Explore Audio Compression Using Machine Learning

A dissertation submitted for the Degree of Master of Business Analytics

N. C Ellepola

University of Colombo School of Computing

2020

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: N C Ellepola

Registration Number: 2018/BA/011

Index Number: 18880112

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr. /Ms. _______________________ under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by,

Supervisor Name: Dr M. I. E Wickramasinghe

____

Signature of the Supervisor & Date

I would like to dedicate this thesis to...

P. P Ellepola

ACKNOWLEDGEMENTS

I would like to express my heartiest gratitude to all those who provided me help to progress my project this far. I convey my special gratitude to Dr M. I. E Wickramasinghe for the guidance and encouragement. Furthermore I would like to acknowledge with much appreciation the crucial role of Dr H. K. T. C Haloluwa in coordinating advising on project timelines.

ABSTRACT

Storage requirements for digital applications are increasing rapidly. There are considerable amount of applications in the world that use audio data. Data compression is one of the key solutions in saving storage space, and has been used for many decades. In this paper we explore the possibility of using machine learning dimensionality reduction methods to compress and represent an original audio data file in storage. Reduced audio file will then be compared with the original audio file. In this research we only focus on speech audio data.

LIST OF PUBLICATIONS

[1] Thambawita, V., Ellepola, N., Ragel, R., Elkaduwe, D., 2012. GPGPU: To Use or Not To Use?,PURSE.

[2] Mohajireen, M., Ellepola, C., Perera, M., Kahanda, I., Kanewala, U., 2011. Relational similarity model for suggesting friends in online social networks. https://doi.org/10.1109/ICIINFS.2011.6038090

Contents

LIST OF FIGURES

s

LIST OF TABLES

Chapter 1 Introduction

During the past decades, machine learning (ML) has spread towards different areas in science and technology bringing machines to interpret things in more human nature. Natural language processing (NLP), automatic speech recognition and computer vision are some of the uprising domains covered in ML. One of the utmost important senses possessed by humans to connect to the environment is the human's auditory system. Initial processing of sound features will be directed to superior olive lateral, lemniscus and inferior colliculus [1] in human brain to decode the incoming signal. Machines on the other hand will use digital signal processing to achieve a similar decoding. Depending on the storage, distribution of audio data, other user application requirements and numerous other parameters, diverse research trends have emerged to analyze audio signals. In this project we want to explore how machine learning can be used to achieve audio data compression.

1.1 Motivation

Introduction of compact disc (CD) in 1980s have open new dimensions of digital audio representation, including high fidelity and dynamic range. Conventional CDs are sampled at either 44.1kHz or 48kHz using pulse code modulation (PCM) with 16bit sample quality. Data rates of the CDs changed depending on the number of channels accommodated. To have the same quality audio in second generation audio and wireless applications, higher bandwidths requirements were often restricted due to the high data rates. Because the end consumers were expecting the same good quality audio, producers have to provide the same quality but reducing the data rates. Owing to this situation, many compressing algorithms which satisfy the audio quality and bandwidth have been introduced during the past decades [2]. Modern day bandwidth requirements in data networks are increasing rapidly around the world [3] .Bandwidth costs more money and congestion can lead to poor quality and services. Streaming of digital media makes 70% of internet traffic, and is projected to reach 80% by 2020, according to the Center of Internet Security (CIS) .Hence many digital applications target to achieve better compression techniques to transmit and store data. Image and video compression using machine learning methods have shown promising results compared to non-machine learning methods [4].

3.1 Statement of the problem

Storage requirement for modern day applications are increasing [3]. There are many applications which process real time speech and music data through networks, which directly correlate to the requirement of the amount of storage needed. Compression of data reduces the network resources need for transmission [5]. Not many machine learning approaches have been taken to compress audio signals to achieve low storage capacity.

3.2 Research Aims and Objectives

1.1.1 Aim

Main focus of the project is to explore, how machine learning techniques can be used to compress audio data.

3.2.1 Objectives

Humans can identify various sounds without putting much effort. Machines, on the other hand need constant training. This leads to a research area known as acoustic scene classification (ASC) [8]. Audible sounds are classified mainly in to three groups, speech, music and environmental [7]. Objective of this project is to focus on compressing speech audio data using machine learning dimensional reduction methods and do a comparison with the original audio data.

3.3 Scope

With the increased research in machine learning, many domains in science and technology are now trying to incorporate ML, AI into discovering new horizons. Given that the data compression is a very broad research area, the scope of this project is to explore the usage of machine learning feature engineering concepts to address the topic of lossy audio compression on speech audio data.

.

3.4 Background of the Study

1.1.1 Audio Signal

It is best to understand the basic physics around an audio signal. Digital audio processing is applied in many life products such as television, radio, cellular phones, video games and the list goes forever. With this wide spectrum, essentially there are three manipulations done on audio signals: acquisition, representation and storage [2]. Audio signals are basically vibrations of particles in the medium it exists. Unique properties of various audio signals that characterize their behavior are amplitude, frequency and time [3].

3.4.2 Sampling

Human hearing range is different to other animals. It is between 20 – 20K Hz ,due to this reason most applications involved in audio signal processing keep its sampling rate near to similar levels. Sampling is the method of converting analog signals in to digital signals. Sampling is needed because of the continuous nature of the audio waves. Computers and most other digital machines are not capable of handling continuous signals but discrete digital signals. Most audio signals are a combination of many individual audio waves. In order to analyse these signals, dimensions of each signal has to be separated for better understanding. We use Fast Fourier Transformation (FFT) to achieve this.

3.4.3 Fast Fourier Transformation

Generally an audio signal is a complex aggregated individual signals with different amplitudes and frequencies. When we capture a recording, what we see is the combination of characteristics of these single waves. Fig 1-1 depicts the amplitudes of a compound wave constructed using the python library librosa[1].

[1] https://librosa.org/doc/latest/index.html

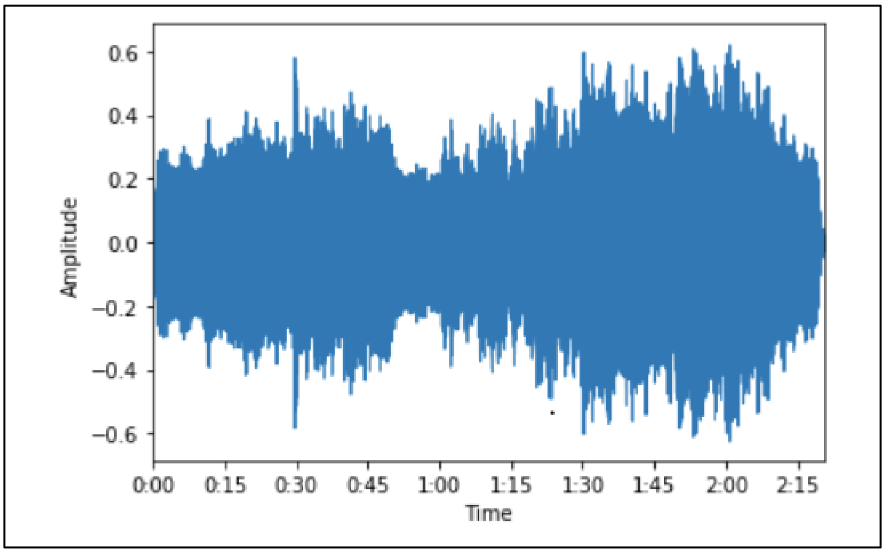


Figure 1-1 Amplitude-Time Plot

FFT is a mathematical model which helps to decompose the wave in to its constituent frequencies. The transformation of the Fig 1-1 is show in below frequency domain graph Fig 1-2

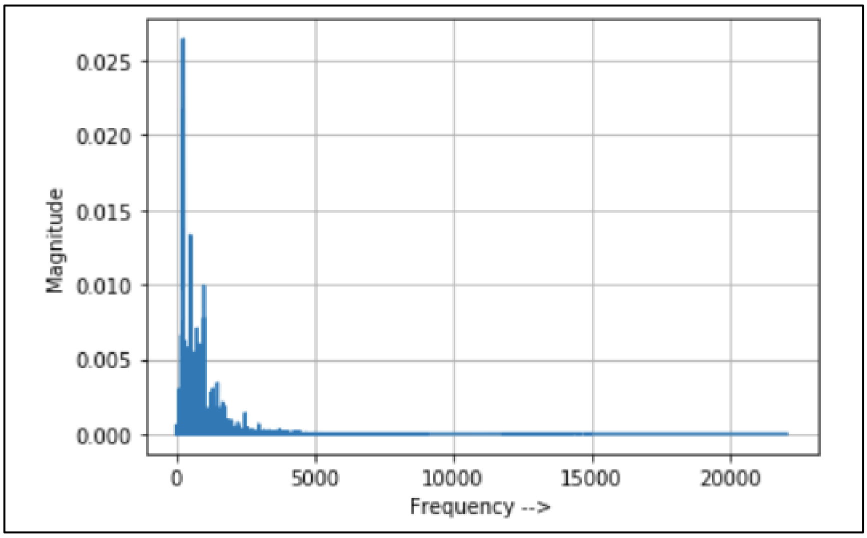


Figure 1-2 Frequency Domain Plot

The drawback of the frequency domain plots is that we lose the sense of time. As a solution for that spectrograms can visualize all three dimensions of a sound wave. Fig 1-3 shows the spectrogram of the earlier sound wave.

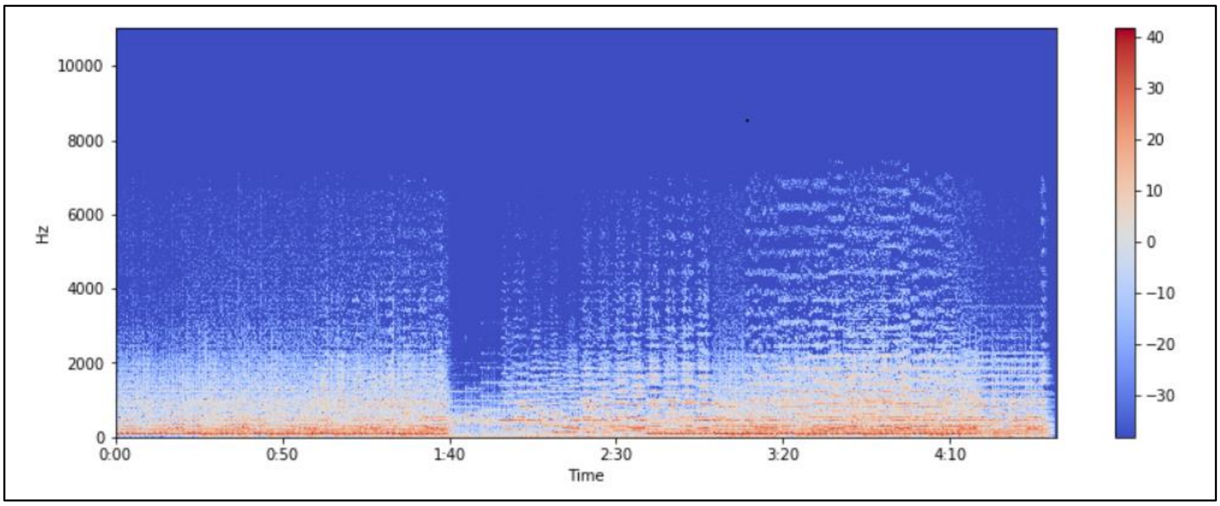


Figure 1-3 Spectrogram

3.4.4 PCA

Due to the increasing number of large datasets, many techniques are being developed to improve the interpretability of these types of data. One of the oldest and most widely utilized methods is the principal component analysis. PCA is a linear method that can be used to extract data from a high-dimensional space. It tries to minimize the number of non-essential parts that can be extracted.

3.4.5 Compression

Audio encoding or compression is used to obtain a compact version of the original signal in order to reduce storage capacity or efficient transmission. Requirement of compression need will vary from the target application using it. Compression algorithms are commonly divided into two stages, namely modeling and coding [9]. Modeling would deal with the redundant data in the audio file and coding would do the necessary encoding.

3.4.6 Lossy Compression

Lossy compression often does certain modification to frequencies that are unlikely to be detected by humans. These unwanted bits will be removed from the original signal and reconstructed original signal would not be the same [9]. There will be a small percentage of distortion. However a better compression rate is obtained by using lossy algorithms compared to lossless algorithms. In certain applications, it is not a must to have the original signal quality to be transferred. A good example of lossy compression is mp3. It is abundantly used in many applications due its compressed nature and the ability to be perceived without many distortion of the original signal.

3.4.7 Lossless Compression

Unlike in lossy compression lossless compression does not allow information loss. There are certain applications that can only use lossless compression due to the nature of its usage. The data must be accurately decompressed in order to get the original message. If a text file is compressed using lossy methods, certain texts can be loss in the original file which would totally give a different message. The only drawback is the amount of storage the file would take [9]. An example of lossless audio compression would be WAV format.

3.4.8 MFCC

Mel Frequency Characteristic Coefficients are commonly used for speech recognition tasks. They take into account the sensitivity of the human perception when converting the conventional frequency into Mel Scale. The feature extraction technique consists of applying the FT (Fourier Transformation), which is a stepwise process that involves taking the log of the frequency, then warping it on a Mel scale [19][20].

Chapter 2 Literature Review

3.5 A Literature Review

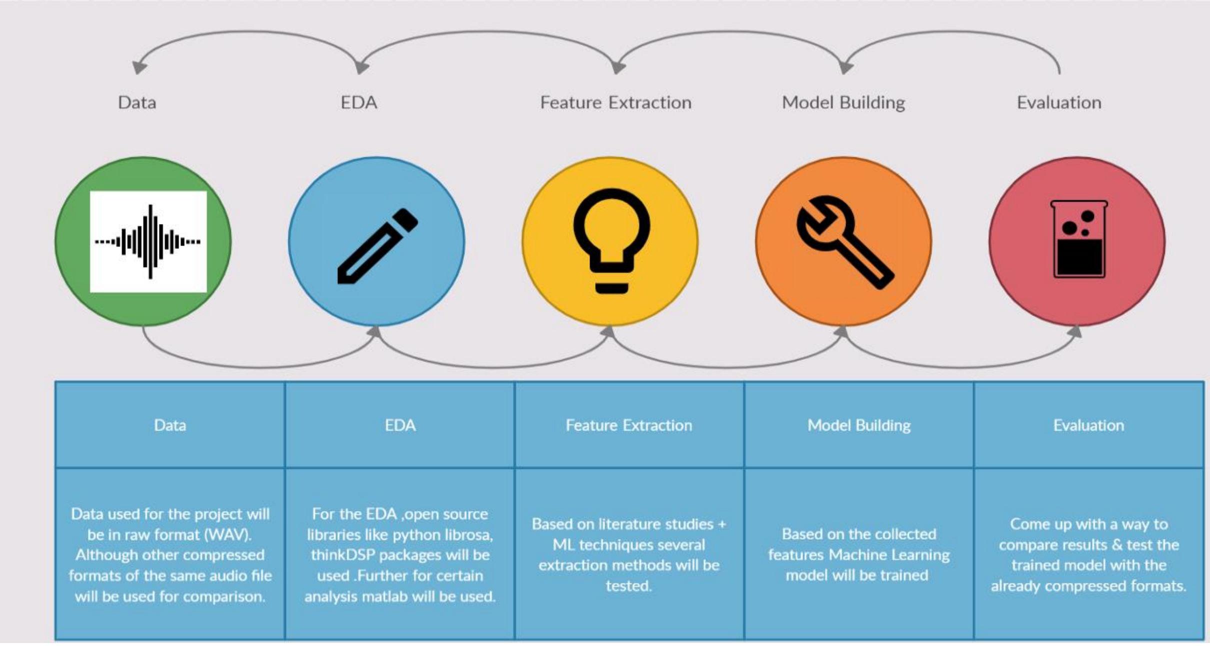
Compression is an experimental science where classical data compression can be divided in to two main parts. Redundant data in the data is described by the first stage and the encoding of the description take part in the second stage. The analogies of lossy and lossless compression depend on the difference between the original data and the encoded data [9] .Work on using auto encoders for sound modeling in [10] has shown PCA (principle component analysis) in contrast to image processing can be used in audio processing giving promising results for using dimensional reduction techniques for compression. Use of Recurrent Neural Networks (RNN) for compression based on the fact its capability to determine long term dependencies have been explored in [11].

Inspired by the image super-resolution algorithms, which uses ML to enhance low-resolution images to high-resolution images, feed forward neural network method has been introduced in study [12] to uplift the quality of a down sampled audio signal containing only a fraction of the original signal. This gives an indirect hint on the fact that neural networks can be used to enhance weak audio signals after down sampling. Artificial Intelligence (AI) has already outperformed traditional image compressions like JPEG and JPG by 100 to 250% when comparing file size as the base line (https://compression.ai). Reconstruction score for compressed file using frequency domain auto encoders have slightly lower value than the state of the art compressors, but the performance of the frequency domain encoders get highly affected when the size of the quantity is increased. Time domain auto encoders on the other hand have characterized lot of white noise in the reconstruction [11]. To classify Audio using machine learning is a one step closer to finding its feature. A novel illustration of audio data is introduced in [12], in order to analyze its behavior. Audio segmentation and classification is done base on the time granularity of the audio signal. The project aimed on defining an efficient audio classification algorithm based on support vector machine method. The semantic layers introduced in this project are audio frame, audio clip, audio slot and audio high level.

In most of the audio classifications, it is problematic to exactly extract the features from and audio signal given there is no best way to do the pre-processing [15]. Mel-Frequency Cepstral Coefficient (MFCC) uses Mel scale, which is a relationship between perceived tone frequency and the actual measured frequency, to represent short term power spectrum which is used in speech recognition research areas [19][20]. Feature engineering for audio signal with short-term Fourier transforms, MFCC, mean and variance from wavelet sub band energies and other ad-hoc features have added up to extraction of nearly 40 and 50 features. In order to extract the features, a frame size is used in the signal like in [14] which will include delta amount of features inside the frame. Different frames will have different features and as a result these frames were also taken as added features for the model increasing the total number of features.

The study in [16] investigates the possibility of reducing the feature dimensionality of a signal by using non-linear unsupervised techniques. Even though the main focus of this study is to synthesis new sounds, it gives a good background literature in dimensionality reduction to compress data. Study compares autoencoders (AE), deep autoencoders (DAE), recurrent autoencoders (LSTM-AE) and various autoencoders (VAE) coupled with principle component analysis (PCA). In contrast to image processing auto encoders do not outperform PCA in regeneration. Most favourable performance was indicated from VAE in audio synthesis. Mel scale is a scale that relates the actual measured frequency of a tone to what the human ear perceived. Mel Frequency Cepstral Coefficients (MFCCs) is a common efficient technique for signal processing. The method is not only used in audio signal processing but also in image data processing [17]. There are other libraries like LIBXTRACT that used MFCC for feature engineering of audio signals [18].

Chapter 3 Methodology



Data	EDA	Feature Extraction	Model Building	Evaluation
Data used for the project will be in raw format (WAV). Although other compressed formats of the same audio file will be used for comparison.	For the EDA ,open source libraries like python librosa, thinkDSP packages will be used .Further for certain analysis matlab will be used.	Based on literature studies + ML techniques several extraction methods will be tested.	Based on the collected features Machine Learning model will be trained	Come up with a way to compare results & test the trained model with the already compressed formats.

Figure 3-1 Methodology

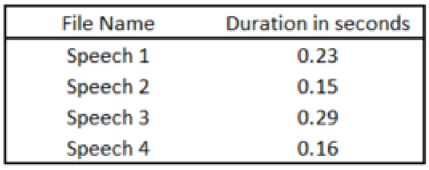
3.1 Data

Speech .WAV audio data files are used in order to do the analysis. These files are different with each other in their nature when comparing the texture.

3.2 EDA

In the exploratory analysis we want to see the basic audio statistics of the data files we are using. We use python librosa library for this particular arrangement. The important thing here to notice is that all the files have the same sampling duration of 0.00045 seconds. Let's look at the duration of the audio files. This is the product of the sample duration into the number of sample in each file. Wave form of each audio file can be seen in Fig 3-2.

Table 3-1 Audio Data Duration



File Name	Duration in seconds
Speech 1	0.23
Speech 2	0.15
Speech 3	0.29
Speech 4	0.16

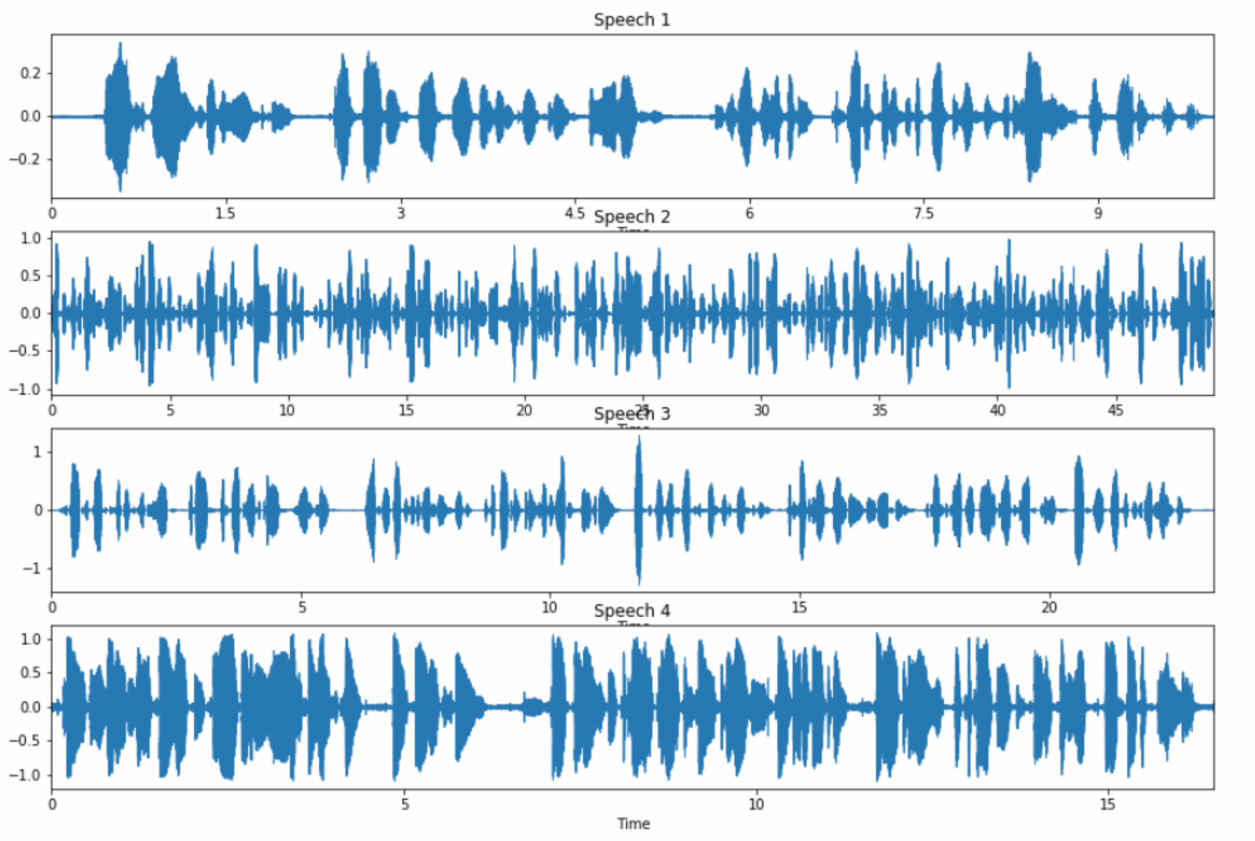


Figure 3-2 Audio Data Time Domain Behavior

For the purpose of this research, we will narrow down to analyze only speech audio data. Fig 3-2 shows some time-domain characteristic differences when we compare speech signal with others. Speech audio signals will always vary from one human to another. The pattern and timing of the notes will be unique for the person who is making the speech.

3.3 Feature Extraction

3.3.1 Mel Frequency Cepstral Coefficient

Mel frequency cepstral coefficients are derived using the librosa library. Librosa load the raw audio file as a floating point array which increases the size of the audio file in memory opposed to using libraries like sci.py.io. Sci.py.io load the audio file in 16 bit integer arrays which reduces the requirement of memory space but has no direct way of calculating MFCCs. For the selected audio file different numbers of coefficients were calculated from a range of 1 to 1024. It was seen that after a certain threshold, the size of the audio representation in memory did not change.

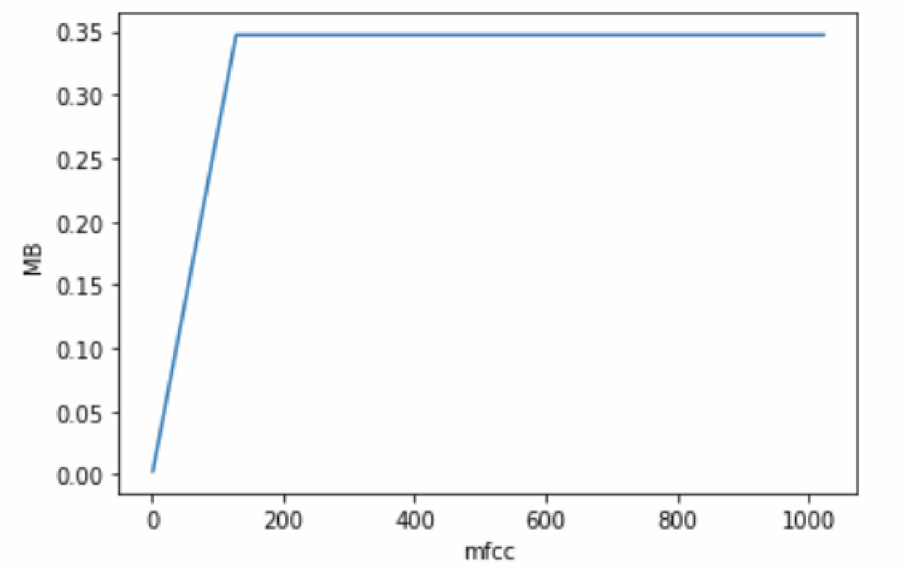


Figure 3-3 In memory MFCC vector size

3.3.2 PCA

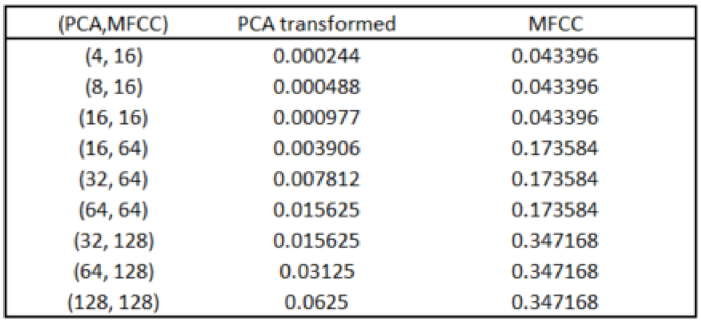
Extracted MFCCs are then subjected to PCA to see how further the audio representation in memory can be compressed. Reconstructed signal from PCA components which is the original mfcc signal has the same memory consumption.

Chapter 4 Analysis

In the first phase of the analysis, audio signal was converted to different feature vectors using mfcc. Different numbers of feature vectors represent the audio with different audio quality and memory representations. MFCC is not an encoding algorithm with its primary objective being to achieve high compression rates. . Before any changes, audio file is represented as a one dimensional array. After applying MFCC to the audio file it becomes multi-dimensional which let up apply PCA. PCA represent the MFCC vector data using components. Each component has its own "sub" section that contains some of the information the well represent the total audio file. For example, a low frequency component might contain all the information that we need to hear the low frequency bass notes. Converted raw audio file to mfcc matrix is not directly useable as audio. It needs to be reverse engineered to an audible format. This can be achieved using librosa.feature.inverse.mfcc_to_audio[2].

Considered a .wav audio speech with a size of .257 MB. Once the audio file is loaded to memory using librosa it becomes 1.388Mb. Linbrosa load audio files in to a floating point array by default, which increase the in memory size. Table 4-1 represents MFCC feature representation for different coefficients with different PCA components. Even though smaller coefficients show promising representations of the audio file using smaller amount of memory compared to the raw value, regenerated audio quality is tremendously degraded. Between the coefficients 64 and 128 audio quality, regenerated audio had better quality and comprehensibility.

Table 4-1 MFCC file compression



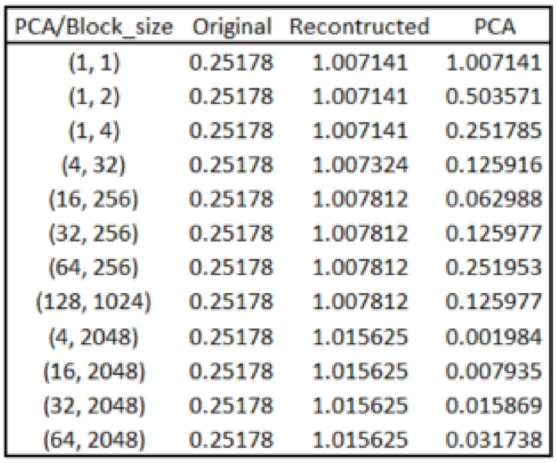
(PCA,MFCC)	PCA transformed	MFCC
(4, 16)	0.000244	0.043396
(8, 16)	0.000488	0.043396
(16, 16)	0.000977	0.043396
(16, 64)	0.003906	0.173584
(32, 64)	0.007812	0.173584
(64, 64)	0.015625	0.173584
(32, 128)	0.015625	0.347168
(64, 128)	0.03125	0.347168
(128, 128)	0.0625	0.347168

[2] https://librosa.org/doc/latest/index.html

Left most column of Table 4-1 indicated the number of PCA components and the number of MFCC coefficients. MFCC column indicates the memory allocation after raw audio is converted to MFCC vectors. PCA transformed column indicates the memory allocation of the MFCC vector after using PCA transformation with different components.

In the second phase of the analysis, we only used PCA in extracting important features. Instead of librosa we used scipy[3] python library which has the ability to load the audio file with minimum memory requirement. Unlike in MFCC, PCA cannot be directly applied to one dimensional data. To mitigate the issue, different block sizes were used with zero padding at the end. By using this method, audio file can be represented by different matrixes with different dimensions.

Table 4-2 PCA file compression



PCA/Block_size	Original	Recontructed	PCA
(1, 1)	0.25178	1.007141	1.007141
(1, 2)	0.25178	1.007141	0.503571
(1, 4)	0.25178	1.007141	0.251785
(4, 32)	0.25178	1.007324	0.125916
(16, 256)	0.25178	1.007812	0.062988
(32, 256)	0.25178	1.007812	0.125977
(64, 256)	0.25178	1.007812	0.251953
(128, 1024)	0.25178	1.007812	0.125977
(4, 2048)	0.25178	1.015625	0.001984
(16, 2048)	0.25178	1.015625	0.007935
(32, 2048)	0.25178	1.015625	0.015869
(64, 2048)	0.25178	1.015625	0.031738

In Table 4-1, PCA/Block_size indicated the number of PCA components and block sizes used for padding. In subsequent columns sizes of original audio file, reconstructed PCA file and PCA representation is shown.

[3] SciPy.org — SciPy.org

Chapter 5 Conclusion

From the results, we can see that there are multiple instances where in memory representation of the audio is well compressed with PCA. Even though there is a higher compression, audio quality is highly degraded. In Table 4-1, when the audio signal is represented with lesser MFCC and PCA components, audio qualities of these signals were highly degraded. We can see a similar behavior in Table 4-2. When a higher number of PCA components were used, a higher compression rates were achieved. On the other hand block size has a negative correlation with the compression. Larger the block size smaller the compression rate.

A real time drawback of this method is, whenever a conversion takes place, PCA or MFCC algorithms consumes considerable amount of space and time which poses a practical problem in streaming audio. Also librosa library load .wav files to a floating point array, which requires more memory for storage posing difficulties in achieving better compression.

For comprehension of speech audio, data should not necessarily have to be high quality. Speech audio storages can adopt dimensionality reduction method to store archived data to save memory. With the correct ratio of PCA components moderate quality speech audio data can be saved using dimensionality reduction methods.

REFERENCES

[1] C. J. Plack (2014) The Sense of Hearing, 2nd edn., Psychology Press: American Phycological Association.

[2] Andreas Spanias, Ted painter, Venkatraman Atti (2017) Audio Signal Processing and Coding, IEEE: A John Wiley & Sons, Inc., Publication.

[3] https://www.statista.com/statistics/638593/worldwide-data-center-storage-capacity-cloud-vs-traditional/

[4] Santurkar, S., Budden, D., Shavit, N., 2017. Generative Compression. arXiv:1703.01467 [cs].

[5] Pane, J.F., Joe, L., Arroyo Center, Force Development and Technology Program, Rand Corporation, United States, Army, 2005. Making better use of bandwidth: data compression and network management technologies. RAND, Santa Monica, CA.

[6] Sharma, G., Umapathy, K., Krishnan, S., 2020. Trends in audio signal feature extraction methods. Applied Acoustics 158, 107020. https://doi.org/10.1016/j.apacoust.2019.107020

[7] Gerhard, D., 2003. Audio Signal Classification: History and Current Techniques. A. temko, C. nadeu / pattern recognition 39 (2006) 682 – 694.

[8] Tzanetakis, G., Cook, P., 2002. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing 10, 293–302.

[9] Nowak, N., Zabierowski, W., 2011. (METHODS OF SOUND DATA COMPRESSION \226 COMPARISON OF DIFFERENT STANDARDS) 5.

[10] Roche, F., Hueber, T., Limier, S., Girin, L., 2019. Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models. arXiv:1806.04096 [cs, eess].

[11] Tatwawadi, K., n.d. DeepZip: Lossless Compression using Recurrent Networks 9.

[12] Kuleshov, V., Enam, S.Z., Ermon, S., 2017. AUDIO SUPER-RESOLUTION USING NEURAL NETS 5.

[13] Introduction [WWW Document], n.d. . Deep Autoencoders for Music Compression and Genre Classification. URL https://pgrady3.github.io/music-compression-web/

[14] Rong, F., 2016. Audio Classification Method Based on Machine Learning. pp. 81–84. https://doi.org/10.1109/ICITBS.2016.98

[15] Parker, C., 2010. An Empirical Study of Feature Extraction Methods for Audio Classification, in: 2010 20th International Conference on Pattern Recognition. Presented at the 2010 20th International Conference on Pattern Recognition (ICPR), IEEE, Istanbul, Turkey, pp. 4593–4596. https://doi.org/10.1109/ICPR.2010.1111

[16] Vallet, G.T., Shore, D.I., Schutz, M., 2014. Exploring the role of the amplitude envelope in duration estimation. Perception 43, 616–630.

[17] Gupta, S., Jaafar, J., Wan Ahmad, W.F., Bansal, A., 2013. Feature Extraction Using Mfcc. Signal & Image Processing : An International Journal 4, 101–108. https://doi.org/10.5121/sipij.2013.4408

[18] Bullock, J., 2007. LIBXTRACT: A LIGHTWEIGHT LIBRARY FOR AUDIO FEATURE EXTRACTION.

[19] Logan, B., 2000. Mel Frequency Cepstral Coefficients for Music Modeling, in: In International Symposium on Music Information Retrieval.

[20] Chauhan, P.M., Desai, N.P., 2014. Mel Frequency Cepstral Coefficients (MFCC) based speaker identification in noisy environment using wiener filter, in: 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE). Presented at the 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), IEEE, Coimbatore, India, pp. 1–5. https://doi.org/10.1109/ICGCCEE.2014.6921394

[21] Bachu, R.G., Kopparthi, S., Adapa, B., Barkana, B.D., 2010. Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy, in: Elleithy, K. (Ed.), Advanced Techniques in Computing Sciences and Software Engineering. Springer Netherlands, Dordrecht, pp. 279–282.