**I. Abstract**

K-mean is a method of vector quantization and used to classify and build speaker model based on the number clusters. In each model, K-mean essentially start randomly with a point in each cluster and find its way to the center of each cluster. Together, those points represent the speaker model as a whole. The problem with K-means algorithm is it is an NP-Hard problem; that’s mean the minimum running time complexity is O(n^k) where k is finite. This will cause the computing time to be really long. As the result of this experimenting with this speaker identification system, the higher the number of codewords the lower the error rate.

**II. Introduction**

The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus is a corpus designed for speech recognition. In using this corpus, the speaker identification is developed and designed to recognize each individual speaker’s voice. In order to implement this speaker identification system, k-means algorithm is used. The corpus provides male and female speakers with all the features has already been extracted. Before classifying any speaker, all of the feature vectors for each speaker will be trained and partitioned into clusters according to the number of codewords. The K-mean is used to represent each speaker model based on the center point of each cluster. After all of the features have been trained, a model is constructed for each speaker. The utterances for each speaker is being tested against with all of the models; whichever model give the minimum distortion, the utterance will be classified as the speaker who own that model. The confusion matrix is used to tell which speaker is being classified as which. The more number of times the test files classify the right speaker, the more accurate this speaker identification system. The error rate is also being used to tell about the accuracy of this system as well.

**III. Methods**

In the DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus, there are 438 male speakers and 192 female speakers. Each speaker has several feature and test files. This classifying system could classify either male or female based on their feature and test files. For performance purposes, female speakers will be used for this experiment since there are less female than male speakers. The algorithm to do this classification are as follow:

1. Use the provided function “partitionspeaker” to obtain the female feature files location from the corpus.
2. Use the provided spReadFeatureDataHTK function to take out all of the feature vectors.
3. Concatenate all of the feature vectors together. This is the training data for each speaker.
4. After obtained the training data for each speaker, use K-means algorithm to classify the training data based on the number of initial codewords (k). These codewords will divide the training data into k-Clusters and each codewords will get recomputed until it get to the center of each cluster. These k-clusters will be the model or the codebook to represent each speaker.
5. In order to classify the speakers, each speakers’ test files will be used to compute the distortion against all of the models. The model that gives the minimum distortion will be classified as the speaker for that tested file.
6. For any given test file, the system might classifies it wrong since this system is not perfect. We would like to see which test file is classified as which speaker; therefore, confusion matrix is used.

**IV. Results**

After running this speaker identification many times with different number of codewords, the result shows that the error rate decreases as the number of codewords increases. The result does make sense since the more number of codewords that representing the model, the more accurate this system will be. Two trials have been conducted with each trial running the number of codewords from 2 to 50. The table 1 and figure 1 show the average result of the two test trials.

|  |  |
| --- | --- |
| **Codewords** | **Error Rate** |
| 2 | 50.6% |
| 4 | 45.3% |
| 6 | 38.1% |
| 8 | 27.1% |
| 10 | 17.6% |
| 12 | 15.2% |
| 14 | 11.9% |
| 16 | 10.0% |
| 18 | 8.3% |
| 20 | 7.0% |
| 25 | 5.6% |
| 30 | 5.0% |
| 35 | 4.3% |
| 40 | 3.4% |
| 45 | 3.1% |
| 50 | 2.5% |

Table 1: Vary number of codewords producing different Error Rate

Figure 1: The graph plots the Error Rate as the function of codewords

Sample of confusion matrix are shown below:

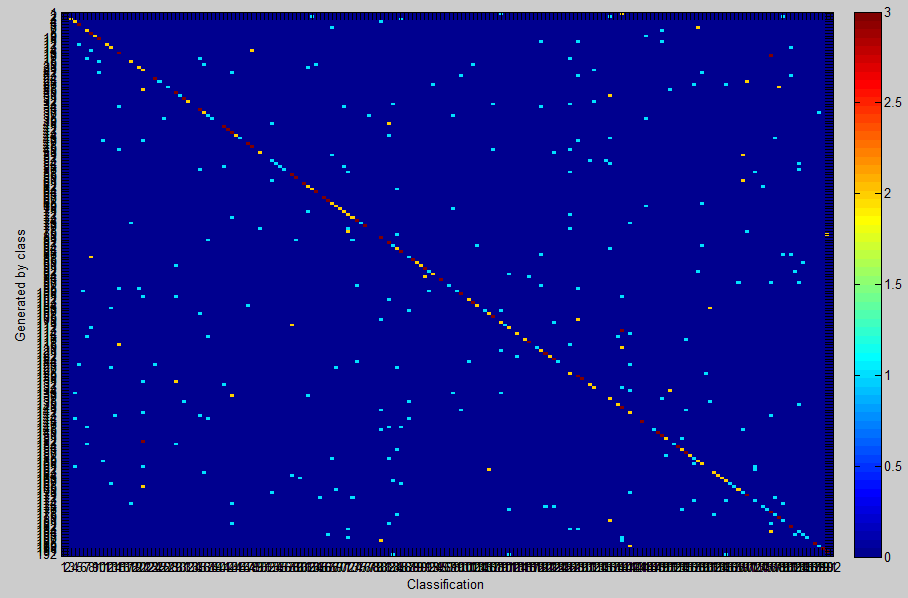


Figure 2: The chosen k is 2. The blue dot shows that many speakers are identified as other speakers. The Error rate is 50.5%

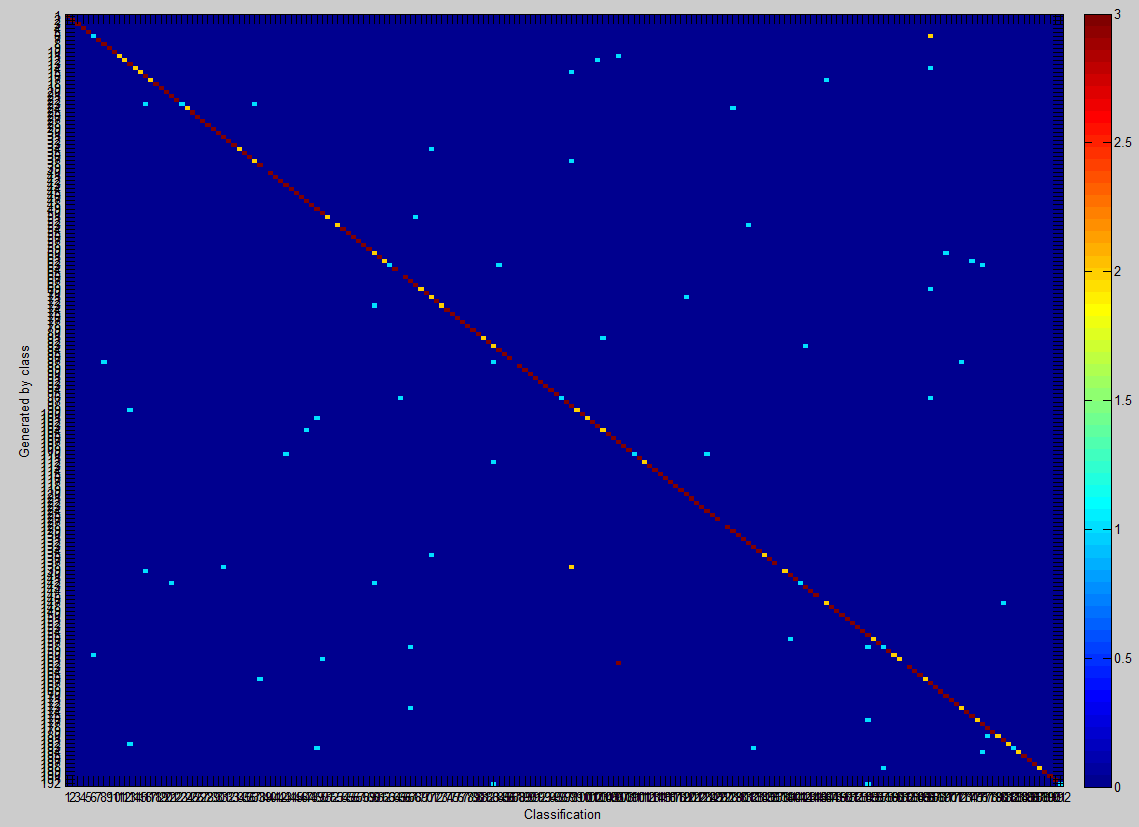


Figure 3: The chosen k is 10. Less speakers are identified as other speaker. The error rate is 16.9%

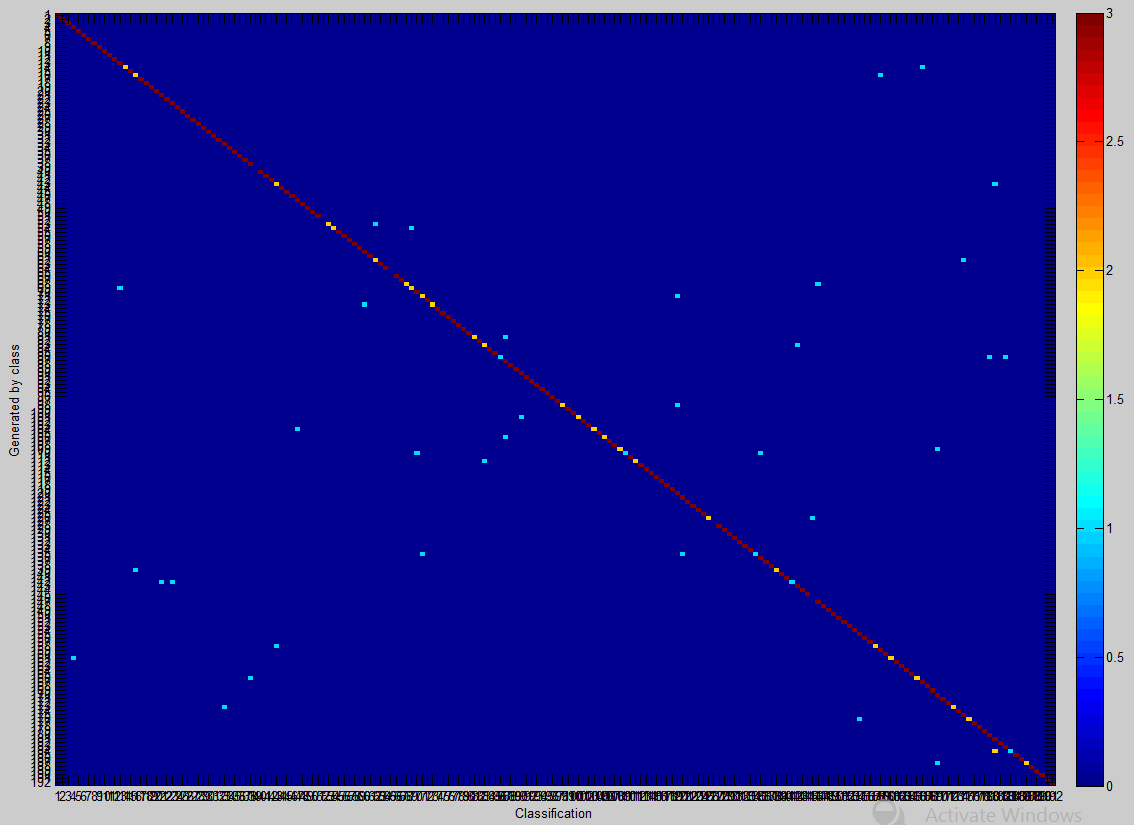


Figure 4: The chosen k is 20. The error rate is 6.4%. The diagonal line is becoming more red as the system classifies the test files to the right speakes

**V. Discussion**

The error rate seems to be really high at the beginning when number of codewords are low. As shown in the table 1 and figure 1, the error rate dramatically decreases as the number of codewords increases. The figure 2 shows when the number of codewords are two, the error rate is about 50.5%. When number of codewords are 10, the error rapidly drops down to 16.9%, more than three times as less as error when k is two; the differences are noticeable. However, comparing figure 3 and figure 4, k = 10 and k = 20, the differences are not much as noticeable as the last comparison. Although, it is also almost three times as less, but the actual value is different by a lot, 50.2% - 16.9% = 33.3% vs. 16.9% - 6.4% = 10.5%. As the data shown, the higher the number of codewords, the higher the chance the tested files will get classified with the correct model. However, as number of codewords increases, there are more clusters have to get classified and partitioned. It tremendously slows the operation down and the computation time dramatically increases. As the result, the more number of codewords, the more accurate this system will get; but the time to do the computation is huge. This is the major disadvantages of using k-mean clustering. Since k-mean is an NP-Hard problem, big running time complexity should be expected. This is where ones should decides where is the boundary line in choosing number of codewords to get good performance and low error rate.

**VI. Summary**

In designing this speaker identification system, k-means is used to classify and build a model for each speaker based on the number of codewords. The features for each speaker will get trained and partitioned into k-clusters, and each k-cluster will have a center point representing each cluster. Together, the center point in each cluster will represent the speaker model as a whole. In using the k-means algorithm, the more number of codewords, the lower the error rate will get. However, the time complex will increase tremendously as number of codewords increases. A good number of codewords should be chose to balance out between the error rate and the performance.