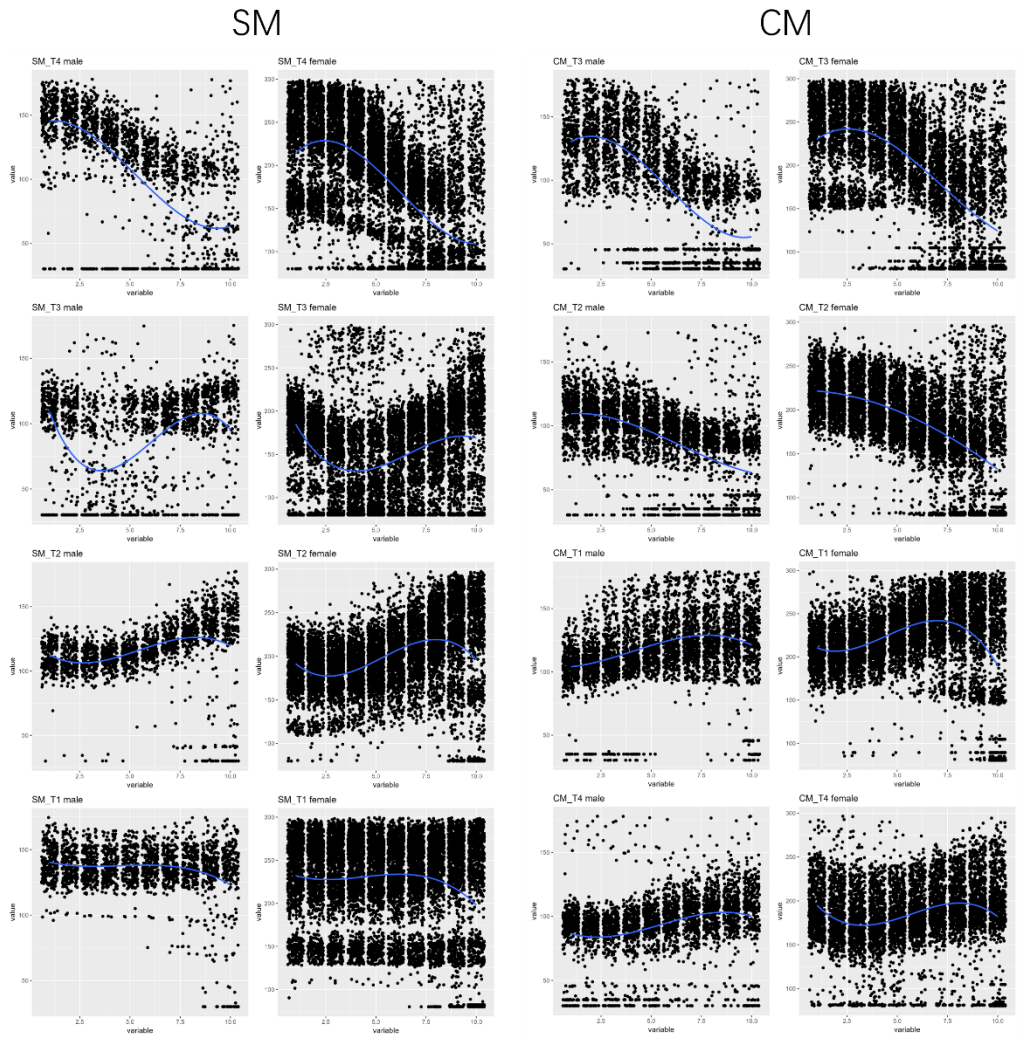
Exploring Machine Learning Applications and Challenges in Tone Recognition

Ying Xiang

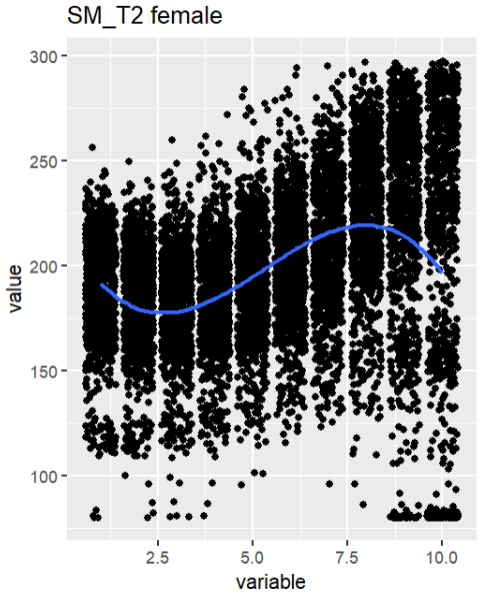
First, we need to address the issue of NAs in the data. Wenqi once mentioned that many people tend to skip over some of the lower tones when pronouncing, primarily because they are either unable to or find it difficult to produce such low tones. Therefore, my method for handling the low tones is to choose the lowest syllable from all the pronunciations of the subject to replace the NA values.
(Theoretically, this lowest value should be the lowest tone that the subject can produce.)

Afterwards, we drew scatter plots of the four different tones of SM and CM for each gender and added smoothing curves.



First, the range of voice frequencies between males and females is different, with males being lower. Even within the same gender, there can be significant differences in voice frequency, so I use scale() to standardize all data to the same standard deviation and mean. Second, the trend varies with different tones. The next step is to add some variables to reflect these trends.

I use cubic linear regression to assess trends. For example, the coefficient corresponding to the blue line in the figure should be as follows:




```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   218.09556    2.49386   87.45   <2e-16 ***
variable      -34.79307    1.86927  -18.61   <2e-16 ***
I(variable^2)   8.78489    0.38557   22.78   <2e-16 ***
I(variable^3)  -0.55231    0.02312  -23.89   <2e-16 ***
```

After adding a new variable, we used machine learning for classification. We randomly split the original data into a training set and a test set at a 70:30 ratio. Initially, we conducted classifications using a simplified version that only retained "CM_T1" and "CM_T2" using three methods: logistics, SVM, and random forest. Comparing accuracy, we found that adding or not adding the new variable representing the trend had little impact on the results.

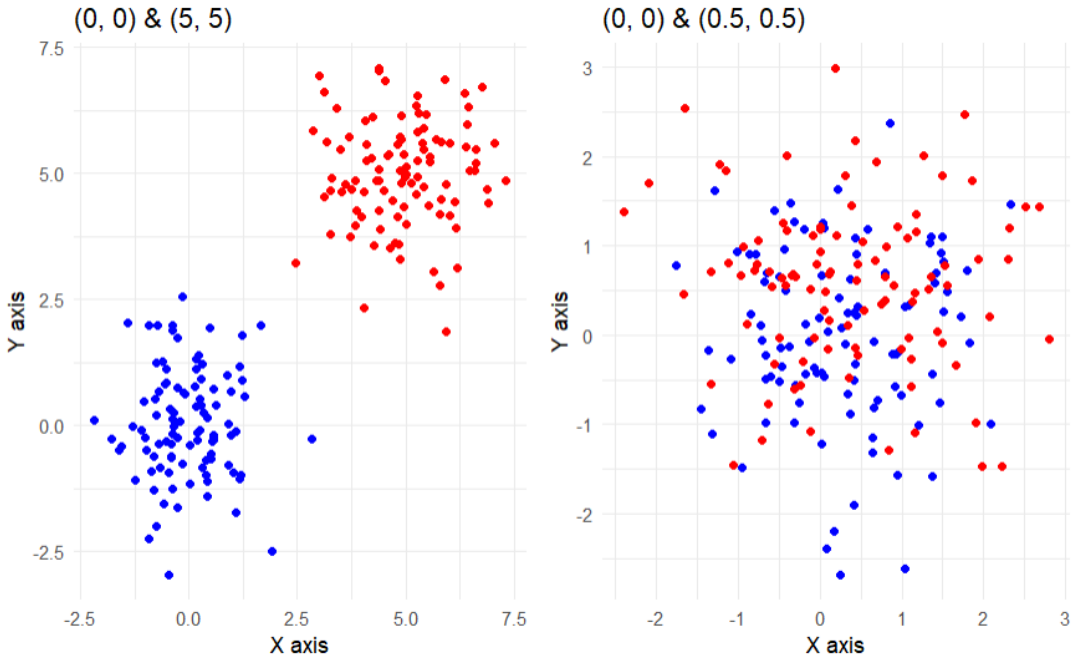
CM_T1 vs CM_T2	SVM	Random forest	Logistics
Without new variable	0.9705449	0.9749632	0.9440353
With new variable	0.9690722	0.9720177	0.9440353

Then, I applied it to a multiple classification problem. The results showed that adding the new variable slightly increased the accuracy of these three methods. Among them, random forest performed the best, with an accuracy of 0.68.

8 Tones	SVM	Random forest	Logistics
Without new variable	0.6026	0.6614	0.463586
With new variable	0.6303	0.6876	0.463586

So far, the analysis has hit a standstill. Using various methods, we have been unable to distinctly separate all eight tones of SM and CM with high precision.

Therefore, I started to consider some issues outside the task requirements. Although I am not very familiar with linguistics, my understanding of machine learning tells me that an ML model can only successfully distinguish different classes if there are actual differences between the datapoints. For example, let's say we have two sets of simulation data: the first group has a mean of (0,0) with a variance of 1 in both x and y and no covariance; the second group has a mean of (5,5) with the same variance and no covariance. We can easily separate the data of these two classes because the difference between them is very obvious.



However, if the first set of data has a mean of (0,0) and the second set a mean of (0.5,0.5), with a high overlap between the two, no matter how we process the data and regardless of the ML method used—

logistics, SVM, KNN, random forest, XGB, neural network—none can effectively separate them. The fundamental reason is that the differences between the two classes are too small.

Returning to our problem, different languages contain a large number of similar or identical pronunciations. CM and SM are inherently very similar, and it is pretty difficult to determine the specific language and tone from a single pronunciation.

In reality, how do we determine what language someone is speaking? Taking myself as an example, my family speaks both Cantonese and Chinese. I usually need to hear a few words from my parents before I can discern which language they are using, rather than determining it from the first word they speak.

Thus, I think in our case, to judge the tone of a word, if we first determine the language being used based on a sentence containing multiple words (considering that people usually do not switch the language they are using), and then judge the tone of each word given the language, could this method achieve higher accuracy?

The next simulation I conducted was randomly selecting 11 datapoints from SM or CM to form pseudo-sentences, resulting in 100 sentences each from SM and CM. Then, for each sentence, I used the random forest model with an accuracy of 0.6832 to predict the tone of each word. The source language of the sentence was determined based on whether the majority of the 11 words came from SM or CM. All 200 simulations were correct, which means that the accuracy of determining the language type through an entire sentence is very high.

Finally, given the language type, we used the CM_random_forest_model and SM_random_forest_model to predict the tone of each word, achieving a higher accuracy rate.

```{r}
CM_accuracy <- rep(NA,100)
for(i in 1:100){
  df <- CM_list[[i]]
  pred=predict(rf_model_CM,df,type = "class")
  CM_accuracy[i] <- mean(pred==df$Tone)
}
mean(CM_accuracy)
```

```
[1] 0.9345455
```

```{r}
SM_accuracy <- rep(NA,100)
for(i in 1:100){
  df <- SM_list[[i]]
  pred=predict(rf_model_SM,df,type = "class")
  SM_accuracy[i] <- mean(pred==df$Tone)
}
mean(SM_accuracy)
```

```
[1] 0.9736364
```