart devices, including phones, smart speakers, as well as home controllers. Since the voice pattern of each user is different, identifying each speaker can improve the accuracy of voice recognition. Moreover, in recent Google and Siri smart speakers, smart assistance can access personal calendars, notes, contacts and reminders based on the detected speakers. Speech recognition technology can utilize the information from speaker’s voice spectrum, talking speed, punctuations or even frequently used words to make a comprehensive identification. Yet, in this project we explored the mechanism of speaker recognition with a simple 3-class classifier only based on speaker’s voice spectrum with logistic regression.

**Data Collection**

3 subjects were recruited in this test. From each subject, around 10-minute book-reading voice data was recorded with numerous voice recorders with different noise backgrounds and the best recorder (R3312, Aigo) was chosen using 16kHz sampling rate and 16-bit data width in wav format which contained minimal noise levels.

**Data Processing and Feature Extraction**

The 2-channel wav files were read with scipy.io.wavfile.read(). Right channel was discarded. Short Time Fourier Transformation was conducted on a 2048-point moving window with 512 points overlap. Energy of these Fourier Transform results was calculated and used as features for later classification. By observation, the frequencies beyond 1kHz shows limited variations between subjects. So we only used frequencies below 1kHz for later training and testing, which equivalent to first ~150 points in the Fourier Transform results. To make the dataset size the same from each subject, only the first 1000 samples from each subject were used. Therefore the total dataset contains 3000 records, where each record contains 150 features.



The fundamental, first and second harmonics of the voice data from Subject 1 and 2 can be clearly observed (green arrows), while only the fundamental and first harmonic can be observed for Subject 3 (black arrows).

**Data Labelling, Shuffling and Splitting**

The data sets from 3 subjects were then stored in panda dataframe and labelled before they were shuffled. The data might be uniformly arranged and can belong to only one subject before shuffling. Due to this the model predicts all the data points to belong to a single class. To avoid this, the data points are randomly shuffled in order to make sure the model is trained well and the predictions are accurate. This means the order of the data points is randomized. The purpose is to ensure that any patterns related to the order of data do not influence the learning process, and that the model remains generalizable and doesn't learn any sequence bias.

After shuffling, 70% of the data is under the training set. The model mainly learns from this data subset, and it tries to learn patterns to get the desired results. 15% of the data is considered for validation for tuning and avoiding over fitting. The testing set also comprises of 15% for which the model is tested for new sets of data samples never encountered by the model yet.

**Model:** ​

​​

g(w)=

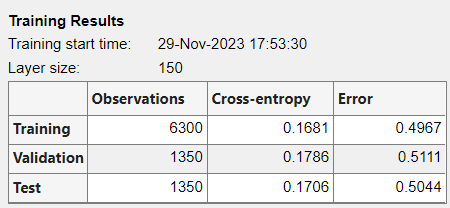
This function computes the regularized cost (loss) using the cross-entropy loss function. It is used to train the logistic regression model by punishing the wrong classification result. The regularization term, controlled by **lambda\_reg**, helps prevent overfitting by penalizing large weights.

Since logistic regression is inherently binary, this function extends it to handle multi-class problems using the one-vs-all (OvA) approach. It trained separate logistic regression weight sets for each class against the rest.

**Result**

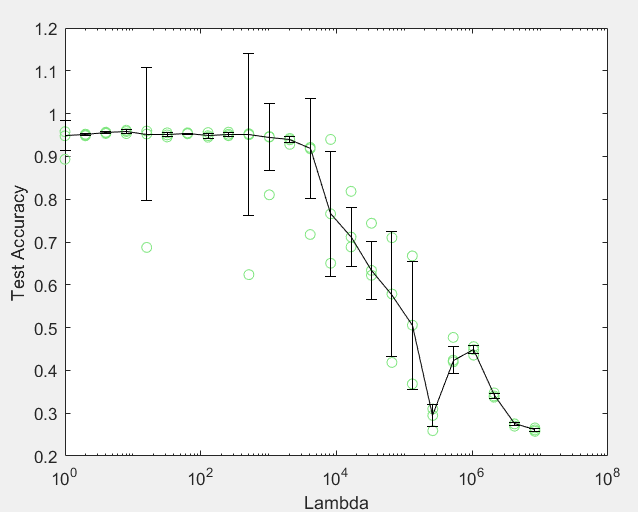
The classifier was trained with 3000 iterations and step size is 0.1. The accuracy is calculated by evaluating the correct classified record rate in the testing data set. Because the data shuffling and splitting is random, the performance of this classifier varies. But each time the accuracy achieved is around 95%. Because the 2048 moving window we used only covered 1/8 second on the 16kHz sampling rate, there are plenty of Fourier Transform windows where no speech was covered. Therefore, we consider the 95% accuracy is good.

We used same data set and fed into single layer neural network of Neural Net Pattern Recognition Toolbox in Matlab. We tried 10, 100 and 150 neurons in hidden layers; and under each setting the neural network model was trained and tested for 3 times. The Neural Network classification accuracy results were all around 95%. We believe our model has reached the best performance given the nature of the data set.

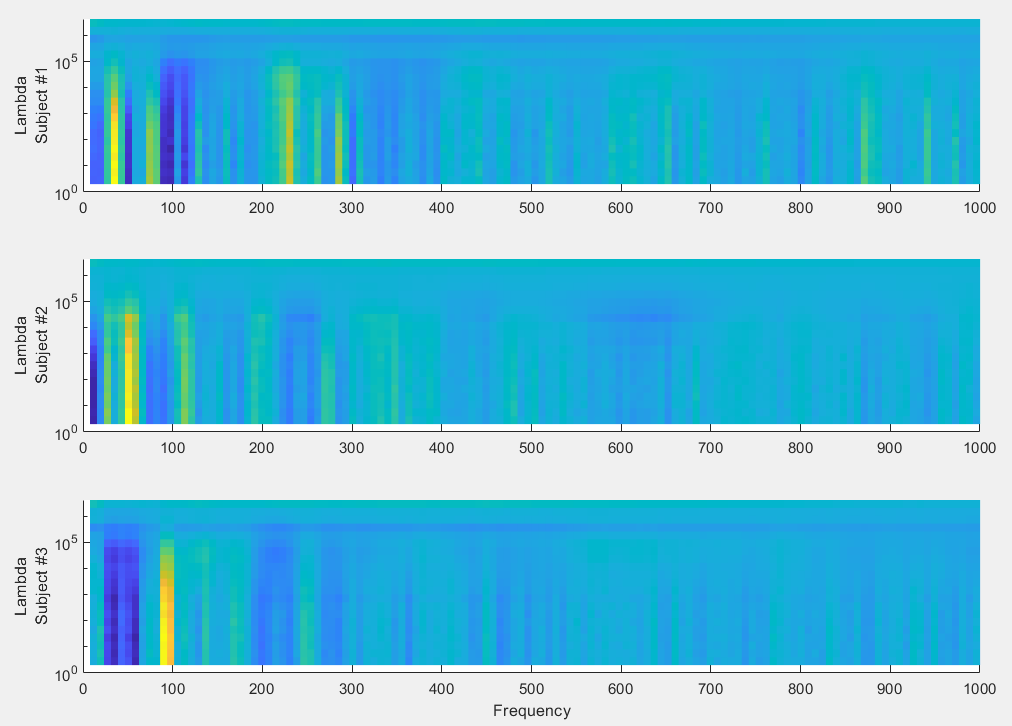


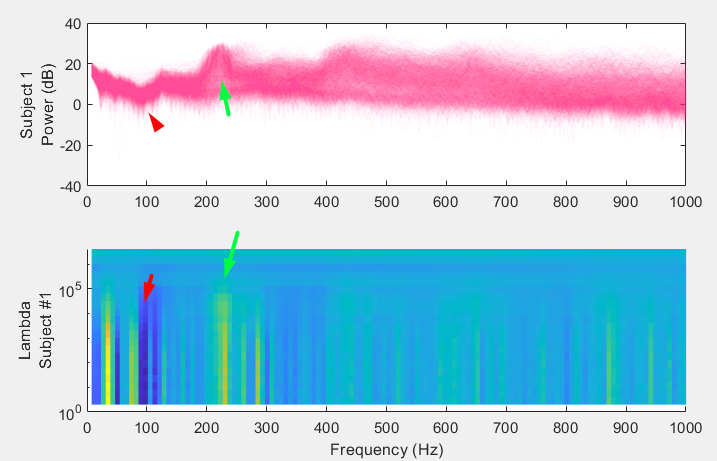
**Boosting**

Signal boosting approaches in machine learning have been gaining ground recently. The key idea behind boosting is to emphasize the learning of difficult or misclassified examples by assigning them higher weights during subsequent iterations of model training. This iterative process focuses on improving the model's performance by correcting errors made by earlier models in the sequence. In order to perform boosting, repeatedly train models by decreasing regularized lambda parameter values to improve performance. When the lambda is reduced from a larger value to smaller value, it resulted in the test accuracy rate approximately from 33% to 90% as shown in below given fig.



After training several weak models, boosting combines their predictions through a weighted majority vote or weighted averaging to create a more robust and accurate final prediction. Since, the outliers significantly impacts the mean, median of weights are used to mitigate the influence of outliers. Fig. Represents the median of weights obtained from boosting, which helped in more stable and robust prediction.





Feature Selection by Boosting

We select 6 frequency bands (7.8 Hz, 31.3 Hz, 46.9 Hz, 54.7 Hz, 85.9 Hz, 93.8 Hz, 109.4Hz) by observing the colors of figure 3 where the blue or yellow color were the strongest, indicating a larger value for classifier weight. Then we use only this 6 features to train our classifier. The result accuracy was around 82%. We were shocked that with only 6 frequencies, the accuracy is unexpected.

**Conclusion**

Overall, the project demonstrated the effectiveness of Fourier Transform for feature extraction, the significance of data shuffling and splitting in model training, the impact of regularization techniques, and the efficacy of boosting methods and feature selection in enhancing classification accuracy and speaker differentiation.

This model was even successfully transferred to perform real-time classification, and it exhibited satisfactory performance.