To appear in IEEE International Conference on Systems, Man and Cybernetics. October 10-13 2004 The Hague, The Netherlands

Automatic Classification of Audio Data*

Carlos H. L. Costa, Jaime D. Valle Jr., Alessandro L. Koerich
Pontifical Catholic University of Paraná
Curitiba, PR, Brazil
alekoe@computer.org

Abstract – *In this paper a novel content–based musical genre classification approach that uses combination of classifiers is proposed. First, musical surface features and beat–related features are extracted from different segments of digital music in MP3 format. Three 15–dimensional feature vectors are extracted from three different parts of a music clip and three different classifiers are trained with such feature vectors. At the classification mode, the outputs provided by the individual classifiers are combined using a majority vote rule. Experimental results show that the proposed approach that combines the output of the classifiers achieves higher correct musical genre classification rate than using single feature vectors and single classifiers.*

Keywords: Audio classification, musical genre classification, information fusion, classifier combination.

1 Introduction

The amount of multimedia now available on–line has created a surge for efficient tools to organize and manage such a huge amount of data [3, 7]. At present, multimedia data is usually classified based on textual meta–information. While such information is very useful for indexing, sorting, comparing and retrieval, it is manually generated. Extracting the information through an automatic and systematic process might overcome such problems.

Digital music is one of the most important data types distributed in the web. How to effectively organize and process such large variety and quantity of musical data to allow efficient indexing, searching and retrieval is a real challenge [1, 4, 5]. There have been many studies on audio content analysis using different features and different methods [5, 7, 16]. In spite of many research efforts, high accuracy audio classification is only achieved for relative simple problems such as speech/music discrimination [2]. Other works attempt to classify audio records into speech, silence, laughter and non–speech sounds. Relatively few works have dealt with musical genre classification [6, 14, 15]. Most of such works have focused on relatively few classes of very distinct musical genres. Furthermore, most of the works have used non–parametric classification strategies and have dealt with small databases.

Musical genre is an important description that has been used to classify and characterize digital music and to organize the large collections available on the web [14, 15]. Genre hierarchies are commonly used to structure the large collections of music available on the web. Furthermore, music genre might be very useful for music indexing and content–based music retrieval. Musical genres are categorical labels created by humans to characterize music clips. These characteristics are related to the instrumentation, rhythmic structure, and harmonic content of the music. However, music genre is a relatively fuzzy concept and even the music industry is sometimes contradicting in assigning genres to music clips. A very common practice is that music clips are categorized according to the artist profile. Furthermore, musical genre annotation is performed manually. In such a way, automatic musical genre classification can assist or replace the human user in this process as well as provide an important component for a complete music information retrieval system for audio signals.

It is extremely more difficult to discriminate musical genres than discriminate music, speech and other sounds. Soltau et al [11] classified music into rock, pop, techno and classic using hidden Markov models and explicit time modeling with neural networks to extract the temporal structure from the sequence of cepstral coefficients. Pye [8] used Mel–frequency cepstral coefficients and Gaussian mixture model to classify music into six types: blues, easy listening, classic, opera, dance, and rock. Tzanetakis and Cook [14] explored features related to the timbral texture, rhythm and pitch. Gaussian mixture model and k–nearest neighbor classifiers were used to classify the extracted features. Shao et al [10] used an unsupervised classification approach based on hidden Markov models. All these works use single classifiers and deal with single feature vectors extracted from the music clips.

In this paper we propose a novel approach for content–based musical genre classification based on the combination of classifiers [12]. In such a way musical surface features and beat–related features are extracted from three different regions of digital music in MP3 format. Musical surface fea-

*0-7803-8566-7/04/$20.00 © 2004 IEEE.

tures include spectral centroid, flux, zerocrossing rate and low–energy. Beat–related features include relative amplitude and beats per minute, etc. These features form 15–dimensional feature vectors which are used to train different classifiers in a supervised approach. Two different classification approaches were investigated: k–nearest neighbor (k–NN) and multilayer perceptron neural network (MLP). At classification mode, the feature vectors extracted from the three different regions of a music clip are classified by single classifiers and the output of these classifiers are combined by a majority voting rule.

This paper is organized as follows. Section 2 presents an overview of the proposed approach for content–based audio data classification. The two types of features extracted from the music clips, that is, musical surface features and beat–related features are described in Section 3. Section 4 presents the details of the classifiers and combination of their outputs to improve the classification performance. The experimental results of the proposed approach on a dataset of 414 music pieces are presented in Section 5. In the last section some concluding remarks are presented.

2 System Overview

The classification system is composed by three main stages as show in Figure 1: feature extraction, classification and combination and decision. At the first stage, feature extraction is carried out from three selected regions of the music clip. From each regions, a 15–dimensional feature vector is generated. Different from previous works, in the proposed approach, three feature vector are extracted from a single music clip. Further, the system operates into two modes: training and testing. In the training mode, the feature vectors together with their labels are used by a learning algorithm to train the classifiers. In this case, the label consists of the musical genre assigned to the audio by a human through subjective evaluation based on his/her hearing perception.

At the classification mode, a music whose genre is unknown, is submitted to the system. From such a music clip are extracted three feature vectors from the corresponding regions which feed the classifiers. Each classifier provides at the output a class (i.e. musical genre) and a confidence score. The output of the classifiers are combined through a majority voting rule to decide the final class to be assigned to the input music clip. Only the class is used in the combination by the majority voting. In the next sections the features, the classifiers and the combination and decision procedure are presented in details.

3 Feature Extraction

In our work we have considered the problem of content–based musical genre classification as a pattern classification problem. In such a way, the methodology that has been developed to tackle such a problem extracts relevant features from music clips.

Feature extraction is the process of representing a segment of audio by a compact but descriptive vector. The choice of such features is one of the main challenges in building pattern recognition systems. Once the features are extracted, several machine learning techniques can be used to manipulated such a vectors.

Since digitized music in good sound quality has an 1MB/minute rate, it would be very time consuming to extract the feature vector from the whole music. In such a way feature extraction is carried out only on segments of the music clip. Three segments are chosen according to the duration and bit rate of the music. The constraints in this choice is that one segment must be from the starting region of the music, another from the middle region and the last segment from the end region of the music. Figure 2 illustrates the feature extraction process.

The feature set used in this paper was originally proposed by Tzanetakis et al [14] and used in other works [6]. We consider two different types of features: musical surface features and beat–related features. Musical surface features include the mean and average of the spectral centroid, flux, zero–crossing rate, and low energy. Beat–related features include relative amplitudes and beats per minute. These features form 15–dimensional feature vectors which are used to train different classifiers in a supervised approach. In this section these features are briefly described.

3.1 Timbral Texture Features

The feature presented in this section are based on the short time Fourier transform (STFT) and are calculated for every short–time frame of sound [9].

Spectral Centroid is defined as the center of gravity of the magnitude spectrum of the STFT and it is computed as

$$C_t = \frac{\sum_{n=1}^{N} nM_t(n)}{\sum_{n=1}^{N} M_t(n)} \tag{1}$$

where $Mt(n)$ is the magnitude of the Fourier transform at frame t and frequency bin n.

Spectral Rolloff is another measure of spectral shape which is defined as the frequency R_t below which 80% of the magnitude distribution is concentrated. It is computed as

$$\sum_{n=1}^{R_t} M_t(n) = 0.8 \sum_{n=1}^{N} M_t(n) \tag{2}$$

Spectral Flux is a measure of local spectral change and it is computed as

$$F_t = \sum_{n=1}^{N} (N_t(n) - N_{t-1}(n))^2 \tag{3}$$

where $N_t(n)$ is the normalized magnitude of the Fourier transform at window t.

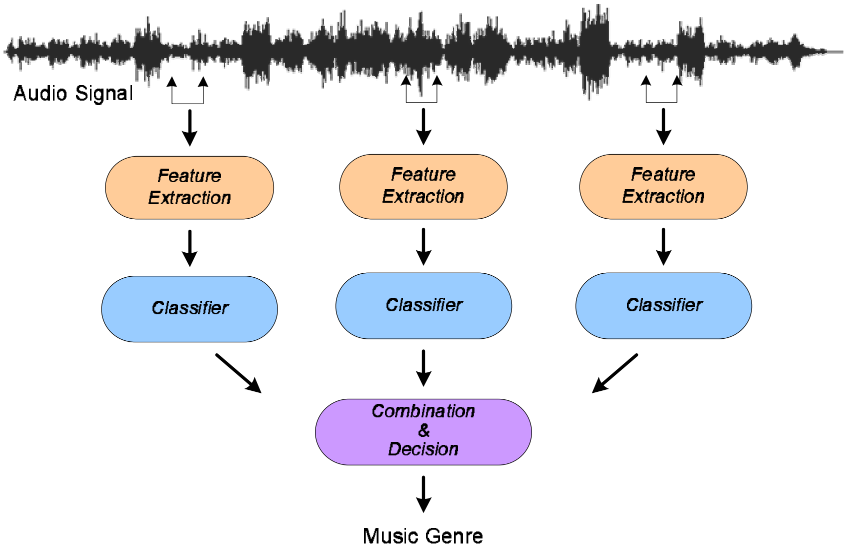

Figure 1: An overview of the proposed musical genre classification approach

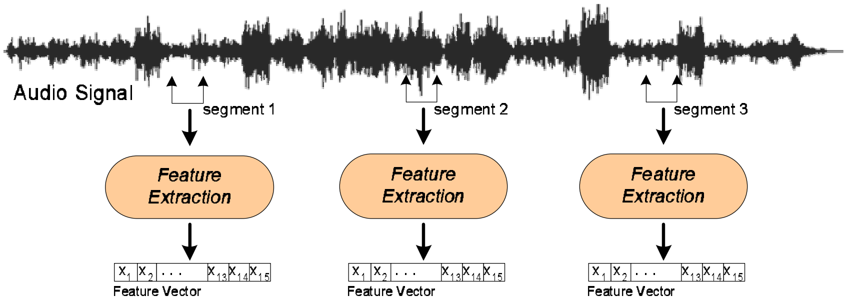

Figure 2: Features are extracted from different regions of the music clip

Time Domain Zero–Crossings is a time–domain feature and occurs when successive samples in a digital signal have different signs. It is computed as

$$Z_t = \frac{1}{2} \sum_{n=1}^{N} |sign(x(n)) - sign(x(n-1))| \qquad (4)$$

where $x(n)$ is the time domain signal, and the $sign$ function is 1 or 0 for positive and negative arguments respectively.

The notion of musical surface is created by a series of spectral states that vary slightly over a short period of time. To capture this information mean and variance of the features listed above are calculated over a number of windows.

Low Energy is defined as the percentage of windows that have less energy than the average energy of all windows. Music that contains silent parts will have a larger low energy value than continuous sounds.

3.2 Beat–Related Features

The beat and rhythmic structure of a song is often a good indication of the genre. The beat feature extractor tries to find the main beat of the song and its period in BPM (beats–per–minute), the second strongest beat, and a number of features concerning the relationship between the first and second beat.

The signal is decomposed into a number of frequency bands using the discrete wavelet transform [13]. After this decomposition, a series of steps for the extraction of the time domain amplitude envelope is applied to each band. These steps are full wave rectification, low pass filtering, downsampling, and mean removal [6, 14].

After the envelope extraction step, the envelopes of each band are summed and the autocorrelation of the resulting envelope is calculated. The result is an autocorrelation function where the dominant peaks correspond to the time lags where the signal has the strongest self–similarity. The first three peaks of the autocorrelation function are added to a beat histogram. Each bin in the histogram corresponds to a beat period in BPM. For each of the three selected peaks, the peak amplitude is added to the histogram. This is repeated for each analysis window. The strongest peaks in the final histogram correspond to the strongest beats in the signal. Six features are calculated from the beat histogram:

- The relative amplitude (i.e. the amplitude divided by the sum of amplitudes) of the first and second peak in the beat histogram. This is a measurement of how distinctive the beat is compared to the rest of the signal.

- The ratio of the amplitude of the second peak divided by the amplitude of the first peak. It expresses the relation between the main beat and the first sub beat

- The period of the first and second peak in BPM, indicating how fast the song is.

- The sum of the histogram, which is an indication of beat strength. The sum of the histogram bins is a measure of the strength of self–similarity between the beats, which in turn is a factor in how rhythmic a song feels.

3.3 Feature Vector

The features proposed before are concatenated to form a 15–dimensional feature vector where nine features (mean and variances of spectral centroid, rolloff, flux, and zero-crossing and low–energy) are related to the music texture and the other six are related to the music rhythm.

4 Classification

The basic problem in musical genre classification is given a music clip to classify represented by a feature vector $x_1^D = (x_1 x_2 \ldots x_D)$ where D is the dimension of the vector, assign a class, i.e. a musical genre $g \in \mathcal{G}$ that best matches to the input vector. For such an aim, we use two different approaches: instance based classification and connexionist approach. These approaches are presented in the next sections.

4.1 Instance Based Classification

Learning in instance based classification algorithms consists of simply storing the presented training data. When a new query instance is encountered, a set of similar related instances is retrieved from memory and used to classify the new query. The most basic instance–based method is the k–nearest neighbor algorithm. This algorithm assumes all instances correspond to points in the *n*–dimensional space $\Re^n$. The nearest neighbors of an instance are defined in terms of the standard Euclidean distance. The distance between two vectors x_i and x_j is denoted as $d(x_i, x_j)$ where

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} [(a_k(x_i) - a_k(x_j)]^2} \qquad (5)$$

To classify the music clips according to the musical genre, first a set of training samples is processed and the corresponding feature vectors are stored in the memory. Another set of feature vectors is used to assess the performance of the classifier. For each testing feature vector, the Euclidean distance to all training feature vectors is computed and the k nearest training feature vectors are selected. The genre of the majority of the selected training feature vectors is assigned to the testing feature vector.

4.2 Neural Network Classifier

We have designed a simple classifier based on a multilayer perceptron (MLP) with one hidden layer. The choice of such a classifier to perform the musical genre classification task is determined by some constraints such as: estimation of a posteriori probabilities at the output and classification speed.

To build an MLP classifier basically we have to determine the number of layers and the number of neurons in each layer. The MLP classifier has 15 neurons in the input layer, 8 neurons in the hidden layer and 2 neurons in the output layer. The number of hidden neurons was determined by a rule of thumb and some exploratory experiments where the error rates on the training and validation sets were used as criteria.

The network was trained using the backpropagation momentum algorithm. The network output estimates a posteriori probabilities and the value of each output necessary remains between zero and one because of the sigmoidal function used.

4.3 Combining Classifiers and Decision

Given that the three feature vector are extracted from the same music clip, the output of the classifiers that take at the input each feature vector can be combined to optimize the classification performance. Several simple combination rules could be employed to combine the output of the classifiers. However, is this paper we have considered only the class provided by each classifier, neglecting the confidence score associated with each class that the classifiers also provide. For such an aim, the majority voting scheme was used.

We have not considered the possibility of rejection because the genres of all music clips used in the experiments are in G. However, this aspect can be easily incorporated in the proposed approach and it will be the subject of future research.

5 Experimental Results

A music collection of more than 1,000 music clips with a total play length of about 50 hours was available for the experiments. This collection contains pieces of almost 40 different musical genres. However, the frequency of some musical genres in the database is too low. The music clips were manually labeled and the genre was assigned according to the profile of the artist or according to the perceptual characteristics evaluated by human beings.

The dataset used in the experiments is composed by 414 music clips where half of them are from the genre rock and half are from the genre classic. From whole dataset, 208 music clips were randomly selected to form the training set. The validation set is composed by 82 samples and the remaining 122 samples form the test set. Three feature vectors were extracted from each music clip as described in Section

3. Therefore, in the experiments, 1,242 feature vectors were used.

Two different experiments were carried out. In the first experiment a single feature vector is extracted from the middlemost region of the music clips. The classifiers were trained using 208 feature vectors and tested using 122 feature vectors. A validation set with 82 feature vectors was used during the training of the MLP to look over the generalization and to avoid overfitting. The correct musical genre classification rates for the MLP classifier and for the k–NN classifier where, $k = (1, 3, 5, 7)$ are shown in Table 1. The correct musical genre classification rate is defined as the number of music clips for which the genre was correctly assigned by the number of music clips tested.

Table 1: Correct musical genre classification rates for single feature vectors and single classifiers

Dataset	Correct Classification Rate (%)				
	1–NN	3–NN	5–NN	7–NN	MLP
Training	—	—	—	—	92.1
Validation	84.0	87.5	90.0	90.0	90.7
Test	83.0	85.5	84.0	83.0	89.5

Table 1 shows that the correct classification rate achieved by the MLP classifier is about 5% better than the classification rates achieved by k–NN classifier for different values of k.

The second experiment considers three feature vectors extracted from three different regions of the music clips: beginning, middle and end (See Figure 2). The classifiers were trained using 624 feature vectors and tested using 366 feature vectors. A validation with 246 feature vectors was also used during the training of the MLP to look over the generalization and to avoid overfitting. The correct musical genre classification rate for the MLP classifier is shown in Table 2. Table 3 shows the correct musical genre classification rate for k–NN classifier where $k = (1, 3, 5, 7)$. In both tables, *Segment 1*, *Segment 2*, and *Segment 3* refers to the results for the feature vectors extracted from the three regions of the music clip (Figure 2).

Table 2: Correct musical genre classification rates for the MLP classifier considering three feature vectors for each music clip

	Correct Classification Rate (%)		
Dataset	Segment 1	Segment 2	Segment 3
Training	90.1	92.1	92.2
Validation	86.3	90.7	88.9
Test	85.5	89.5	88.7

In Tables 2 and 3 it is clear the difference in correct musical genre classification rates achieved on the feature vectors extracted from different segments of a single music clip. Higher correct classification rates were achieved for the feature vectors extracted from the middlemost part (segment 2) of the audio clips. Another common behavior is that the worst correct classification rates were achieved on the feature vectors extracted from the first segment of the music clips. For the MLP classifier, this difference is about 4% while for the k–NN classifier the difference is about 8%. In fact, such a variability reflects problems of robustness in the proposed approach, possibly due to the feature set. It is expected a uniform performance, that is, similar correct classification rates for features extracted from different regions of the music clips.

In an attempt to alleviate this problem and to optimize the performance of the proposed classification approach, combination of classifiers is introduced. As described in Section 4, the unweighted majority voting rule is considered. For instance, if the first two segments of a music clip are classified as rock and the third segment is classified as classical, the majority wins and the genre rock is assigned to such a music clip. Table 4 shows the correct musical genre classification rates obtained by combining the outputs of the single MLP classifiers and the results obtained by combining the outputs of the k–NN classifiers.

Table 4: Correct musical genre classification rates for the combination of classifiers output using the majority voting rule

Dataset	Correct Classification Rate (%)				
	1-NN	3-NN	5-NN	7-NN	MLP
Validation	83.3	89.1	84.2	79.9	91.3
Test	82.3	86.3	83.1	81.5	90.3

The improvements in the correct musical genre classification obtained by combining the outputs of the classifiers are moderated. Table 4 shows some improvement relative to the performance of single classifiers shown in Tables 2 and 3. For the combination of the MLP classifiers, the improvement in the correct classification is about 0.8% over the performance of the best single MLP classifier. For the k–NN classifier, the improvements achieved are inconsistent. For $k = 1, 5, 7$ the observed correct classification rate was worst than the performance achieved by the best single k–NN. The performance was slightly better only for $k = 3$.

6 Concluding Remarks

Automatic musical genre classification is a difficult pattern recognition task. In this paper we have presented a novel approach to musical genre classification that combines three feature vectors extracted from different regions of music clips. The feature vectors are combined at classification level through the combination of the outputs of single classifiers. A slight improvement in the correct musical genre classification was achieved. However, the combination rule used is very elementar. Future work will include other combination strategies that take into account the confidence scores provided by the classifiers as well as a rejection mechanism to further improve the reliability of the system.

Table 3: Correct musical genre classification rates for the k–NN classifier considering three feature vectors for each music clip

Dataset	Correct Classification Rate (%)											
	Segment 1				Segment 2				Segment 3			
	1	3	5	7	1	3	5	7	1	3	5	7
Validation	85.5	91.5	92.5	90.5	84.0	87.5	90.0	90.0	80.5	85.0	88.0	90.5
Test	78.0	79.0	76.0	74.0	83.0	85.5	84.0	83.0	79.0	84.5	82.5	80.5

The results achieved by the proposed approach are similar to some recent results from the literature [6, 14]. However, it should be stressed that these studies have used different datasets and experimental conditions, which makes a direct comparison very difficult.

Acknowledgments

The authors would like to acknowledge Ms. K. Kosina for sharing the source code for feature extraction.

References

[1] W. Birmingham, R. B. Dannenberg, G. H. Wakefield, M. Bartsch, D. Bykowski, D. Mazzoni, C. Meek, M. Mellody, and W. Rand. Musart: Music retrieval via aural queries. In *International Symposium on Music Information Retrieval*, pages 73–81, 2001.

[2] M. J. Carey, E. S. Parris, and H. Lloyd-Thomas. A comparison of features for speech, music discrimination. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 149–152, 1999.

[3] M. Fingerhut. The ircam multimedia library: A digital music library. In *IEEE Forum on Research and Technology Advances in Digital Libraries*, pages 19–21, 1999.

[4] J. T. A. Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):42–51, 1999.

[5] G. Guo and S. Z. Li. Content–based audio classification and retrieval by support vector machines. *IEEE Transactions on Neural Networks*, 14(1):209–215, 2003.

[6] K. Kosina. Music genre recognition. Technical report, Fachlochschul Hagenberg, 2002.

[7] E. Pampalk, A. Rauber, and D. Merkl. Content–based organization and visualization of music archives. In *ACM International Conference on Multimedia*, 2002.

[8] D. Pye. Content–based methods for the manegement of digital music. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2437–2440, 2000.

[9] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[10] X. Shao, C. Xu, and M. S. Kankanhalli. Applying neural network on the content–based audio classification. In *Fourth International Conference on Information, Communications and Signal Processing*, volume 3, pages 1821–1825, 2003.

[11] H. Soltau, T. Schultz, and M. Westphal. Recognition of music types. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1137–1140, 1998.

[12] C. Y. Suen and L. Lam. Multiple classifier combination methodologies for different output levels. In *Proc. 1st International Workshop on Multiple Classifier Systems*, pages 52–66, Cagliari, Italy, 2000.

[13] W. Sweldens and R. Piessens. Wavelet sampling techniques. In *Proc. of the Statistical Computing Section*, pages 20–29, 1993.

[14] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[15] C. Xu, N. Maddage, X. Shao, F. Cao, and Q.Tian. Musical genre classification using support vector machines. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 429–432, 2003.

[16] T. Zhang and C. C. J. Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on Speech and Audio Processing*, 9(4):441–457, 2001.