Symbolic Musical Genre Classification based on Repeating Patterns

Ioannis Karydis Alexandros Nanopoulos
Yannis Manolopoulos

Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
{karydis,alex,manolopo}@delab.csd.auth.gr

ABSTRACT

This paper presents a genre classification algorithm for symbolic music data. The proposed methodology relies on note pitch and duration features, derived from the repeating patterns and duration histograms of a musical piece, respectively. Note-information histograms have a great capability in capturing a fair amount of information regarding harmonic as well as rhythmic features of different musical genres and pieces, while repeating patterns refer to segments of the piece that are semantically important. Detailed experimental results on intra-classical genres illustrate the significant performance gains due to the proposed features.

Keywords

Genre, Features, Repeating Patterns, Classification, Music Information Retrieval.

1. INTRODUCTION

Musical genres are categories of musical pieces which have in common a certain style. Genre taxonomies are established by a pertinent community, such as the creators, the critics and the commercial industry. Genre is an important research criterion for Music Information Retrieval (MIR). Since manual appointment of metadata inhibits difficulties and potential inconsistencies [11], the need for an effective automatic means of music classification unfolds.

Not only challenging, due to the inherent subjectivity in the notion of genre, automatic genre classification is also highly required, as the collections of symbolic digital music files increase at a rapid rate. Musical files can be broadly categorised according to their content in symbolic and acoustic. In this paper, we focus on automatic genre classification from musical files in symbolic format, as they engulf an excess of information that may not always be perceivable in the respective acoustic piece[1]. In order to process all the information included in the music files, one can rely on perceptual criteria (features) related to pitch, rhythm, timbre, etc of the music in order to characterise a musical genre [3, 10, 13]. We focus on the note pitch and duration information of the musical data.

AMCMM'06, October 27, 2006, Santa Barbara, California, USA.

1.1 Motivation

Until now, research on symbolic music genre classification has focused on numerous features extracted from the entire of the musical datum. In contrast, we propose to focus the feature-extraction procedure on important areas of a music piece, which are characteristic of the musical composition and act as basic construction elements of the piece. Important areas are designated through the notion of Repeating Patterns (RPs) [6], which are recurring fragment or succession of notes and are connected to motifs and themes [2]. Moreover, most of the existing features in prior work do not consider the factor of interdependence between pitch information. The features we utilise take into account the aforementioned factor, thus achieving significant improvement in accuracy of genre classification.

Notice that the objective of this paper is not to abolish any existing approaches. In contrast, we only want to demonstrate the value of features extracted by RPs as an additional source of information for effective genre classification. The same remark applies for the statistical features we utilise, which can be considered as one of many alternatives that related research has indicated and can operate complementary with them. Therefore, this paper highlights the importance of taking into account RPs for the purpose of genre classification, which is sustained by the very good results of our experimental evaluation.

1.2 Contribution

Our contributions are summarised as follows:

- A novel method for extracting features from note information, which is focused on RPs, motivated by the fact that RPs represent characteristic parts of the musical data and avoid noise during genre classification.
- The use of (a) several statistical features extracted from RPs, which exploit the correlation of the occurrences of the fundamental music characteristics, and (b) of an automatic feature-selection process that keeps only the best features.

[1]Extension to acoustic data can be possible, by extracting pitch information as proposed by Tzanetakis et al. [13].

- We additionally employ features extracted from information about duration, and a weighting scheme so as to harness the full discriminating capability of pitch and duration.

- Experimental results which illustrate the superiority of the proposed method against a previous method [13] that is based on pitch histograms without taking into account the proposed considerations.

The rest of the paper is organised as follows. Section 2 is devoted to related research as far as symbolic music genre classification is concerned, while Section 3 provides a complete account of the proposed method of this paper. Subsequently, Section 4 presents and discusses the experimentation results obtained. Finally, the paper is concluded by a summary in Section 5.

2. RELATED WORK

Despite the fact that musical genre classification research is mainly oriented towards acoustic data, approaches for symbolic data have interesting results to demonstrate.

Tzanetakis et al. [13] introduced pitch histograms as a way to represent the pitch content of music signals both in symbolic and acoustic form. The authors of [13] considered two versions of the pitch histogram according to whether the octave discrimination of notes is taken into consideration or not. The rationale these choices rely on is that unfolded histograms can capture the pitch range of a piece while folding supports octave independency. In order to minimise the search space, four one-dimensional features were extracted from the two histograms (folding and non-folding). Based on features extracted from these histograms the authors of [13] managed a 50% accuracy for 5 genres. Compared to this work, we additionally include the notion of pitch dependency as well as the orientation in repeating patterns, which focus on significant parts of the piece as far as semantics and efficiency are concerned [6, 7].

In [10] an account of a system that extracts 109 musical features from symbolic recordings and uses them to classify the recordings by genre, is presented. The features used are based on instrumentation, texture, rhythm, dynamics, pitch statistics, melody and chords, while the system described therein requires training for the"fittest" set of features, a cost that trades-off the generality of the approach with the overhead of feature selection. The achieved reported classification reaches 90% for intra-category subcategories and 98% for categories.

Finally, Basili et al. [3] presented five features based on melody, timbre and rhythm for the purposes of symbolic music genre classification. In their investigation, the comparison of different machine learning algorithms (including decision-tree, Bayesian and rule-based classifiers) for the purpose of genre classification have been considered.

Repeating Patterns (RPs) have been extensively used throughout the history of music [2] as well as in the modern music [1], as they comprise a compact form for indexing the original formats (e.g., raw audio, MIDI, etc), a reference point for the discovery of music *themes* [9, 12] and *characteristic signatures* of music objects, which have the notion of a quantitative measure for music similarity [4]. The existence of RPs in musical pieces has been supported by both composers (a starting point for development, which may be repeated possibly in the form of variations) and listeners [2]. As the process of mining musical RPs presents important challenges, several methods, e.g, [6, 7], have been proposed in the MIR literature for their extraction.

The use of RPs for the purpose of musical genre classification has been proposed by Lin et al. [8], where RPs are used *per se* for the characterisation of musical categories. For each pattern discovered from a group of music data, Lin et al. employ a series of measurements to estimate its usefulness for classifying this group of music data. According to the patterns contained in a music piece, they determine which class it should be assigned to.

The approach considered in [8] can be thought of as complementary to our proposed methodology. Lin et al. collectively characterise a musical genre by the RPs that are common to this genre. Accordingly, each new, characterised-to-be piece is judged by the similarity of its RPs with the repeating patterns of each genre. In our approach, RPs determine musical segments that contain valuable information, while the parts of the piece lying outside the RPs are considered less important and thus not taken into consideration. Consequently, the RPs of a piece contribute for the creation of a set of features that characterises the piece. Accordingly, each new, characterised-to-be piece is judged by the similarity of the resulting features using the k-NN approach.

2.1 The RP-Tree Algorithm

Hsu et al. [6] proposed two different techniques for the discovery of non-trivial repeating patterns. Herein, we focus on the string-join approach, which is denoted as RP-Tree, which will be epigrammatically considered in this section.

RP-Tree utilises $\{X, freq(X), (pos_1, pos_2, \ldots)\}$ to represent a repeating pattern found in a music sequence S, where X is the repeating pattern, $freq(X)$ is the repeating frequency of X and each $pos_i, 1 \leq i \leq freq(X)$, is a starting position of X in S. According to [6] the string-join operation is defined as follows: Assume that $\{\alpha_1\alpha_2 \ldots \alpha_m, freq(\alpha_1\alpha_2 \ldots \alpha_m), (p_1, p_2, \ldots, p_i)\}$ and $\{\beta_1\beta_2 \ldots \beta_n, freq(\beta_1\beta_2 \ldots \beta_n), (q_1, q_2, \ldots, q_j)\}$ are two repeating patterns in the music feature string of a music object. We define order-k string-join ($k \geq 0$) of the two repeating patterns as follows:

$$\begin{gathered}\{\alpha_1\alpha_2 \ldots \alpha_m, freq(\alpha_1\alpha_2 \ldots \alpha_m), (p_1, p_2, \ldots, p_i)\} \quad \bowtie_k \\ \{\beta_1\beta_2 \ldots \beta_n, freq(\beta_1\beta_2 \ldots \beta_n), (q_1, q_2, \ldots, q_j)\} = \\ \{\gamma_1\gamma_2 \ldots \gamma_l, freq(\gamma_1\gamma_2 \ldots \gamma_l), (o_1, o_2, \ldots, o_h)\}\end{gathered}$$

where

- $i = freq(\alpha_1\alpha_2 \ldots \alpha_m)$, $j = freq(\beta_1\beta_2 \ldots \beta_n)$, $h = freq(\gamma_1\gamma_2 \ldots \gamma_l)$,
- $\gamma_t = \alpha_t$ for $1 \leq t \leq m$, $\gamma_t = \beta_{t-m+k}$ for $m+1 \leq t \leq l = m+n-k$,
- $o_t = x = y - m + k$, where $x \in \{p_1, p_2 \ldots, p_i\}$ and $y \in \{q_1, q_2 \ldots, q_j\}$[2],
- $o_t < o_{t+1}$, for $1 \leq t \leq h-1$,
- if $k > 0$, $\alpha_{m-k+s} = \beta_s$, for $1 \leq s \leq k$.

[2]This condition refers to how the position of elements in sequence γ relates to the positions of appearance of the sequences α and β.

Name	Value	Name	Value
Entropy	$-\sum_i \sum_j C(i,j) log C(i,j)$	Variance	$\frac{1}{2}\sum_i \sum_j ((i-\mu)^2 C(i,j) + (j-\mu)^2 C(i,j))$
Energy	$\sum_i \sum_j C^2(i,j)$	Correlation	$\sum_i \sum_j \frac{(i-\mu)(j-\mu)C(i,j)}{\sigma^2}$
Contrast	$\sum_i \sum_j (i-j)^2 C(i,j)$	Maximum Probability	$\max_{i,j} C(i,j)$
Homogeneity	$\sum_i \sum_j \frac{C(i,j)}{1+\lvert i-j\rvert}$	Inverse Difference Moment	$\sum_i \sum_j \frac{C(i,j)}{\lvert i-j\rvert^k},\ i \neq j$
SumMean	$\frac{1}{2}\sum_i \sum_j (iC(i,j) + jC(i,j))$	Cluster Tendency	$\sum_i \sum_j (i+j-2\mu)^k C(i,j)$

Table 1: Statistical Features

RP-Tree develops in two stages: In the first stage, repeating patterns of length 2^k (initially, $k = 0$) are found, while repeating patterns of length 2^{k+1} are then found by joining repeating patterns of length 2^k. The search, during the first stage, proceeds until a k_l is reached for which no repeating pattern exists. At this point, RP-Tree has to determine the length L of the longest repeating pattern, which is unknown in advance. Though, the length of the maximum repeating pattern L is known to be between $2^{k_{l-1}} \leq L < 2^{k_l}$. Therefore, RP-Tree performs a binary search of the patterns the length of which is in the range $[2^{k_{l-1}}, 2^{k_l})$. At the end of the first stage, the RP-Tree has determined the L and the corresponding maximum-length patterns. Proceeding to the second stage, in order to ensure that all repeating patterns found in the previous step are nontrivial, a tree structure called RP-Tree is introduced, each node of which represents a repeating pattern found. After the removal of all trivial repeating patterns, a refining procedure identifies all repeating patterns the length of which is not a power of two (if any). The resulting repeating patterns of the refining process are added to the RP-Tree. Finally, all trivial repeating patterns are discarded, leaving the RP-Tree to contain only the maximum and the non-trivial repeating patterns, completing thus the second stage of RP-Tree.

3. THE PROPOSED METHOD

This section presents the three key parts of this work: (i) the extraction of statistical features from the pitch of RPs (Section 3.1), (ii) the extraction of note duration features from the pieces (Section 3.2), and (iii) the proposed weighting scheme that balances the two previously mentioned feature sets (Section 3.3).

3.1 Statistical features extraction

To extract statistical features, we first have to find all non-trivial RPs of each provided musical piece. In this work, RPs are found using the algorithm of [6], as described in Section 2.1.

Next, for each music piece we compute the *co-occurrence matrix*, C, of the discrete pitches appearing within its RPs. Element $C(i,j)$ denotes the number of times that pitches i and j co-appear in the same RP. This way, we take into account the inter-dependencies between pitch appearances. Additionally, to consider locality we use the constraint that pitches i and j should co-appear in a window of length at most w in the RP (i.e., there are no more than $w-1$ pitches between them). Larger musical pieces tend to include more RPs, which contribute more to the co-occurrence matrix of the piece. To get the relative values of occurrences and reveal the actual contribution of each $C(i,j)$ value, we perform normalisation. Each $C(i,j)$ is divided by n^M, where n is the number of RPs that participated in the creation of C and M is a parameter that controls the amount of normalisation.

From each C matrix we can extract several statistical features. The features we considered are summarised in Table 1. Automatic feature selection is performed using the hybrid feature-selection algorithm of [5], which uses three statistical measurements to evaluate features and applies Bayesian Expectation Maximisation to the features ranked by the three measurements to select the feature set.

3.2 Duration features extraction

For each music piece we construct a histogram that indicates the frequency of occurrence of each discrete note duration. Intuitively, duration histograms offer a means to capture the structure and rhythmic part of a piece. This is especially true in classical music where musical genres were created and evolved based on rules. For example, it is quite common for fugues to have several parts where the durations of the notes therein are significantly shorter than the other parts, in order to convey a sense of tenseness, since the original theme of fugues was an escape.

We extract three one-dimensional features from the duration histograms: (a) the duration that has the greatest frequency of appearance in a piece, (b) the number of appearances of the duration with the highest frequency, and (c) the distance between the two highest frequency durations in terms of relative temporal duration.

Accordingly, the proposed selection of features was based on the specific characteristics required to retain such as the note duration that appears more often as well as the second (indirectly through the distance) and the appearances of the most frequent duration. Additionally, features of the same style have successfully been employed in the literature for the purposes of symbolic genre classification [13], although on differentiated characteristic of the musical data.

3.3 Feature-set weighting scheme

Until now, we have described two sets of features, those extracted from pitch information (Section 3.1) and those from duration information (Section 3.2). To create a single set of features that will consider both these aspects, we combine them using the following weighting scheme.

We use a k-NN algorithm to perform genre classification.

First, we perform classification separately for the feature set of pitch information and the feature set of duration information. Thus, we find two separate sets of k nearest neighbors. Next, we perform a voting procedure by taking into account both sets of neighbors and using a different weight for each of the two sets. For example, in the $0\% - 100\%$ case, the neighbors from the feature set of pitch information are entirely ignored and only the ones from duration information are considered. In the $100\% - 0\%$ case, the opposite holds and we only consider the neighbors from pitch information. Between these two extremes we have to find the optimum balance. During our experimental evaluation we found that pitch information is in general more valuable and has to be given higher weight. For this reason, in the presented experimental results in the following section, we use the $70\% - 30\%$ combination.

4. PERFORMANCE EVALUATION

This section presents a concise description of the experimentation platform and data sets, followed by a performance analysis based on experimentation on the proposed method.

4.1 Experimental Set-up

We measure the performance of the proposed method, which is henceforth denoted as RPH (Repeating-Pattern Histograms). For comparison purposes we also consider the method of [13], because it is also based on pitch histograms but not on RPs (this method is denoted as PH). All algorithms were implemented with MS Visual C++. The performance measure was the precision accuracy of genre classification. The data sets employed for the experiments include real music objects, that originated from **kern Humdrum files acquired from the Humdrum website [14], while ground truth is provided by the very same website. It should be noted that our proposed approach is by no means bound to the specific file format, as any symbolic music file format (such as midi, musicXML, etc.) can be used instead. Each **kern file was stripped in order to retain only the note pitch and duration information. All the music objects pertain to classical works. The following five sub-categories were selected: ballads, chorales, fugues, mazurkas and sonatas. Fifty songs were randomly selected by each category, adding up to a total corpus of 250 pieces.

After the feature extraction is completed, the distinguishing capability of the feature vectors is examined by means of the k-NN classifier, using the leave-one-out method. That is, one musical piece in the database is assumed to be of unknown genre and the rest of the pieces are considered as training data.

We have experimented with a set of tuning parameters in order to establish the levels of accuracy achieved by the proposed methodology. In particular, we experimented with the number k of nearest neighbors during the k-NN search, the size w of the window performed on the co-occurrence matrix, the frequency of appearance and the size threshold for each RP in order to be included in the co-occurrence matrix, the M parameter for the normalisation step and the weights of the two feature sets. Due to space restrictions, we do not present results on the tuning of these parameters. Table 2 summarises the default values that resulted from the tuning process. In each measurement we also tune PH and use the best parameters for it.

Parameter	Default value
window	100
min RP size	1
min RP frequency	2
M	3
weight	70%-30%
number of neighbors (k)	8

Table 2: Experimental parameters.

4.2 Results

In our first experiment we compared RPH and PH. The results are given in Tables 3a and b, in terms of confusion matrixes (for example the cell of row 1 and column 3 in Table 3a denotes that 2% of ballads were classified as fugues). The main diagonal represents the correct classifications. As shown, RPH attains significantly better accuracy in all genres. In contrast, PH is outperformed in all cases and in some cases (e.g., for sonatas) it leads to high error.

To further evaluate the merits of the proposed method, we tested two variations of it: (a) one not using RPs (that is, we extract the proposed features from the entire piece and not from its RPs) and (b) one not using the second order dependencies between pitches (that is, it uses RPs but with the constraint that $w = 1$). These two variations help quantify the performance gain obtained by RPH, which uses both the aforementioned characteristics. For purposes of comparison we also measured the accuracy of PH. Figure 1 illustrates the average accuracy (for all genres). As shown, there is a 21% increase in performance due to the use of both characteristics, which results from the comparison of RPH against the two variations. It is interesting, however, that both variations perform better than PH. The reason is that they have at least one good characteristic compared to PH, which is the use second-order dependencies and the consideration of RPs, respectively.

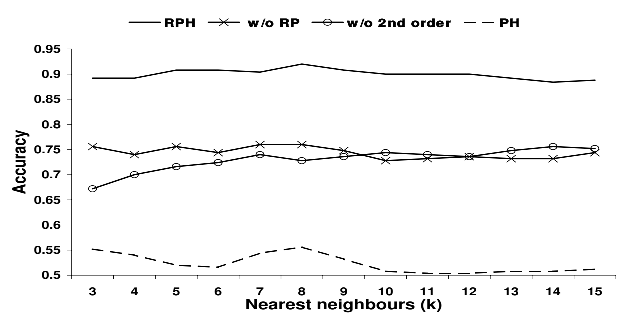

Figure 1: Evaluation of the characteristics of RPH.

Finally, in our last experiment, we tested the accuracy for variable size, L, of the musical piece regarded as unknown, in terms of the number of pitches it includes. This was done in order to test the capability of the proposed algorithm to withstand escalation as far as the size of the piece regarded as unknown is concerned and to indicate the capability of the proposed scheme to work in an "on-line" version. Figure 2 depicts the accuracy of RPH and PH versus L. RPH performs favorably against PH and, as shown, an acceptable performance is reached at the one third of the size that meets the full potential.

	ballad	choral	fuga	mazurka	sonata
ballad	47	0	2	1	0
choral	0	49	0	0	1
fuga	1	0	48	0	1
mazurka	1	0	1	41	7
sonata	0	1	0	4	45

(a)

	ballad	choral	fuga	mazurka	sonata
ballad	31	13	3	0	3
choral	6	37	2	2	3
fuga	2	9	28	6	5
mazurka	5	4	6	26	9
sonata	7	14	3	9	17

(b)

Table 3: Genre classification confusion matrix (percentage values): (a) RPH (proposed), (b) PH ([8]).

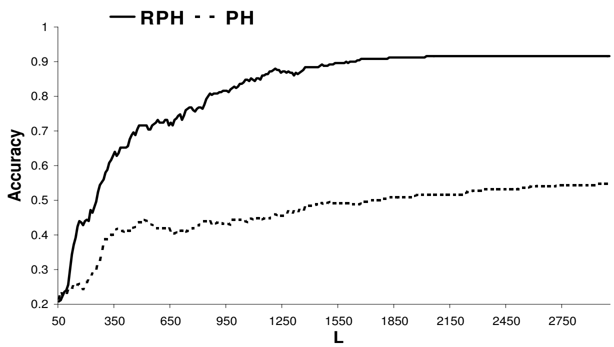

Figure 2: Accuracy for varying L.

5. CONCLUSIONS

This paper proposes the use of note pitch statistical features, derived from repeating patterns and features from duration histograms, for the purposes of symbolic music genre classification. Note-information histograms have a great capability in capturing a fair amount of information regarding harmonic as well as rhythmic features of different musical genres and pieces, while repeating patterns refer to segments of the piece that are semantically important.

This is verified through extensive experimental results, which illustrate the suitability of the proposed feature, reaching an accuracy level of 92% for an intra-category dataset.

6. REFERENCES

[1] J.-J. Aucouturier and M. Sandler. Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing, 2002.

[2] H. Barlow and S. Morgenstern. *A Dictionary of Musical Themes*. Crown, 1975.

[3] R. Basili, A. Serafini, and A. Stellato. Classification of musical genre: a machine learning approach. In *Proceedings of ISMIR*, 2004.

[4] T. Crawford, C.S. Iliopoulos, and R. Raman. String matching techniques for music similarity and melodic recognition. *Computing in Musicology*, 11:73–100, 1998.

[5] H. Guo and Y. L. Murphey. Automatic feature selection - A hybrid statistical approach. In *Procceedings Conf. on Pattern Recognition (ICPR)*, page 2382, 2000.

[6] J.L. Hsu, C.C. Liu, and A.L.P. Chen. Discovering non-trivial repeating patterns in music data. *IEEE Transactions on Multimedia*, 3(3):311–325, 2001.

[7] I. Karydis, A. Nanopoulos, and Y. Manolopoulos. Finding maximum-length repeating patterns in music databases. *Multimedia Tools and Applications, to appear*.

[8] C.-R. Lin, N.-H. Liu, Y.-H. Wu, and A.L.P. Chen. Music classification using significant repeating patterns. In *Procceedings Database Systems for Advanced Applications*, pages 506–518, 2004.

[9] C.C. Liu, J.L. Hsu, and A.L.P. Chen. Efficient theme and non-trivial repeating pattern discovering in music databases. In *Proc. 15th IEEE International Conf. on Data Engineering (ICDE)*, pages 14–21, 1999.

[10] C. McKay and I. Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Proceedings of ISMIR*, pages 31–38, 2004.

[11] F. Pachet and D. Cazaly. A taxonomy of musical genres, 2000.

[12] L. Smith and R. Medina. Discovering themes by exact pattern matching. In *Proc. of International Symposium on Music Information Retrieval*, pages 31–32, 2001.

[13] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. In *Proceedings of ISMIR*, pages 31–38, 2002.

[14] The Humdrum website. A library of virtual musical scores in the humdrum **kern data format. http://kern.humdrum.net.