Analysis of Compressed Speech Signals in an Automatic Speaker Recognition System

Richard A. Metzger and John F. Doherty
Department of Electrical Engineering
Pennsylvania State University
Email: {ram476, jfd6}@psu.edu

David M. Jenkins
Applied Research Laboratory
University Park, Pennsylvania
Email: dmj114@arl.psu.edu

***Abstract*—This paper analyzes the effects popular audio compression algorithms have on the performance of a speaker recognition system. Popular audio compression algorithms were used to compress both clean and noisy speech before being passed to a speaker recognition system. The features extracted from each speaker were 19-dimensional Mel-Frequency Cepstrum Coefficients (MFCC) and the corresponding features were modeled using a 16 mixture Gaussian Mixture Model (GMM). Our experiments show that compression will have a negative effect on recognition rates if the compressed speech is clean. However, if small amounts of white Gaussian noise are added before the speech is compressed, recognition rates can be increased by as much as 7% with certain compression algorithms.**

***Index Terms*— Speaker Recognition, Gaussian Mixture Models, Mel-Frequency Cepstrum Coefficients, Audio Compression**

I. Introduction

The field of Automatic Speaker Recognition (ASR) has been well researched for several years. It is because of this intensive research that ASR is now making its way into our everyday lives. Phone banking and audio indexing in multimedia are popular areas incorporating speech recognition technology [1]. The implementation of ASR in real-world scenarios presents a set of unique challenges. One challenge is the presence of noise. Anywhere outside of a laboratory setting, noise will presumably be present in a recording. The noise present in the recording can have a detrimental effect on ASR performance [2]. Secondly, it is known that an increase in the amount of test speech can increase the performance of an ASR system. In order for practical application, compression would be necessary for maximizing the amount of space needed to store these files. The popular compression algorithm MP3 has been shown to work well in speaker recognition applications as long as the compression rate remains above 32kbps [3].

Many previous studies make the assumption that either noise is present but the speech is uncompressed, or the speech is compressed but there is no noise present. This study was undertaken to examine the effect of both compression and noise on a speaker recognition system. In addition, entropy calculations are also preformed in order to gain a better insight into the effects these distortions have on the MFCCs.

In Section II we will outline the processing techniques used to extract and compare the speech data. The experimental setup, including compression rates will be explained in Section III. Finally, the results of the experiment will be presented in Section IV.

II. Methods

The objective of any ASR system is to take in speech signals, and produce a decision based on the input. However, any sort of distortion present on the signal will casuse a decrease in performance. This experiment was set up to test severity of these distortions, specifically if the signal is compressed before being passed to the ASR system, will distortion be present.

A. Compression Algorithms

Today there exists many algorithms that are capable of high audio quality using very low bit rates. These algorithms make use of psychoacoustic models that better model speech by allocating bits to the frequency bins more sensitive to human hearing [4]. We selected three popular audio compression algorithms that make use of this model and are freely available. The schemes chosen were MP3, WMA, and Vorbis Ogg. The audio data was compressed using the program WinFF for WMA, lamedropXPd3 for MP3, and oggdropXPd for Ogg. The compression programs used above are open source and published under the GNU public license.

B. Speech Data Set

A good representation of the variation in human speech was desired for this experiment. To satisfy this requirement the Linguistic Data Consortium's Switchboard-1 data set was chosen [5]. The data set comprised of both male and female speakers, as well as regional dialects (e.g. New York City, Mid-Western). Being that these speech files were recorded from a telephone, all instances of cross-talk were removed as well as the silence preceding the voiced segments. This silence was removed using a simple energy threshold criterion.

C. Feature Extraction

Given the speech data above, the next step is extract the spectral features. There currently exists different techniques to do this, including Linear Prediction Coefficients (LPC) and MFCCs [6]. Due to its robustness and implementation into modern ASR systems, we chose to extract the features using MFCCs. The computation of of MFCC used in this experiment is as follows:

978-1-4799-8428-2/15/$31.00 ©2015 IEEE

1) Framing: A defining characteristic of speech signals is its non-stationarity. To accurately extract the short-time spectral characteristics from the non-stationary signal, multiple overlapping frames of a small duration will be used. Given the data set contains speech sampled at 8kHz, each frame will contain 128 samples with 64 samples of overlap from the previous frame. These small sample sizes are chosen to keep with convention that speech becomes quasi-stationary around a 20ms frame length [7].

2) Windowing: Within each frame, there are discontinuities at the first and last sample points. To minimize these discontinuities, each frame is multiplied by a Hamming window. The impulse response of the window for the k^{th} frame is

$$w_k[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \tag{1}$$

where N is the number of samples per frame..

3) Fast Fourier Transform: The Fast Fourier Transform (FFT) is used to convert each windowed frame from the speaker into the spectrum domain. The conversion into this domain allows the spectrum coefficients to be filtered by the Mel-filter described below.

4) Mel-Filter Bank: The Mel-filter bank will convert the power spectrum to a mel spectrum by passing it through a series of overlapping triangular filters. This triangular filter bank is based on the mel-scale shown below

$$M(f) = 2595\log_{10}\left(1 + \frac{f}{700}\right) \tag{2}$$

The mel-scale is modeled after the human auditory system and is linear up to 1000Hz, then logarithmic beyond that. To accurately model our system based on modern techniques, a total of 20 filters were implemented.

5) Cepstrum: Human speech can be modeled as a simple filter: the source (air from lungs) passes through a filter (the larynx) and speech is produced. In order for the system to identify each individual speaker, there must be a deconvolution of the source form the filter. Taking the cepstrum of the mel spectrum above will produce this deconvolution, and is found using the expression

$$\hat{x}_k = \mathcal{F}^{-1}\{\log(|\,\mathcal{F}\{x_k(t)\}|^2)\} \tag{3}$$

for the k^{th} frame of the speech signal $x(t)$.

D. Speaker Modeling

Once the features have been isolated, a model must be created for each individual so a comparison can be made and recognition preformed. There exist several methods to model features. Hidden Markov Models (HMM), Vector Quantization (VQ), and Neural Netowrks are examples of pattern recognition techniques that have been used to model features [6] [8] [9]. The most popular technique though, used in modern systems and the one used throughout this paper is the GMM. The GMM applies a probabilistic model to each frame passed to it by the feature extraction algorithm [10]. The assigning of probabilities to each of the feature frames allows for relative robustness when distortions are present in the speech.

The Gaussian mixture model is a weighted sum of K component densities, given by the equation

$$p(\boldsymbol{x}_j|\lambda_j) = \sum_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{x}_j, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{4}$$

where $\boldsymbol{x}_j$ is the feature vector for speaker j and w_i is the mixture weight satisfying the condition $\sum_{i=1}^{K} w_i = 1$. Each component is a D-dimensional multivariate Gaussian that takes the form

$$\mathcal{N}(\boldsymbol{x}_j, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{\exp\{-\frac{1}{2}(\mathbf{x}_j - \boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_j - \boldsymbol{\mu}_i)\}}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}} \tag{5}$$

with the mean vector $\boldsymbol{\mu}_i$, and covariance matrix $\boldsymbol{\Sigma}_i$. Speaker j is characterized by their model λ_j,containing the parameters

$$\lambda_j = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\} \qquad i = 1, \ldots, K \tag{6}$$

In keeping with modeling our ASR system after modern techniques, we selected K to be equal to 16, which is a model order that performs the best for our given speech utterance length [8].

E. Entropy and Spectrum Analysis

Compression, by definition, reduces the amount of bits available to represent the original signal. Therefore, when the compressed signal is compared to the uncompressed signal, there should be a loss of information. To show this, we calculated the entropy of both the uncompressed and compressed signals and compared the amount of information lost. The noise-free GMM served as the base speaker model to which both the compressed and uncompressed signals were compared. The calculation of the entropy of each signal was performed by taking the framed signal and calculating the posterior probability of a frame belonging to a mixture component in the speaker model. Hence, the entropy for each frame of the signal can be represented by the equation:

$$h_n = -\sum_{i=1}^{K} p_i \log_2(p_i) \tag{7}$$

where K is the total number of mixtures in the model, and p_i is the probability of the frame belonging to mixture i. Each frame entropy can be then constructed into the vector

$$\mathbf{H}_j = [h_1, h_2, \ldots, h_N] \tag{8}$$

for the N total frames for speaker j. The vector $\mathbf{H}$ was constructed for all speakers present in the experiment. The means of these vectors at the lowest compression rates for each algorithm shown in Table I are then calculated along with the Standard Error of the Mean (SEM).

To obtain a better understanding of effects compression had on the speech signals, we decided to analyze both the entropy and spectrum of the speech signals after they were compressed. As stated above in Section II-A, all the compression algorithms tested in this experiment utilized a psychoacoustic model in their bit-allocation. To investigate, we analyzed the spectrum of the phonemes /OW/ and /Z/, supplied by [11]. These phonemes were a recording of a male voice and sampled at 8kHz. The signals were then framed into sections containing 256 samples

TABLE I. TABLE OF COMPRESSION ALGORITHMS WITH CORRESPONDING COMPRESSION RATES

Compression Algor.	Data Rates (kbps)			
Ogg	12	16	24	32
MP3	16	24	32	64
WMA	24	32	64	128

per frame. This was done to insure the time series was quasi-stationary. After the signals were framed, noise was added and the signals were then compressed. This was done for each of the algorithms.

III. EXPERIMENT

The data set consisted of speakers taken from the 1993 NIST database distributed by the Language Data Consortium, with each speaker sampled at 8kHz with data rate of 128kbps. Four data partitions each with 37 speakers comprising of both males and females were created in order to provide a statistical measure of the recognition rates. Each of the 4 sets compressed with various compression algorithms and statistical average taken. The compression rates for each of the partitions is shown in Table I.

In order to create a text-independent training set, the speech signals would need to be segmented. The segmentation was implemented such that the first 15 seconds of speech would be used to train the speaker models. Then, a 1 second buffer of zeros was inserted to de-correlate each of the training and testing pairs. The corresponding testing segment was constructed from the next 15 seconds after the buffer.

Our first experiment will test the performance of the different compression algorithms when the tested speech is free of any external distortions. Each algorithm will be compressed to the different rates shown in Table I.

Our second experiment will test the same compression algorithms and associated rates, but the tested speech will be corrupted with Additive White Gaussian Noise (AWGN). Four different Signal-to-Noise (SNR) ratios will be tested: 0dB, 10dB, 20dB, and 30dB. When working with speech signals, there exist different criteria for what constitutes SNR, such as Local SNR and Segmental SNR [12]. Throughout the paper we will be using Global SNR, defined as

$$SNR_G = 10\log_{10}\frac{\sigma_s^2}{\sigma_n^2} \tag{9}$$

where σ_s^2 is the power of the speech signal and σ_n^2 the noise power. The AWGN was then added to each the test segments and then compressed.

IV. RESULTS

Several notable results were obtained by this experiment, most intriguing of which is the effect WMA has on a speech signal. The spectrum of a single frame of the phonemes /OW/ and /Z/, both with and without noise, and compressed as well as uncompressed is shown in Figure 1. Closer inspection of Figure 1(a) shows, at very high frequencies, the noisy compressed speech is nearly identical to the noise-free speech. On the other hand, the fricative phoneme /Z/ fails to match when compressed and exhibits a noise-like structure in the higher band. These results lead us to believe that the increase in performance for noisy speech compressed with WMA, shown in Figure 4(c) and Figure 4(b), is caused by the matching of certain vocal characteristics. We speculate that the cause of the accurate spectral matching in phoneme /OW/ is due to WMA's use of efficent spectral coding. This coding separates the signal into a baseband and extended band, and encodes the extended band as a scaled version of the already coded baseband if a match exists between them [13]. If there is no match, the coder looks to a fixed codebook of spectral shapes and sees if a match is there. If no match is present in the codebook, the coder will insert scaled random noise for the extended band. Assuming that the high frequencies greater than 3.5 kHz constitute the extended ban, the implementation of this efficient coding could account for WMA's accuracy in replicating the noisy speech signal and thus increasing the recognition rate. To validate our hypothesis, the last three MFCC's (coefficients 17, 18, and 19) were removed from the WMA compressed speech signals, and were then re-entered into the ASR system. The last three coefficients were removed because these three coefficients correspond to the high frequency components in the spectrum. As Figure 5 shows, the removal of these coefficients has caused the recognition rate to more closely follow the uncompressed noisy speech. We can thereby assume the increase in recognition rates can be attributed to the accurately modeled high frequency component. On the other hand, the noisy signal compressed with Ogg and the noisy uncompressed signal are nearly identical. We can then infer from this result that compressing speech with Ogg, when noise is present, provides no increase in performance. This can be validated by viewing Figure 4, which shows that compressing the speech signal with anything other than WMA will provide at best the same recognition rate as the uncompressed signals.

To measure the total loss attributed to the compression alone, the mean of the entropy vectors of all 148 speakers were taken, as well as their SEM. As is evident by the plots in Figure 2, there is a slight loss of information present in all three compression algorithms. The decrease in recognition rates, shown in Figure 3, can be attributed to this loss of information.

To obtain a clear picture of compression effects on an ASR performance, we ran noise-free speech signals compressed with different compression algorithms through our ASR system. As shown in Figure 3, all three compression algorithms show a clear degradation in recognition performance at specific data rates. It is interesting to note the performance of MP3, which seems to suffer from a slight degradation even at relatively higher compression rates.For the other two however, there seems to be no effect on the recognition rates until a specific data rate is reached. When noise is added to both the compressed and uncompressed test segments, the ASR performance is negatively effected. Though recognition rates are only slightly effected at 30dB, there is a steep drop off in performance between 20 and 10 dB. Figure 4 shows the recognition rates for various compression rates when noise is added. Of the three algorithms, Ogg performed the worst. As Figure 4 shows, compressing at any rate when noise is present in the signal will yield no increase in the ASR performance. In fact, compressing the corrupted signal with Ogg at a data rate below 32kbps will decrease the recognition rate by as much as 20%. This result presents an interesting

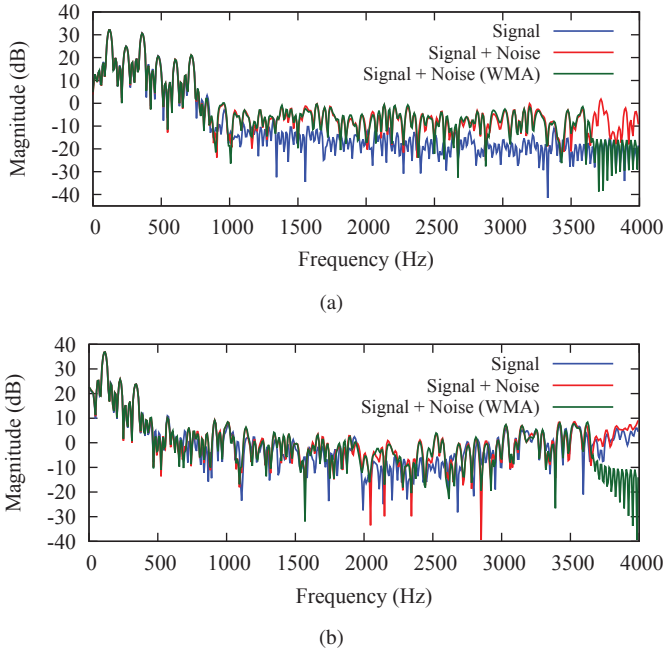

Fig. 1. Spectrum of the phoneme (a) /OW/ and (b) /Z/, both at an SNR of 20dB and compressed with WMA

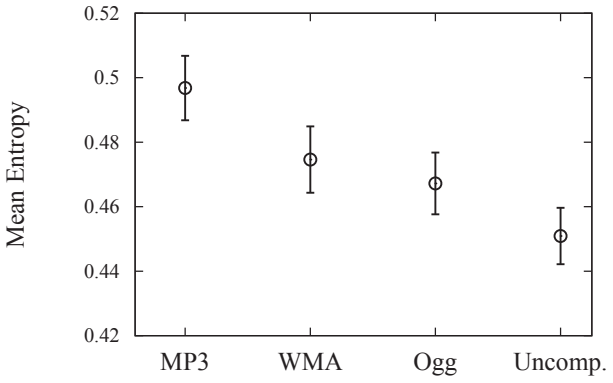

Fig. 2. Mean and SEM of the entropy vectors for the compression algorithms

dynamic with recognition rates and noise. When noise is not present, Ogg clearly preforms better; maintaining a near optimum ASR performance while reducing the bit rate to 16 kbps (a compression ratio of 8:1). When even a slight amount of noise is present during compression however, this benefit is completely lost and performance is worse off than if compression was not utilized.

WMA on the other hand, will increase the recognition rate of the ASR system if noise is present in the test segment. If the SNR is 10dB or 20dB at the time of compression, the recognition rate will improve compared to if the signal was at the same SNR level but compression was not utilized, as shown in Figure 4(c) and 4(b). The amount of improvement is around 7%. The compression rate at which WMA is used also seems to make no significant difference in the performance. This is in direct contrast to both MP3 and Ogg, which at some compression rates, the performance improved when SNR is 20dB but at other rates performance diminished. The

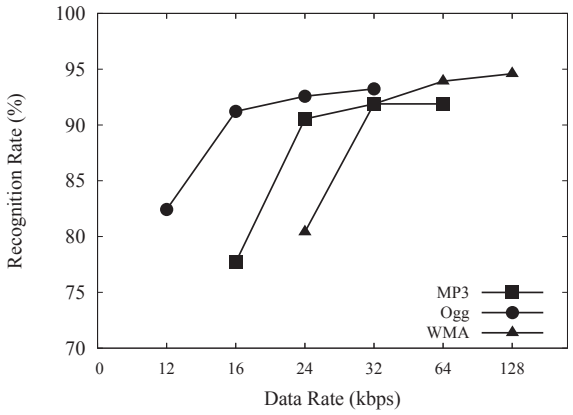

Fig. 3. Recognition rates for different compression algorithms without noise

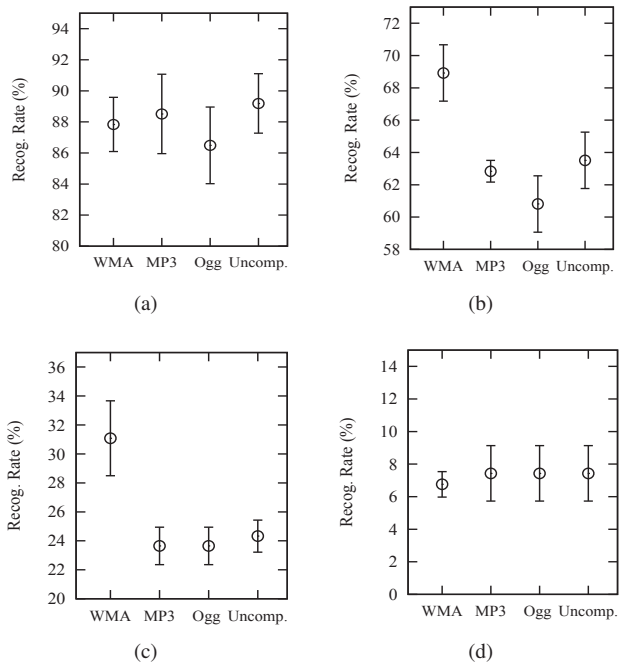

Fig. 4. Comparison of the mean and SEM of the recognition rates for SNR values of (a) 30dB, (b) 20dB, (c) 10dB, and (d) 0dB, all at a compression rate of 24kbps

downside to the WMA compression is the inability to achieve a compression rate as significant as Ogg or MP3. The lowest rate achievable for WMA given the speech signals was 32kbps, corresponding to a compression ratio of 4:1. This is twice as high as MP3, which achieved a compression ratio of 8:1.

V. Conclusion

This study has shown that while compression will have an effect on the ASR performance, that effect is dependent on the quality of the signal before compression. If the signal is noise-free, then Ogg is desirable because of its ability to compress the signal to a lower bit rate without loss of performance compared to the other algorithms. However, if noise is present in the signal, then WMA would be a better choice. In fact, if noise is present and the signal is compressed

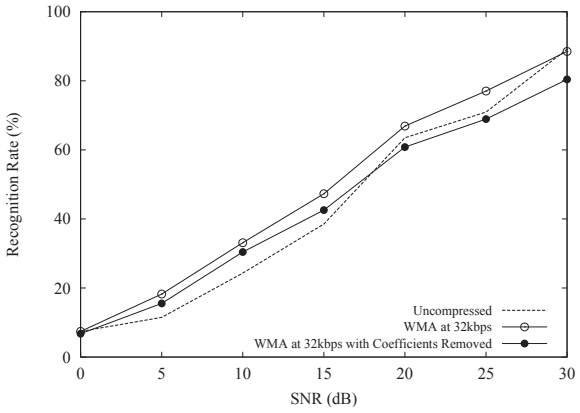

Fig. 5. Recognition rates for WMA and uncompressed speech signals for various SNR values

with WMA, then the performance of the ASR will actually increase. This increase can be as high as 7% when certain levels of noise are present. Therefore, as long as the signal is not compressed beyond one of the specified data rates, compression will not degrade performance. In fact, instances where noisy speech signals were compressed the recognition rates actually improved when compared to noisy speech signals that were not compressed.

REFERENCES

[1] M. Viswanathan, H. S. M. Beigi, A. Tritschler, and F. Maali, "Information access using speech, speaker and face recognition," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 1, 2000, pp. 493–496 vol.1.

[2] J. Saastamoinen, Z. Fiedler, T. Kinnunen, and P. Fränti, "On factors affecting mfcc-based speaker recognition accuracy," in *International Conference on Speech and Computer (SPECOM2005), Patras, Greece*, 2005, pp. 503–506.

[3] P. S. Ng and I. Sanches, "The influence of audio compression on speech recognition systems," in *9th Conference Speech and Computer*, 2004.

[4] D. Wiese and G. Stoll, "Bitrate reduction of high quality audio signals by modeling the ears masking thresholds," in *Audio Engineering Society Convention 89*, Sep 1990. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=5723

[5] E. H. John Godfrey. (1993) Switchboard-1 release 2 ldc97s62. DVD. Philadelphia: Linguistic Data Consortium,.

[6] A. Paul, D. Das, and M. Kamal, "Bangla speech recognition system using lpc and ann," in *Advances in Pattern Recognition, 2009. ICAPR '09. Seventh International Conference on*, 2009, pp. 171–174.

[7] M. Sinith, A. Salim, K. Gowri Sankar, K. Sandeep Narayanan, and V. Soman, "A novel method for text-independent speaker identification using mfcc and gmm," in *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, Nov 2010, pp. 292–296.

[8] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, Jan 1995.

[9] J. Campbell, J.P., "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, Sep 1997.

[10] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *Circuits and Systems Magazine, IEEE*, vol. 11, no. 2, pp. 23–61, 2011.

[11] D. G. Childers, *Speech Processing*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.

[12] M. Vondrášek and P. Pollak, "Methods for speech snr estimation: Evaluation tool and analysis of vad dependency," *Radioengineering*, vol. 14, no. 1, pp. 6–11, 2005.

[13] S. Mehrotra and W. Chen, "Efficient coding of digital media spectral data using wide-sense perceptual similarity," Dec. 2 2008, uS Patent 7,460,990. [Online]. Available: https://www.google.com/patents/US7460990