A shallow description framework for musical style recognition

Pedro J. Ponce de León, Carlos Pérez-Sancho and José M. Iñesta

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante,
Ap. 99, E-03080 Alicante, Spain
{pierre,cperez,inesta}@dlsi.ua.es

Abstract. In the field of computer music, pattern recognition algorithms are very relevant for music information retrieval (MIR). One challenging task within this area is the automatic recognition of musical style, that has a number of applications like indexing and selecting musical databases. In this paper, the classification of monophonic melodies of two different musical styles (jazz and classical) represented symbolically as MIDI files is studied, using different classification methods: Bayesian classifier and nearest neighbour classifier. From the music sequences, a number of melodic, harmonic, and rhythmic statistical descriptors are computed and used for style recognition. We present a performance analysis of such algorithms against different description models and parameters.

Keywords: music information retrieval, Bayesian classifier, nearest neighbours.

1 Introduction

The computer music research field is an emerging area for pattern recognition and machine learning techniques to be applied. The content-based organisation, indexing, and exploration of digital music databases (digital music libraries), where digitised (MP3), sequenced (MIDI) or structurally represented (XML) music can be found, is known as music information retrieval (MIR).

One of the problems to be solved in MIR is the modelization of the music style. The computer could be trained in the user musical taste in order to look for that kind of music over large musical databases. The same scheme is suitable to learn stylistic features of composers. Other applications of such a system is to be used in cooperation with automatic composition algorithms to guide this process according to a stylistic profile provided by the user.

A number of recent papers explore the capabilities of machine learning methods to recognise music style. Pampalk et al. [1] use self-organising maps (SOM) to pose the problem of organising music digital libraries according to sound features of musical themes, in such a way that similar themes are clustered, performing a content-based classification of the sounds. Whitman and Flake [2] present a system based on neural nets and support vector machines, able to classify an audio fragment into a given list of sources or artists. Also in [3], the authors describe a neural system to recognise music types from sound inputs. In the work by Thom [4] pitch histograms (measured in semitones relative to the tonal pitch and independent of the octave) are used to describe blues fragments of the saxophonist Charlie Parker. Also pitch histograms and SOM are used in [5] for musicological analysis of folk songs.

In a recent work, Cruz et al. [6] show the ability of grammatical inference methods for modelling musical style. A stochastic grammar for each musical style is inferred from examples, and those grammars are used to parse and classify new melodies. The authors also discuss about the encoding schemes that can be used to achieve the best recognition result. Other approaches like hidden Markov models [7] or multilayer feedforward neural networks [8] have been used to solve this problem.

2 Objectives

Our aim is to develop a system able to distinguish musical styles from a symbolic representation of melodies (digital scores) using shallow structural features, like melodic, harmonic, and rhythmic statistical descriptors. Our working hypothesis is that melodies from a same musical genre may share some common features that permits to assign a musical style to them. We have chosen two music styles, jazz and classical, as a workbench for our experiments. The initial results have been encouraging (see [9]) but now we want to explore the method performance for different classification algorithms, descriptor models, and parameter values.

First, our methodology will be presented, describing the musical data and the description models and classifiers we have used. Then, the classification results obtained with each classifier and an analysis of the recognition results against the different description parameters will be presented. Finally, conclusions and current and future lines of work are discussed.

3 Methodology

In this section we first present the musical sources from which the experimental framework has been established. Second, we will go into the details of the description models we have chosen to describe those musical data. Next we will discuss the free parameters space that sets up the whole experimental framework, and then the classifier implementation and tuning will be presented.

3.1 Musical data

MIDI files from jazz and classical music, were collected. Classical melody samples were taken from works by Mozart, Bach, Schubert, Chopin, Grieg, Vivaldi, Schumann, Brahms, Beethoven, Dvorak, Haendel, Paganini and Mendehlson. Jazz music samples were standard tunes from a variety of authors like Charlie Parker, Duke Ellington, Bill Evans, Miles Davis, etc. The MIDI files are composed of several tracks, one of them being the *melody track* from which we actually extract our data[1]. The corpus is made up of a total of 110 MIDI files,

[1] Without loosing generality, all the melodies are written in the 4/4 meter. They are monophonic sequencies (at most one note is playing at any given time.)

45 of them being classical music and 65 being jazz music. This is a somewhat heterogeneous corpus, not specifically created to fit our purposes but collected from different sources ranging from web sites to private collections.

The monophonic melodies consist of a sequence of paired events that can be either note onsets or note endings. The onset event encodes the note *pitch*, that can take a value from 0 to 127. Each onset event at time t has its corresponding ending event at time $t+d$, being d the note *duration*. Time intervals between an ending event and the next onset event are *silences*.

3.2 Description model

Instead of using an explicit representation of the melodies, we have chosen a description model based on statistical descriptors that summarise the content of the melody in terms of pitches, note durations, silence durations, harmonicity, etc.

The datasets are vectors of musical descriptors computed from fixed length segments of the melodies found in the MIDI files (See section 3.4 for a discussion about how these segments are obtained). Each segment is labelled with the style of the melody it belongs to. We defined an initial set of descriptors based on three groups of features that assess the melodic, harmonic and rhythmic properties of a melody, respectively. Then, from this initial set of descriptors a selection procedure has been performed based on a per-feature separability test. This way, some reduced models have been constructed and their classification ability tested.

The features are computed using a time resolution of $Q = 48$ pulses per bar[2].

The initial model is made up of the following 22 descriptors:

- Overall descriptors:
  - *Number of notes* and *number of significant silences* (those larger than a sixteenth note) in the fragment.
- Pitch descriptors:
  - *Pitch range* (The difference in semitones between the highest and the lowest note in the melody), *average pitch*, and *standard deviation of pitch* (provide information about how the notes are distributed in the score).
- Note duration descriptors (these descriptors are measured in pulses):
  - *Minimum*, *maximum*, *average*, and *standard deviation* of note durations.
- Significant silence duration descriptors (in pulses):
  - *Minimum*, *maximum*, *average*, and *standard deviation.*
- Interval descriptors (distance in pitch between two consecutive notes):
  - *Minimum*, *maximum*, *average*, and *standard deviation.*
- Harmonic descriptors:

[2] This is call quantisation. $Q = 48$ means that if a bar is composed of 4 times, each time can be divided, at most, into 12 pulses.

  - *Number of non diatonic notes.* An indication of frequent excursions outside the song key[3] or modulations.
  - *Average degree of non diatonic notes.* Describes the kind of excursions. It is a number between 0 and 4 that indexes the non diatonic notes of the diatonic scale of the tune key, that can be major or minor key[4]. It can take a fractional value.
  - *Standard deviation of degrees of non diatonic notes.* Indicates a higher variety in the non diatonic notes.
- Rhythmic descriptor: *number of syncopations*: notes that do not begin at the rhythm beats but in some places between them (usually in the middle) and that extend across beats. This is actually an estimation of the number of syncopations, but is enough for this task. Syncopations are supposed to appear more frequently in Jazz music than in classical music.

With this set of descriptors, we assume the following hypothesis: melodies of the same style are closer to each other in the description space than melodies from different styles. We will test the performance of different classifiers to verify this hypothesis.

This kind of statistical description of musical content is sometimes referred to as *shallow structure description* [10]. It is similar to histogram-based descriptions, like those found in [5], that try to model the distribution of musical events in a music fragment. Computing the minimum, maximum, mean and standard deviation from the distribution of musical features like pitches, durations, intervals and non-diatonic notes we reduce the number of features needed (each histogram may be made up of tens of features). Other authors have also used some of the descriptors presented here to classify music [11].

3.3 Feature selection procedure

The utilised features have been designed according to those used in musicological studies but there is no theoretical support for them. We have devised a selection procedure in order to keep those descriptors that actually contribute to make the classification. The method doesn't account for possible correlations between descriptors, but tests the separability provided by each descriptor independently, and uses this separability to obtain a descriptor ranking. For a detailed discussion on how descriptors are ranked and selected, see [9].

Four additional description models have been constructed with selected descriptors, as shown in table 1. Each model number denotes the number of descriptors included in that model.

We have chosen four reduced model sizes: 6, 7, 10 and 13 descriptors. The 7-descriptor model includes the best rated descriptors. The 6-descriptor model excludes syncopation from the former model, to test the contribution of the

[3] We used the *key* meta-event present in each MIDI file to compute the harmonic descriptors. The correctness of its value was verified for each file prior to the feature extraction process.

[4] Non diatonic degrees are: 0: ♭II, 1: ♭III (♮III for minor key), 2: ♭V, 3: ♭VI, 4: ♭VII.

rhythm descriptor on its own. The other two models include other average rated descriptors.

Table 1. Description models. For each model the descriptors included are shown in the right column.

Model	Descriptors
6	Pitch range, max. interval, dev. note duration, max. note duration, dev. pitch, avg. note duration
7	+syncopation
10	+avg. pitch, dev. interval, number of notes
13	+number of silences, min. interval, num. non-diatonic notes
22	All the descriptors computed

3.4 Free parameter space

Given a melody track, the statistical descriptors presented above are computed from fixed length segments of that melody. These segments are extracted defining a window of size ω. One segment is extracted and the window is shifted δ measures towards the end of the melody to obtain the next segment to be described. Given a melody with $m > 0$ measures, the number of segments s of size $\omega > 0$ obtained from that melody is

$$s = \begin{cases} 1 & \text{if } \omega \geq m \\ 1 + \left\lceil \frac{m-\omega}{\delta} \right\rceil & \text{otherwise} \end{cases} \tag{1}$$

showing that at least one segment is always extracted (ω and s are positive integers; m and δ may be positive fractional numbers).

Taking ω and δ as free parameters in our methodology, we have setup a framework where the style classification task is achieved, for different datasets of segments derived from the particular values for ω and δ. The goal is to investigate if there is an optimal combination of these parameters that gives the best segment classification results. The exploration space for this parameters would be referred as $\omega\delta$-space.

ω is the most important parameter in this framework, as it determines the amount of information available for the descriptor computations. A value around 1 would produce windows with a few notes inside, making statistical descriptors less reliable. A large value for ω would lead to merge the –probably different– principal parts of a melody into a single window and also produces datasets with too few samples for training the classifiers. The value of δ would affect primarily the number of samples in a dataset. A small δ value combined with somewhat large values for ω can produce datasets with a large number of samples. The details about the values we used for these parameters can be found in section 4.

3.5 Classifier implementation and tuning

Two different classifiers are used in this paper to automatic style identification. They are supervised methods: The Bayesian classifier and the nearest neighbour (NN) classifier [12].

For the Bayesian classifier, we assume that individual descriptor probability distributions for each style are normal, with means and variances estimated from the training data. This classifier computes the squared Mahalanobis distance from test samples to the mean vector of each style in order to obtain a classification criterion.

The NN classifier uses an Euclidean metrics to compute the distance between the test sample and the training samples. The style label of the nearest training sample is assigned to the test sample.

4 Experiments and results

4.1 The $\omega\delta$-space experiment framework

The melodic segment classification framework has been defined as follows:

$$\omega = 1, ..., 100 \tag{2}$$

and, for each ω

$$\delta = \begin{cases} 1, ..., \omega & \text{if } \omega \leq 50 \\ 1, ..., 20 & \text{otherwise} \end{cases} \tag{3}$$

The range for δ when $\omega > 50$ has been limited to 20 due to the very few number of samples obtained with larger δ values for this ω range. This setup let us with a total of 2275 points in the $\omega\delta$-space. We will denote a point in such a space as $\langle w, d \rangle$. A set of experiments have been done for each of these points. An experiment with each classifier (Bayesian and NN) has been prepared for each of the five description models discussed in section 3.3, in order to classify melodic segments. We therefore have 10 different experiments for each $\omega\delta$-point, denoted by $(\omega, \delta, \mu, \gamma)$ where $\mu \in \{6, 7, 10, 13, 22\}$ indicates the description model and $\gamma \in \{Bayesian, NN\}$ the classifier used in that experiment.

For obtaining reliable results a scheme based on *leave-k-out* has been carried out at the level of the source MIDI files for each of the $(\omega, \delta, \mu, \gamma)$ experiments. We want to end up with 10 sub-experiments, that is making $k \simeq 10\%$. The reason behind choosing to separate the files for testing and training rather than first extracting the segments from the files and then perform the leaving-k-out separation is that we want to minimise the probability of having identical segments in the test and training sets. Intuitively, compelling training samples to come from different sources (different MIDI files) than test samples would reduce such a probability to a minimum.

For each sub-experiment 5 jazz style files and 5 classical style files out of a total number of 110 files are kept for testing. Once they have been chosen, segments of ω measures are extracted from the melody tracks and test and training datasets containing μ-size descriptor vectors are constructed.

The segment classification sub-experiments are performed training the γ classifier with the corresponding training set. Classification tests are done with the trained classifiers and the success ratio is averaged over all the sub-experiments.

Thus, for each $(\omega, \delta, \mu, \gamma)$ experiment, we obtained an average classification success rate.

Summarising, 22750 experiments, each consisting of 10 sub-experiments, have been carried out. The maximum number of segments extracted is 8985 for the $\omega\delta$-point $\langle 3, 1\rangle$. The maximum is not located at $\langle 1, 1\rangle$ as expected, due to the fact that segments not containing at least two notes are discarded. The minimum number of segments extracted is 119 for $\langle 100, 20\rangle$, as expected. The average number of segments in the whole $\omega\delta$-space is 775. The average proportion of jazz segments is 36% of the total number of segments, with a standard deviation of about 4%. This is a consequence of the classical MIDI files having an average melody length greater than jazz files, although there are less classical files than jazz files.

4.2 Classification results

In Fig. 1 the results in the $\omega\delta$-space for the Bayes classifier with the 10-descriptor model are displayed. Note that recognition percentages range from 76 to 91%, although the values below 80% concentrate in the low-left corner when the window width is very small. All the results with $\omega > 5$ are above 82%. The best results were obtained for $\omega \approx 70$ and $\delta \approx 15$ (more than 91%). Although this behaviour is not exactly the same for the other classifiers and models.

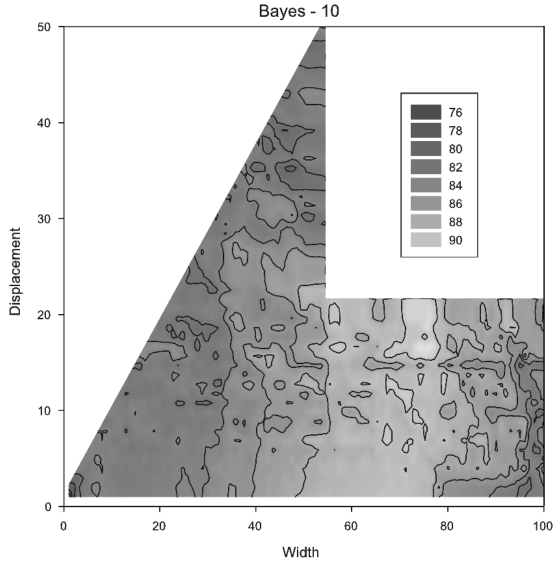

Fig. 1. Illustration of the recognition percentage in the $\omega\delta$ space. The best results (around 91.5 %) are found in the lighter area, with large widths and moderate displacements.

Figure 2 summarizes the behaviour of the Bayes and NN classifiers for the different values of ω, given a fixed value of $\delta = 1$. Note that the trend is to rise rapidly for small values of ω and then to be more or less stable. The different experiments for NN have provided higher differences in recognition percentages than the Bayes classifier. Also, note that NN performs slightly better in average for large ω values. As for the entire $\omega\delta$-space, the NN classifier scored an 83.3% overall average success rate, while a 76.6% success rate was obtained for the Bayesian classifier.

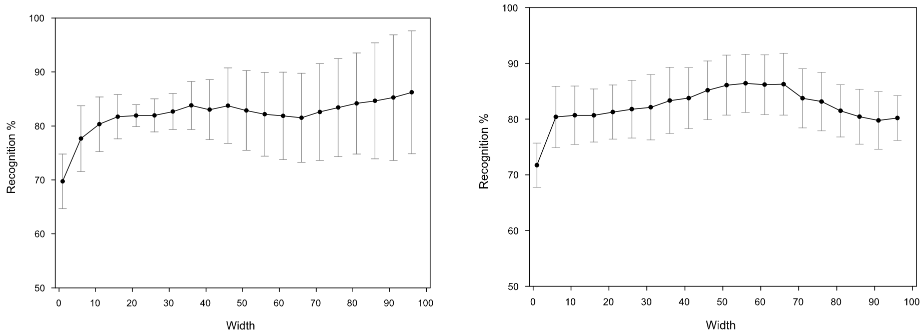

Fig. 2. Recognition results averaged for the different models against the window width, with a fixed $\delta = 1$. Bars indicate the deviation obtained for the different experiments. Only one point every five points is displayed for clarity. (left) Nearest neighbour; (right) Bayes.

5 Conclusions and future work

We have shown the ability of two classifiers to map symbolic representations of melodic segments into a set of musical styles using melodic, harmonic and rhythmic statistical descriptors. The experiments have been carried out over a large parameter space defined by the size of melodic segments extracted from melody tracks of MIDI files of both styles and the displacement between segments consecutively extracted from the same melodic source. A total of 227500 classification experiments have been performed.

Our main goal in this work has been to establish a framework for musical style recognition experiments, while concluding an answer for the following two questions:

1. Which classifier works better for this task?
2. Are there any optimal ω and δ values for style classification?

Answer to the first question is the NN classifier, as seen in previous section, with an 83.3% overall average succes rate, with a best success rate of 94% for model 10 and point $\langle 98, 1 \rangle$. Answer to the second question is a somewhat disappointing one. The best results were obtained with very large segment sizes with the NN classifier, leading us to a new question: Would we achieve even more better results if we take as samples single descriptor vectors that represent the whole melodies? This issue must be investigated further with larger corpora.

New description models and different classifers can be easily incorporated to this framework, as well as different corpora and the exploration of different ranges for ω and δ. An extension to the framework is under development, where a voting scheme for segments will be used to obtain classification results for whole melodies. Connectionist approaches are also to be tested with the models and parameters presented here. Finally, different descriptor sets are currently under study in our search for a good statistical description for musical styles.

Acknowledgements

This paper has been funded by the Spanish CICyT project TIRIG; code TIC2003–08496–C04–04,

References

1. E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, pages 201–208, Baltimore, USA, 2003.
2. B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568. Falmouth, Massachusetts, September 10–12 2001.
3. H. Soltau, T. Schultz, M. Westphal, and A. Waibel. Recognition of music types. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-1998).* Seattle, Washington, May 1998.
4. B. Thom. Unsupervised learning and interactive jazz/blues improvisation. In *Proceedings of the AAAI2000*, pages 652–657, 2000.
5. P. Toiviainen and T. Eerola. Method for comparative analysis of folk music based on musical feature extraction and neural networks. In *III International Conference on Cognitive Musicology*, pages 41–45, Jyvskyl, Finland, 2001.
6. P. P. Cruz-Alcázar, E. Vidal, and J. C. Pérez-Cortes. Musical style identification using grammatical inference: The encoding problem. In *Proceedings of CIARP 2003*, pages 375–382, La Habana, Cuba, 2003.
7. W. Chai and B. Vercoe. Folk music classification using hidden markov models. In *Proc. of the Int. Conf. on Artificial Intelligence*, Las Vegas, USA, 2001.
8. G. Buzzanca. A supervised learning approach to musical style recognition. In *Music and Artificial Intelligence. Additional Proceedings of the Second International Conference, ICMAI 2002*, Edinburgh, Scotland, 2002.
9. P. J. Ponce de León and J. M. Iñesta. Feature-driven recognition of music styles. In *1st Iberian Conference on Pattern Recognition and Image Analysis. Lecture Notes in Computer Science, 2652*, pages 773–781, Majorca, Spain, 2003.
10. Jeremy Pickens. A survey of feature selection techniques for music information retrieval. Technical report, Center for Intelligent Information Retrieval, Departament of Computer Science, University of Massachussetts, 2001.
11. S. G. Blackburn. *Content Based Retrieval and Navigation of Music Using Melodic Pitch Contours.* PhD thesis, Department of Electronics and Computer Science, University of Southampton, UK, 2000.
12. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition).* Wiley-Interscience, 2000.