**Gender Recognition by Voice   
using Linear Regression**

Phuong Dinh, Nguyen Thi Kim Phuong, Nguyen Nguyen Hoang Thy, Nguyen Anh Thu

Advanced Program in Computer Science

Faculty of Computer Science, HCMUS, VNU-HCM

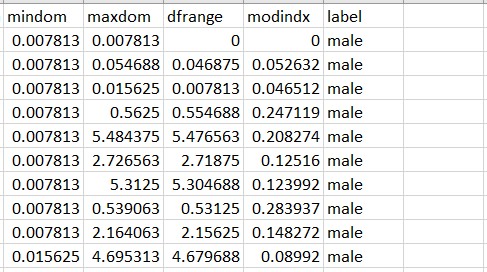
**1. Abstract**

The number of products and services having speech-recognition interfaces keeps growing. Gender misrecognition of those services may lead to wrong product/article suggestions. Many wrong suggestions decrease users’ satisfaction, which hurts the business ultimately. In this project, we try to determine the acoustic property that has the biggest impact on gender recognition by voice. Knowing that property, we will be able to optimize the categorical result and enhance the products/services that include speech-recognition interfaces.

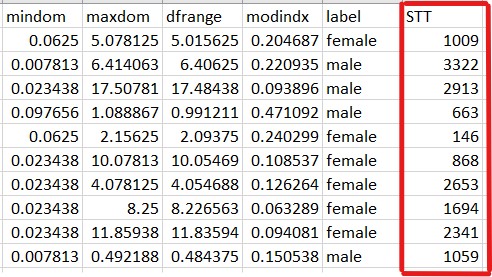
**2. Introduction**

Gender recognition is one of the first steps in identifying a person via their voice. In this project, we try to distinguish between a male and a female voice. A voice can be recorded and stored in a \*.wav file. From our observation, it is hard to train the machine to recognize the gender from these raw files. So the voice data set is stored in a csv file, which contains comparable figures. Those figures are generated from raw files using Fast Fourier Transform. Based on acoustic properties of a voice, mainly the frequency, we can tell apart if a voice is of a male or a female person.

The data set we get from Kaggle is already converted to \*.csv file and have more than 3000 voice sample labeled “male” or “female”. The available code gives pretty high accuracy. So we do not try to improve the accuracy anymore. Instead, we rearrange voice records and train the data set and deploy 5-Fold Cross Validation method. That way, we can make use of the data set in both traing and testing phases.



[*Data-set-before]*



[*Data-set-after]*

[Explanation: add STT field and mix up the data set]

**3. Proposed method**

Wav → csv file

Training

(cross validation)

Comparison

Using Linear Regression

Testing

**3.1 Overview**

**Step 1:** Transform to speech signal

**Step 2:** Training phase (8-2) cross validation

**Step 3:** Using **Linear Regression**

**Step 4:** Testing

**Step 5:** Comparison

In the step transform, we suggested doing Fourier transform to transmit from wav to csv file (for more detail, please refer to section 3.2). Taking the dataset from a similar Gender Recognition project on Kaggle [1] which contain more than 1000 samples in csv file, we mix and change the dataset so as to best fit our model. Turning to next step, we split the given data into 8:2 ratio, meaning 80 percent of which will be used for training purpose and the remain 20 percent is for testing.

The Linear Regression we pick up will used nearly 20 x-dimension (freq, kurt, Q25, etc.) and its respective y-dimension (the gender) to predict the equation between x variables and y result.

Since we use cross validation, the dataset order will vary so at each time training, we can use test set to measure the accuracy of this method and compare its mean accuracy with other method’s.

**3.2 Transform to speech signal**

We use the dataset on Kaggle containing more than 3000 samples. The dataset constructor uses raw sound data in wav file and then transform into csv file. By our observation, that transformation can be deployed using Fast Fourier Transform [2]. Using Fast Fourier Transform, we might convert the time domain in wav file into the frequency domain of the csv file.

Initially, the given dataset arranges female data and male data into to top-end of the file. To make the result more accurate, we try to mix the data of male and female by hand. We also try to parameterize the male label to 0 and female label to 1. After using Seaborn toolkit in Python to draw the relation between each pair of the data, it can be observed that some features have linear relation with each other. Hence, we omit IQR feature since it has the mentioned relation with Kurt feature and observe the change of the algorithm’s mean accuracy.

**3.3 Training phase (8-2) cross validation**

With the training section, we decide to use the ratio 8:2 between the train set and the test set to support for the comparison step (3.5). Furthermore, we also apply cross validation to the dataset. The idea of cross validation is to mix and pick the latest dataset so it will create a new train set and test set each time user repeat the algorithm, instead of, doing so by hand.

**3.4 Using Linear Regression**

The main reasons for choosing the algorithm Linear Regression varies. Linear Regression is an explanable algorithm and performing well in price predicting problem. Furthermore, there are already a lot of related works about the Gender Recognition project by using other algorithms such as Logistic Regression and Random Forest. Hence, we want to build another approach and broaden the choose to solve this problem.

Linear Regression will try to draw a linear relation from the output y with other components (k from 1 to n):

y =++...+

From the train set, it will predict the n coefficients a and curve the equation with the most possible accuracy gauging in train set, then applies them to predict y from test set.

**3.5 Testing**

Having done with the train set, we will have the equation to predict y. Next step is to apply x-component from test set to predict y and compare it with the actual y value, in this case, male or female. We then can measure the accuracy in this run.

Repeat the algorithm more than 5 times, we gain the relatively mean accuracy of this algorithm. The more repetition, the more concrete the accuracy is.

**3.6 Comparison**

Since the accuracy of the algorithms we find on Kaggle are pretty high (greater than 90%), we are not trying to optimize the result. Instead, we train and test the data set with different algorithms to see if there is any big difference. Those algorithms include Decision Tree, Random Forests, Gradient Boosting, Support Vector Machine, Multilayer Perceptron.

**4. Suggested work**

Next step, we try to calculate the confusion matrix. Since there are an unsolved output region having in common female and male, ranging from 0.1 to 0.9, we will research for a method to split the common region.

***Reference***

[1] Gender Recognition by Voice, Kory Becker on Kaggle

[2] <https://www.researchgate.net/publication/236982475_Spectrum_analysis_of_speech_recognition_via_discrete_Tchebichef_transform>