Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music

Sigurdur Sigurdsson, Kaare Brandt Petersen and Tue Lehn-Schiøler
Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads - Building 321
DK-2800 Kgs. Lyngby - Denmark
{siggi,kbp}@imm.dtu.dk

Abstract

In large MP3 databases, files are typically generated with different parameter settings, i.e., bit rate and sampling rates. This is of concern for MIR applications, as encoding difference can potentially confound meta-data estimation and similarity evaluation. In this paper we will discuss the influence of MP3 coding for the Mel frequency cepstral coeficients (MFCCs). The main result is that the widely used subset of the MFCCs is robust at bit rates equal or higher than 128 kbits/s, for the implementations we have investigated. However, for lower bit rates, e.g., 64 kbits/s, the implementation of the Mel filter bank becomes an issue.

Keywords: Mel frequency cepstral coefficients, MFCC, robustness, MP3.

1. Introduction

The use of Mel frequency cepstral coefficients (MFCCs) for music information retrieval has become standard since the seminal paper [4] in 1997. But only little effort has been put into investigating the applicability of the MFCC's as features for music, with [6] as a rare exception. In this paper we investigate how MP3 encoding of music files is influencing the signal information content of the MFCC's.

2. Mel Frequency Cepstral Coefficients

We will use the Intelligent sound implementation (ISP) to explain the computation of MFCCs. First the music signal is divided into short time windows, where we compute the discrete Fourier transform (DFT) of each time window for the discrete-time signal $x(n)$ with length N, given by

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n)\exp(-j2\pi kn/N) \qquad (1)$$

for $k = 0, 1, \ldots, N-1$, where k corresponds to the frequency $f(k) = kf_s/N$, f_s is the sampling frequency in Hertz and $w(n)$ is a time-window. Here, we chose the popular Hamming window as a time window, given by $w(n) = 0.54 - 0.46\cos(\pi n/N)$, due to computational simplicity.

The magnitude spectrum $|X(k)|$ is now scaled in both frequency and magnitude. First, the frequency is scaled logarithmically using the so-called Mel filter bank $H(k,m)$ and then the logarithm is taken, giving

$$X'(m) = \ln\left(\sum_{k=0}^{N-1} |X(k)| \cdot H(k,m)\right) \qquad (2)$$

for $m = 1, 2, \ldots, M$, where M is the number of filter banks and $M \ll N$. The Mel filter bank is a collection of triangular filters defined by the center frequencies $f_c(m)$, written as

$$H(k,m) = \begin{cases} 0 & \text{for} \quad f(k) < f_c(m-1) \\ \frac{f(k)-f_c(m-1)}{f_c(m)-f_c(m-1)} & \text{for} \quad f_c(m-1) \le f(k) < f_c(m) \\ \frac{f(k)-f_c(m+1)}{f_c(m)-f_c(m+1)} & \text{for} \quad f_c(m) \le f(k) < f_c(m+1) \\ 0 & \text{for} \quad f(k) \ge f_c(m+1). \end{cases} \qquad (3)$$

The center frequencies of the filter bank are computed by approximating the Mel scale with

$$\phi = 2595\log_{10}(\tfrac{f}{700} + 1), \qquad (4)$$

which is a common approximation. Note that this equation is non-linear for all frequencies. Then a fixed frequency resolution in the Mel scale is computed, corresponding to a logarithmic scaling of the repetition frequency, using $\Delta\phi = (\phi_{\max} - \phi_{\min})/(M+1)$ where $\phi_{\max}$ is the highest frequency of the filter bank on the Mel scale, computed from $f_{\max}$ using equation (4), $\phi_{\min}$ is the lowest frequency in Mel scale, having a corresponding $f_{\min}$, and M is the number of filter banks. The values for the ISP implementation is $f_{\max} = 11.025$ kHz, $f_{\min} = 0$ Hz, and $M = 30$. The center frequencies on the Mel scale are given by $\phi_c(m) = m \cdot \Delta\phi$ for $m = 1, 2, \ldots, M$. To obtain the center frequencies in Hertz, we apply the inverse of equation (4), given by $f_c(m) = 700(10^{\phi_c(m)/2595} - 1)$, which are inserted into equation (3) to give the Mel filter bank. Finally, the MFCCs are obtained by computing the DCT of $X'(m)$ using

$$c(l) = \sum_{m=1}^{M} X'(m)\cos(l\tfrac{\pi}{M}(m - \tfrac{1}{2})) \qquad (5)$$

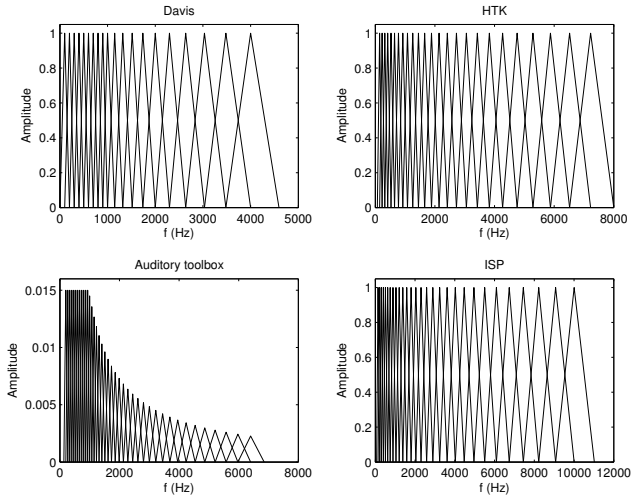

Figure 1. The figure shows 4 different implementations of the Mel filter bank. Note the different scaling of the frequency axes in the plots.

for $l = 1, 2, \ldots, M$, where $c(l)$ is the lth MFCC.

In this paper we will focus on 4 different implementations of the MFCCs; the algorithm due to Davis [2], the Auditory toolbox [8], the hidden Markov model toolkit (HTK) [9], and the ISP implementation given above. The implementations have different Mel filter banks, shown in Figure 1. Note the different characteristics of the filter banks. Davis' implementation has linear spacing up to 1 kHz and then logarithmic spacing, where the filter amplitude is constant. HTK has logarithmic spacing and constant amplitude. The Auditory toolbox suppresses frequencies below approximately 133 Hz, has linear spacing up to 1 kHz and then logarithmic spacing, where the energy in all filters is fixed to unity. The ISP implementation is similar to HTK, using the same definition of the Mel filter bank with different number of filters and filter center frequencies. Also, the ISP implementation does not use liftering.

3. MP3 Encoding

The compression used for MP3 files is based on perceptual encoding, where the goal is to apply efficient coding while, at the same time, obtaining a perceptually good coding of the signal. The main building blocks of an MP3 encoder are: An analysis filter bank which decomposes the signal into subsampled spectral bands, a perceptual model which controls the quantization and coding scheme for the decomposed signal, and finally a bitstream coding. It is the perceptual model that determines the quality of the signal, as compression is obtained by adapting the amount of quantization noise, based on the amplitude and frequency content of the signal. Despite of this advanced scheme for coding the music signals, some artifacts are encountered. The most common is pre-echo where a noise signal is observed before the music signal that actually causes the noise. This is due to the temporal resolution of the decoder, given by the synthesis window length, where the quantization error is distributed over the full window. Thus, a sudden signal attack increases the quantization error, which includes the music signal before the attack. Another artifact is the loss of signal bandwidth when the encoder runs out of bits for a given quality of the signal. For an introduction to MP3 coding, see e.g. [1].

In this paper we have used the LAME 3.96.1 encoder, which is very popular and often acclaimed being the best encoder for bit rates at 128 kbit/s or higher. We have used the popular Madplay 0.15.0 (beta) for decoding the MP3 files. The choice of encoder/decoder were based on their popularity and that they are freely available. The encoder specifications for the experiments were; stereo mode, variable bit rates at 64, 128 and 320 kbit/s, sampling rate of 44.1 and 22.05 kHz. The most commonly used bit rate is 128 kbit/s, where both good compression and reasonable sound quality may be obtained. The 64 and 320 kbit/s are used to show results at very low and good quality. The reason to use a lower sampling rate than 44.1 kHz is to show improvement in quality at low bit rate.

4. Evaluating Robustness with Correlation

In order to evaluate the effect of different MFCC approaches and different MP3 encodings, we need a measure of difference. We have chosen the so-called Pearson's correlation coefficient to compare MFCCs. By using this simple scheme, we avoid selecting a classifier for a specific MIR task and choosing a temporal coding scheme for the MFCCs, e.g. Gaussian mixture model.

The Pearson's correlation coefficient r_{xy} for two variables x and y, is a measure of the correlation between them given a linear model and Gaussian noise [3]. Here we will use the squared correlation r_{xy}^2, which indicates the percentage of variation in the data that can be explained with the linear model. For $r_{xy}^2 = 1$ the relation is exact, and as r_{xy}^2 becomes smaller, the relation becomes weaker.

It is well known that Pearson's correlation coefficient should be used as a measure of regression rather than correlation, and in the case of the MFCCs we are doing exactly that: Estimating the noise variance under the linear assumption. To be sure that the assumption about the linear relation and Gaussian noise is not too restrictive, we conduct a Kolmogorov-Smirnov test (KS-test) on the noise residuals, see e.g. [7] for details.

5. Experiments

All experiments were conducted using a data set of 46 songs from 46 different rock and pop artists. WAV files were generated from compact disks using CDex 1.51. MP3 files were generated from the WAV files using the LAME encoder. To avoid noise due to time difference between the WAV and

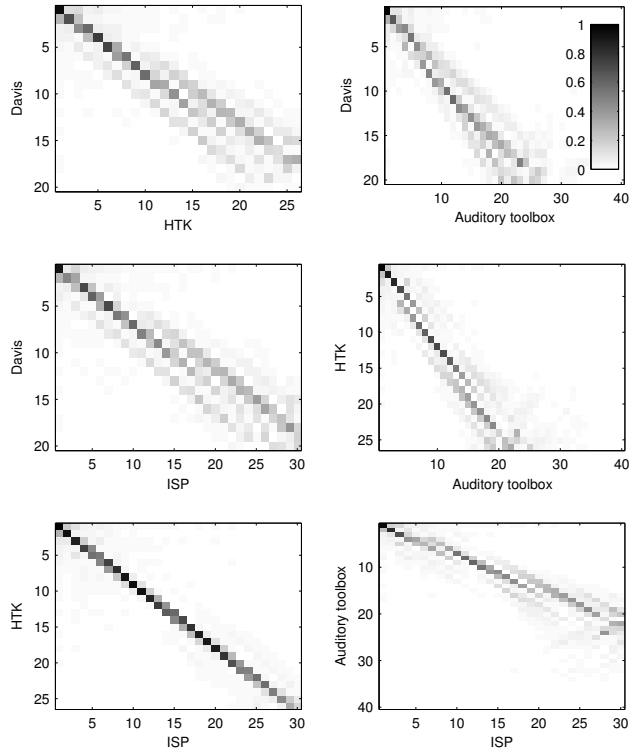

Figure 2. The figure shows the squared Pearson's correlation coefficient (r^2) between single MFCCs for the 4 selected implementations, where the values on the axes indicates MFCC number. Note that the images are different in size, due to different number of MFCCs for each implementation.

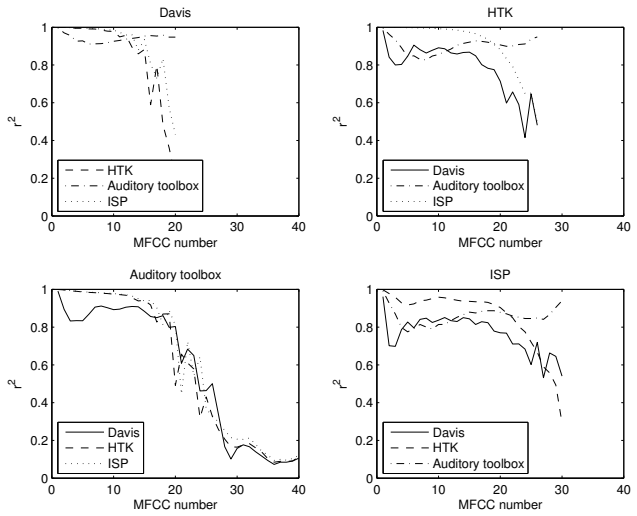

Figure 3. The figure shows the squared Pearson's correlation coefficient (r^2) where each MFCC of one implementation (title of plot) is conditioned on all the MFCCs for the other implementations (legend of plot).

MP3 files, the signals were aligned in time prior to MFCC computation. Various window sizes are suggested to compute MFCCs, ranging from 5-100 ms and often around 20 ms, with overlap 30-50 %. On the basis of this, the MFCCs for the songs were computed using a fixed window size of 20 ms with 50 % overlap. As the music files contain stereo music, we generate a single channel signal by averaging over both channels prior to MFCC computation.

5.1. MFCC Implementations

The implementation comparison used only WAV files for evaluation. MFCCs were computed for each song for all 4 implementations. The squared Pearson's correlation coefficient r^2 was computed between all MFCCs for all methods and for each song. The result shown in figure 2 is the average over all songs. From the figure we observe that approximately the first 15 MFCCs are quite correlated between implementations. This varies somewhat between implementations, e.g. the HTK and ISP are very correlated as they are based on the same implementation of the Mel filter bank with different specifications. In practical applications only the first 5-15 MFCCs are in general used, which could explain similar performances using different implementations. For instance, investigations of different MFCC implementation schemes for speaker verification have shown very similar results [5]. The MFCCs above approximately 15, have lower r^2 and become more diffused, as information spreads out to neighboring MFCCs.

It should be noted that the assumption of the relation between MFCCs from different implementations are modeled linearly with Gaussian noise is highly unlikely. This is due to the fact that each MFCC implementation is a highly nonlinear process. On the other hand, high r^2 means that much of relation may be explained with the linear model, while the noise is not Gaussian distributed. This was confirmed with the KS-test.

The results shown in figure 2 may be confirmed by computing the r^2 between a single MFCC conditioned on all MFCCs from other implementations. Figure 3 shows the results for all implementations. The figure shows that the r^2 is approximately 0.8 or higher for MFCCs up to 15 for all implementations. Again it should be noted that the KS-test rejects in many cases the hypothesis of a linear model with Gaussian noise, although the r^2 is high.

5.2. MFCC Robustness to MP3 Coding

The influence of MP3 coding was evaluated by computing the MFCCs for WAV and MP3 files at different bit rates and sample rates, and then evaluating the squared Pearson's correlation coefficient r^2 between the WAV generated MFCCs and the MP3 generated MFCCs. The KS-test accepted in almost all cases the hypothesis of a linear relation with Gaussian noise. The results are shown in figure 4. At a fixed sampling rate of 44.1 kHz and bit rate of 320 kbits/s the r^2 between WAV and MP3 MFCCs are approximately 1, indicating little or no loss. At 128 kbits/s, r^2 drops similarly

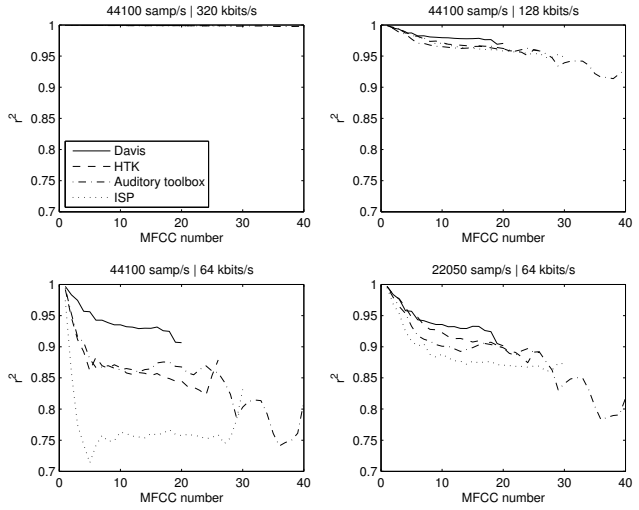

Figure 4. The squared Pearson's correlation coefficient (r^2) as a function of MFCC number for the 4 MFCC implementations, using different sampling rate and bit rate.

for all implementations, but is higher than approximately 0.95 for the first 15 MFCCs. Interestingly, r^2 is dependent on the MFCC number, showing that higher MFCCs have lower sample correlation, indicating that they are less robust to MP3 encoding of music. At 64 kbits/s the sample correlation has decreased significantly and is now dependent on implementations. The largest single factor is the highest frequency included in the Mel filter bank. The most robust implementation is Davis' with the highest frequency 4.6 kHz, while the least robust is the ISP implementation with highest frequency 11.025 kHz. The HTK and Auditory toolbox implementations are in between the other two, having the highest included frequency of 8 kHz and 6.9 kHz.

Figure 4 shows also that it is possible to improve the robustness by reducing the sample rate from 44.1 kHz to 22.05 kHz. This is due to the MP3 encoding, where higher frequencies are more expensive to code and deviate more from the original. Thus, by disregarding higher frequencies, both by removing higher frequencies in the Mel filter bank implementation and reducing the sampling rate, more robust MFCCs are obtained.

6. Conclusion

In this paper we have evaluated the robustness of MFCCs with the squared Pearson's correlation coefficient. The results show that the different MFCC implementations are very correlated for approximately the first 15 MFCCs. This supports experiments for speaker verification [5], showing similar performance for different MFCC implementations and settings.

MFCCs were shown to be very robust at bit rates of 320 and 128 kbit/s for all implementations at a fixed sampling rate of 44.1 kHz. At 64 kbits/s, using the same sampling rate, the implementations are less robust and the robustness is dependent on implementation. The robustness decayed more rapidly for implementations that included higher frequencies in the Mel filter bank. Also, we showed that the robustness at lower bit rates, e.g. 64 kbits/s, may be improved by reducing the sampling rate, especially for implementations that included higher frequencies in the Mel filter bank. Finally, we illustrated that higher order MFCCs are less robust than lower order for MP3 encoding.

This paper shows that MFCC features are very robust to MP3 encoding and thus applicable in MIR tasks. However, the MFCC implementation should take into account the encoding distortion in MP3 files at low bit rates.

7. Acknowledgements

This work is supported by the Danish Technical Research Council, through the framework project 'Intelligent Sound', www.intelligentsound.org (STVF No. 26-04-0092). We thank Anders Meng, Jan Larsen and Lars Kai Hansen for discussions and comments.

References

[1] Karlheinz Brandenburg. MP3 and AAC explained. In *AES 17th International Conference on High Quality Audio Coding*, 1999.

[2] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

[3] Allen L. Edwards. *An introduction to linear regression and correlation*. W. H. Freeman and Company, 1976.

[4] J. Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, volume 3229, pages 138–147, 1997.

[5] Todor Ganchev, Nikos Fakotakis, and George Kokkinakis. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proceedings of the 10th International Conference on Speech and Computer (SPECOM 2005)*, volume 1, pages 191–194, 2005.

[6] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of International Symposium on Music Information Retrieval (ISMIR)*, 2000.

[7] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, Chapter 14, pp. 623–628, 2nd edition, 2002.

[8] Malcolm Slaney. Auditory toolbox, version 2. Technical Report #1998-010, Interval Research Corporation, 1998.

[9] Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. The HTK book (for version 3.2). Cambridge University Engineering Department, December 2002.