AUDIO GENRE CLASSIFICATION USING PERCUSSIVE PATTERN CLUSTERING COMBINED WITH TIMBRAL FEATURES

Emiru Tsunoo, George Tzanetakis[†]*, Nobutaka Ono, Shigeki Sagayama*

Graduate School of Information Science and Technology, The University of Tokyo, Japan
{tsunoo,onono,sagayama}@hil.t.u-tokyo.ac.jp
[†]Conputer Science Department, University of Victoria, Canada
gtzan@cs.uvic.ca

ABSTRACT

Many musical genres and styles are characterized by distinct representative rhythmic patterns. In most automatic genre classification systems global statistical features based on timbral dynamics such as Mel-Frequency Cepstral Coefficients (MFCC) are utilized but so far rhythmic information has not so effectively been used. In order to extract bar-long unit rhythmic patterns for a music collection we propose a clustering method based on one-pass dynamic programming and k-means clustering. After extracting the fundamental rhythmic patterns for each style/genre a pattern occurrence histogram is calculated and used as a feature vector for supervised learning. Experimental results show that the automatically calculated rhythmic pattern information can be used to effectively classify musical genre/style and improve upon current approaches based on timbral features.

Index Terms— Audio genre classification, Percussive sound, Dynamic programming, Pattern clustering method, Feature extraction

1. INTRODUCTION

Interest in music information retrieval (MIR) has recently surged due to the increasing size of digital music collections available on computers and portable music players. Automatic genre classification from audio is a traditional topic of MIR and provides a structured way of evaluating new representations of musical content. In this task, not only instrumental information but also a rhythmic information is thought to be important. For instance, the distinction between samba and tango exists primarily in their bar-long rhythmic patterns. If such representative unit rhythmic patterns in music can be identified automatically they can potentially be used to automatically characterize different genres and styles of music directly from audio signals.

In previous research, timbral features, rhythmic features and pitch features have been used for audio genre classification[1]. However the rhythmic features were based on overall statistics of periodicities and did not represent directly temporal information. Research more closely related to our are work with rhythmic patterns includes Dixon [2] which extract a periodical pattern from acoustic signals heuristically and Peeters [3] which extracts features based on the periodicity of the spectrum. These approaches can successfully discriminate styles such samba or tango primarily based on rhythmic information.

In this paper, we describe an approach for extracting unit rhythmic patterns of percussive sounds from a number of audio tracks and propose a pattern occurrence histogram as a feature for genre classification. Finally, the effectiveness of the proposed percussive pattern information for genre classification is verified experimentally.

2. RHYTHM PATTERN CLUSTERING

2.1. Challenges in Rhythm Pattern Clustering

Bar-long percussive patterns are frequently common and characteristic of a particular genre or style. Automatically detecting these patterns is a "chicken-and-egg" problem in that sets of bar-long unit rhythm patterns may be determined only after their corresponding unit boundaries in the music pieces are given, and vice versa. This is complicated by tempo fluctuations which might cause the unit pattern to stretch or shrink. An additional problem is that pieces comprise of both harmonic and percussive sounds especially in modern popular music. Harmonic sounds sometimes disturb rhythm analysis based on the spectrogram. In the next subsections, we describe our approach to solving these challenges.

2.2. Emphasizing Percussive Components

Generally, harmonic and percussive sounds are mixed in the observed spectrograms of audio pieces. Therefore, in order to perform percussive pattern analysis it is useful to separate these components. We utilize the method described in Ono [4] that is based on the difference of general timbral features. By looking at the left figure in Fig. 1, a typical instance of spectrogram, one can observe that harmonic components

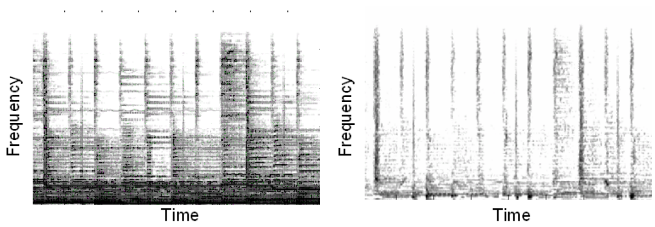

Fig. 1. The original spectrogram (left) and the percussion-emphasized spectrogram (right) of a popular music piece (RWC-MDB-G-2001 No.6[5]).

tend to be continuous along the temporal axis in particular frequencies. On the other hand, percussive components tend to be continuous along the frequency axis and temporally short. Mask functions for separating the two components (harmonic and percussive) are calculated following a Maximum a posteriori (MAP) estimation approach using the Expectation Maximization (EM) algorithm. Applying this algorithm to the shown spectrogram, harmonic and percussive components are separated (percussive components are shown in the right of Fig. 1).

2.3. Iterative Update of Percussive Pattern Cluster and Segmentation

If the true set of unit patterns are given as templates, the problem of unit segmentation is analogous to the problem of continuous speech recognition where the one-pass dynamic programming (DP) algorithm [6] can be used to find the sequence of uttered words. In addition because dynamic programming is flexible in terms of time alignment this also simultaneously deals with the problem of tempo fluctuation during a performance. On the other hand, if the boundaries are given initially, the percussive patterns can be easily clustered by k-means clustering and the set of unit patterns can be calculated. We have proposed an approach to solve these problems iteratively in [7].

First, a set of initial seed templates is provided such as typical percussive spectrogram patterns in modern music. Then, according to the initial seed templates, the alignment is calculated using a one-pass DP algorithm and the optimal segmentation of the input spectral patterns is calculated. Based on the calculated alignment and an approach similar to k-means clustering the input templates are adapted by averaging segments that belong to the same cluster. By iterating these two steps, the total summation of distance cost gradually converges. Fig. 2 illustrates the flow of this algorithm.

Mathematically, the template updating and convergence proof are as follows: Considering a probabilistic model where the output probability of the spectrum $\boldsymbol{r}_x$ with size N from the spectrum of the frame i in template m is defined as:

$$p_{m,i}(\boldsymbol{r}_x) = \frac{1}{(2\pi)^{\frac{N}{2}}|\boldsymbol{\Sigma}_{m,i}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{e}_{m,i,x}^T\boldsymbol{\Sigma}_{m,i}^{-1}\boldsymbol{e}_{m,i,x}\right) \quad (1)$$

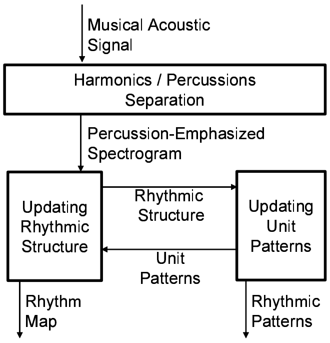

Fig. 2. The flow diagram of the system.

where $\boldsymbol{e}_{m,i,x} = (\boldsymbol{\mu}_{m,i} - \boldsymbol{r}_x)$ and $\boldsymbol{\Sigma}_{m,i}$ is a diagonal covariance matrix of the time i and the template m, logarithmic likelihoods $\ln(p_{m,i}(\boldsymbol{r}_x))$ can be multiplied by some weight w and summed up one after another according to one-pass DP algorithm.

The alignment calculated above gives a correspondence between the spectrum $r_{x(a)}$ of the time index $x(a)$ and the template spectrum $\boldsymbol{\mu}_{m(a),i(a)}$ of the time index $i = i(a)$ in template $m = m(a)$. Therefore, the distance cost in one-pass DP algorithm can be written as

$$D_A = -\frac{1}{2}\Big(\sum_{a=1}^{A}\left(Y\log(2\pi) + \log|\boldsymbol{\Sigma}_{m(a),i(a)}|\right)\cdot w(a) + \sum_{a=1}^{A}\boldsymbol{e}_{m,i,x}(a)^T\boldsymbol{\Sigma}_{m(a),i(a)}^{-1}\boldsymbol{e}_{m,i,x}(a)\cdot w(a)\Big) \quad (2)$$

where $\boldsymbol{e}_{m,i,x}(a) = (\boldsymbol{\mu}_{m(a),i(a)} - \boldsymbol{r}_{x(a)})$.

When the template patterns are updated, the update rules are solved based on the maximum likelihood estimation. The updated template spectrum is written as

$$\hat{\mu}_{m,i} = \frac{\sum_{a\in A_{m,i}}\boldsymbol{r}_{x(a)}\cdot w(a)}{\sum_{a\in A_{m,i}} w(a)} \quad (3)$$

where $A_{m,i} = \{a|m(a) = m, i(a) = i\}$, and the covariance matrix of it is written as

$$\hat{\boldsymbol{\Sigma}}_{m,i} = \frac{\sum_{a\in A_{m,i}}\boldsymbol{e}_{m,i,x}(a)\boldsymbol{e}_{m,i,x}(a)^t\cdot w(a)}{\sum_{a\in A_{m,i}} w(a)}. \quad (4)$$

Thus, the total likelihood calculated after this update, D'_A satisfies

$$D'_A \geq \hat{D}_A = \max_{\mu,\Sigma} D_A \geq D_A \quad (5)$$

and this iterative update never reduces the total likelihood, so the convergence is guaranteed.

When used in the context of genre classification the above algorithm needs to be adapted so that it can be applied to a collection of music pieces rather than a single one. In this case the same set of templates is used for the one-pass DP alignment calculation for all the pieces. In addition all corresponding segments of all music pieces are collected and averaged in the template update phase. That way a certain number of representative percussive templates common to a particular genre or style can be identified.

3. RHYTHM PATTERN FEATURE EXTRACTION

3.1. Genre Classification via Percussive Pattern Clustering

Ideally percussive patterns for a particular genre or style would be fixed and would be automatically extracted perfectly. If that was the case then automatic genre classification could be performed simply by looking at which particular rhythm patterns are used in a music piece. However in practice there is no guarantee that patterns are fixed for a particular genre/style or that their automatic extraction will be perfect. Therefore in many cases the percussive patterns of a particular music piece will belong to more than one genre. To address these problems simultaneously we utilize a pattern occurrence histogram representation followed statistical machine learning to automatically classify music genre. Supervised learning classifiers such as Support Vector Machines (SVM)[8] can be used for this purpose.

One possible way to extract feature vector is count up which percussive pattern templates are contained in a song and calculating the genre pattern occurrence histogram, similarly to Latent Semantic Indexing approach [9].

If M percussive pattern templates are learned from genre g ($g = 1, \ldots, G$), an alignment can be calculated using dynamic programming to calculate the templates $T_{m,g}$ that exist in the song s. Then, the occurrence number of the patterns from genre g can be simply calculated by summation as follows:

$$c_{s,g} = \sum_{m=1}^{M} c_{s,m,g} \tag{6}$$

where $c_{s,m,g}$ is the number of the template $T_{m,g}$ in the song s, and the eventual G dimensional pattern occurrence histogram features vector $\boldsymbol{x}$ of song s can be written as

$$x_g = \frac{c_{s,g}}{N_s} \tag{7}$$

which is normalized by N_s, the number of measure in the song s.

4. PROCEDURAL SUMMARY OF THE ALGORITHM

The overall algorithm can be summarized as follows:

1. Emphasis of percussive components using Ono's method
2. Initial templates are provided
3. The optimal segmentation is calculated using one-pass DP
4. The templates patterns are updated with k-means like clustering
5. Steps 3 and 4 are iterated until convergence
6. The alignment from the learned templates of all genres is calculated using one-pass DP algorithm
7. The pattern occurrence histogram is used as a feature vector characterizing a music pieces
8. Classification into genres is performed using a machine learning technique

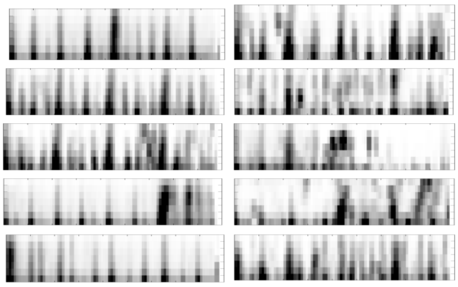

Fig. 3. Example of learned 10 common percussive spectrogram patterns (Blues)

5. EXPERIMENTAL RESULTS

5.1. Dataset

Experiments with the proposed algorithms were conducted on both the GTZAN dataset [1] as well as a dataset of ballroom music [10]. The former had 10 genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock. The latter contains 8 dance styles: chacha, foxtrot, quickstep, rumba, samba, tango, viennesewaltz, and waltz. Both of the datasets have 100 songs per genre all of which are single-channel and sampled at 22.05kHz.

5.2. Template learning and feature extraction

First, common percussive pattern templates were learned using the proposed algorithm for each genre. The proposed algorithm was implemented using the audio processing framework, *Marsyas*[1] which is open source software with specific emphasis on Music Information Retrieval (MIR) [11].

To ensure that the template learning didn't become accidentally good for classification, we divided each dataset into two parts and obtained two sets of templates for each genre. In this experiment, 10 templates were learned from each particular genre or dance style, and the number of iterations was fixed to 30 times because it was enough to converge. The example of learned templates of blues is shown in Fig. 3.

After template learning there are a total of 100 templates for the GTZAN dataset and 80 templates for the ballroom

[1] http://marsyas.sness.net/

dancing dataset. Using the one-pass DP algorithm all segments are labeled and the pattern occurrence histograms from Eq. 7 are calculated. That way 10 dimensional and 8 dimensional feature vectors are obtained in each case.

5.3. Classification results

To train a classifier in feature space, the “Weka” machine learning toolkit [12] was employed. All the results shown are based on 10-fold cross-validation using a linear SVM as a classifier. To generalize the feature extraction part, we divided the datasets into two parts, though, in 10-fold cross-validation to generalize the classification, the labeled data was split into 10 folds and each fold is used once for testing with the remaining 9 folds used for training the classifier to generalize the classification. The results using only the rhythmic pattern features (10 dimensional or 8 dimensional vector) are shown in Table 1. As can be seen the proposed features have enough information for genre classification because this classification accuracy is significantly above the baselines of random classification, which is the naive classification based on the most likely genre/style based on the number of instance.

An existing state-of-the-art genre classification system which uses 68 dimensional timbral features like MFCC and spectrum centroids proposed by Tzanetakis was used for comparison. The system performed well on several audio classification tasks in MIREX 2008 [13]. This system achieved 72.4% on the GTZAN dataset and 57.6% on the ballroom dataset. Songs in the ballroom dataset tend to have similar timbral characteristics and therefore rhythm patterns are more significant for recognizing dance style.

Merging the timbral features and rhythmic features, the classification accuracies shown in Table 2 were obtained. These results are higher than existing genre classification systems that rely on timbral information and verify the effectiveness of the proposed percussive patterns.

6. CONCLUSIONS

We discussed an approach to extracting common percussive patterns for particular genres/styles and using pattern occurrence histograms as features for genre classification. We used Ono’s method to extract percussive components from audio signals and clustered percussive patterns using a combination of one-pass DP and k-means clustering algorithm. Experiments over music pieces from various genres confirmed that the proposed algorithm can improve the accuracy of classification systems based on timbral information.

Future work includes using n-gram model approach rather than only looking at the uni-gram histogram. In addition more experiments with the parameters of the algorithms such as the number of templates to be learned need to be conducted. Other features like bass-line pattern can be used for genre classification as well.

Table 1. Genre classification accuracy using only rhythmic pattern features

Features	GTZAN	Ballroom
Baseline (random classifier))	10.0%	12.5%
Rhythmic (from template set #1)	43.6%	54.0%
Rhythmic (from template set #2)	42.3%	55.1%

Table 2. Genre classification accuracy using merged features with existing timbral features

Features	GTZAN	Ballroom
Existing (Timbre)	72.4%	57.6%
Merged (from template set #1)	76.1%	70.1%
Merged (from template set #2)	76.2%	69.1%

7. REFERENCES

[1] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transaction on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[2] S. Dixon, F. Guyon, and G. Widmer, “Towards characterization of music via rhythmic patterns,” in *Proc. of the 5th Int. Conf. on Music Information Retrieval*, 2004, pp. 509–516.

[3] G. Peeters, “Rhythm classification using spectral rhythm patterns,” in *Proc. of the 6th Int. Conf. on Music Information Retrieval*, September 2005, pp. 644–647.

[4] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, “A real-time equalizer of harmonic and percussive componets in music signals,” in *Proc. of the 9th Int. Conf. on Music Information Retrieval*, September 2008, pp. 139–144.

[5] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “Rwc music database: Music genre database and musical instrument sound database,” in *Proc. of the 4th Int. Conf. on Music Information Retrieval*, October 2003, pp. 229–230.

[6] H. Ney, “The use of a one-stage dynamic programming algorithm for connected word recognition,” in *Int. Conf. on Acoust., Speech, Signal Processing*, 1984, pp. 263–271.

[7] E. Tsunoo, N. Ono, and S. Sagayama, “Rhythm map: Extraction of unit rhythmic patterns and analysis of rhythmic structure from music acoustic signals,” in *Accepted for ICASSP*, 2009.

[8] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[9] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[10] “Ballroomdancers.com,” http://www.ballroomdancers.com/.

[11] G. Tzanetakis, *Marsyas-0.2: A Case Study in Implementing Music Information Retrieval System*, chapter 2, pp. 31 – 49, Idea Group Reference, 2007, Shen, Shepherd, Cui, Liu (eds).

[12] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005.

[13] “Mirex 2008,” http://www.music-ir.org/mirex/2008/.