Music Genre Recognition and Classification

Milos Djuric[1] and Milena Stankovic[2]

Abstract – **This work describes a system for the automatic recognition and classification of music according to genres, based exclusively on audio content of the signal. Described system is based on sound characteristics that should have influence on human spontaneous and natural ability to classify music into genres. In order to examine the needed characteristics, every melody is divided into segments which are analyzed individually and final result for each characteristic is presented as arithmetic mean or variance of all proper segment values. A Support Vector Machine classifier is chosen for training and classification, reaching a classification accuracy of between 85 % - 98 % for the three test genres: Heavy Metal, Techno Dance and Classical music.**

Keywords – **Music genre recognition, Music genre classification, Support Vector Machine method.**

I. Introduction

The amount of multimedia now available online has created an impact for efficient tools to organize, search and manage this huge amount of data [1, 2]. At present, multimedia data is usually classified based on textual meta–information. Although such kind of information is very useful for indexing, sorting, comparing and retrieval, the problem is that it is manually generated and, as so, the process is more expensive and very likely arbitrary and subjective. Extracting the information through an automatic and systematic process might overcome most of these problems. So, a challenge for the pattern recognition community is to develop intelligent algorithms for searching and indexing what in recent years became a huge amount of data. While there has obviously been a great deal of work around speech recognition and music processing, in this work, we will focus on the music genre recognition.

Distinguishing between music genres is a trivial task for human beings. A few seconds of music is usually sufficient to allow us to do a rough classification, such as identifying a song as punk or rap, or rock or classical music. The question that this paper attempts to address is whether it is also possible for a machine to make such a classification. The present work investigates ways to automatically classify music files according to genre, based exclusively on the audio content of the files.

Machine can “experience” music only in a digital format, and for it, it is nothing but a sequence of bits whose values correspond to the sound-pressure levels in an analogue acoustic waveform. These bits are for instance easily interpreted by a machine to find out certain facts, such as the overall amplitude of the signal at a given time, which is, of course, impossible to the humans. But understanding music, like humans do it all the time without effort, is far more complex matter. The recognition of music genres undertakes these advanced tasks.

II. The Approach

A. Main Idea

For this discussion, a music genre can be considered as a specific class of music with a set of common properties that in the perception of the average listener distinguish music in that category from other songs.

Human listeners have remarkable music genre recognition abilities. This was shown in a study conducted by R.O. Gjerdigen and D. Perrot [3]. They used ten different genres and eight sample songs for each genre. Five excerpts with different durations were taken from each song. The subjects of the study who did not have any higher-level knowledge in musical theories, were presented with the short excerpts and asked to decide on one of the ten genres for each excerpt. The accuracy of genre prediction for the longest samples was around 70%, compared to the CD companies’ original classification.

Taking into account that music genres are a fuzzy concept, and that even the music industry is sometimes contradictory in assigning genres, this percentage is, according to [3] unexpectedly high. The results of the study are especially interesting, since they show that it may be possible to accurately recognize music genres without using any higher-level abstractions. So, the basic assumption of this paper is that some form of classification is possible based on spectral and timbral characteristics alone, because music samples used in [3] are much too short for recognizing the rhythm, melody or conceptual structure of a song.

In order to increase classification success of 70 % for the longest samples, we divided each signal into windows and so we analyzed every part of the song and calculated arithmetical mean and variance of all feature values for every sound signal (song). In this way the possibility that the part of the song that is atypical for genre is chosen is smaller then it was the case in [3].

B. Features used for recognition

For characterization of music songs a feature set originally proposed by Tzanetakis [4] is used. This feature set in combination with other types of features (e.g. from cepstral analysis) is also used in many other works from related area, e.g. musical instrument recognition [5].

Those features are based on the short time Fourier transform (STFT) and are calculated for every short–time

[1] Milos Djuric is with the Metropolitan University, Vozd Karadjordje 47, 18000 Nis, Serbia, E-mail: djura042@gmaill.com.

[2] Milena Stankovic is with the University of Nis, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: milena.stankovic@elfak.ni.ac.rs

frame of sound [6]. The basic ideas and knowledge for creating MatLab algorithms for calculating values of the following features are extracted from [7] and [8].

Spectral Centroid is the balancing point of the spectrum. It is a measure of spectral shape, and is often associated with the notion of spectral brightness, or in this case – sound brightness. The spectral centroid can be calculated as:

$$C = \frac{\sum_{n=1}^{N} M_t[n]\cdot n}{\sum_{n=1}^{N} M_t[n]} \tag{1}$$

where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n.

Spectral Flux is a measure of local spectral change, and it is defined as:

$$F_t = (N_t[n] - N_{t-1}[n])^2 \tag{2}$$

where $N_t[n]$ is the normalized magnitude of the Fourier transform at window t. In this particular case spectral flux could be considered as a flow rate of musical events in one song (sound variations).

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

$$\sum_{n=1}^{R_t} M_t[n] = 0.8\sum_{n=1}^{N} M_t[n] \tag{3}$$

Time Domain Zero–Crossings occurs when successive samples in a digital signal have different signs. The zero-crossing rate is a simple measure of the noisiness of a signal. It can be calculated as

$$Z_t = \sum_{n=1}^{N} | s(x[n]) - s(x[n-1]) | \tag{4}$$

where $x(n)$ is the time domain signal, and the s function has value 1 or 0 for positive and negative arguments respectively. Unlike spectral centroid, rolloff and flux, which are frequency-domain features, the zero-crossing rate is a time-domain feature.

There is one additional feature, called **Low Energy**. It 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.

C. Forming Feature Vector

The features proposed before are concatenated to form a 9–dimensional feature vector. Eight features present mean values and variances of *spectral centroid*, *rolloff*, *flux* and *zerocrossing*, that are calculated from all windows, and there is also an additional *low–energy* feature. So in this case, each signal is divided in 4.5 *s* windows for the sample rate of 44.100 kHz and by accounting all of them there is a good possibility that this model will have better results than in the case of Gjerdigen–Perrot study.

Every song, wheter it has a role in training or in testing part of the process, is presented with one feature vector.

III. Classification Method

The basic problem in musical genre classification is to assign a class, i.e. a musical genre $g \in G$, that best matches to the input vector representing one music clip $X_D = (x_1\ x_2\ \ldots\ x_D)$ where D is the dimension of the vector. For such an aim, we use method called Support Vector Machine (SVM). SVM is a set of related supervised learning methods invented by Vladimir Vapnik that analyze data and recognize patterns, used for classification and regression analysis.

SVM performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories. So the goal of SVM modeling is to find the optimal hyperplane that in idealized example separates clusters of vectors in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the support vectors. As the Fig.1. below depicts, the optimal hyperplane is oriented in a way that maximal margin between the clusters is present.

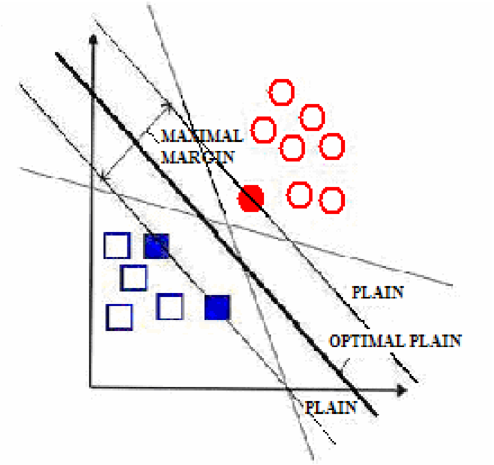

Fig. 1 Optimal plain that separates two clusters of vectors in SVM - method

In cases like the one we are analyzing there are more than two categories (three music genres). For resolving this kind of classification several approaches have been suggested, but two are the most popular. (1) "one against many" where a data point would be classified under a certain class if and only if that class's SVM accepted it and all other classes' SVMs rejected it (the classifier with the highest output function assigns the class). (2) "one against one" where the classification is done by max-wins voting strategy, in which every classifier assigns the instance to one of the two classes when the vote for the assigned class is increased by one vote, and the class with the most votes finally determines the instance classification. While accurate for tightly clustered classes (as is not the case in problem analyzed in this paper), method (1) leaves regions of the feature space undecided where more than one class accepts or all classes reject. For this reason approach used in this case is “one against one”.

For more detailed information about the Support Vector Machine method see [9].

IV. System and Experiment Overview

A. System Overview

System described in this paper consists of two parts or subsystems. The first one is the *extractor* that transforms signal on the entry if necessarily and extracts features crucial for the particular problem, and is denoted as *E* on Fig. 2. Second one is the *classifier* and it classifies data given from the output of the extractor, and is denoted as *C*. So, the two steps can be clearly separated: The output of the feature extraction step is the input for the classification step. We can substitute this subsystem into the black box introduced above, resulting in the model shown in Fig. 2. This is the basic music genre recognition system in its most simple representation.

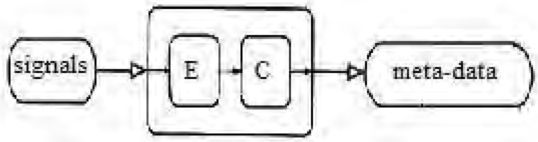

Fig. 2 Global view of the music genre recognition system

B. Experiment Procedure

After forming feature vectors the system operates into two modes: training and testing. For the implementation of SVM-method, MatLab LSSVMLab1.6 library is used, and feature vectors are imported through standard *.xls* document.

In the training mode, the feature vectors are used by a learning algorithm to train the classifier. For this purpose 80 songs from each of three genres are used.

Because of the fact that the problem of music genre recognition is fuzzy and is often matter of subjective arbitration, it is important to underline that modality of forming the training kernel is of the crucial importance in this experiment. This means that 240 training vectors should present the songs that are “very typical” for what today is called classical, metal and dance music, i.e. classical music often has relatively small spectral flux values (less local spectral changes) compared to the dance music, and relatively small spectral centroid values (less brightness) than heavy metal, or high values of zero-crossing-rate feature (simple measure of noisiness of the music) should be expected in music genres that could be described as “noisy”.

At the classification mode, a music whose genre is unknown is submitted to the system. From such a music clip feature vector is extracted and for this purpose 100 songs are used, i.e. 30 for heavy metal, 40 for classical music and 30 for techno dance (this songs were chosen randomly). The results of this classification follow in next section.

V. Test Results

Special attention was paid to the necessity of variance in the selection of test music. For instance, the Metal collection contains songs from the Trash Metal, Speed Metal, Classic Metal, Hard Rock, Gothic Metal, Black Metal, Punk, Grunge, and Death Metal categories. The same is true for the Dance genre, which covers a spectrum from Eurodance to Rave. Test music data for the classical genre is somewhat more limited. This is due to the fact that classical music is based on different concepts than popular music. It requires the listener to actively pay attention in order to fully appreciate a piece. So, the limitation for the case of the classical music mentioned above, refers to the fact that both test and training classical music sets mostly consist of the compositions made by three different and, in musical terms, especial classic composers: Strauss (orchestral music and his famous waltzes were used only), Bach (only his concertos were used), and Vivaldi (also with his concertos and Four Seasons). Almost 80 % of both training and test sets consist of Strauss’s, Bach’s and Vivaldi’s compositions, and the rest of the sets are filled with compositions that, according to author’s opinion, belong to the same classical music subgenres. This does not seem to jeopardize scientific approach to the problem because, as mentioned above (I and II. *A*), the classification of the music into genres is a fuzzy concept that often generates a lot of uncertainty and requires arbitrary solutions. So, it is clear that in this case classical music set is reduced to three subsets (subgenres), and it does not fully represent current understanding of what term “classical music” really means.

The results obtained from recognition and classification process presented in this paper are given in Table 1.

Table 1.

Examination of training and classification of music genres.

	Classical	Dance	Metal
Training vectors	80	80	80
Test vectors	40	30	30
Correct hits	39	25	28
Misses	1	5	2
Classification accuracy	97.5%	83.3%	93.3%

As can be seen from the Table 1. (and as noted in IV.*B*), 240 training vectors were used, 80 for each of three genres. For the testing process 100 vectors were used (40 – 30 – 30, for classical, dance and metal music, respectively).

Subgenres are, in classical music, mostly based on certain composer, particular epoch and sometimes are totally arbitrary. It should be stated that without dividing the classical music set on subgenres in both training and testing sets, classification results had 10 % to 20 % lower values.

Achieved classification accuracy (Table 1.) is 97.5% for classical music, 83.3% for dance music, and 93.3% for metal music. These results are in rank with other classifiers that operate with similar problems [4].

VI. Conclusion

Automatic musical genre classification is a difficult pattern recognition task. In this paper we have presented an approach to musical genre classification that combines Support Vector Machine Method with basic assumption that music can be successfully classified into genres without involving complicated music theory and by proposed set of nine stated features.

The results achieved by the proposed approach are similar to some results from the literature [4]. However, it should be stressed that these studies have used different datasets and experimental conditions, which makes a direct comparison very difficult.

Future work should include other combinations of strategies that include inserting some different kind of features, e.g. rhythm features.

Also, in this case only tree very typical music genres are considered. It will be interesting to apply similar approach for classification of music songs from larger number of genres. In this case, sound recognition of some specific musical instruments could also be helpful in achieving better results.

Acknowledgement

The work presented here was supported by the Serbian Ministry of Education and Science (project III44006).

References

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

[2] 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.

[3] D. Perrot and R. O. Gjerdigen. Scanning the dial: An exploration of factors in the identification of musical style. In Proceedings of the 1999 Society for Music Perception and Cognition, 1999. S. Haykin, *Neural Networks,* New York, IEEE Press, 1994.

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

[5] J. A. Charles, D. Fitzgerald, E. Coyle, "Violin Timbre Space Features", *IEE Irish Signals and Systems Conference, Dublin, 2006.*

[6] L. Rabiner and B. H. Juang, “Fundamentals of Speech Recognition”, Prentice Hall, 1993.

[7] S. Theodoridis and K. Koutroumbas, “Pattern Recognition – fourth edition”, 2009, Elsevier Inc.

[8] Richard G. Lyons, “Understanding Digital Signal Processing”, Prentice Hall PTR, 2004.

[9] Corinna Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning, 20, 1995.*