International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:9, 2009

Feature-Driven Classification of Musical Styles

A. Buzzanca, G. Castellano and A.M. Fanelli

***Abstract*—** In this paper we address the problem of musical style classification, which has a number of applications like indexing in musical databases or automatic composition systems. Starting from MIDI files of real-world improvisations, we extract the melody track and cut it into overlapping segments of equal length. From these fragments, some numerical features are extracted as descriptors of style samples. We show that a standard Bayesian classifier can be conveniently employed to build an effective musical style classifier, once this set of features has been extracted from musical data. Preliminary experimental results show the effectiveness of the developed classifier that represents the first component of a musical audio retrieval system..

***Keywords*—**Musical style, Bayesian classifier.

I. Introduction

MUSICAL style as well as the mechanisms underlying style recognition are relatively ill-defined [1]. Several definitions of musical style have been formulated so far. Cope [3] defines musical style as "the identifiable characteristics of a composer's music which are recognizably similar from one work to another". Another definition of musical style is given in [5]: "Style is a replication or patterning, either in human behavior or in the artifacts produced by human behavior, that results from a series of choices made within one set of constraints". In [4] theoretical comprehensive guidelines for style analysis are provided by dissecting musical style into three dimensions: large (groups of works, work, movement), middle (part, section, paragraph, sentence) and small (motive, subphrase, phrase).

Whatever the definition, musical style (and its recognition) is something related to human nature: the average layperson can recognize the difference among simple stylistic features. But things change when it comes to computers. Automatic recognition of musical style is not an easy task. Even relatively simple stylistic manners of playing an instrument, such as playing "energetically", playing "lyrically" or playing "with syncopation" are difficult to detect reliably by automatic classification tools.

Nevertheless, automatic classification of musical styles is gaining more and more importance since it may serve as a way to structure and organize the increasingly large number of music files available on the Web. Actually, styles and genres, typically created manually by human experts, are currently one of the ways used to structure music content on the Web. Automatic musical style classification can potentially automate this process and provide an important component for a complete music information retrieval system. Therefore, building a classifier that can recognize different musical styles is of primary interest.

In this paper, we explore the automatic classification of musical styles by means of a feature-driven approach. More specifically, a feature set for representing a musical monophonic excerpt is considered. The significance of the proposed features is investigated by training a Bayesian classifier using real-world data collected from actual performances of a musician playing the piano. Using the proposed feature set, good classification for all the considered musical styles is achieved. These results are comparable to those reported from human musical style classification.

The paper is structured as follows. Section II describes the process to extract a set of features useful to describe the content of a musical excerpt played according to a specific style. Section III deals with the automatic classification of styles using a statistical classifier applied on the extracted features. In Section IV preliminary results are given. Finally, in Section V some conclusions and future directions are drawn.

II. Feature extraction

The first step for automatic musical style classification is feature extraction, that is the process of computing a compact numerical representation to be used for characterizing a musical excerpt. This is a very important task, since the identification of descriptive features for a specific application is the main challenge in building pattern recognition systems.
To study the problem of musical style recognition, we consider a number of styles that a performer could improvise playing a musical instrument like the piano and using only one melodic line. We assume standard MIDI files as the source of monophonic melodies.

In order to extract a number of samples for each style, and a number of features for describing each sample, we first transform the MIDI file in text format using the Midi2txt tool [7] that allows to extract information such as midi key number, duration (in ms) and volume. Then, we apply a parser that analyzes the text file and extracts useful information from the file. The parser was developed in Borland Delphi 7.0 and presents a graphical interface (see fig. 1) that allows the user to visualize useful information extracted from the MIDI file.

A. Buzzanca is with the Department of Computer Science at the University of Bari, Via Orabona, 4, 70125 Bari.Italy (corresponding author phone:0039 0805442476 fax 0039 080 5442476 e-mail: abuzzanca@ di.uniba.it).

G. Castellano is with the Department of Computer Science at the University of Bari, Via Orabona, 4. 70125 Bari Italy (e-mail: castellano@ di.uniba.it)

A.M. Fanelli is with the Department of Computer Science at the University of Bari, Via Orabona, 470125 Bari.Italy (e-mail: fanelli@ di.uniba.it)..

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:9, 2009

Specifically, the parser extracts information contained in the MIDI file header:

- Midi File Type (in our case always 0 because we assume only one melodic line)
- Number of tracks;
- Number of tics;
- Tempo (here we deal only with melodies written in 4/4);
- Beats.

as well as information concerning the MIDI tracks:

- Key Number: it is the note that was pressed;
- Duration: it is the distance in pulses from the event that onsets the sound of a note to the finishing event;
- Duty Factor: it means the ratio of the time between midi note_on and midi note_off;
- Pitch: it is the value each note can take and ranges from 0 to 127;
- Volume;
- Counts of notes.

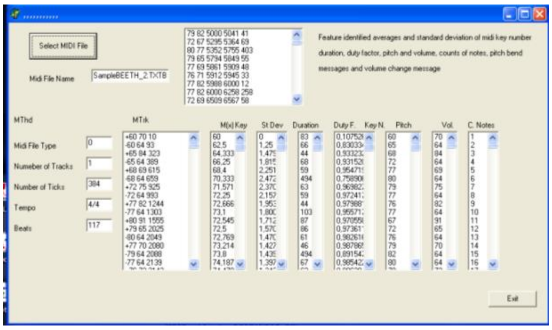

Fig. 1 Graphical interface of the parser developed to extract information from MIDI files

Summarizing, the parser produces in output two text files, a file named "general_info" containing general information extracted from the midi file (note on, note off, volume, duration in ms) and a file named "track_info" containing information extracted from the midi tracks.

In order to derive a number of samples for each style, data contained in the "track_info" file of each musical improvisation are processed according to the following scheme.

Each minute of performance is divided into six blocks of 10 seconds, as proposed in [4]. Each 10-sec segment is then divided into six overlapping intervals, as depicted in fig. 2, with duration of 5 seconds. On the overall, from each minute of performance, we extract 36 samples.

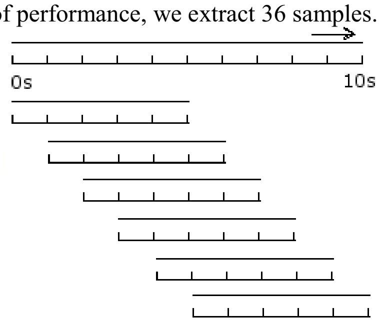

Fig. 2 – Overlapping intervals extracted from the 10-sec segment

Then, for each 5-sec interval the following statistics are computed:

- average and standard deviation of the Key Number
- average and standard deviation of the Duration
- average and standard deviation of Duty Factor
- average and standard deviation of Pitch
- average and standard deviation of Volume
- count of notes

To summarize, each sample is a 11-dimensional vector describing a 5-sec fragment of a played excerpt. Since the problem of musical style classification is mainly a supervised one, each sample is labeled with a number indicating the class (musical style) it belongs to.

III. Musical Style Classification

Once features are extracted, standard pattern recognition techniques which are independent of the specific application area can be used. In this work, as a preliminary approach, we adopt the standard naïve Bayesian Classifier [10]. As for any other statistical pattern recognition classifier, the basic idea is to estimate the probability density function (pdf) for the feature vectors of each class. In supervised learning a labeled training set is used to estimate the pdf for each class. The naïve Bayesian classifier assumes that the features are uncorrelated and normally distributed.

Depending on the precise nature of the probability model, naïve Bayesian classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naïve Bayesian models uses the method of maximum likelihood; in other words, one can work with the naïve Bayesian model without believing in Bayesian probability or using any Bayesian methods.

In spite of their naive design and apparently over-simplified assumptions, naïve Bayesian classifiers often work much better in many complex real-world situations than one might expect. Recently, careful analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naïve Bayesian classifiers [10]. An advantage of the naïve Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Since independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Formally, the probability model for a classifier is a conditional model $p(C|F_1,\ldots F_n)$ over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables F_1 through F_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. Therefore the model is reformulated to make it more tractable.

Using Bayes' theorem, we write $p(C|F_1,\ldots F_n) = p(C)\, p(|F_1,\ldots F_n|C)/\, p(F_1,\ldots F_n)$. In practice we are only interested in the numerator of that fraction, since the denominator does not

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:9, 2009

depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$p(C|F_1,\dots,F_n) = \frac{1}{z} p(C) \prod_{i=1}^{n} p(F_i|C) \qquad (1)$$

where Z is a scaling factor dependent only on $F_1,\dots F_n$), i.e., a constant if the values of the feature variables are known.
Models of this form are much more manageable, since they factor into a so-called class prior p(C) and independent probability distributions $p(F_i|C)$). If there are k classes and if a model for p(Fi) can be expressed in terms of r parameters, then the corresponding naïve Bayes model has $(k - 1) + n\,r\,k$ parameters. In practice, often k = 2 (binary classification) and r = 1 (Bernoulli variables as features) are common, and so the total number of parameters of the naïve Bayes model is 2n + 1, where n is the number of binary features used for prediction.
So given a vector of features F, we would know which one is the nearest classification C, that is the classification that has the least normalized distance from the vector F, i.e.

$$\Delta_C = \sqrt{\sum_{i=1}^{n} \left(\frac{F_i - \mu_{c,i}}{\sigma_{c,i}}\right)^2} \qquad (2)$$

where c indexes classes, i indexes features, $\mu_{c,i}$ represent means and $\sigma_{c,i}$ are standard deviations.

IV. Experimental Evaluation and Analysis

In order to test our approach, we have chosen classical music styles obtained by playing the piano. In particular, we have chosen the following seven musical score. The exact meaning of these terms does not matter. What is really important is the ability of the performer to produce different styles of playing the piano.
Data have been collected at the 'N. Piccinni' State Conservatory of Music in Bari, Italy, where a musician performed for us, playing the piano using only one melodic line, a total amount of about 35 minutes of improvisation into the seven selected styles. For each style, the musician played an excerpt corresponding to a part of an opera written by a not coeval composer according to that style. Table 1 reports details about the excerpts played. Melodies have been sequenced in real time using the piano Yamaha 48" Mark III Series Upright Disklavier that allows the recording of the played excerpt directly in MIDI format.

TABLE I
DETAILS CONCERNING THE EXCERPT PLAYED BY A MUSICIAN ACCORDING TO DIFFERENT MUSICAL STYLES

Author	Opera	Historical period	Duration (in ms)	Number of samples	Style
Czerny	Etude op.740 nr.3	1791–1857	183669	107	1
Chopin	Studio Opera 10 nr.1	1810-1849	195000	113	2
Bach	English Suite BWv 808: Prelude	1685–1750	235708	138	3
Beethoven	Piano Sonata op. 27 nr.2 (1st movement)	1770-1827	274111	161	4
Bach	English Suite BWv 807: Prelude	1685–1750	361242	215	5
Schubert	Impromptu op. 90 nr. 3	1797–1828	386961	228	6
Beethoven	Sonata op.57 "Appassionata" 3rd movement	1770-1827	430480	251	7

For each of the seven musical styles, a number of representative samples were derived, by processing the corresponding excerpt according to the feature extraction scheme described in Section 2. Due to different duration of played excerpt, a different number of samples was derived for each style (see Table 1). In order to have a uniform distribution of samples for each style, we decided to extract exactly 100 samples for each style.
Once the feature extraction process was completed, we built different datasets, in order to evaluate the classification accuracy with different number of musical styles. Specifically, we constructed the following datasets:

- 21 two-class datasets: obtained by considering all possible combination of two styles among seven.
- 35 three-class datasets: obtained by considering all possible combination of three styles among seven.
- 35 four-class datasets: obtained by considering all possible combination of four styles among seven.
- 20 five-class datasets: obtained by considering all possible combination of five styles among seven.
- 7 six-class datasets: obtained by considering all possible combination of six styles among seven.
- one seven-class dataset: this is the whole dataset considering samples of all styles

To evaluate the classification performance, a scheme based on leave-k-out was carried out. In our case k=20% of the size of the dataset. In other words, each dataset was randomly partitioned so that 4/5 of data were used as training set and the remaining 1/5 as testing set. The Bayesian classifier, implemented in Matlab 2008a, was applied to five different random partitions and the results were averaged. This ensures the calculated accuracy to be not biased because of a particular partitioning of training and testing. Indeed, if the datasets are representative of the corresponding musical styles then these results are also indicative of the classification performance with real-world unknown performances.
Classification results for each classifier are plotted in figures 3-7. As concerns the 2-style classifier, the best average performance was obtained with styles 1 and 6 , with a classification rate of 98,57%. For 3-style classification, the best result is achieved with styles 2,4 and 5, with a classification rate of 98,43%. The best result for 4-style classifier was obtained with styles 1,2,4 and 6, with a classification rate of 96,47 %. Classifiers for 5 and 6 styles perform quite worst. Specifically, for 5 styles the best result was obtained with style 1,2,3,4 and 6, with a classification rate of 95,14%. As concerns the 6-style classifier, the best average

performance was obtained with style 1,2,3,4,5 and 6, with a classification rate of 86,25%.

As concerns classification of all the seven musical styles, figure 8 shows more detailed information about the classification accuracy in the form of a confusion matrix. In a confusion matrix, the columns correspond to the actual style and the rows to the classified style. For example, the cell of row 7, column 3 with value 19 means that 100% of the style 3 (column 3) was wrongly classified as style 7 (row 7). The percentages of correct classification lie in the diagonal of the confusion matrix. The confusion matrix shows that the misclassifications of the system are similar to what a human would do For example, samples of style 3 (Bach) are misclassified as style 7 (Beethoven). By analyzing the confusion matrix, it can be seen that Style 3 has the worst classification accuracy since it is easily confused with other styles. This result was somehow expected because of the broad nature of Style 3, which includes very general rules of composition.

Summarizing, the classifier developed on the basis of the extracted features has produced very interesting results for all the styles recognized, as depicted in table 2.

TABLE II
AVERAGE CLASSIFICATION ACCURACY FOR EACH CLASSIFIER

	average performance
2-style classifier	89,37 %
3-style classifier	79,93 %
4-style classifier	71,55 %
5-style classifier	65,32 %
6-style classifier	57,26 %
7-style classifier	51,94%

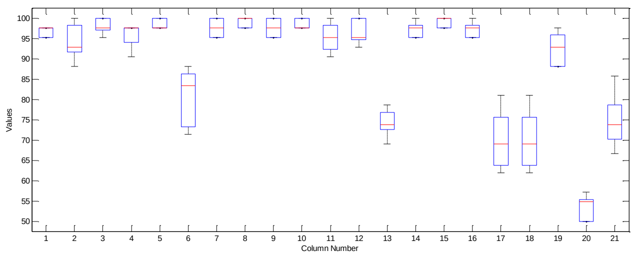

Fig. 3 CLASSIFICATION ACCURACY OF THE 2-STYLES CLASSIFIER

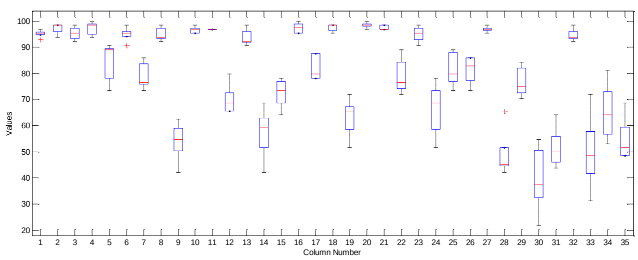

Fig. 4 CLASSIFICATION ACCURACY OF THE 3-STYLES CLASSIFIER

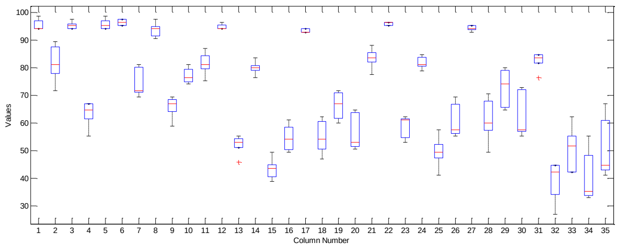

Fig. 5 CLASSIFICATION ACCURACY OF THE 4-STYLES CLASSIFIER

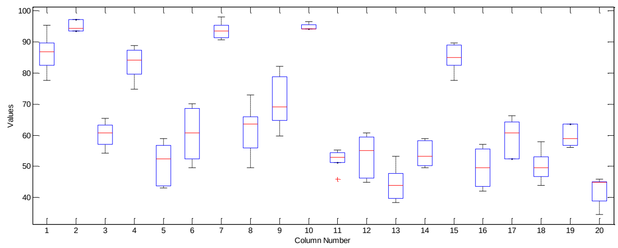

Fig. 6 CLASSIFICATION ACCURACY OF THE 5-STYLES CLASSIFIER

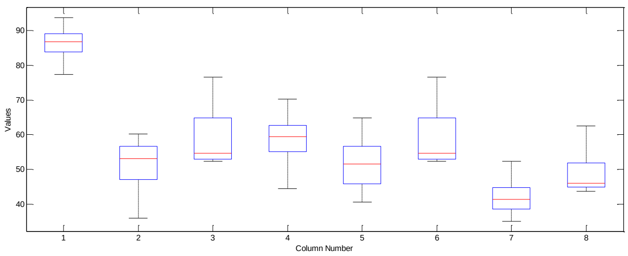

Fig. 7 CLASSIFICATION ACCURACY OF THE 6-STYLES CLASSIFIER

	Style 1	Style 2	Style 3	Style 4	Style 5	Style 6	Style 7
Style 1	15	0	0	0	0	0	0
Style 2	0	18	0	0	0	0	0
Style 3	0	0	0	0	0	0	0
Style 4	0	0	0	15	0	0	0
Style 5	0	0	0	0	2	0	2
Style 6	0	0	0	0	0	11	0
Style 7	10	2	19	11	14	7	23

Fig. 8 Confusion matrix concerning the 7-style classifier

V. CONCLUSION

This work has shown that a standard classifier can be used to classify the musical style of an author once significant features are properly extracted. In our case the approach used has been tested with a free interpretation (improvisational style) with one melodic line.

Some of the misclassification can be caused by the lack of a smart method for melody segmentation. The music samples have been arbitrarily restricted to a duration of 5 sec, getting fragments not necessarily related to musical motives. The main goal of this work was to test the feasibility of the feature extraction approach, and an average recognition of 89,37% for 2 styles is a very encouraging result keeping in mind these limitations. This is just a work in progress, and a number of possibilities are still to be explored, such as the study of other features as significant descriptors. Moreover, in order to draw significant conclusions about the validity of the proposed approach, a large database of musical excerpts representing different styles has to be created and tested using our approach.

REFERENCES

[1] M. Crump, “A principal components approach to the perception of musical style”, Banff Annual Seminar in Cognitive Science (BASICS), Banff, Alberta May 10, 2002.

[2] S. Brook, Barry, “Style and Content Analysis in Music” The simplified Plaine and Easie Code in The Analysis of Communication Content, George Gerbner et Al. New York, Wiley 1969

[3] D. Cope, “Computers and Musical style”, Madison, Wisconsin: A-R Editions, Inc. 1991

[4] J. La Rue, “Guidelines for style analysis”, New York: W. W. Norton & Company 1970;

[5] Leonard Meyer, “Style and Music” Philadelphia: University of Pennsylvania Press, 1989.

[6] G. Buzzanca, “A supervised learning approach to musical style recognition” Additional proceedings of the Second International Conference ICMAI, Edinburgh, Scotland. 2002

[7] MIDI2TXT v1.14 midi binaries to text mnemonic by Guenter Nagler 1995.

[8] D. J. C. MacKay, “Information-based objective functions for active data selection. Neural Computation 4” (4), 590-6041992c 1992a

[9] D. J. C. MacKay, “Bayesian methods for backpropagation networks”. In E. Domany, J. L. van Hemmen, and K. Schulten (Eds.), Models of Neural Networks III, Chapter 6. New York: Springer-Verlag 1994a

[10] C. M. Bishop, “Neural Networks for pattern Recognition”. Clarendon Press, 1995.