2012 11th International Conference on Machine Learning and Applications

Feature Mapping and Fusion for Music Genre Classification

Haythem Balti
Multimedia Research Lab
University Of Louisville
Louisville, USA
Email: h0balt01@louisville.edu

Hichem Frigui
Multimedia Research Lab
University Of Louisville
Louisville, USA
Email: h.frigui@louisville.edu

***Abstract*—We propose a feature level fusion that is based on mapping the original low-level audio features to histogram descriptors. Our mapping is based on possibilistic membership functions and has two main components. The first one consists of clustering each set of features and identifying a set of representative prototypes. The second component uses the learned prototypes within membership functions to transform the original features into histograms. The mapping transforms features of different dimensions to histograms of fixed dimensions. This makes the fusion of multiple features less biased by the dimensionality and distributions of the different features. Using a standard collection of songs, we show that the transformed features provide higher classification accuracy than the original features. We also show that mapping simple low-level features and using a K-NN classifier provides results comparable to the state-of-the art.**

***Keywords*-Music genre classification; clustering; feature mapping; fusion**

I. Introduction

Browsing and indexing large music collections requires new efficient and automated tools. Recent research has shown that music information such as genre, mood, and rhythm can be extracted from songs [1], [2] and used by machine learning algorithms to efficiently index and retrieve music by content [3], [4], [5]. Music genre is one of the most efficient ways to represent music since it can group heterogeneous music collections into clusters that can be shared by many users. Genre classification has received significant attention lately in the music information retrieval community [3], [2], [4], [5], [6]. The general approach involves three main steps: preprocessing, feature extraction, and classification. Preprocessing uses some criteria to enhance the music signal. Feature extraction transforms the music signal to a reduced representation by keeping only relevant information for the desired application. In the last step, a classifier is trained to predict the genre of a new music signal.

For feature extraction, Researchers have focused mainly on developing new ways to represent information like rhythm and timbre. For instance, in [7] the authors proposed timbre and rhythm related features based on beat histogram and pitch related features. In [8] the authors used the Daubechies wavelet coefficient histogram (DWCH) to capture both local and global information. In [9], Ahrendt et al. proposed the use of co-occurrence models where, instead of considering the whole song as an integrated part of a probabilistic model, they considered it as a set of independent co-occurrences. In [10] Elis proposed the use of beat-synchronous chroma features, designed to reflect melodic and harmonic content that are invariant to instrumentation.

The performance of the different audio features depends on the music genre and other unknown factors, and there is no one feature that consistently outperforms all others. Consequently, researchers have investigated the possibility of fusing multiple features to take advantage of their individual strengths. For instance, in [11] the authors combined features derived from static and transitional information (delta-MFCC, delta-OSC, delta-delta-MFCC, and delta-delta-OSC) of cepstral (MFCC) and spectral (OSC) features to improve the classification accuracy. Feature level and decision level fusion were used to combine the extracted features. In the feature level approach, the features are concatenated to form one global feature vector. In the decision level fusion, each feature is independently processed by a different classifier, and the outputs of the multiple classifiers are fused using summation and product rules. In [12], the authors proposed an approach for music genre classification based on multiple classifier fusion. First, MFCCs and four features from MPEG-7 standard [13] are extracted. Then, random forest and multilayer perceptron neural networks were applied to the data and a weighted voting strategy was used to fuse the results of the two classifiers.
One of the limitations of feature level fusion is the lack of a robust approach to deal with the different dimensions and distributions of the multiple features. In fact, these can be the dominant factor in influencing the fusion results. In this paper, we propose a feature level fusion that is based on mapping the original low-level audio features with different dimensions to histogram descriptors of equal dimensions. Our approach is based on two main components. The first one consists of clustering each set of features and identifying a set of representative prototypes. The second component

978-0-7695-4913-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ICMLA.2012.59

uses the learned prototypes within possibilistic membership functions to transform the original features into histograms.

II. Feature Level Fusion of Multiple Audio Features

A. Feature Mapping

Our proposed representation is inspired by the "bag of words" concept used in natural language processing and information retrieval [14]. In the bag of words representation, each document is described by a histogram that represents the count of every term in the document. For our application, we assume that songs are documents and we learn a set of clusters that could be used as terms. The mapped descriptor would reflect the response of the audio features of a song to the set of learned clusters. An illustrative block diagram of the proposed mapping is shown in Fig.1. It consists of two main steps. The first one clusters raw features and identifies a set of representative prototypes. The second step consists of mapping the features to histograms and concatenating.

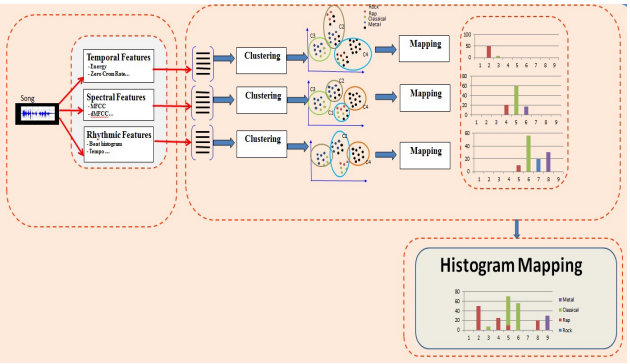

Figure 1. Overview of the proposed feature mapping and fusion

1) Feature Clustering: We assume that we have L classes of music genre and for each class ,i, we have N^i training samples. We also assume that for each sample we extract F low level audio features such as MFCC [7], dMFCC [7], etc. Thus, the training data can be represented by $X = \{X^{if} \text{for} \quad i = 1...L \quad , and \quad f = 1...F\}$ where $X^{if} = \{x_j^{if} \text{for} \quad j = 1...N^i\}$ and x_j^{if} corresponds to the j^{th} feature vector of class i represented by the f^{th} feature set. The first step of our approach consists of identifying a set of prototypes $\{C_1^{if}, .., C_{K^i}^{if}\}$ that is representative of training data X^{if}. This can be achieved by using a clustering algorithm to partition X^{if} into K^i clusters and letting C_k^{if} be the centroid of the k^{th} partition. In particular, we partition X^{if} by minimizing

$$J(U; X^{if}) = \sum_{j=1}^{N^i} \sum_{t=1}^{K^i} \mu_{tj}^m d^2(x_j^{if}, C_t^{if}) \qquad (1)$$

In (1), $d^2(x_j^{if}, C_t^{if})$ refers to the Euclidean distance between data sample x_j^{if} and the centroid of cluster t, and $U = [u_{tj}]$ represents the membership of feature vector x_j^{fi} in the t^{th} cluster.
After clustering, we end up with K^i clusters for each class and for each feature set.

2) Feature Mapping: In a second step, The features X are projected into a new space H defined by the clusters. The new space captures the distribution of the training data set using the cluster centers computed in the clustering step. Each feature, f, is independently projected into a new space defined by its prototypes using

$$\begin{aligned} M^f : \quad & X^{if} \longrightarrow H^f \\ M^f(x_j^{if}) \quad = \quad & h_j^f = [\mu_1(x_j^{if}), ..., \mu_K(x_j^{if})] \end{aligned} \qquad (2)$$

In (2), $\mu_t(x_j^{if})$ is the membership of feature vector x_j^{if} in the t^{th} cluster of feature f, and $K = \sum_{k=1}^{L} K^k$. The membership could be crisp, fuzzy, or possibilistic. In this paper we use and compare crisp and possibilistic membership functions. The crisp function is defined as

$$\mu_t(x_j^{if}) = \begin{cases} 1, & \text{if } t = \arg\min_k(d((x_j^{if}), C_k^{if}))) \\ 0, & otherwise \end{cases} \qquad (3)$$

This function considers only the closest cluster to the input feature vector x_j^{if}.
For the possiblistic mapping, we use the membership function

$$\mu_t(x_j^{if}) = exp(-\frac{||x_j^{if} - C_t^{if}||^2}{a * (\sigma_t^{if})^2}) \qquad (4)$$

where σ_t^{if} is the variance of the data samples belonging to the t^{th} partition using feature f. In equation (4), a is a constant chosen empirically.

The possibilistic mapping defined by (4) is a soft mapping where each sample x_j^{if} contributes to each of the K bins of the histogram by a value within [0,1]. This is in contrast to the crisp mapping in (3) where each sample votes for only one bin. In possibilistic mapping, the closer the samples are to cluster k, the higher their contribution to bin k. This soft voting makes the mapping less sensitive to the selection and the number of clusters.

After mapping each data sample (song) in X to a histogram h_k^f where k denotes the k^{th} data sample in X, the F histograms are fused through a simple concatenation. That's $H_j = [h_j^1, ..., h_j^f, ..., h_j^F]$.

One advantage of the proposed mapping is that all features get mapped to histograms of equal number of bins in a similar data driven way. Thus, the fused feature will not be biased by the dimensionality of the original features.

B. Classification

We use a simple K-nearest neighbor classifier [15]. Our choice is motivated by two main factors. First, KNN is simple and its results are easily interpretable. Second, KNN is flexible and can integrate any distance measure.

Various distances were proposed in the literature to compare histograms [16],[17]. In general, these distances can be classified into two categories: Bin to bin and cross-bin. The bin-to-bin distances assume that the histograms are aligned, and each bin in one histogram is compared to only the corresponding bin in the other histogram. Examples of such distances include the cosine distance[18], and Kullback-Leibler divergence [19].

The cross-bin distances allow bins at different locations to be (partially) matched. The partial matching alleviates the quantization effect. Examples of such distances include the earth mover's distance (EMD) [20], and the diffusion distance [16].

In this paper, we compare and report the results using the cosine distance, the diffusion distance, and the KL divergence.

III. Experimental Results

The GTZAN dataset [7] is used to illustrate the performance of the proposed feature mapping and fusion. This collection contains 10 music genre classes, each class contains 100 songs of 30 seconds length.

To extract features that describe the songs, we divide the stream of each song into 50ms windows with a 50% overlap. This will results in 1995 windows per song. For each window, we extract 9 sets of features (i.e F=9). These are the MFCC [7], dMFCC [7], RMS, spectral brightness, spectral flatness, spectral spread, spectral centroid, spectral roll-off and, spectral flux [21].

The data collection is divided into training and testing sets using 10% cross validation. That is, we keep 10% for testing and use the remaining 90% for training. This procedure is repeated 10 times, each time using a different 10% for testing. The reported results are the average over the 10 runs. For each run, the training data is used to learn the prototypes and the parameters needed for the mapping. We cluster the training samples of each class using the k-means clustering algorithm[22]. We use $K^i = 10$ clusters for each feature, except for the MFCC and dMFCC where $K^i = 30$. Then, we map each feature using (2). Thus, each data sample (extracted from a 50 ms window) is represented by 9 histograms. The MFCC and dMFCC histograms have 300 bins while histograms of the other features have only 100 bins. The histograms of the 1995 windows of each song are added to create one histogram that describes each song for each feature. Fig.2 displays the histogram representation of 4 songs taken from 4 different classes (Classical, Metal, Hip Hop, Blues) using dMFCC. As it can be seen, songs from the different classes have a different response to the set of prototypes. For example, the classical song have a high response to the first 30 bins (these correspond to the prototypes learned from the training data of the class) but uniform response to all the other prototypes. Similarly, the hip hop sample have a high response to the clusters of its own class and low response to the prototypes of classical music.
Finally, for the feature level fusion, we concatenate the 9 histograms and obtain one global histogram with 1300 bins.

To test the proposed feature mapping, we first compare the performance to the original features using a simple K-NN classifier and the Euclidean distance. Every window is classified independently then a song is assigned to the most frequent class among its windows. We used Euclidean distance because most of the features used in this paper are one dimensional. As a consequence, the cosine, diffusion, and KL distances are not suitable. For the mapped features, we also compare the performance using Euclidean, cosine, diffusion and KL distances.
Table I. presents the classification results of the mapped and original features. As it can be seen, the proposed mapping improves the classification rate of every feature. For example, for the MFCC features, the mapping improved the classification accuracy by 10% when the diffusion distance is used. The main reason for the improved performance is that, using the original features, every feature vector extracted from one window is classified independently. This approach is not robust when the music signal contains many silence or noise segments. Those segments will heavily influence the classification accuracy and can be misleading. Our approach can alleviate this problem by automatically grouping silence or noise segments into few prototypes. These prototypes are then part of the feature representation.

Another advantage of the proposed approach is that the diffusion distance was able to outperform the Euclidean, cosine, and KL distances. Being a cross-bin distance, it alleviates the quantization effect in histograms. Thus, it matches similar histograms even if they are shifted. Our mapped features are also more compact than the original features. Every song is represented by one histogram after mapping compared to a 1995xd matrix (d denotes the dimension of the feature in the original space). For example, a song is represented by 1995x13 in the original space for MFCC and dMFCC. When we apply our feature mapping, the representation is reduced to a 300 bins histogram.

Another major advantage of the proposed approach is that it facilitates feature level fusion. The fact that each set of feature is mapped to a histogram with comparable distributions and dimensions makes the fusion less sensitive to these factors. Thus, as it can be seen in table I, a simple

concatenation of the 9 histograms can provide a significant improvement to the classification rate.

In the second experimental setup, we tested our feature fusion approach against the results of many classification methods compared and reported in[2]. Table II. shows the results of state of the art methods in music genre classification. Most of the methods in Table II use high level and low level features combined with robust classifiers like support vector machines(SVM) and sparse representation classifiers (SRC). We used the same experimental setup described in [2] to make sure that the results are comparable. Our best classification accuracy was 81.14%, achieved using the diffusion distance and KNN. Thus, using only low level features and the classic KNN, our approach was able to outperform most of the state of art methods mentioned in Table II.

The last 4 methods in table II outperform our classifier. These methods integrate high level features and more robust classifiers. Currently we are investigating applying the same mapping to these high level features. We are also investigating using more robust classifiers such as SVM and SRC. The highest

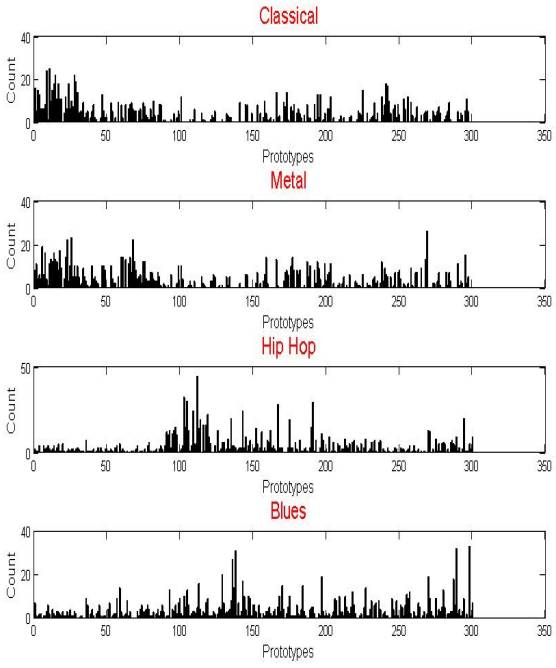

Figure 2. The histogram representation of the dMFCC of 4 songs taken from 4 different classes.

Features	Cosine	Diffusion	KLDIV	Euclidean	Ori. Features
MFCC	59.1%	61.2%	59.9%	52%	51.6%
dMFCC	62.5%	61%	61.14%	64.6%	52.4%
Flatness	34.2%	35.1%	37.7%	33%	24.2%
Spread	35.1%	35.4%	35.1%	34.9%	31.2%
C	32.7%	35.2%	33.3%	33.9%	28.8%
Brightness	35.5%	35.1%	35.8%	34.2%	34.4%
R	37.8%	38.6%	38.8%	36.8%	27.8%
RMS	24.7%	24.9%	25.7%	24.9%	24%
F	28.2%	32%	30.9%	28.2%	25.8%
Fused Features	72.71%	**81.14%**	79%	62%	59.7%

Table I: Comparaison of the classification rate of the proposed mapping and the original features .

Reference	Features	Classifier	Accuracy (%)
[7]	STFT+MFCCxMuVar +beat+pitch	K-NN	60
[7]	STFT+MFCCxMuVar +beat+pitch	GMM	61
[23]	MFCCxFP	SVM	77.7
[24]	MFCCxGMM	K-NN	70.6
[24]	MFCCxGMM	SVM	70.4
[25]	STFT+MFCCxMuVar+beat+pitch	SVM	72
[25]	STFT+MFCCxMuVar	SVM	71.8
[25]	DWCH+STFT+MFCCxMuVar	SVM	78.5
[26]	MFCCxMuCov	SVM	78.6
[27]	STFT+MFCCxMuVar	SVM	79.8
[28]	STFT+FFT+MFCC+LPC	AdaBoost.DT	82.4
[29]	CRxNTF	SVM	78.2
[30]	MFCC+ASE+OSCxFPxLDA	NC	90.6
[31]	CRxNTF	SRC	92.4
[32]	MFCC+ASE+OSCxMuCov,FP+beat+chord	SVM(MKL)	90.4
[32]	MFCC+ASE+OSCxMuCov,FP+beat+chord	SVM(SG)	90.9

Table II: Performance comparison of genre classification algorithms on the standard GTZAN data set .

IV. Conclusions

We proposed a data driven mapping that transforms audio features into histogram descriptors. The first step consists of learning a set of prototypes that is representative of the training data. Then, using a soft voting approach, the low-level features of a song get mapped to a histogram that reflects their proximity to the learned prototypes. The dimensionality of the mapped histogram is determined by the number of prototypes used to summarize the data. Thus, features with different dimensionality and distributions get mapped to histograms with equal dimensions and comparable distributions. This makes fusion of multiple features less sensitive to these factors.

Using a standard collection of songs, we showed that the mapped features provide higher classification accuracy than the original features. We also showed that mapping simple low-level features and using a K-NN classifier provides results comparable to the state-of-the art.

The proposed mapping could be improved by extracting additional information during the clustering process. For instance, the validity of each cluster could be used to assign a relevance weight to each bin of the histogram. Similarly, clusters that do not discriminate between samples from the difference classes could be ignored.

REFERENCES

[1] F. Aucouturier, J. J. & Pachet, "Improving timbre similarity: How high is the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[2] Z. G. Scaringella, N. and D. Mlynek, "Automatic genre classification of music content: a survey," *Signal Processing Magazine, IEEE*, 2006.

[3] O. M. L. Q. Li, T., "A comparative study on content-based music genre," *Proc. SIGIR*.

[4] G. L. . K. M. T. . D. Z. Zhouyu Fu, "A survey of audio-based music classification and annotation," *Multimedia, IEEE Transactions*, 2011.

[5] Y. S. C. Zhang, "Content-based information fusion for semi-supervised music genre classification," *Multimedia, IEEE Transactions*, 2008.

[6] F. A. . W. G. Pampalk, E., "Improvements of audio-based music similarity and genre classification," *ISMIR*, 2005.

[7] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, July 2002.

[8] G. Li, T. & Tzanetakis, "Ieee workshop on applications of signal processing to audio and acoustics (waspaa), new paltz, ny," *IEEE Trans. Speech and Audio Processing*, 2003.

[9] L. J. Ahrendt, P. and C. Goutte, "Co-occurrence models in music genre classification," *Machine Learning for Signal Processing, IEEE Workshop*, pp. 247–252, 2005.

[10] D. Ellis, "Classifying music audio with timbral and chroma features," *ISMIR*, 2007.

[11] C.-H. L. J.-L. S. K.-M. Y. H.-S. L. M.-H. Wei, "Fusion of static and transitional information of cepstral and spectral features for music genre classification," *Asia-Pacific Services Computing Conference, IEEE*, 2008.

[12] S. H. . S. W. . J. L. . B. X. Lei Wang, "Music genre classification based on multiple classifier fusion," *Natural Computation*, 2008.

[13] P. S. B.S. Manjunath, P. S. Thomas SikoraB.S. Manjunath, and T. Sikora, "Introduction to mpeg-7: Multimedia content description interface," 2002.

[14] D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," *Proceedings of ECML-98, 10th European Conference on Machine Learning.*, 1998.

[15] H. P. Cover TM, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, 1967.

[16] H. Ling and K. Okada, "Diffusion distance for histogram comparison," *IEEE Computer Society Conference*, 2006.

[17] R. . B. J. Gibson, S. Harvey, "Multi-dimensional histogram comparison via scale trees," *Image Processing Proceedings*, 2001.

[18] M. S. . V. K. P.-N. Tan, "Introduction to data mining," *Annals of Mathematical Statistics*, 2005.

[19] R. Kullback, S.; Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, 1951.

[20] C. T. Y. Rubner and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, 2000.

[21] P. T. Olivier Lartillot, "A matlab toolbox for musical feature extraction from audio," *International Conference on Digital Audio Effects*, 2007.

[22] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[23] A. R. E. Pampalk and D. Merkl, "Content-based organization and visualization of music archives," *ACM Multimedia*, 2002.

[24] S. D. E. Pampalk and G.Widmer, "On the evaluation of perceptual similarity measures for music," *Music Information Retrieval*, 2003.

[25] T. Li and M. Ogihara, "Toward intelligent music information retrieval," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 564–574, 2006.

[26] M. Mandel and D. Ellis, "Song-level features and svms for music classification," *Int. Conf. Music Information Retrieval*, 2005.

[27] G. Tzanetakis, "Marsyas-0.2: A case study in implementing music information retrieval systems," *Intelligent Mobile Information Systems*.

[28] D. E. D. E. J. Bergstra, N. Casagrande and B. Kegl, "Aggregate features and ada boost for music classification," *Mach. Learn. vol. 65, no. 23, pp. 473484, 2006.*

[29] E. B. I. Panagakis and C. Kotropoulos, "Music genre classification: A multilinear approach," *Int. Conf. Music Information Retrieval*, 2008.

[30] K.-M. Y. C.-H. Lin, J.-L. Shih and H.-S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Trans. Multimedia*, 2009.

[31] C. K. I. Panagakis and G. R. Arce, "Music genre classification using locality preserving non-negative tensor factorization and sparse representations," *Int. Conf. Music Information Retrieval*, 2009.

[32] K. T. Z. Fu, G. Lu and D. Zhang, "On feature combination for music classification," *Int. Workshop Statistical Pattern Recognition*, 2010.