Speaker Recognition for Multi-Source Single-Channel Recordings

Jose Krause Perin, Maria Frank, and Neil Gallagher

***Abstract*— We have applied speaker recognition algorithms to the problem of speaker classification in multi-source (multiple speakers) of speech signals recorded on a single channel (one microphone). The goal is to separate the speakers in a single-channel recording by classifying short time frames of the recording as one of the speakers. We have evaluated three different supervised learning techniques commonly used in the speaker recognition: non-negative matrix factorization, vector quantization, and Gaussian mixture models. The feature space of the first technique is the spectrogram representation of speech signals, whereas the last two are based on mel-frequency cepstral coefficients. Initially, we compared these techniques based on the classification error rate for a designed recording. Then, we selected the best-performing technique and applied it to noisy recordings of seven speakers in a teleconference meeting.**

I. Introduction

In today's globalized work place an increasing number of team projects are executed using online communication tools for meetings. Those online meetings often include a mixture of native and non-native speakers and multiple speakers talking simultaneously, and are recorded on a single channel. In addition, different network speeds and locational background sounds create varying sound qualities for those speakers. In order to be able to retrace meeting topics organizations have an increased interest in using recordings to locate specific parts of the conversation. For this purpose, among others, it is useful to separate speakers as sources from those single-channel recordings.

The aforementioned scenario is an extension of the classic cocktail party problem [1] where, typically, the number of microphones is greater or equal than the number of sources. In our problem, however, only one microphone is used to record a mixture of different independent sound sources.

The global shape of the DFT magnitude spectrum, known as spectral envelope, contains information about the resonance properties of the vocal tract and has been found out to be the most informative part of the spectrum in speaker recognition [2]. Thus, speech signals are typically analyze in the frequency domain. We have studied two different approaches to representation in the frequency domain. The first one is based on spectrogram, made up of Fourier transforms of short time frames from the speech signal. This representation tells us how the spectral shape evolves over time. The second approach is based on mel-frequency cepstral coefficients (MFCC). The key idea behind the calculation of MFCC is to use a filter bank spaced according to the mel frequency scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands [2].

We have evaluated three different supervised learning techniques commonly used in the speaker recognition: non-negative matrix factorization (NMF), vector quantization (VQ), and Gaussian mixture models (GMM). The feature space of the first technique is the spectrogram representation of speech signals, whereas the last two are based on MFCCs.

Initially, we compared these techniques based on the classification error rate for a designed recording. Then, we selected the most efficient technique and evaluated performance for varying training sequence lengths. Furthermore, we applied it to actual recordings of team meetings. One of the project members was a participant in those recordings. Those data contain up to 7 speakers, some of whom are non-native speakers, per recording at different locations that are recorded as one monaural video.

The remainder of this paper is organized as follows. In Section II we described our approach to speech separation by speaker identification of short time frames of speech. In Section III, we described in more detail the frequency domain representation based on spectrograms and MFCCs. In Section IV we described the three different machine learning algorithms used in this project. In Section V, we compare the performance of these different algorithms and perform more in-depth analysis and simulations one the most computationally efficient of these. Moreover, we apply this algorithm to actual meeting recordings with up to seven participants. In Section VI, we discuss the impact of the results we found and possible areas for future work. Section VII concludes the paper.

II. Speech Separation Strategy

When the number of microphones is equal or greater than the number of speakers we have the well-known cocktail party problem. This problem is generally solved by the unsupervised learning algorithm called independent component analysis (ICA). Although ICA is a powerful algorithm, the requirement of a number of microphones equal or greater than the number of speakers is rarely met in practice. Thus different strategies must be adopted. Several algorithms, both assuming supervised and unsupervised learning, have been proposed towards that goal e.g., [3]–[5]

Supervised learning techniques are typically based on breaking the training data of different speakers into dictionaries and trying to find the dictionary that best fit a given sample of recording. This is either accomplished by variations of non-negative matrix factorization or support vector machines as discussed in [2], for instance.

For unsupervised learning, speech separation is typically done by independent subspace analysis (ISA) [5]. Roughly

speaking, in ISA the spectrogram of an audio recording is calculated. The magnitude of the spectrogram for the different time frames is used as different outputs for the ICA algorithm that separates the independent components of the data. After this the data must be clustered so that each frame can be assigned to the right speaker.

For this application, however, where we wish to separate speakers in a meeting recording (i.e., in a dialog), we can count with a further simplification that speakers do not overlap each other for long periods of time. Thus, this simplifies the problem to a speaker recognition problem. That is, given a training sequence for each speaker we can take frames of the single-channel recording and classify them as one of the speakers.

Fig. 1 illustrates this process. The sound waves of the speech signals of each speaker is additively combined at the microphone. The combined speech signal is then passed to the speaker recognition algorithm that based on the training data classifies a certain time frame to be spoken by one of the speakers.

A shortcoming of this approach is that it will not be able to correctly separate frames where more than one speaker is speaking. Rather, it will classify the entire frame as one speaker. However, we can minimize the errors induced by this overlapping by selecting very short time frames at which only one speaker is speaking or there is one clearly dominant speaker. Indeed, our simulations were based in time frames of the order of 30ms.

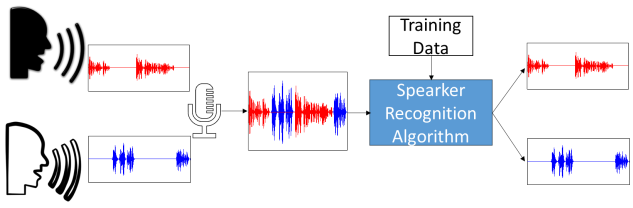

Fig. 1 – Diagram illustrating speaker-recognition approach to speech separation.

III. Speech Signal Representations

The global shape of the DFT magnitude spectrum, known as spectral envelope, contains information about the resonance properties of the vocal tract and has been found out to be the most informative part of the spectrum in speaker recognition [2]. We will use two different approaches to frequency domain representation of speech signals: one based on spectrograms and the other based on mel-frequency cepstral coefficients (MFCCs).

A. Spectrogram

The spectrogram is a widely used frequency domain representation of speech signals. It is based on the short-time Fourier transform. That is, instead of taking the Fourier transform of the entire signal (which could be hours long), we take the Fourier transform of short time frames (typically tens of microseconds). This enables us to analyzed localized frequency domain characteristics that are more useful for speaker recognition and other speech processing tasks.

More formally, the spectrogram of a discrete-time signal $s(n)$ is defined as

$$S(f,n) = \text{FFT}\{s(n+m)w(m)\}, \quad (1)$$

where $w(m)$ is a window function (typically the Hamming window), and f are the discrete set of frequencies. Fig. 2 illustrates a spectrogram of a 20s speech signal. This graph shows how the spectrum information changes with time. For this plot we chose the number of samples per window, $N_{FFT} = 512$ (which corresponds to approximately 35 ms time frame for the sampling frequency of 14.8 KHz), Hamming window, and no overlapping between frames. Note that most of the signal energy is confined within 3500 kHz.

For analyzing the speech signals we normally work only with the modulus square values of $S(f,n)$. This is motivated by the fact that our auditory system does not perceive differences in the phase of speech signals. Moreover, $|S(f,n)|^2$ is real valued, which facilitates the analysis. However, the speech signals of different speakers are modeled as additive signals in the time domain, and therefore in the frequency domain. However, this condition is not generally true for the modulus (i.e., the modulus of the sum is normally different from the sum of the modulus). Nonetheless, algorithms based on the modulus typically work fairly well.

Since $|S(f,n)|^2$ is discrete in both frequency and time we can write it in matrix form:

$$S_{ij} = \left|S(f_i, n_j)\right|^2, \quad (2)$$

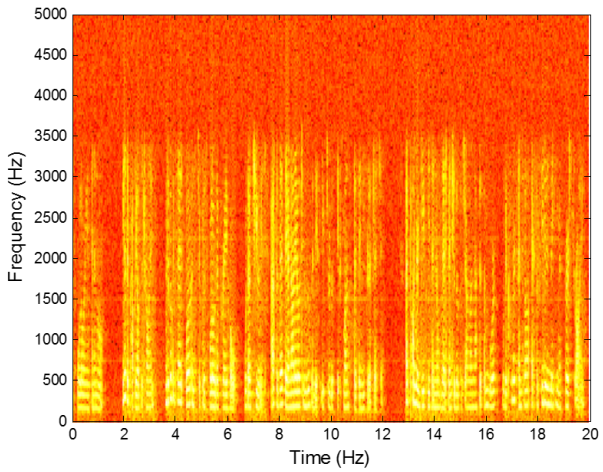

Fig. 2 – Spectrogram of a 20s speech signal. Note that most of the signal energy is confined within 3500 kHz.

where f_i corresponds to the ith non-negative frequency and n_j corresponds to the jth time frame. Thus, S is a $\left(\frac{N_{FFT}}{2}+1\right) \times N_{frames}$, where N_{frames} is the number of time frames. Moreover, each column of S corresponds to the spectral information of a particularly time frame.

B. Mel-frequency Cepstral Coefficients (MFCCs)

A more sophisticated approach to representing signals in the frequency domain is based on mel-frequency cepstral coefficients (MFCCs). The key idea to MFCC representation is to use a set of bandpass filters to do energy integration over neighboring frequency bands. The filter spacing is set accordingly to the mel-frequency scale which better approximates the human auditory system response [2]. Lower

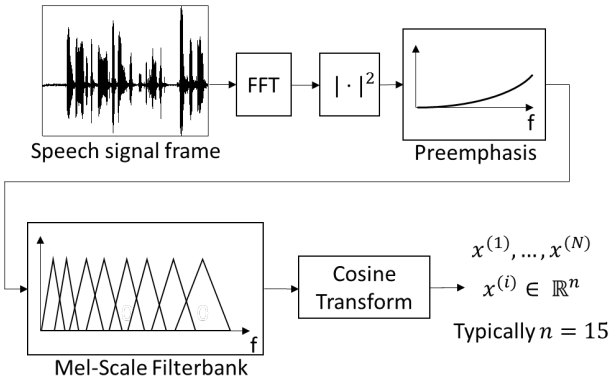

Fig. 3 – Calculation of mel-frequency cepstral coefficients.

frequencies carry more energy; thus they are represented with higher resolution by allocating more filters with narrower bandwidths.

Fig. 3 illustrates the process of calculating MFCC. Initially a time frame from the speech signal is selected (similarly to spectrogram calculation), and the modulus square of its Fourier transform is calculated. After this the preemphasis filter is used to mitigate the low-pass frequency characteristic of the vocal tract and also from the microphone. This is intended to enhance the power of high-frequency components that are naturally more attenuated. After this, we have a mel-frequency filter bank. Roughly speaking, this filter bank integrates the signal energy in certain important frequency ranges that mainly characterize the speaker. Intuitively, more filters are put in the low scale frequency where most of the signal energy is confined. As the frequency increases the spacing between filters is reduced as less discriminative speaker characteristics are presented in high frequencies. Lastly, a discrete cosine transform (DCT) is calculated and we have the MFCC coefficients. Typically, in speech signal processing no more than 15 coefficients are used. In our simulations we have used 13 and a group of 20 mel-frequency bandpass filters. An example of MFCC coefficients is shown in Fig. 4. This representation the matrix has dimensionality $N_{coeff} \times N_{frames}$, where N_{coeff} is the number of cepstral coefficients (typically $N_{coeff} \sim 15$). This reduction in dimensionality allows us to use techniques such as vector quantization and Gaussian mixture model.

IV. Models

We have studied and implemented three different models for speaker recognition: (i) non-negative matrix factorization (NMF), (ii) vector quantization (VQ), and (iii) Gaussian mixture models (GMM). These methods are commonly used in speaker recognition applications. Methods based on support vector machine (SVM) are also commonly used in speaker recognition applications.

Our goal is to identify the best performing algorithm for the application and data set we have, and then proceed to more in-depth analysis of that algorithm. The next subsections describe each one of these methods.

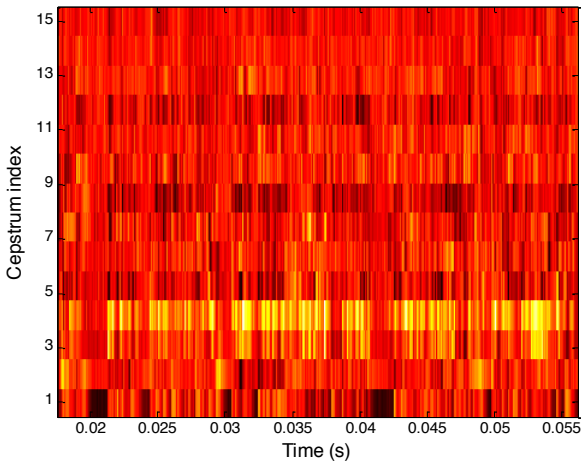

Fig. 4 – Mel-frequency cepstral coefficients for a speech signal. The color indicates the intensity of a certain coefficient.

A. Non-negative Matrix Factorization (NMF)

This method is based on a factorization of the spectrogram matrix given in (2). The non-negative S matrix is factorized into two non-negative matrices

$$S = DW. \tag{3}$$

Where D is interpreted as the dictionary matrix that characterizes a speaker, and W is the weights matrix. This way, a certain sound uttered by a speaker is decomposed as a sum of weighted sounds from a dictionary. The size of the dictionary is an important design parameter. Moreover, note that this method cannot be applied to the MFCC matrix because it is not necessarily non-negative.

The W and S matrices are obtained through the update equations [6]:

$$W := W * \frac{D^T S}{DWW^T}$$
$$D := D * \frac{SW^T}{DWW^T}, \tag{4}$$

where $*$ and $\frac{\cdot}{\cdot}$ denote element-wise product and division respectively.

For a certain frame s_n (a column of the spectrogram matrix) we calculate the weighting vector $w^{(i)}$ corresponding to the dictionary $D^{(i)}$ of the ith speaker according to

$$w^{(i)} = \arg\min_{w^{(i)}} ||s_n - D^{(i)} w^{(i)}||^2 \tag{5}$$

The frame s_n is then classified as being from the speaker who led to the minimum mean square error (i.e., minimization over i).

Results in the literature suggest that $w^{(i)}$ (or the corresponding W matrix) should be sparse so that utterances are decomposed as a combination of just a few dictionary sounds [2]. However, requiring sparsity is difficult in optimization problems. Nonetheless, $||w^{(i)}||^2$ is commonly minimized instead; even though this condition does not necessarily imply sparsity it typically leads to good performance. As a result, the optimization problem in (5) is solved by regularized least squares instead of conventional least squares.

B. Vector Quantization (VQ)

Vector quantization is a simple classification technique that assumes the probability distribution of each class is well modeled by the distribution of training examples. Classically in speaker recognition problems, this technique is applied to an entire utterance in which it is known that only one speaker is talking [2]. We denote the feature vectors representing the frames of the test segment as $S = \{s_1, \dots, s_T\}$, and the reference features vectors of speaker i as $R_i = \{r, \dots, r_K\}$. The test segment given by S is classified as the speaker that minimizes,

$$D_Q(S, R_i) = \frac{1}{T}\sum_{t=1}^{T} \min_{1 \le k \le K} d(s_t, r_k) \tag{6}$$

for some distance measure, $d(\cdot,\cdot)$. Often, a clustering algorithm is used to reduce the number of feature vectors in each reference set, R_i. When this is done, each vector r_k is a summary of each cluster, rather than an individual frame. In the case of k-means clustering, each r_k would be a cluster centroid.

In our case, we assume that the times when each speaker begins and ends a segment of speech is unknown. So instead of averaging an entire segment of speech, we chose to classify each test frame individually, as if it were at the center of a segment one second in length. The frame is classified as the speaker that minimizes,

$$D_Q(s_j, R_i) = \frac{1}{T}\sum_{t=j-\frac{F_s}{2}}^{j+F_s} \min_{1 \le k \le K} d(s_t, r_k) \tag{7}$$

, where F_s is the number of frames per second. K-means clustering with 128 clusters was used to decrease the number of reference vectors for each speaker.

C. Gaussian Mixture Model (GMM)

GMM are widely used in speaker and speech recognition tasks e.g., [7],[8]. GMM can be considered as an extension of the VQ model, in which the clusters are overlapping. That is, a feature vector is not assigned to the nearest cluster as in VQ, but it has a nonzero probability of originating from each cluster.

The MFCC of each speaker is modeled as a mixture of N multivariable Gaussian random variables [8]:

$$p(x^{(i)}|\lambda) = \sum_{k=1}^{N} \phi_k g_k(x^{(i)}) \tag{8}$$

where $x^{(i)}$ is a M-dimensional vector corresponding to the MFC coefficients of a certain time frame, ϕ_k are the mixture weights, and $g_k(x^{(i)})$ is well-known multivariate Gaussian distribution (i.e., $g_k(x^{(i)})$ is the distribution of $\mathcal{N}(\mu_k, \Sigma_k)$). Each speaker is characterized by a set of parameters $\lambda_k = \{\phi_j, \mu_j, \Sigma_j\}, j = 1, \dots, N$. These sets of parameters are estimated for speaker user based on their training sequence using the EM algorithm [9].

After every user is modeled we wish to classify what speaker $\{1, \dots, \mathcal{S}\}$ spoke a certain frame $x^{(i)}$. This is done by maximum likelihood according to

$$\begin{aligned} \hat{\mathcal{S}}^{(i)} &= \arg \max_{1 \le k \le \mathcal{S}} \Pr(\lambda_k | x^{(i)}) \\ &= \arg \max_{1 \le k \le \mathcal{S}} \frac{\Pr(x^{(i)}|\lambda_k)\Pr(\lambda_k)}{\Pr(x^{(i)})} \\ &= \arg \max_{1 \le k \le \mathcal{S}} \Pr(x^{(i)}|\lambda_k) \end{aligned} \tag{9}$$

Where the last equality follows by assuming that all speakers are equally likely (i.e., $\Pr(\lambda_k) = 1/\mathcal{S}$). This is a reasonable assumption in long recordings, and since $\Pr(x^{(i)})$ is the same for all speakers.

V. Results

Initially, we implemented and tested all these methods for test case of one male and one female speaker. The training sets were 120 seconds from recordings of each user reading a segment of a book. The test data was 30 seconds of the same users reading from a play script. The play was designed so that each speaker would speak for the same amount of time. Due to time constraints, the parameters for each algorithm were not systematically optimized.

Table 1 gives a summary of the comparative results of the different methods presented.

Table 1 – Comparative Results Summary

Method	Error rate (%)
NMF	20
VQ	23
GMM	33

A learning curve was generated for the VQ algorithm (Fig. 5). The test set was 219 seconds of the same play reading as used in the comparative tests. The training sets were also taken from the same book recordings as in the comparative tests. The learning curve displays a clear minimum at 330 seconds of training data, with a corresponding error rate of 11.0%.

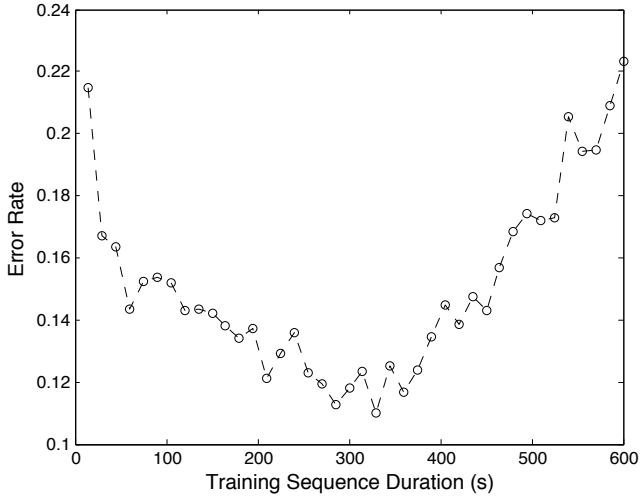

Fig. 5 – Learning Curve for Vector Quantization Algorithm

Finally, vector quantization was tested on team meeting data made up of seven different participants. Training data was generated by splitting 25 minutes of the recording into seven separate recordings, each containing only one speaker. These training recordings were then truncated so that all speakers had the same amount of training data. The VQ algorithm was then used to classify an additional 5 minutes of the original recording, using the truncated training data. In this setting, our implementation of VQ does not perform much better than chance.

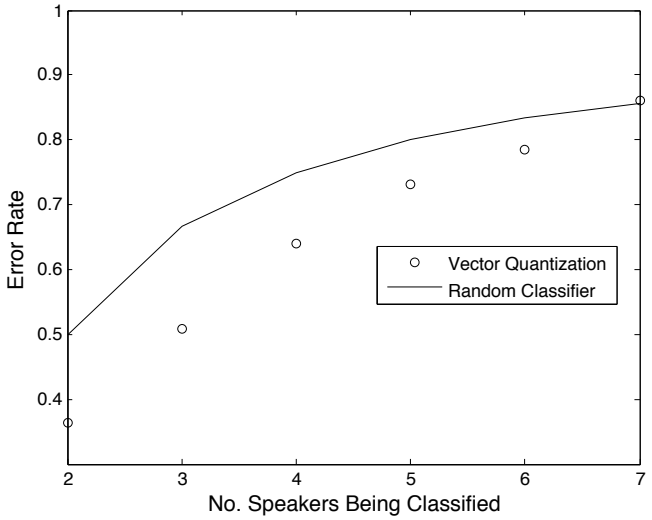

Fig. 6 – Error Rates for Multi-speaker Classification on Teleconference Recording

We also tested the effect of increasing the number of speakers in the recording on performance. This was done by using removing portions of the test data where one or more speakers are talking, then running VQ on the modified test set without taking into consideration the training sets of those speakers that were eliminated. A plot of performance for VQ with each possible number of speakers classified is given in Figure 6. The error rate when each frame is classified randomly is given for comparison.

VI. Discussion

In our comparative test, we found that of the three algorithms tested, vector quantization and non-negative matrix factorization demonstrated similar levels of performance, while classification using a Gaussian mixture model performed significantly worse. When analyzing these results, we must take into consideration that parameters for each model were not systematically optimized.

We hypothesize that the superior relative performance of the VQ and NMF algorithms is due to the fact that performance is less heavily dependent on variations in model parameters.

The poor performance of the GMM and was somewhat surprising. We, again, attribute this to sub-optimal parameters. In the case of GMM we also conjecture that performance suffered because we were able to run the algorithm only for a small number of Gaussians (~10), while results in the literature show that best results are obtained for mixture of a larger number of Gaussians (>30).

It is interesting that the learning curve for vector quantization displayed a clear minimum, indicating overfitting. Lastly, we note that the performance of VQ on the meeting recording was worse than on the male-female play recording. It is particularly important to note that performance was worse versus the male-female recording even when classifying only two speakers in the meeting recording. Likely reasons for lower performance are noticeable background noise in the meeting recording, overlap of individual speakers talking, and presence of more than one speaker of the same gender. It is also worth noting that performance of the VQ algorithm gets closer to that of a random classification as the number of speakers increases. This makes sense, as one would think that classifying more speakers makes for a more difficult problem.

The most important area of future work is to re-test the algorithms using a cross-validation framework to optimize model parameters. We hypothesize that doing this would significantly improve the performance of all three algorithms. Other additions that might help performance when dealing with complex recording environments and multiple speakers would be techniques to classify when multiple speakers are talking at the same time, and when no speakers are talking.

VII. Conclusion

We tried three different algorithms for speaker separation. The vector quantization as the simplest algorithm yielded very good results. The assumption is that this algorithm requires the least optimization. Under the time constraints we were able to generate error plots for speakers and time of training data for the vector quantization but not the other approaches.

We achieved reasonable error rates for the male-female speaker data however the current algorithms were not able to achieve comparable performance for the seven speaker data. This appears to be an issue of different audio qualities of the individual speakers as well as due to a lack of dedicated training data and more overlap.

References

[1] Z. Haykin, S.;Chen, "The Cocktail Party Problem," *Neural Comput.*, vol. 17, no. 9, pp. 1875–1902, 2005.

[2] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Commun.*, vol. 52, no. 1, pp. 12–40, Jan. 2010.

[3] H. Makino, S.; Lee, T. W.; Sawada, *Blind Speech Separation*. 2007, pp. 387–407.

[4] M. A. Casey and A. Westner, "Separation of Mixed Audio Sources by Independent Subspace Analysis," *Mitsubishi Electr. Res. Labs*, 2001.

[5] T. Virtanen, "Unsupervised Learning Methods for Source Separation in Monaural Music Signals," .

[6] B. Wang and M. D. Plumbley, "Musical Audio Stream Separation By Non-Negative Matrix Factorization."

[7] M. N. Stuttle, "A Gaussian Mixture Model Spectral Representation for Speech Recognition," 2003.

[8] R. Reynolds, Douglas; Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, 1995.

[9] A. Ng, "Mixtures of Gaussians and the EM algorithm," in *CS229 Lecture Notes*, 2014, pp. 1–4.