Content-Based Methods for the Management of Digital Music

David Pye

AT&T Laboratories Cambridge,
24a Trumpington Street,
Cambridge, CB2 1QA, UK

ABSTRACT

The literature on content-based music retrieval has largely finessed acoustic issues by using MIDI format music. This paper however considers content-based classification and retrieval of a typical (MPEG layer III) digital music archive. Two statistical techniques are investigated and appraised. Gaussian Mixture Modelling performs well with an accuracy of **92%** on a music classification task. A Tree-based Vector Quantization scheme offers marginally worse performance in a faster, scalable framework. Good results are also reported for music retrieval-by-similarity using the same techniques. Mel-Frequency Cepstral Coefficients parameterize the audio well, though are slow to compute from the compressed domain. A new parameterization (MP3CEP), based on a partial decompression of MPEG layer III audio, is therefore proposed to facilitate music processing at user-interactive speeds. Overall, the techniques described herein provide useful tools in the management of a typical digital music library.

1. INTRODUCTION

Personal archives of digital music have become increasingly common over recent years, with the popularity of MPEG Layer III (MP3) [5][6] and other proprietary formats. With this trend set to accelerate, tools will be needed to help navigate and manage these archives. This paper looks at two such tools for performing content-based classification and retrieval of music. In doing so, it contributes to the relatively sparse literature on content-based retrieval of digital sampled music.

Music genre information can prove useful in helping users focus on music of their particular preference. Whilst many compression formats come complete with suitable genre annotations, these are frequently inappropriate, inaccurate, over-generalized or simply omitted. These text annotations can be supplemented or replaced with the user's own carefully chosen categories that encapsulate exactly the set of music desired. Of potentially more use is content-based music retrieval. The ability to retrieve music acoustically similar to a song or a short extract is an invaluable tool to find new songs matching the user's taste in large (possibly web-wide) archives. Alternative techniques such as analyzing buying patterns require considerable data and are only applicable some time after release. The same basic technology has other applications. For instance, storing song likes and dislikes for a user allows prediction of new songs into one of these two sets. This allows filtering of potential additions to an archive, or for instance, digital radio broadcasting personalised to musical taste.

Two core techniques are investigated for music processing in this paper. Firstly, Gaussian Mixture Modelling (GMM) which models the distribution of the acoustics and has proved itself for audio classification. Secondly, a Tree-Based Vector Quantization (TreeQ) method [4] is considered that takes a discriminative approach. Each of these techniques requires parameterization of the audio examples into feature vectors. The baseline results use Mel-Frequency Cepstral Coefficients, which are popular in the speech community. An alternative method is introduced that avoids fully decompressing the audio and parameterising the resulting waveform. In this scheme, MP3CEP, cepstral parameters are rapidly derived from the partially decompressed MPEG layer III audio. This paper discusses these techniques and parameterizations in more detail. Subsequently, their use in classification and retrieval experiments is described and results are reported.

2. TECHNIQUES

2.1 Gaussian Mixture Modelling (GMM)

Gaussian Mixture Modelling has been successfully used for a variety of audio classification tasks such as speaker recognition [8]. In this paper, the technique is used to model a song or musical genre as a probability density function (PDF), using a weighted combination of Gaussian component PDFs (mixtures). As the feature vectors in this work have reasonably uncorrelated components, computationally convenient diagonal covariance matrices can be used. The GMM parameters are estimated to maximize the probability of the training observation sequence. Although there is no known way to solve this in a closed form, the likelihood can be locally maximized using the Baum-Welch (Expectation-Maximization) method. This iterative process continues until the parameter values converge at hopefully an optimal solution. A GMM is used to determine the probability a test feature vector belongs to that model. By iterating over all observations, a score for a whole song can be produced. Rather than simply summing this score directly, however, a form of frame normalization is used here to limit the contribution of any single frame.

2.2 Tree-based Vector Quantization (TreeQ)

The second technique discriminately trains a Vector Quantizer rather than trying to model the acoustics directly [4]. Results have been reported using this technique for speaker identification as well as retrieval of short clips of audio and music [3]. The training data is parameterized first into feature vectors. Each training example is associated with a class such as artist or genre. A quantization tree is grown that automatically partitions the feature space into regions that have maximally different class populations. Once the tree has been constructed, it can be used to form a histogram template for a song or genre. All feature vectors

fall through the tree into the leaf cells. The relative quantity of the samples in each cell forms a histogram template. These templates are a compact representation of the acoustics of a song and the distance between any two provides an estimate of acoustic similarity. Although many techniques are available to compare two vectors, the cosine distance measure [10] is used throughout this paper.

3. PARAMETERIZATION SCHEMES

The first stage of music processing requires the MP3 compressed digital music to be parameterized into suitable feature vectors. These vectors should retain salient information whilst discarding unnecessary acoustic detail. Mel-Frequency Cepstral Coefficients (MFCCs) [2][7] were tested first. This process requires the audio to be completely decompressed. The resulting digital waveform is returned to a spectral domain, albeit a different one, during parameterization. In the second scheme, MP3CEP, this partial redundancy is eliminated in using parameters derived from MP3 subband data directly.

3.1 MFCC

MFCCs are popular in the speech processing community and give good discriminative performance with reasonable noise robustness. Although designed for speech rather than music, MFCCs are nevertheless a good starting point. In our scheme, windows of audio of 25msec are considered every 10msec. A discrete Fourier transform converts each window to a spectrum. The spectral coefficients are accumulated into bins on the mel-scale – a non-linearity emphasizing the perceptually important lower- and mid-frequency regions. The log bin values are then transformed using the discrete cosine transform into twelve reasonable uncorrelated cepstral coefficients. A log energy term is typically appended. A thirteen component feature vector is produced 100 times a second.

3.2 MP3CEP

This process starts as normal MP3 decompression including bit-stream parsing and frequency sample de-quantization. As subband data becomes available, this is used as a source for parameterization rather than to synthesize actual samples with the synthesis filter. Each MP3 frame (corresponding to 1152 PCM samples) consists of two granules. With a standard 44.1 kHz song, a granule occurs approximately every 13ms, of the same order as the 10ms MFCC frame rate. To replicate the MFCC window size (25ms), feature vectors are produced using subband data across two adjacent granules (13*2 = 26 ms).

Two granules contain 36 subband samples, where each subband sample is a vector of 32 (equally spaced) frequency band amplitudes. A subband magnitude vector is produced by summing the magnitudes of these thirty-six subband sample vectors. The resulting 32 component vector is reduced to 20 components by a mel-like warping. The lower frequency components are unaltered while the higher ones are combined, in increasingly larger numbers. After taking the log of each component value, twelve cepstral coefficients are produced using a discrete cosine transform. Finally, summing the magnitude across all subbands estimates an energy term, the log of which is appended. A thirteen component feature vector results 76.6 times a second.

3.3 Parameterization Times

The MFCC scheme is intended for good performance, whilst the MP3CEP scheme is designed primarily for speed. Figure one shows the time taken for a typical MP3 song to be parameterized for both schemes. The actual values are less important than the ratio between them, since various optimizations may apply to both strategies. It can be seen that MP3CEP parameterization is approximately six times faster. It is potentially less accurate however and is format specific. Whether these drawbacks outweigh the speed advantage is an application specific matter.

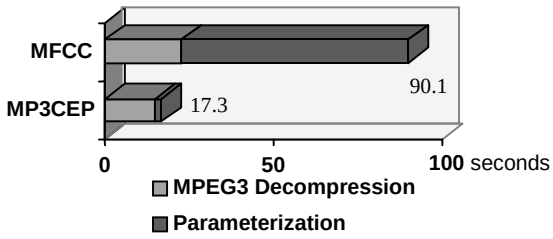

Figure 1. A comparison of processing time (seconds), to decompress and parameterize into MFCC & MP3CEP feature vectors, for a typical MP3 song (3m. 45s.)

4. EXPERIMENTS

4.1 Music Genre Classification

The two techniques and parameterizations were first tested on the task of music genre classification. A typical personal corpus of MP3 music was used to provide music for these experiments. For classification, six music genres have been determined that contain sufficient material for these experiments. These genres are *blues, easy listening, classical, opera, dance (techno)* and *indie rock*. The music from each category was split evenly into test and training data. The final test set across all genres consists of 175 songs. In order to identify music from out of these set categories, additional music was used to form a generic music category or 'garbage model'.

4.1.1 Gaussian Mixture Modelling

To evaluate the GMM approach for music classification, a GMM is estimated from the training material for each genre (including a generic one). Each test song in turn is scored with all genre models, with the song being classified in accordance with the best scoring GMM. A number of experiments were run to evaluate the performance of various compositions of MFCC feature vectors. Figures for percentage classification accuracy are shown in table 1 for 4, 8, 32 and 64 mixture component sizes respectively. In the first line, the basic cepstral components are used without an energy term, which is added in the second. The third and fourth lines are similar but with the deltas of each

component appended (doubling the vector size). The table shows that the energy term improves classification accuracy quite significantly on this highly matched task. Furthermore, using deltas also provides consistent gains in accuracy. The best configuration of using 12 MFCC components along with energy and second derivatives forms the last line of table one. This configuration is used for all subsequent experiments, both MFCC and MP3CEP. The final line also shows that good performance can be achieved using as little as four mixture components, though performance increases steadily to peak at 32. With the best system, **92**% of 175 test songs are correctly classified.

GMM Systems	Number of mixture components			
	4 mix	8 mix	32 mix	64 mix
MFCC	**79.4**	**81.1**	**84.0**	-
MFCC+*e*	**86.2**	**88.0**	**90.2**	**89.7**
MFCC+δ	**81.7**	**84.0**	**87.4**	-
MFCC+*e*+δ	**88.0**	**90.2**	**92.0**	**91.4**

Table 1. Percentage of songs accurately classified using various MFCC **GMM** systems. Comparable results with energy (e) and delta parameters (δ) are included.

4.1.2 TreeQ System

To use decision tree vector quantization, all the training data for a particular music genre forms a single class. The seven classes (including generic) are used to construct a quantization tree that attempts to put samples from different training classes into different leaf nodes. The leaf node contents are used to form a histogram template for each genre and test file. Each test file is classified by using the cosine distance metric to determine the genre with the closest matching histogram template. Trees of size 100, 500 and 1000 leaves have been compared. The percentage accuracy figures using this technique are in table 2. The first line shows the results using an MFCC parameterization (including energy and deltas). As decision tree construction is linear in terms of the number of dimensions, better modelling of time variation effects is possible by forming super-vectors by concatenating local context. The second line shows classification results therefore using a sliding window of five concatenated vectors. The results indicate a tree generated using context and to a size of 500 leaves is the most effective classifier – producing a classification accuracy of **89.7**%.

TreeQ Systems	Number of leaf nodes		
	100	500	1000
MFCC	**86.3**	**88.6**	**88.6**
MFCC *(ctx 5)*	**86.8**	**89.7**	**89.1**

Table 2. Percentage of songs accurately classified using **TreeQ** systems. Firstly, using MFCC (+e+δ) vectors and subsequently with five frame super-vectors.

4.2 Comparison of GMM and TreeQ Techniques

The first column of Table 3 summarizes the best results from the tables above for the GMM and TreeQ systems respectively. The best TreeQ figure of **89.7**% is marginally lower than the GMM system best of **92%.** One issue of using GMMs is however the computational cost of estimating and testing them. Although schemes have been proposed to reduce this problem such as using pooling, it remains a serious drawback. The TreeQ approach is however fast, especially in quantizing using the tree. Furthermore, the process of finding similar templates lends itself to highly efficient indexing structures such as M-trees [1]. Overall, for applications where speed and scalability are an issue, it would be the method of choice.

System	Feature vectors	
	MFCC	MP3CEP
GMM 8	**90.3**	**86.9**
GMM 32	**92.0**	**90.9**
TreeQ 500	**89.7**	**85.1**

Table 3. Percentage of songs accurately classified using **GMM 8** & **32** mixture systems and the best **TreeQ** system, using both MFCC and MP3CEP feature vectors.

4.3 Comparison of Parameterizations

The second column of Table 3 are the results of replicating the experiments with MP3CEP feature vectors rather than MFCC. The trend of GMM marginally outperforming the TreeQ system is again evident. An interesting result is that the classification accuracy using MP3CEP parameters, for the best performing GMM32 system, is **90.9**% just **1.2**% behind the MFCC figure. This is a surprisingly good result particularly as over **20**% less feature vectors are used. For the TreeQ system, the performance drop from MFCC to MP3CEP increases to **5**%. The performance degradation, however, may be a small price to pay to allow music processing at interactive speeds. The major drawback of MP3CEP is its MP3 specificity. Whilst similar techniques are possible for alternative audio compression schemes, such as MPEG2-AAC format, compatibility is unlikely and deployment in a mixed format archive is inappropriate.

4.4 Music Retrieval

In a further experiment, the two techniques have been compared on a music retrieval-by-similarity task. The corpus for this task is drawn from the same source as the classification experiment. Ten songs from each “album” are split into two collections: a random five into a test set and the remaining five into a set of retrievable candidate songs. The retrievable candidate set was supplemented by many other songs to hide the relevant ones, making the task more difficult. The final test set consists of one hundred and fifty songs.

4.4.1 Gaussian Mixture Modelling

In order to retrieve music using GMMs, a 16 component GMM is estimated for each possible retrieval candidate. For each test song, it is parameterized and scored by all candidate GMMs. The output is a list of all retrievable songs in order of acoustic similarity.

4.4.2 TreeQ System

In constructing a decision tree, a class is created for each artist. As before, a 500-leaf tree is constructed to discriminate between classes. The tree is used to generate a histogram template for all test and candidate songs. A test song is compared using the cosine metric to each candidate in turn, again producing a list of candidates in order of acoustic similarity.

4.4.3 Retrieval Performance

Due to the impracticality of determining genuine relevance assessments, two simplistic relevance assumptions have been used. Firstly, only the five songs by the same artist are considered relevant. Secondly, all songs of the same genre are relevant. The TREC evaluation software [9] produced the results that are summarized in Table 4. For each test song, the precision is calculated after each relevant song is retrieved. These precision values are averaged for all relevant songs for the entire test set to produce the average retrieval precision.

System	Feature Vectors	Average Precision	
		Artist	Genre
GMM 16	MFCC	**0.60**	**0.63**
	MP3CEP	**0.60**	**0.59**
TreeQ	MFCC	**0.54**	**0.62**
	MP3CEP	**0.51**	**0.59**

Table 4. Average retrieval precision on a music retrieval task for the **GMM 16** and **TreeQ** systems using both parameterizations. Results are included for same artist & same genre relevance cases.

The figures demonstrate that music retrieval works reasonably well and practical experimentation reinforces this. It is rare to find examples where two songs are declared similar without discernable acoustic similarity. It is worth noting that all average precision figures reported are essentially underestimates due to the nature of the relevance assumptions, though they do serve to highlight the relative performance of the various systems. With the stricter same artist relevance case, GMMs perform rather better than the TreeQ system. At the best average precision figure of **0.60**, the top retrieved song was one of the five candidates by the same artist **75%** of the time. In the genre relevance case, no significant difference exists between the two techniques. Moreover, for music retrieval, the average precision falls on average **4%** across all systems when moving to the MP3CEP parameterization. Again, this is acceptable for applications that demand interactive speeds.

5. SUMMARY

The results in this paper demonstrate that music processing techniques can provide useful tools for the navigation and maintenance of digital music libraries. The performance of music classification is very good with a best accuracy of **92%** discriminating between six musical genres. Furthermore, an average precision figure of **0.63** was achieved for music retrieval. Of the techniques tested, the TreeQ vector quantization is outperformed slightly by Gaussian Mixture Modelling, though the former offers a fast and scalable solution. The MFCC parameterization typically performs well and is independent of music compression scheme, yet it is a little slow for user-interactive application of these techniques. A considerably faster parameterization technique, MP3CEP, has therefore been proposed based on the subband data from partially decompressed MPEG Layer III music. It is anticipated these techniques have further application in any domain where learning a user's musical taste is of benefit.

ACKNOWLEDGEMENT

This research used the TreeQ package developed by Jonathan T. Foote.

REFERENCES

[1] P. Ciaccia, M. Patella and P. Zezula. *M-tree: An Efficient Access Method for Similarity Search in Metric Spaces.* Proc. Of VLDB, 1997.

[2] S.B. David and P. Mermelstein. *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-28 (4), 1980.

[3] J.T. Foote. *Content-Based Retrieval of Music and Audio*. Proc. Of SPIE, Multimedia Storage and Archiving Systems II, pp 138-147, 1997.

[4] J.T. Foote. *The TreeQ Package.* Available from ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/tools/treeq1.3.tar.gz

[5] ISO/IEC International Standard IS 11172-3. *Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbit/s – Part 3: Audio"*

[6] D. Pan. *A Tutorial on MPEG/Audio Compression*. IEEE Multimedia Journal, summer 1995.

[7] L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[8] D.A. Reynolds. *Speaker identification and verification using Gaussian mixture speaker models.* Speech Communication, 17:98-108, 1995.

[9] G. Salton and C. Buckley. *TREC evaluation software distribution.* ftp://ftp.cs.cornell.edu/pub/smart, 1991.

[10] G. Salton, C.S. Yang and A. Wong. *A Vector Space Model for Automatic Indexing.* Communications of the ACM, 18(11), 1975.