|  |
| --- |
| **Music Genre Recognition Using Weighted Majority Algorithm** |

**Jivjot SinghMangeshBhangare**

Department of Computer Science Department of Computer Science

Simon Fraser University Simon Fraser University

[*jivjots@sfu.ca*](mailto:jivjots@sfu.ca)[*mbhangar@sfu.ca*](mailto:mbhangar@sfu.ca)

**Anuj Issar Bikramdeep Singh**

Department of Computer Science Department of Computer Science

Simon Fraser University Simon Fraser University

[*aissar@sfu.ca*](mailto:aissar@sfu.ca)[*bsa61@sfu.ca*](mailto:bsa61@sfu.ca)

**Abstract**

This project presents an approach for the music genre classification problem. The proposed approach uses temporal feature vector and weighted voting to improve the prediction accuracy. Classical machine learning algorithms such as Naïve-Bayes, k Nearest-Neighbors, and Support Vector Machines are employed and weighted voting procedures were employed in order to enhance final prediction results. Experiments were carried out on a dataset obtained from Music Analysis, Retrieval and Synthesis for Audio Signals (MARSYAS) which is an open source software framework and data collection of audio files. The dataset contains 1,000 audio files categorized in 10 musical genres. Experimental results show that the proposed ensemble approach produces better results than the ones obtained from individual classifiers.

**1. Introduction**

Duke Ellington who once very wisely said: There are simply two kinds of music, good music and the other kind. The only yardstick by which the result should be judged is simply that of how it sounds. If it sounds good, it's successful; if it doesn't it has failed. Exploring different genres of music is about wanting to know a little bit more about some of the things we humans can do, the feelings we didn't want to leave unsaid; the messages we’ve wanted to get across. Today with the amount of music we have, automatic procedures capable of dealing with large amounts of music in digital formats are imperative, and Music Information Retrieval (MIR) has become an important research area. An important task in MIR is Music Genre Classification problem, music genres are categorical labels created by experts in order to identify the style of the music. The music genre is a descriptor that is largely used to organize collections of digital music. It is not only a crucial metadata in large music databases and electronic music distribution (EMD).

The music can be considered as a high-dimensional digital time-variant signal and considering the amount of music data we have today; it is a good opportunity to automate music genre classification using temporal feature vectors. The approach involves classical machine algorithms such as Naïve-Bayes, k Nearest-Neighbors, and Support Vector Machines and using weighted voting procedures to improve final prediction results. The Naïve Bayes is implemented with 10-fold cross validation using ‘e1071’ package in R. For Support Vector Machines, we have implemented ‘libSVM’ with 10-fold cross validation. The K-Nearest Neighbor is implemented with ‘kkNN’ package of R.

**2. Dataset and Features**

**2.1 Dataset**

The dataset we are using is from Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals) which is an open source software framework for audio processing with specific emphasis on Music Information Retrieval applications. The dataset consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav format.

**2.2 Preprocessing**

The first step in preprocessing is to divide the dataset in training, testing and validation sets. For this we have written a python script, which divides the dataset for each genre into 70 songs for testing, 30 songs for testing and 30 songs for training. So finally we have 150 songs in training set and 150 each in validation and training set. After that we have used ‘mkcollection’ which is a simple utility provided by Marsyas framework for creating collection files. The ‘mkcollection’ utility takes the folder containing all songs as an input and the soundfiles residing in that directory or any subdirectories are added to the collection.

**2.3 Feature Extraction**

For features extraction, we are using MARSYAS which is implemented in C++ and retains the ability to output feature extraction data to ARFF format. With the tool bextract.exe, the following features are extracted: Zero Crossings, Spectral Centroid, Spectral Flux, Spectral Rolloff, Mel-Frequency Cepstral Coefficients (MFCC), and chroma features (). The total number of features extracted are 124.

A zero-crossing is a point where the sign of a mathematical function changes (e.g. from positive to negative), represented by a crossing of the axis (zero value) in the graph of the function. The spectral centroid is a measure used in digital signal processing to characterize a spectrum. It indicates where the "center of mass" of the spectrum is. Perceptually, it has a robust connection with the impression of "brightness" of a sound. Spectral flux is a measure of how quickly the power spectrum of a signal is changing, calculated by comparing the power spectrum for one frame against the power spectrum from the previous frame. Spectral rolloff point is defined as the Nth percentile of the power spectral distribution, where N is usually 85% or 95%. The rolloff point is the frequency below which the N% of the magnitude distribution is concentrated. In sound processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave.

**2. Contributions**

|  |  |
| --- | --- |
| **Task** | **Member** |
| Appropriate Data Set search | Jivjot, Mangesh, Anuj, Bikram |
| Feature Set Extraction Library Comparisons | Jivjot, Mangesh, Anuj, Bikram |
| Naïve Bayes implementation | Mangesh |
| SVM implementation | Jivjot |
| Neural Networks implementation | Anuj |
| K- Nearest Neighbor implementation | Bikram |
| Weighted Majority Algorithm | Jivjot |
| Poster | Jivjot, Mangesh, Anuj, Bikram |
| Report | Jivjot, Mangesh, Anuj, Bikram |

**2. Conclusion and Future Work**

Our approach of ensembling over Naïve Bayes, SVM and K-Nearest Neighbour using **Weighted Majority Algorithm (WMA)** improved the overall accuracy on Validation and Testing set

**References**

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to ‘small’ (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*. New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hiippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.