ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

Codebook Based Vector Quantization Technique for Song-to-Song Retrieval

Ms. Shabnam R. Makandar[1], Mrs. V. L. Kolhe[2]

Department of Computer Engg. , D.Y. Patil College of Engg. , Akurdi, Savitribai Phule Pune University, Pune, India.[1,2]

Abstract: Digital music is widely used now-a-days. The systems are required for user to find the music the need. The attention has been an increasing on learning feature representations from audio data used in various MIR problems. The key element for relevant retrieval is the audio content representation. Good representation should be short, terse, efficient, and easy and fast to compute. The Bag-of-Frames (BoF) approach is evaluated. In this approach, low-level MFCC and PLP features are explored from the audio signal of songs. The encoding stage is added with pre-computed codebook and pooling stage gives compact representation for the feature vector. A Vector Quantization (VQ) encoding method using Online Dictionary Learning (ODL) algorithm performs well in query-by-example task of MIR to decrease the runtime of relevant retrieval. Experimental result shows that PLP performs better than MFCC.

Keywords: Audio content representation; music information retrieval; MFCC; PLP; sparse coding; bag-of-frames; vector quantization.

I. INTRODUCTION

Over the recent years, digital music has become more friendly and huge on the web and large scale systems for relevant music information retrieval. Content based methods, which present the actual content of music and extract significant information from it. The content based systems for music information retrieval (MIR) tasks such as music classification, semantic annotation and music similarity for song-to-song recommendation are previously researched topics.

The idea behind content-based approaches is to extract information directly from the audio signal, more precisely, from a digital representation of a recording of the acoustic wave, which needs to be accessible. To compare two pieces, their signals are typically cut into a series of short segments called frames which are transformed from the time-domain representation into a frequency-domain representation by choice. Next, feature extraction is performed on each frame in some approach-specific manner. Finally, the extracted features are summarized for each piece. Between these summarizations, pairwise similarities of audio tracks can be computed [3].

The audio feature extraction and representation is focused on the well-organized methods to represent whole song in compact way that ease efficient storage and communication for large music repositories. Suitable processing for fast search and relevant retrieval is necessary. Sparse coding (SC) and deep belief networks (DBNs) algorithms have been mostly utilized for constructing the codebook for music. Inspired by the human's sensory system, SC aims at forming codes that are sparse in support (with most coefficients being zero) but are sufficient to reconstruct or to interpret the input signal. The codebook of SC can be pre-defined using standard bases such as wavelet, Gabor, or Gammatone functions, but can also be learnt from a collection of music signals using algorithms such as matching pursuit and online dictionary learning (ODL)[2].

In this approach, given a codebook, any acoustic feature vector can be replaced by the instance of codewords in the corresponding music signal, leading to the so-called *bag-of-frames* (BoF) representation of music. This technique suspects that a vocabulary consists of finite words and that documents are unordered sets of word instances. Audio events local in time (e.g., guitar solo or riffs) can be represented by different codewords in the BoF model, instead of being spread out as in the case of taking mean or median pooling over the entire feature sequence. Moreover, as the feature representation is like text, one can recast MIR as text IR and benefit from the lessons and techniques that have been learnt and developed for text.

As audio codewords are usually computed from a single or a limited number of consecutive frames (e.g., less than 0.5 second). Encoding stage is used which frame feature vectors with pre-computed codebook and pooling stage perform the temporal integration. The encoding diagnoses informative local patterns and represents the frames at a higher level. The pooling stage makes the compact representation and creates the representation that has the same dimension for all songs, regardless of their durations [1].

The structure of the paper is followed with the Section II reviews the related work on deep belief network, sparse coding, bag-of-frames and dictionary training. Section III explains the bag-of-frames approach and query-by-example task. Section IV describes the implementation details of system architecture, processing of features, temporal pooling and dictionary training. Section V details results of relevant retrieval of song using MFCC features and VQ technique. In section VI future work is described and concludes this study.

II. RELATED WORK

Bag-of-Frames models have been popular in music information retrieval. This vector quantization (VQ) technique has been widely used. A standard approach to form an audio similarity of a word is by clustering a

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

collection of frame-level feature vectors and using the cluster centres to form the codebook. In recent years, sparse coding (SC) and deep belief networks (DBNs) algorithms have been used for building up the codebook for music.

J. T Foote [4] presented an idea for the representation of an audio object by a template that identifies the object. For construction of a template; an audio signal is first divided into overlapping frames of constant length then using simple signal processing techniques, for each frame a13-dimensional feature vector is extracted (12 Mel-Frequency Cepstral Coefficients plus Energy) at a 500Hz, and then these feature vectors are used to generate templates using tree-based Vector Quantized trained to maximize mutual information (MMI). For retrieval, query is first converted in to template in the same way described earlier then for its similarity search template matching is applied which uses distance measure, and finally a ranked list is generated based on minimum distance. In this system performance of the system with Euclidean distance as well as Cosine distance, is also compared, and experimental results show that cosine distance performs slightly better than Euclidean distance. This system may break down for music retrieval if one or other query is inconstant with noise or bad quality recorded.

Muscle fish *et al.* [5] in this system an audio object is represented by its frame level and global acoustical and perceptual parameters. These features are extracted at frame level using signal processing techniques and globally using statistical analysis based on frame level features and musical features (for music signals only) using musical analysis. Frame level features consist of loudness, pitch, tone (brightness and bandwidth), MFCCs and derivative. Global features are determined by applying statistical modeling techniques on the frame level features that is, using Gaussian and Histogram Modeling techniques to analyze audio objects. For musical objects, musical features (i.e. rhythm, events and distance (interval)) are extracted using simple signal processing techniques like pitch tracking, voiced and unvoiced segmentation and note formation. For indexing, multidimensional features space is adopted. For retrieval, distance measure is used and to improve the performance, a modified version of query-point-expansion technique is adopted, but here expansion for the processing of the concept if achieved by standard deviation and mean of the objects in the expected region. This system again limited by its inherited limitation, and works for QBE only.

Riley *et al.* [6] shown that a Bag-of-Audio-Words approach to audio retrieval can be both inclined in a large data set and robust to common signal distortions. Three clustering algorithms for VQ are analysed and found that means achieves competitive result with relatively less estimated cost in ten-class classification for the GTZAN data set [7], [8]. Seyerlehner *et al.* [9] proposed a multi-level approach to advanced VQ, whereas McFee *et al.* [10] proposed a VQ audio representation allow efficient and compact illustration of the acoustic content of music data and used a soft variant of means to decrease quantification errors.

Lee *et al.* [11] reported the first study that applies DBN to MIR problems and feature representations acquire information from unlabelled audio data show very good performance for multiple audio classification tasks. Using the features learnt by DBN beats standard acoustic features such as spectrogram and MFCC. Hamel *et al.* [12] noted that DBN requires large number of hyper parameters to be adapted and possible longer training times.

M. D. Plumbley et al. [13] has presented their work, which depict an approach to musical audio analysis based on a sparse representations search, where any coefficient in such a representation has only a little probability of being far from zero. It is easier to use pre-defined codebooks for SC, which has been shown beneficial over learnt codebooks. For other MIR tasks such as similarity estimation and auto-tagging, it has been shown that the performance of DBN and SC is similar.

Smith *et al.* [14] demonstrated audio codewords learnt by using the matching pursuit (MP) algorithm for sparse decomposition show striking similarities to time-domain cochlear filter estimates. Moreover, as shown by Henaff *et al.* [15], the dictionary is learned assuming the sparse representations of new inputs is very effective, making the system scalable and acceptable for real-time applications and codewords learnt from Constant-Q representations (CQT) using SC correlate well to the specific chords or pitch intervals such as minor thirds, perfect fifths, sevenths, major triads, etc.

Scholler and Purwins [16] found that using a Gammatone dictionary as exemplar codewords leads to better accuracy in drum sound classification than the codewords learnt by using matching search. Yeh *et al.* [17] found that using log-power spectrogram for low-level feature representation and ODL for feature learning cause the best performance. This result suggest that then way the codewords are assigned may be more important than the way the codebook is generated, which is in line with the observations made in [18]. Coates and Ng [18] examined the usage of different sequences of dictionary training algorithms and encoding algorithms to better explain the successful performance of sparse coding in earlier works. They concluded that the dictionary training stage has less of an affect the final performance than the encoding stage and that the main benefit of sparse coding may be due to its nonlinearity, which can be achieved also with simpler encoders that exercise some nonlinear soft thresholding.

Y. Vaizman *et al.* [1] examined the sparse representations were assigned directly to time domain audio signals, with a trained codebook. The three encoding techniques LASSO, Vector Quantization and Cosine Similarity are compared for dictionary training using ODL and found that VQ is the efficient encoding method can successfully compete with the more sophisticated method (the LASSO), achieving better performance, with much less computing resources.

DOI 10.17148/IJARCCE.2015.4667

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

III. IMPLEMENTATION DETAILS

The system is to retrieve songs from the repository and rank them in order of relevance to the query. In query-by-example (or "song-song retrieval") the query is a song by itself, enabling an online radio or other interfaces. Efficient content analysis methods could admit for a real-time query-by- example interface, where the user may upload an unfamiliar song to the system, and get similar/relevant songs in return.

A. System Architecture:

Figure 1 shows the proposed system architecture and the processing of songs. At first data is trained. The MFCC and PLP features are extracted which are further reduced using PCA. The Online Dictionary Learning Algorithm and Vector Quantization technique, both are used to generate codebook.

For compact audio content representations for full-length songs that will be effective for MIR application query-by-example. A large scale calculation is carried out. The result of design in the "low-level-feature, encoding, pooling" scheme is determined, and ultimately retrieve a representation "recipe" (based on vector quantization) that is efficient to estimate, and has consistent high performance in MIR application.

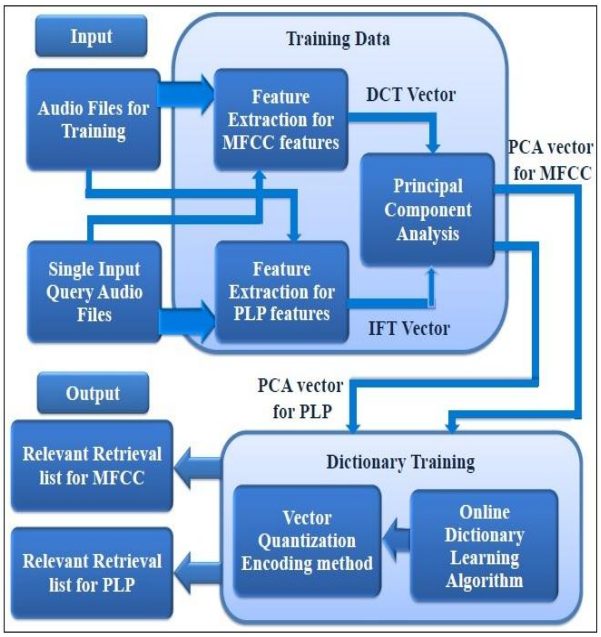

Fig.1: Proposed System

B. Bag-of-Frames Approach:

The encoding-pooling scheme to get a compact representation for each song (or musical piece) is examined. The scheme is consist of three stages:

1) *Short time frame features:* each song is processed to time series of low-level feature vectors $X \in \mathbb{R}^{d \times T}$, ($T$ time frames, with a dimensional feature vector from each frame).
2) *Encoding:* each feature vector $x_t \in \mathbb{R}^d$ is then encoded to $\alpha_t \in \mathbb{R}^k$ a code vector, using a pre-calculated dictionary $D \in \mathbb{R}^{d \times k}$, a codebook of"basis vectors" of dimension. We get the encoded song $C \in \mathbb{R}^{k \times T}$.
3) *Pooling:* the frame vectors coded are together pooled to a single compact vector $v \in \mathbb{R}^k$.

C. Query-by-Example (QbE):

Given a query song, whose audio content is represented as vector $\mathbf{q} \in \mathbb{R}^{\mathbf{k}}$, query-by-example system calculates its distance $\mathbf{dist}(\mathbf{q}, \mathbf{r})$ from each repository $\mathbf{r} \in \mathbb{R}^{\mathbf{k}}$ and the relevant retrieval result is the repository songs ranked in increasing order of distance from the query song. The Euclidean distance is a probable simple distance measure between songs' representations. However, it allocates equal weight to each of the vectors' dimensions, and it is possible that there are dimensions that import most of the relevant information, while other dimensions carry just noise. For that reason, we use a more general metric as a distance measure, the Mahalanobis distance:

$$dist(q,r) = \sqrt{(q-r)^T W (q-r)}, \qquad (1)$$

when $W \in \mathbb{R}^{k \times k}$ is the parameter matrix for the metric (W has to be a positive semi definite matrix for a valid metric)[1].

D. Processing of features:

Feature extraction is the process of estimating a compact numerical representation that can be used to describe a segment of audio. Once the features are extracted standard machine learning techniques which are not dependent of the specific application area can be used.

1)Mel-Frequency Cepstral Coefficients Features (MFCC): The mel-frequency spectrum (MFC) is a representation of the short-term power cepstrum of a sound. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively form an MFC. The cepstrum can be defined as information about rate of change in the different spectrum bands (MFS). Spectral features that are commonly assumed to catch timbral qualities are used. Since we are interested in general sound similarity, so assume timbral features to be relevant here. Low-level features are based on mel frequency spectra (MFS), which are calculated by computing the short time Fourier transform (STFT), sum up the spread of energy along mel scaled frequency bins, and compressing the values with logarithm. Mel frequency cepstral coefficients are the result of additional processing MFS, using discrete cosine transform (DCT), in order to both create uncorrelated features from the correlated frequency bins, and reduce the feature dimension [19].

2)Perceptual Linear Prediction Features: H. Hermansky has developed the PLP model. For transformation of a power speech spectrum to a comparing auditory spectrum the PLP combines three components from the psychophysics of hearing: the critical-band spectral selectivity, the equivalent loudness curve and the intensity-loudness power law. PLP is alike LPC except that its spectral characteristics have been transformed to match characteristics of human auditory system [20].

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

3) *Principal Component Analysis (PCA):* For QbE PCA decorrelation and dimensionality reduction is performed on the data: in each split the PCA matrix is estimated from the train set and the song representation vectors (of train, validation and test set) are projected to a predetermined lower dimension (so the trained matrices are in fact not but smaller). PCA whitening uncorrelates the mel scaled spectral features and thus encase most information in the diagonal of the covariance matrix. In reaction, relevant information flows better through the pooling functions, which gives better pooled features and allows faster and more efficient training [1].

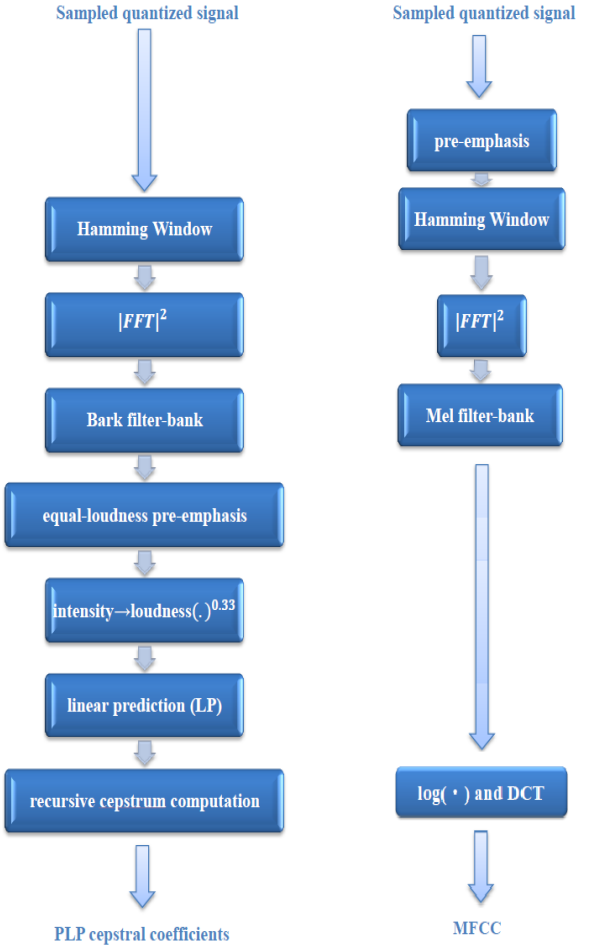

Fig. 2: The computation steps of PLP (left) and MFCC (right)[23].

E. Temporal Pooling:

Pooling is one of the simplest ways of aggregating time varying information. Temporal pooling is getting a compact representation of a song by generative modelling. The pooling stage allows us to have a unified dimension to the representations of all songs, regardless of the songs' durations. A simple way to pool the low-level frame vectors together is to take some statistic of them, typically their mean. For a monotonic, short song, such a statistic may be a good representative of the properties of the song. The pooling of the coded vectors is meaningful using mean pooling:

results in a histogram representation, stating the frequency of occurrence of each sound pattern [22].

$$v = \frac{1}{T}\sum_{t=1}^{T}\alpha_t \qquad (2)$$

F. Dictionary Training:

The training of the dictionaries (codebook construction) is performed with the online learning algorithm for sparse coding and the encoding method used is vector quantization.

1) *Online Dictionary Learning (ODL):* is a first-order stochastic gradient descent algorithm proposed by Mairal *et al.* [22] to solve the following optimization problem,

$$arg\min_{D\in C}\frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}\left||x_i - D\alpha_i\right||_2^2 + \lambda\left||\alpha_i\right||_1\right), \quad (3)$$

where $x_i \in \mathbb{R}^d$ denotes the (observed) i-th signal among a set of t signals, $\alpha_i \in \mathbb{R}^k$, C is a set of (unknown) convex matrices $D \in \mathbb{R}^{d\times k}$ satisfying $d_j^T, d_j \leq 1$, a constraint that is dictating to limit the energy of the codewords and λ is a pre-set parameter for the trade-off between the sparsity of α and the representation accuracy. A natural solution to this joint optimization problem is to solve for the two variables D and α in an alternating fashion: reduce one while keeping the other fixed. The optimization D of uses block coordinate descent with warm restarts, which aggregates the past information computed during the previous steps of the algorithm. The optimization of α involves a typical sparse decomposition problem. Several optimization steps are made until convergence [22].

Algorithm 1 Online Dictionary Learning

Require: $x \in \mathbb{R}^d \sim p(x)$ (random variable and an algorithm to draw i.i.d samples of p), $\lambda \in \mathbb{R}$ (regularization parameter), $D_0 \in \mathbb{R}^{d\times k}$ (initial dictionary), T (number of iterations).

1: $A_0 \in \mathbb{R}^{k\times k} \leftarrow 0$, $B_0 \in \mathbb{R}^{d\times k} \leftarrow 0$ (reset the "past" information).

2: for $t = 1$ to T do

3: Draw x_t from $p(x)$.

4: Sparse coding: compute using Vector Quantization (α_t).

5: $A_t \leftarrow A_{t-1} + \alpha_t\alpha_t^T$.

6: $B_t \leftarrow B_{t-1} + x_t\alpha_t^T$.

7: Compute D_t using Algorithm 2, with D_{t-1} as warm restart, so that

$$D_t \triangleq arg\min_{D\in C}\frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}\left||x_i - D\alpha_i\right||_2^2 + \lambda\left||\alpha_i\right||_1\right),$$
$$= arg\min_{D\in C}\frac{1}{t}\left(\frac{1}{2}\left||x_i - D\alpha_i\right||_2^2 + \lambda\left||\alpha_i\right||_1\right). \qquad (4)$$

8: end for

9: Return D_t (learned dictionary).

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

Algorithm 2 Dictionary Update

Require: D = $[d_1, \ldots, d_k] \in \mathbb{R}^{d \times k}$ (input dictionary), A = $[a_1, \ldots, a_k] \in \mathbb{R}^{k \times k}$,
B = $[b_1, \ldots, b_k] \in \mathbb{R}^{d \times k}$
1: repeat
2: for $j = 1$ to $k$do
3: Update the j-th column to optimize for (4):

$$u_j = \frac{1}{A[j,j]}\left(b_j - Da_j\right) + d_j,$$
$$d_j \leftarrow \frac{1}{\max(\left|\left|u_j\right|\right|_2, 1)} u_j. \quad (5)$$

4: end for
5: until convergence
6: Return D (updated dictionary).

2) *Top-τ Vector Quantization Encoding method:*

Given the codebook, any input signal can be represented by a linear combination of the codewords. The vector of combination coefficients can be either sparse or dense, rely upon how the encoding algorithm manipulates the loss function and the sparsity constraint.

In vector quantization (VQ) a continuous multi-dimensional vector space is partitioned to a discrete finite set of bins, each having its own representative vector. The training of a VQ codebook is essentially a clustering that describes the distribution of vectors in the space.
During encoding, each frame's feature vector is quantized to the closest codeword in the codebook, meaning it is encoded as, a sparse binary vector with just a single "on" value, in the index of the codeword that has smallest distance to it (we use Euclidean distance)[17].

It is also possible to use a softer version of VQ, selecting for each feature vector the nearest neighbors among the codewords, creating a code vector α_t with τ "on" values and $k - \tau$ "off" values:

$$\alpha_t(j) = \frac{1}{\tau}\mathbb{1}\left[D_j \in \tau - nearest\ neighbors\ of\ x_t\right],$$
$$j \in \{1,2,\ldots,k\}. \quad (6)$$

The hard threshold of selecting just one codeword will result in distorted, noise-sensitive code, while using top-τ quantization will be stronger.
Of course, if τ is too large, may end up with codes that are not important—all the songs will have similar representations and all the discriminating information will be lost.
The encoding parameter is a density parameter τ, with larger values causing denser codes. By adjusting we can directly control the level of sparsity of the code For VQ, using mean pooling results in a codeword histogram representation with richer values.

G. Mathematical Model:

1) Let S be the system or application.

$$\boldsymbol{S = \{I, O, \delta, A, q_0, f\}}$$

Where,
I = set of inputs = {Dataset, Single file Input},
O = set of output = {Relevant Retrieval List},
δ = transition = {Feature extraction, Dictionary training},
A = set of algorithms = {Online Dictionary Learning, Dictionary Update},
q_0 = initial set = {Feature Vector Set},
F = final set = {Single Compact Vector Set}.

2) Feature Extraction

$$F = \{p, f_0, t, a_0, w\}$$

Where,
- p – Power spectrum,
- f_0 – frequency,
- t – time or period,
- a_0 – amplitude,
- w – wavetimelength.

$$wavetimelength = \frac{\frac{streamlength}{samplerate \times (\frac{bitrate}{8})}}{channel} \quad (7)$$

3) MFCC Features

$$M = \{w_1, f_1, d_0\}$$

- w_1 – Breaking input file into window frames,
- f_1 – Fast Fourier Transformation,
- d_0 – Discrete Cosine Transformation.

4) PLP Features

$$P = \{h_1, f_2, b, l_1, i\}$$

- h_1 – Breaking input file into window frames,
- f_2 – Fast Fourier Transformation,
- b – Discrete Cosine Transformation,
- l_1 – Loudness,
- i – Intensity

H. Experimental Setup:

The standard configuration required to build the system is using Java framework (jdk 1.7). SWING is used to form front end. Netbeans 7.4 is used as a developer tool. Each experiment regards to a different type of audio-content representation.
We experiment with combinations of the following parameters:
- low-level features: MFCCΔ or MFSΔPC and PLPΔ,
- codebook size $k \in 1024$,
- encoding method: the VQ,
- encoding parameters: $\tau = \{1,2,4,8,16,32\}$,
- pooling function: mean.

IV. RESULTS

A. Data Set:

A dataset is collected from the GTZAN. In the dataset 120 tracks exists, each 30 seconds long. Each class (music/speech) contains 60 examples.
The tracks are all 22050Hz Mono 16-bit audio les in .wav format. Size of the dataset is approximately 297MB.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

B. Results:

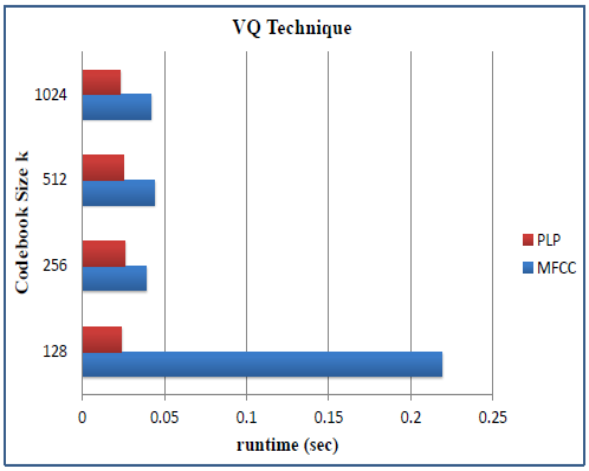

Fig. 3: Empirical runtime test for different codebook sizes for VQ

Figure 3 shows average runtime for VQ encoding a song as a function of *k* (log-scale / codebook size), and standard deviation in error-bars. Multiple points represent the vector quantization values.

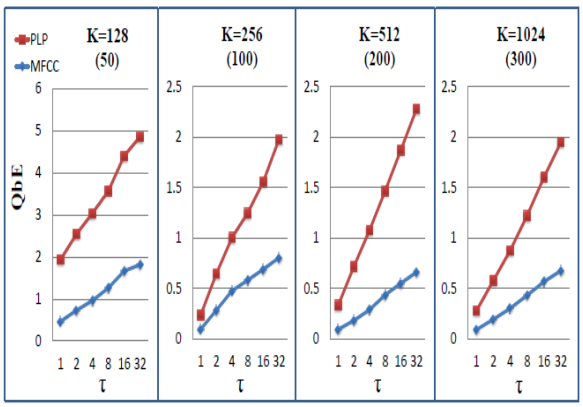

Fig. 4: Query-by-example with Vector Quantization for MFCC and PLP features.

Figure 4 shows Query-by-example with Vector Quantization for MFCC and PLP features. τ (log-scale / density parameter) for VQ. These results are when using PCA dimensionality reduction from k = 128, 256, 512, 1024 to d_{PC} = 50, 100, 200, 300 respectively. The number below codebook is the codebook size k is reduced dimension used for QbE.

V. CONCLUSION AND FUTURE WORK

In this paper we presented an advantage to using PCA decorrelation of MFSΔ features over MFCC and PLP. Our revised setup with PLP applies a pre-emphasis to the signal, and employs a Mel filter-bank with a large number of filters (e. g. 257) and a band-width around 230 Mel. Equal-loudness weighting and duplication of the boundary values of the filter-bank are discarded which improves the accuracy. The difference is analytically significant, but small, showing that the data-agnostic DCT manages to compress music data well. Increasing the codebook size gives better performance for the VQ method. Performance may degrade for the encoding method when too low values of encoding parameters are used. VQ is more powerful, having smooth and controlled change in performance when adjusting its density parameter. The resulting representations are concise, easy to work.

For future work, Perceptual Linear Prediction (PLP) features can be used for other applications of MIR such as query-by-tag to represent various aspects of musical audio.

ACKNOWLEDGEMENT

The authors would like to thank the researchers as well as publishers for making their resources available and the teachers for their guidance. We also thank the college authorities for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to all friends and family members.

REFERENCES

[1] Vaizman, Y.; McFee, B.; Lanckriet, G., "Codebook-Based Audio Feature Representation for Music Information Retrieval," Audio, Speech, and Language Processing, IEEE/ACM Transactions on , vol.22, no.10, pp.1483,1493, Oct. 2014.
[2] Li Su; Yeh, C.-C.M.; Jen-Yu Liu; Ju-Chiang Wang; Yi-Hsuan Yang, "A Systematic Evaluation of the Bag-of-Frames Representation for Music Information Retrieval," Multimedia, IEEE Transactions on , vol.16, no.5, pp.1188,1200, Aug. 2014.
[3] Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M., "Content-based music information retrieval:Current directions and future challenges,". Proc. IEEE 96, 668–696, 2008.
[4] J. Foote., "An overview of audio information retrieval," In Multimedia Systems 7, pp 2- 10. ACM, January 1999.
[5] A. Khokhar, G. Li "Content-based Indexing and Retrieval of Audio Data using Wavelet," ICME, 2000.
[6] M. Riley, E. Heinen, and J. Ghosh, "A text retrieval approach to content- based audio retrieval," in Proc. ISMIR, 2008, pp. 295–300.
[7] Z. Fu, G. Lu, K.-M. Ting, and D. Zhang, "Music classification via the bag-of-features approach," Pattern Recognit. Lett., vol. 32, pp. 1768–1777, 2011.
[8] J. Wülfing andM. Riedmiller, "Unsupervised learning of local features for music classification," in Proc. ISMIR, 2012, pp. 139–144.
[9] K. Seyerlehner, G.Widmer, and P.Knees, "Framelevel audio similarity: A codebook approach," in Proc. Int.Conf.Digital Audio Effects, 2008.
[10] B. McFee, L. Barrington, and G. R. G. Lanckriet, "Learning content similarity for music recommendation," IEEE Trans. Audio, Speech, Lang. Process., vol. 20, no. 8, 2012.
[11] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in Proc. NIPS, 2009, pp. 1096–1104.
[12] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in Proc. ISMIR, 2010, pp. 339–344.
[13] C.-T. Lee, Y.-H.Yang, and H. H. Chen, "Multipitch estimation of piano music by exemplar-based sparse representation," IEEE Trans. Multimedia, to be published.
[14] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," Nature, vol. 439, no. 7079, pp. 978–982, 2006.
[15] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in Proc. ISMIR, 2011, pp. 681–686.
[16] S. Scholler and H. Purwins, "Sparse approximations for drum sound classification," IEEE J. Select. Topics Signal Process., vol. 5, no. 5, pp. 933–940, 2011.
[17] C. Yeh, M. C. , and Y. H. Yang, "Supervised dictionary learning for music genre classification," in Proc. ICMR, 2012.
[18] A. Coates and A. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in Proc. ICML, 2011, pp. 921–928.
[19] B. Logan, "Mel frequency cepstral coefficients for music modeling," in Proc. Int. Soc. Music Inf. Retrieval Conf. (ISMIR), 2000, vol. 28.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

[20] Josef Psutka, "Comparison of MFCC and PLP parameterizations in the Speaker Independent Continuous Speech Recognition Task," Eurospeech- Scandinavia, 2001.
[21] C. J. S. Essid and G. Richard, "Temporal integration for audio classification with application to musical instrument classification," IEEE Trans. Audio, Speech, Lang. Process., vol. 17, no. 1, pp. 174–186, Jan. 2009.
[22] J.Mairal, F. Bach, J. Ponce, andG. Sapiro, "Online learning for matrix factorization and sparse coding," J. Mach. Learn. Res., vol. 11, pp. 19–60, 2010.
[23] Hönig, Florian, Georg Stemmer, Christian Hacker, and Fabio Brugnara. "Revising Perceptual Linear Prediction (PLP)." In INTERSPEECH, pp. 2997-3000. 2005.

BIOGRAPHIES

Shabnam R. Makandar received the BE degree in Computer Science and Engineering from B.V.C.O.E.K, Kolhapur which is under Shivaji University, in 2013, and currently pursuing Master of Engineering degree in Computer Engineering from D. Y. Patil College of Engineering, Akurdi, Pune under Savitribai Phule Pune University. Her research interests includes Data Mining and Soft Computing.

V. L. Kolhe is an Assistant Professor in the Computer Engineering Department, D. Y. Patil College of Engineering, Akurdi, Pune. She received the BE degree in Computer Technology from Nagpur University, and ME degree in Computer Engineering from Savitribai Phule Pune University.