ECE 480 PROJCET

SPOKEN DIGIT RECOGNITION

JAEYOUNG KIL

1. PROJECT INTRODUCTION

1-1 Project Goal

The goal of this project is to create a speech recognition system that can identify spoken Arabic digits (0 - 9) based on Mel-Frequency Cepstral Coefficients (MFCCs).

The project aims to explore how Gaussian Mixture Models (GMMs) can be used to capture the distributions of MFCCs for each digit. Two approaches to estimate GMM parameters and their impact on system performance will be investigated - parameter extraction from k-means clustering and expectation maximization (EM) algorithm. The system will exclusively use maximum likelihood classification to select the identified digit.

The impact of different modeling choices, such as restrictions on the GMM covariance structure, number of GMM components, and dimension reduction will be additionally explored in this project.

1-2 Significance

This project is highly interesting as it explores the effect of different design choices in creating a probabilistic model for speech recognition. Speech recognition is a highly relevant technology in our modern lives, with virtual assistants like Apple's Siri and already changing how we interact with modern-day devices.

The probabilistic model investigated in this project is further relevant to domains beyond speech recognition. The GMM and maximum likelihood classification framework that is developed in this project can be adapted to applications we might want to make meaningful predictions given a set of observations.

For example, similar probabilistic models might be deployed in finance where future asset prices might be estimated given historical and current relevant market data. A highly analogous application would also be image recognition, where, given a set of pixel data, a system can predict what the image is of.

1-3 Background Info. - Dataset

The dataset used for this project is the Spoken Arabic Digit dataset available at the UC Irvine Machine Learning Repository. The dataset contains time series MFCCs corresponding to spoken Arabic digits from 0 to 9 by 44 male and 44 female native Arabic speakers.

Each digit is spoken 880 times (880 utterances). The training dataset contains 660 utterances for each digit, and the test dataset contains 220 utterances for each digit.

A single utterance of each digit is represented as a ‘block’ in the dataset. Each ‘block’ comprises a sequence of frames (short segments of the audio signal). Each frame is represented by 13 MFCCs that capture the characteristic of the audio signal specific to that frame. It also must be noted that each utterance of a digit is composed of a varying number of frames due to variations in the length of each digit’s pronunciation.

Figure 1-3 provides, for a single utterance of digit 0 and digit 1, plots showing the progression of the 13 MFCCs with time. **[1]**

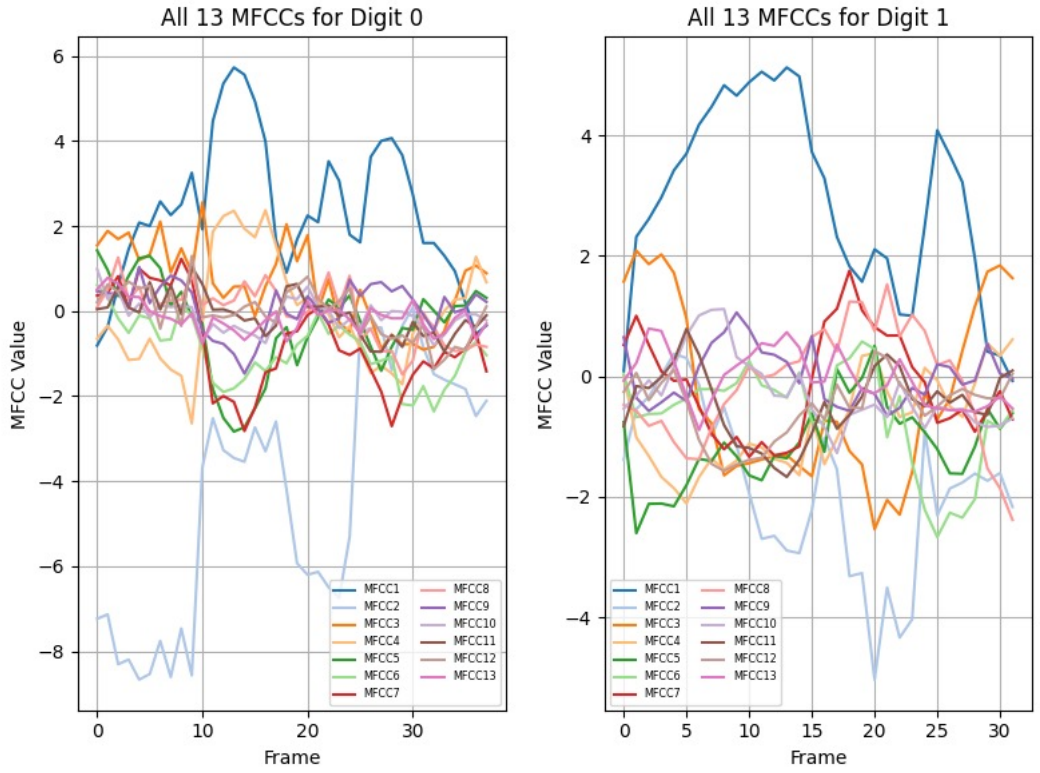

Figure 1.3 - 13 MFCCs plotted for each frame for a single utterance of digit 0 and digit 1

1-4 Background Info. – Arabic Digit Phonemes

A phoneme is an abstract representation of the smallest unit of perceivable sound **[2]** . From listening to how Arabic digits are pronounced, the number of unique phonemes and the number of syllables in each Arabic digit has been estimated and summarized in Table 1-4. The Table also provides the phonetic pronunciations of the Arabic digits from 0 to 9 **[3].**

Table 1.4: Number of unique phoneme estimates for Arabic digits

Digit	*Phonetic Pronunciation*	*# of Unique Phonemes*	*# of Syllables*
0	sĕfr	4	2
1	wâ-hĕd	4	2
2	aâth-nayn	5	2
3	thâ-la-thâh	6	3
4	aâr-bâ-‘aâh	5	4
5	khâm-sâh	5	2
6	sĕt-tâh	4	2
7	sûb-‘aâh	4	3
8	thâ-mă-ni-yĕh	6	4
9	tĕs-‘aâh	4	2

1-5 Background Info. – GMM

A Gaussian Mixture Model (GMM) is a density model that is composed of a convex combination of 'K' number of Gaussian distributions $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ such that

$$f(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$0 \leq \pi_k \leq 1 \, , \sum_{k=1}^{K} \pi_k = 1$$

where $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k : k = 1 \ \ldots K\}$. **[4]**
The parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are respectively the mean and covariance of the k^{th} Gaussian component, and the parameter π_k represents the weight of each Gaussian component.

sklearn's GMM will be used in this project **[5]**

GMM in this project:

Previously, Figure 1-3 showed the variation in the distribution of the 13 MFCCs between different digits. We can, therefore, take each frame data as a single independent observation - as a feature vector in 13 dimensions (assuming we are using the whole 13 MFCCs to represent the observation).

As suggested by the presence of sequences of frames during which the distributions of the MFCCs remain relatively unchanged, we can infer the frame data (or observations) within such sequence will form clusters in the 13-dimensional space.

We can, therefore, capture the unique distribution of the 13 MFCCs for each digit by designing a GMM model for each digit.

1-6 Background Info. – MLC

This project will exclusively use maximum likelihood classification (MLC) to identify the spoken digit. Given we have N frames, K GMM components, the likelihood of a sequence of frames is under the GMM for digit d is given by the equation:

$$f(\boldsymbol{X} \mid \boldsymbol{\theta_d}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_{k,d} \, \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_{k,d}, \boldsymbol{\Sigma}_{k,d})$$

where $\boldsymbol{\theta_d} = \{\boldsymbol{\pi_{k,d}}, \boldsymbol{\mu}_{k,d}, \boldsymbol{\Sigma}_{k,d}\}$ **[4].** Note how the GMM parameters are specific to the GMM for the digit d.

For digit classification, given a sequence of frames that represent a single utterance of a spoken digit, the digit d that maximizes $f(\boldsymbol{X} \mid \boldsymbol{\theta_d})$ will be the final classification.

$$d = argmax_d(f(\boldsymbol{X} \mid \boldsymbol{\theta_d}))$$

MLC in this project:

MLC is an appropriate choice for this project, mostly for its ease of implementation , together with the GMMs developed for each digit. Once we obtain the GMM parameters, MLC can be seamlessly implemented by simple numerical calculation of $f(\boldsymbol{X} \mid \boldsymbol{\theta_d})$ using different $\boldsymbol{\theta_d}$.

In this project, to mitigate the effect of numerical underflow, the log likelihood has been calculated instead.

2. PRELIMINARY MODELING DECISIONS

2-1 Frame Aggregation

In this project, each frame data will be taken as a single independent observation, resulting in a 13-dimensional feature vector corresponding to the 13 MFCCs. This choice will allow the model to capture details in the MFCC distribution down to the frame-level temporal resolution, providing sufficient observations per utterance of a digit.

Aggregating all frames into a single observation will introduce complications such as the significantly increased dimension of each observation and the need to handle variances in the total frame number per utterance. GMM clustering is also incompatible with this choice of frame aggregation, as an utterance of a digit is represented only by a single data point.

2-2 # of GMM Clusters

Figure 2.2 displays the time-series progression of the MFCCs for digit 4 pronounced as “theh-lah-theh” in Arabic. From Figure 2.2 it is evident that the repeated “theh” sound at the start and end of the sample is reflected by similar patterns in the MFCCs.

This confirms there is a correspondence between the quality of the sound we perceive to a unique distribution of MFCCs.

Hence, this motivates the number of unique phonemes in a digit to be a suitable estimate for the number of unique MFCC features and, synonymously, a suitable choice for the number of GMM clusters for each digit.

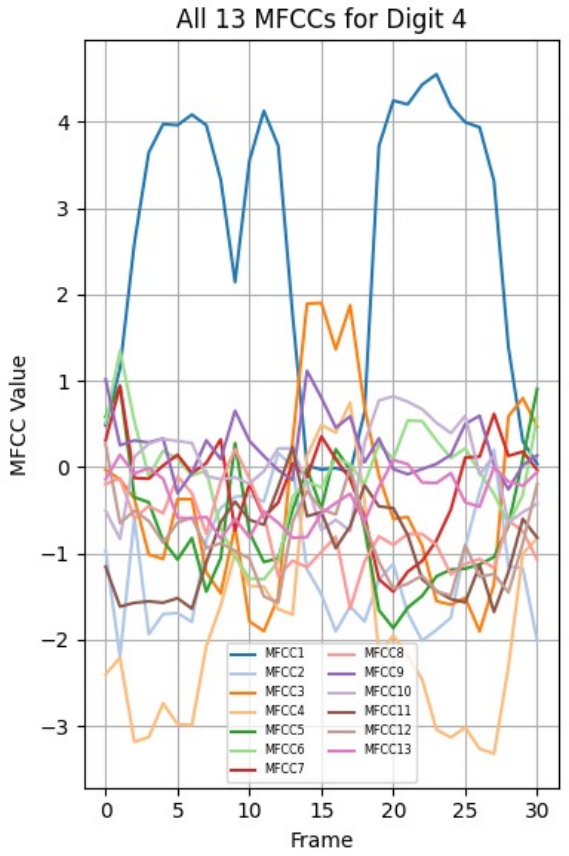

Figure 2.2: plot of MFCCs versus frame for digit 4

2-3 # of MFCCs

Initially, all 13 MFCCs will be used such that GMM models for each digit will be created in a 13-dimensional space. However, as seen in Figure 1.3, while the values of the lower-order MFCCs varied largely, the higher-order MFCCs were mostly concentrated near zero.

This suggests that higher-order MFCCs contributes little to the variance in the MFCC distribution captured by the GMMs. Hence, we will later investigate the effect of reducing the number of MFCCs used in the model on system performance, equivalent to reducing the dimension of a single observation

2-4 Covariance Structure

Initially, no restrictions in the covariance structure will be placed for the GMMs.

That is, for the K-Means-based GMM, there will be no restriction on the sample covariance matrix calculated from the data points contained within the same cluster. Note that although the initial clustering of the data points from K-Means is bound to a spherical cluster, the sample covariance matrix calculated from data points within that cluster will not necessarily be spherical.

For EM algorithm-based GMM, no restriction will be placed on the covariance matrix that is estimated by the EM algorithm.

3. K-MEANS
GMM ESTIMATION

3-1 Intro to K-Means

K-Means clustering is an unsupervised learning algorithm that is used to group a dataset into distinct clusters with hard boundaries. K-Means is performed as follows:

1. The number of clusters K is chosen, and the initial cluster centroids are randomly assigned.
2. For each data point in the dataset, its distance to each of the cluster centroids are calculated, and the data point is assigned to the cluster whose centroid is the closest.
3. The cluster centroid is recomputed as the mean of all the data points assigned to that cluster.
4. Steps 2 and 3 are repeated until the total distance between each dataset to its cluster centroid converges (until cluster assignments of data points no longer changes). **[6]**

K-Means in this project is performed by using sklearn's K-Means **[7]**. By measuring Euclidean distances, K-Means implicitly assumes the clusters are spherical.

From the result of K-means clustering, parameters from GMM can be estimated using the following equations.

$$\pi_k = \frac{\text{number of observations in cluster k}}{\text{total number of observations}}$$

$$\mu_k = \text{sample mean of observations in cluster k}$$

$$\Sigma_k = \text{sample covariance of observations in cluster k}$$

3-2 K-Means Based GMM

The K-Means-based GMM is visualized in Figures 3.2.1 and 3.2.2. Figure 3.2.1 illustrates the hard cluster assignments, where each frame is assigned to the cluster with the highest responsibility. These assignments are represented as distinct colors and plotted against MFCC1 and MFCC2. For digit 0, consistent with the estimated number of unique phonemes, the observations are clearly divided into four well-defined clusters.

In contrast, Figure 3.2.2 presents the same observations and cluster assignments as Figure 3.2.1, but plotted against MFCC11 and MFCC12. Unlike the lower-order MFCCs, the higher-order MFCCs show significant overlap between the four clusters, indicating that these dimensions do not provide clear separability for clustering. This visualization highlights that higher-order MFCCs do not contribute significantly to distinguishing phonetic variations and may therefore be unnecessary for effective digit classification. Reducing or omitting these coefficients could simplify the model without compromising performance, as the essential spectral features are captured by the lower-order MFCCs.

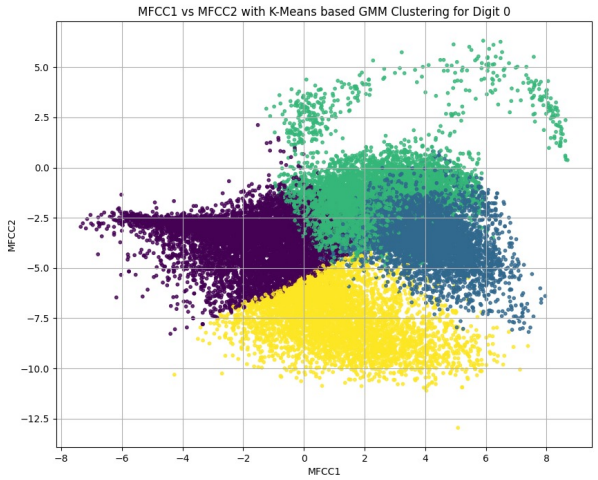

Figure 3.2.1 - Cluster assignments plotted on MFCC1 vs MFCC2

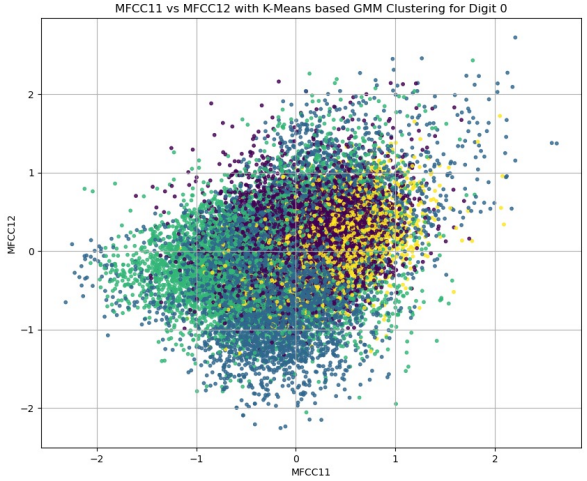

Figure 3.2.2 - Cluster assignments plotted on MFCC11 vs MFCC12

3-2 K-Means Based GMM

Figure 3.2.3 further visualizes the GMM that has been estimated by K-Means. The background color of each frame represents the cluster with the highest responsibility for that frame. We observe that, in general, sequences of frames with similar distributions of MFCCs are assigned to the same cluster.

However, across different digits, it is common to observe a "striped" pattern in the clustering. For instance, in the plot for digit 9, frames assigned to the blue clusters do not form a continuous sequence, which is unusual given the expectation that each cluster should correspond to a phoneme—a unit of speech that is contiguous in time.

This captures the limitations of the K-Means-based GMM in that the initial hard-bounded and spherical clusters of K-Means might make the clustering highly sensitive to local variations in MFCCs and fail to capture more meaningful MFCC distributions that require a more flexible cluster shape.

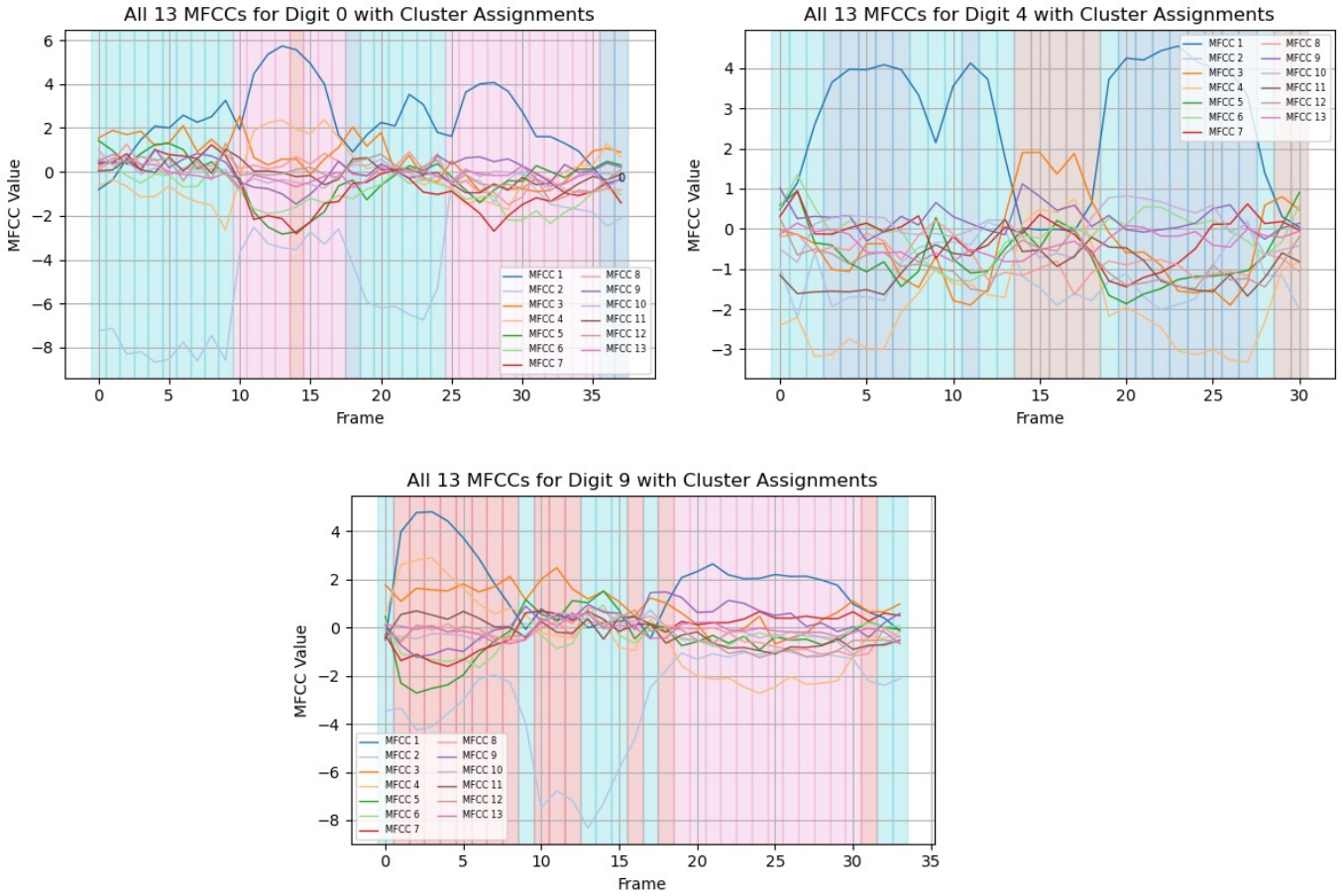

Figure 3.2.3 – Cluster assignment overlay on MFCCs vs frame plot for a single utterance of a) digit 0, b) digit 4, and c) digit 9

3-3 K-Means Based GMM Performance

Using the GMM created with parameters estimated from K-Means, maximum likelihood classification was performed on the test dataset. Each utterance of a digit in the test dataset was passed into the GMM created for each digit. Then, ach utterance has been classified as the digit whose GMM yielded the maximum likelihood for the given sequence of frames.

Figure 3.3 is a confusion matrix of the classification result that displays both the count of utterances classified as a particular digit and the percentage relative to the total utterance of each digit (220 utterances for each digit in the test dataset).

It is observed that the system performed poorly in classifying the digits 2, 7, and 9 correctly with a below 80% accuracy, while it classified the digits 1 and 6 extremely well with an above 95% accuracy.

The total accuracy of the system was **86.23%**

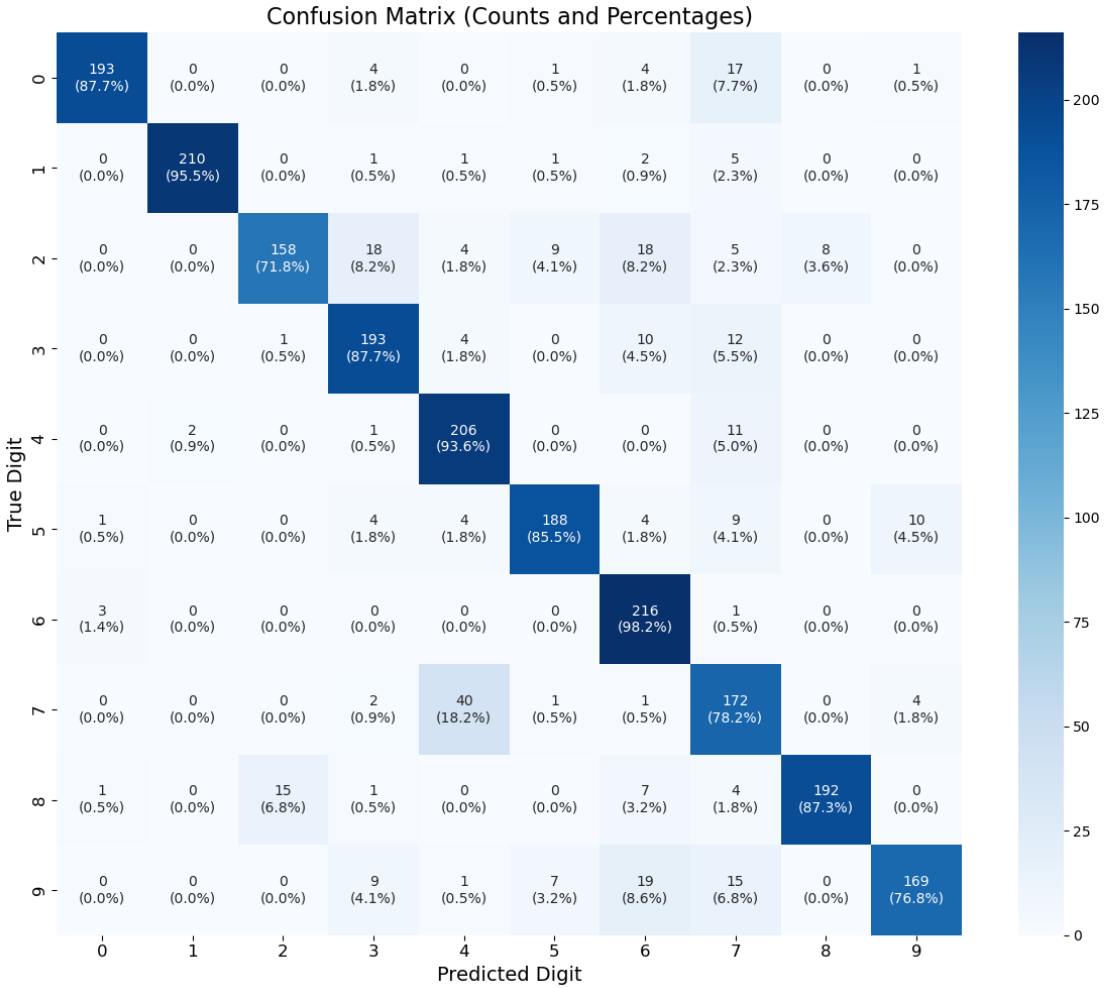

Figure 3.3 – Confusion matrix for K-Means based GMM digit classifier

4. EM ALGORITHM GMM ESTIMATION

4-1 Intro to. EM Algorithm

EM algorithm is an iterative algorithm that uses log-likelihood maximization to compute GMM parameters.

Once the initial GMM parameters are estimated from K-means clustering, in the **expectation** step (E-step), the parameters are used to calculate the responsibility r_{nk}, which explains the probability that the Gaussian component k is responsible for generating the data point x_n.

Then, in the **maximization** step (M-step), the GMM parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ are updated based on the calculated responsibilities to better fit the data using the equations on the right.

E and M steps are repeated until the log likelihood converges. EM algorithm, like K-Means clustering, is subject to converging to the local optima and is therefore sensitive to initialization. **[4]**

Responsibility Calculation :

$$r_{nk} := \frac{\pi_k \mathcal{N}(\boldsymbol{x_n} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x_n} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

Parameter Update:

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} r_{nk}$$

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} \cdot x_n}{\sum_{n=1}^{N} r_{nk}}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} r_{nk} \cdot (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} r_{nk}}$$

4-2 EM Algorithm Based GMM

The difference between the EM algorithm-based GMM and the K-Means-based GMM is most well visualized in Figure 4.2, which overlays the cluster assignment as colors on the time-series plot of the MFCCs.

In contrast to the same type of plots generated for the K-Means-based GMM, it is immediately evident that the "striped" pattern in the clustering is greatly reduced. We now observe that touching frames are more likely to be assigned to the same cluster, which is consistent with our intuition that each cluster should ideally represent a phoneme that is also contiguous in time.

However, we additionally notice that not all clusters are present in the data for a single utterance. This might indicate that some clusters are being used to capture the variability of the MFCCs introduced from dependencies other than the common phonemes for a digit, such as differences in pronunciation between speakers.

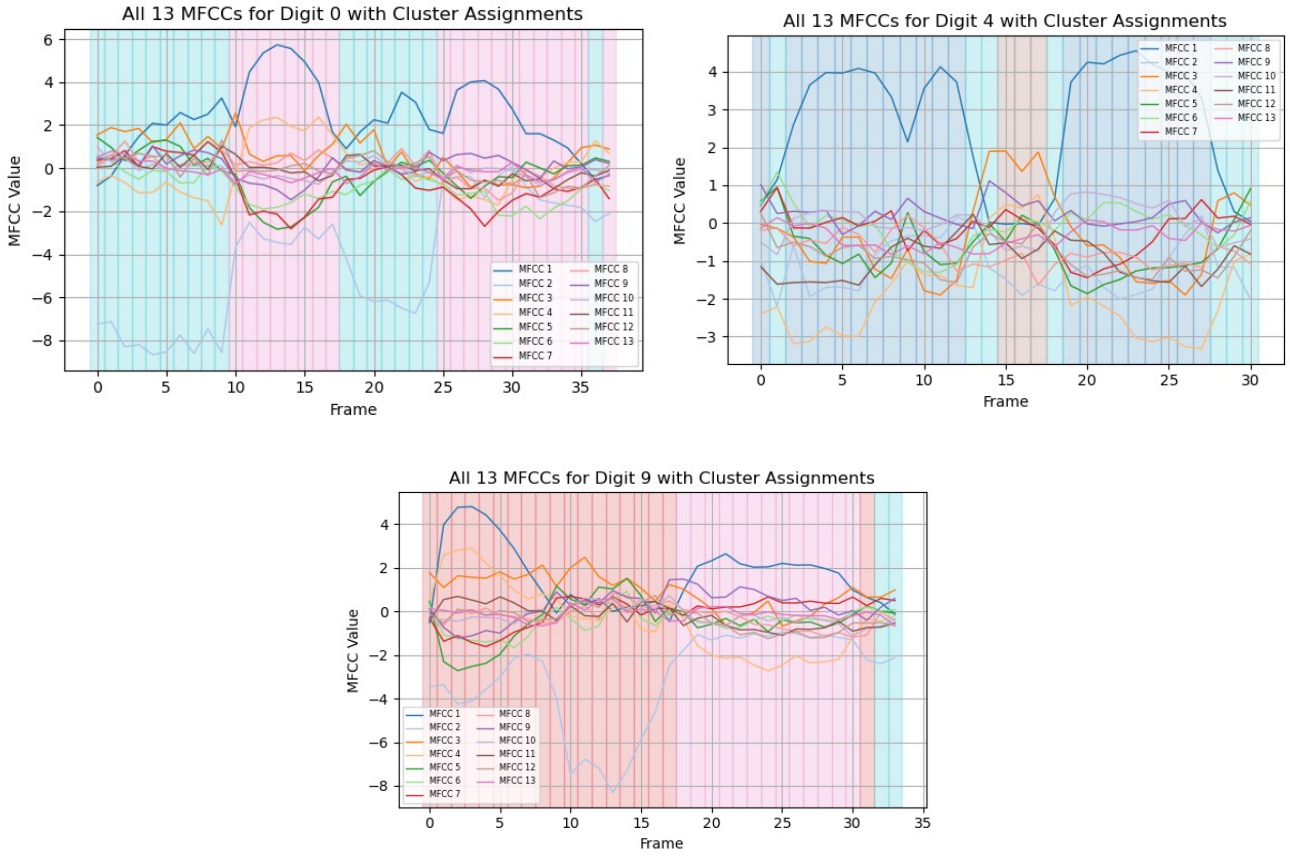

Figure 4.2– Cluster assignment overlay on MFCCs vs frame plot for a single utterance of a) digit 0, b) digit 4, and c) digit 9

4-3 EM Algorithm Based GMM Performance

The classification result is summarized by the confusion matrix in Figure 4.3, which displays both the count of utterances classified as a particular digit and the percentage relative to the total utterance of each digit.

The total accuracy of the system is now **89.05%,** which is nearly a 3% improvement from the classifier that used K-Means-based GMM.

Although much improved than the K-Means based GMM, the system still performs poorly for the same digits as before: 2, 7, and 9. The system, again, is most accurate for digits 1 and 6, with an accuracy of 96.4% in correctly classifying both digits.

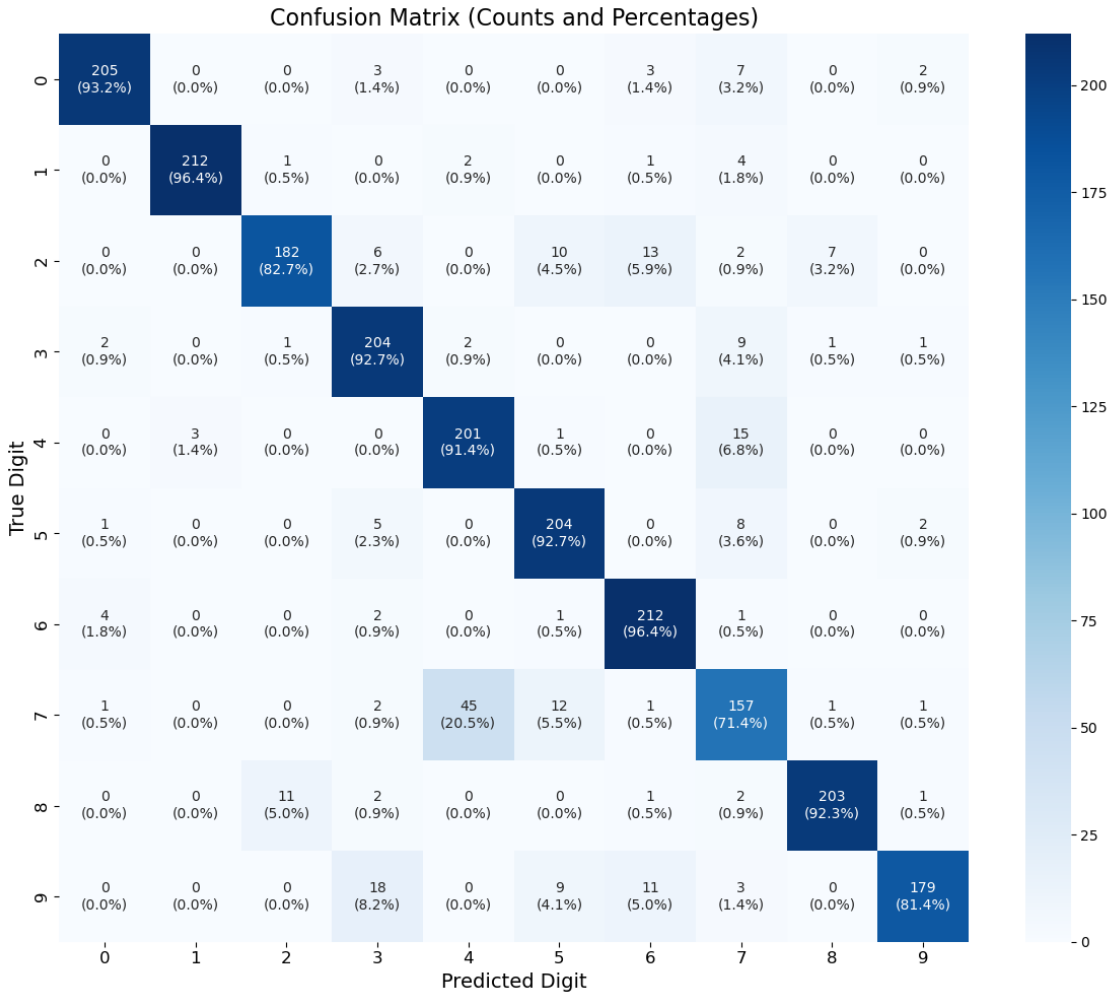

Figure 4.3 – Confusion matrix for EM algorithm based GMM digit classifier

4-4 Digit Specific Performances

High Accuracy: Digits 1, 6

Both models performed exceptionally well in classifying digits 1 and 6. A simple phonetic analysis provides insight into why these digits were accurately recognized. Referring to Table 1.4, digit 1 is pronounced as “waah-heet,” and digit 6 as “sith-tah.” Comparing these pronunciations with those of other digits reveals that both 1 and 6 contain distinctive phonemes or sounds that are absent in other digits.

These unique phonemes translate directly into distinctive MFCC features, which make digits 1 and 6 highly separable in the feature space. Once these unique features are identified by the models, they allow for confident and consistent classification, minimizing ambiguity and misclassification for these digits.

Low Accuracy : Digits 2,7, 9

Both models struggled to classify digits 2, 7, and 9, with frequent misclassifications. Specifically, digit 2 was often misclassified as 6, digit 7 as 4, and digit 9 as 3 or 6. These errors can be explained by, again, analyzing the phonetic similarities between these digits and their respective misclassifications.

Digit 2 shares a similar “ht” sound with digit 6, digit 7 shares the vowel sounds “uh” and “ah” with digit 4, and digit 9 shares similar “t” and “i” sounds with digits 3 and 6. Such overlaps in phonetic characteristics create ambiguity in the MFCC feature space, leading to confusion in classification.

Interestingly, misclassification was largely one-directional. For instance, while digits like 2 and 9 were frequently misclassified as digit 6, the reverse rarely occurred. This asymmetry is because digit 6 possesses more distinct and well-defined phonemes, which are absent in the digits commonly misclassified as 6.

5. ADDITIONAL MODELING DECISIONS

5-1 Covariance Restriction

This section will investigate the effect of covariance restriction on classification accuracy. The EM algorithm based GMMs are exclusively used in this section as it has been previously demonstrated that, on average, they perform better than the K-Means based GMM. Furthermore, it has been showed that the EM algorithm-based GMMs provide clusters that better align with the temporal continuity of speech signals.

Using the same number of GMM components as before, we explore the effect of restricting the covariance structure to three configurations:

- **Tied:** All GMM components share the same full covariance matrix, assuming similar shapes and orientations across clusters.
- **Diagonal:** The covariance matrices are constrained to be diagonal, allowing variances only along each feature dimension.
- **Spherical:** The covariance matrices are constrained to be a single variance value along the main diagonal.

Table 5.1– System accuracy based on covariance restriction types

Covariance Type	***Classification Accuracy (%)***
Full	89.05
Tied	*88.14*
Diagonal	*87.41*
Spherical	*74.05*

From Table 5.1, we observe that the classification accuracy remains relatively high despite the covariance matrix is constrained to a tied or diagonal covariance matrix. The investigation shows that system only loses up to 2% accuracy as a trade of for having to compute and store significantly less covariance computations with a diagonal covariance. The accuracy of the system, however, degrades significantly once a spherical restriction is imposed.

5-2 Dimension Reduction

Here, we investigate the impact of reducing the number of MFCC coefficients on the systems' accuracy. This is equivalent to reducing the dimensionality of the GMM as now each frame is represented by a reduced vector. The MFCCs are removed sequentially starting with the highest-order MFCCs based on prior observations that lower-order MFCCs capture the most significant variation in the MFCC distribution. We continue to use the EM algorithm-based GMM with full covariance matrix.

From Figure 5.2, we observe that the system remains largely unaffected when the number of MFCCs is reduced to 11, at which the system accuracy is 88.77%.

By reducing the number of MFCCs significant computational advantage is attained as it streamlines and reduces the computations required for the EM algorithm and the covariance matrices.

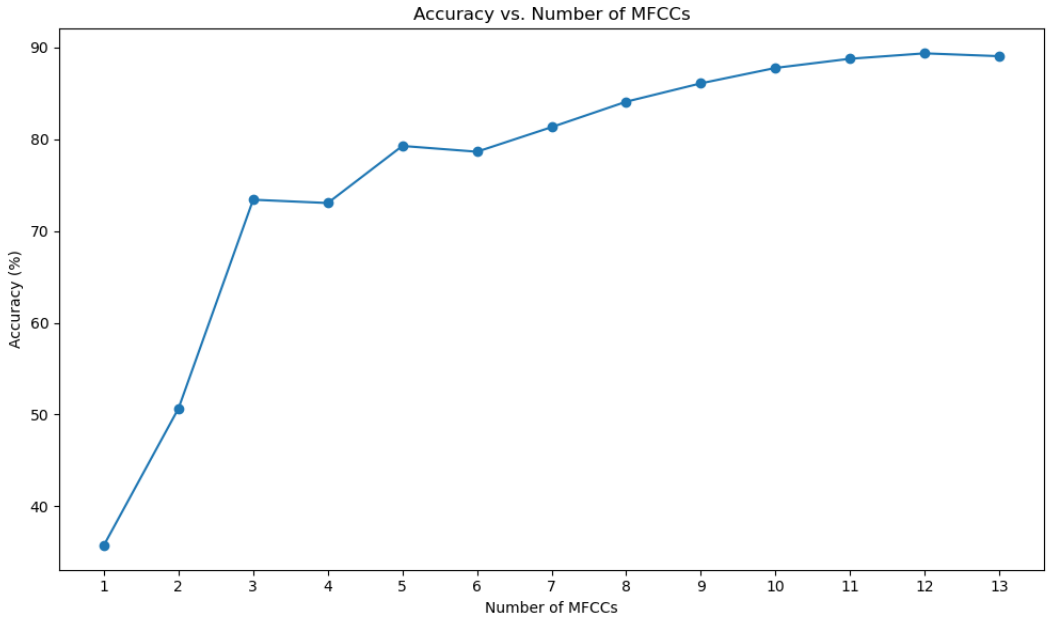

Figure 5.2– System accuracy versus the number of MFCCs used

5-3 # of GMM Components

The final additional parameter to explore is the number of cluster components used in each digits GMM. Since MFCCs are designed to capture the spectral characteristics of speech, the selection of the number of clusters remained rooted to the natural structure of speech units. Specifically, this section investigates two alternative approaches:

1. Using the number of syllables in each digit's pronunciation

2. Doubling the number of unique phonemes for each digit to account for intermediate frames between distinct phonemes.

From Table 5.3, it is notable to see that using number of syllables per digits, resulting in only 2 to 3 clusters per GMM, yielded similar accuracy to comparable to those when double the unique number of phonemes was used. In the context of bias-variance trade-off, test error increases as the model moves away from this optimal cluster configuration. Using fewer clusters (the syllable-based approach) increases bias while increasing the number of clusters excessively (the doubled phoneme approach) raises variance. Hence, we conclude the number of unique phoneme in each digit roughly does provide a balanced estimate for the number of GMM components.

Table 5.3– System accuracy based on cluster number choices

Number of Clusters	***Classification Accuracy (%)***
Number of Unique Phonemes	89.05
Number of Syllables	*85.09*
Double the Number of Unique Phonemes	*85.82*

6. CONCLUSION

6-1 Evaluation of Modeling Choices

This project has successfully demonstrated the use of GMM and maximum likelihood classification to recognize spoken Arabic digits. In particular, the effect of the following modeling choices has been explored:

K-Means vs EM based GMM

Two methods for estimating GMM parameters were explored in this project. While K-Means is computationally simpler, we demonstrated that the more rigorous probabilistic framework of the EM algorithm enabled the EM algorithm-based GMM to better capture the variability of MFCCs for a given digit.

This was particularly evident in the fact that EM algorithm-based GMMs were more likely to classify adjacent frames into the same cluster, aligning with the contiguous nature of phonemes. As a result, the EM algorithm-based model outperformed the K-Means-based model.

Covariance Structure

Although using a full covariance matrix provided the highest accuracy, restricting the covariance structure to a diagonal or tied covariance resulted in only a minimal sacrifice in accuracy while drastically reducing computation time and memory overhead. This makes covariance structure restriction suitable for applications where runtime performance is critical, such as real-time systems.

The exception to this is the spherical covariance structure, which significantly reduced the model's accuracy

6-1 Evaluation of Modeling Choices

Number of MFCCS

The selection of the number of MFCCs directly impacted both classification accuracy and computational efficiency. We demonstrated that the first 11 MFCC coefficients were sufficient to capture the distribution of MFCC features for accurate digit classification, as the lower-order MFCCs accounted for most of the variation in MFCC features within a given digit.

The inclusion of higher-order MFCCs provided only minimal improvements in accuracy, while significantly increasing computational costs.

Number of GMM Components

The number of GMM components determines the granularity with which the phonetic characteristics of speech are modeled. In this project, three different estimates for the number of GMM components were used: the number of unique phonemes, the number of syllables, and twice the number of unique phonemes. The system that used the number of unique phonemes achieved an accuracy roughly 4% higher than the other two models.

Therefore, if we were to specify a single system that balances classification accuracy and computational load, the model of choice would be an EM algorithm-based GMM, with tied covariance matrices, 11 MFCCs representing each speech frame, and number of GMM components equal to the number of unique phonemes in each digit.

6-2 Takeaways and Improvements

Great things about the system

The system facilitates the easy modification of parameters and supports discussions on modeling decisions throughout this document. It can also be generalized to other speech datasets, as the cornerstone of the system's model relies on MFCC features of speech, which can be computed from any speech signal. One aspect that worked particularly well during system development was caching large intermediary computations as local files. This significantly reduced the system's runtime, as numerical values such as covariance matrices, GMM parameters, and log-likelihoods only needed to be computed once and could be quickly fetched later when needed. Lastly, in developing this model, I learned that simple visualizations of the dataset can often provide meaningful insights into the project's direction and better communicate ideas and design motivations to others

Improvements

To improve the system, a more methodical approach could have been pursued to optimize different model parameters. For example, cross-validation could have been used to determine the optimal number of GMM components and MFCC features prior to evaluating the test dataset against the system. Additionally, the current system does not incorporate any information about temporal dependencies, such as the fact that certain phonemes always precede others for a given digit. To address this, in addition to the MFCCs, we could calculate additional features, such as the derivatives of the MFCCs, to add feature dimensions that capture how MFCCs change over time. Finally, knowing that the sample includes both male and female speakers, the system could have been improved by accounting for such latent variables. Incorporating these variables into the probabilistic models could condition the system's performance and lead to better overall results

References

[1] D. Dua and C. Graff, "*Spoken Arabic Digit Dataset,*" UCI Machine Learning Repository, 2011. Available: https://archive.ics.uci.edu/dataset/195/spoken+arabic+digit. [Accessed: Dec. 6, 2024].

[2] J.-P. Hosom, "Speech Recognition," in *Encyclopedia of Information Systems*, H. Bidgoli, Ed., Elsevier, 2003, pp. 155–169. Available: https://doi.org/10.1016/B0-12-227240-4/00164-7. [Accessed: Dec. 6, 2024].

[3] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980. Available: https://ieeexplore.ieee.org/abstract/document/1433720. [Accessed: Dec. 5, 2024].

[4] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge University Press, 2020, pp. 348–359.

[5] "GaussianMixture," *scikit-learn: Machine Learning in Python*, Scikit-Learn Developers, 2024. Available: https://scikit-learn.org/dev/modules/generated/sklearn.mixture.GaussianMixture.html. [Accessed: Dec. 6, 2024].

[6] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, 2023. Available: https://doi.org/10.1016/j.ins.2022.11.139. [Accessed: Dec. 7, 2024].

[7] "KMeans," *scikit-learn: Machine Learning in Python*, Scikit-Learn Developers, 2024. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html. [Accessed: Dec. 7, 2024].

Collaborations

In completing this project, I shared and debated high-level ideas with Brian Kim. We discussed what to anticipate in terms of how the model's behavior might change with different design choices and parameters.

Rudimentary results, such as the outcome of parsing the data or simple MFCC vs. MFCC plots, were compared with Brian Kim to ensure the dataset was parsed correctly. We supported each other in overcoming obstacles by suggesting ideas and seeking clarifications when we were confused about conceptual aspects, such as what MFCCs represent or how the dataset is structured.

I did not share code with anyone else in this class, as my code was difficult to read and because I anticipated that others would take significantly different implementation approaches, such as using different data structures to store the GMM parameters.